



AUTODESK ARNOLD (MAYA)

AUTODESK ARNOLD 튜토리얼 가이드

[이 문서는 Autodesk사의 Arnold 렌더러 한글 튜토리얼 가이드 문서이며, 아놀드 공식 사이트의 내용을 번역한 것입니다..]

[\[Autodesk Korea\]](#)



Table of Contents

Arnold Tutorials (Maya)	6
Arnold for Maya 소개	7
Learning scenes	12
머트리얼 라이브러리	13
Arnold에서 물리적 기반의 렌더링 이해하기	14
감마 보정 및 리니어 워크플로우	20
AOV 자습서	24
Ai Write Color - 별도의 렌더링 AOV 생성	25
Cryptomatte	27
Motion Vector 쉐이더가 있는 사용자 정의 Motion Vector AOV	30
Light Path Expression 소개	33
Motion Vector AOV	38
사용자 정의 AOV를 사용하여 매트 생성	43
Z Depth AOV	46
Kick	49
Kick 시작하기	50
렌더 로그를 읽는 방법	55
조명	64
인테리어 장면을 MtoA로 변환	65
플래시 사진 효과	71
IES 조명	73
방 조명	77
방 쉐이딩	78
Ambient Occlusion으로 렌더링	80
Ai Area Lights	85
Emissive Geometry	91
Physical Sky 조명 적용	95
Ai Sky Dome Light 조명 적용	97
자연사 박물관 조명 효과	100
스튜디오 자동차 렌더링	112
스튜디오 조명	118
Sky Dome Light 사용하기	121

Skydome & The Table 설정하기.....	122
Ai Standard 쉐이더를 사용하여 자동차 쉐이딩 하기	125
최종 렌더링 설정하기.....	130
Shading	132
Anisotropic Brushed Metal	133
Normal과 Bump 맵 결합하기	136
Displacement 튜토리얼	138
<udim> Token을 사용한 Displacement 맵핑	139
Displacement로 바다 렌더링하기	143
Mudbox에서 Arnold로 Vector Displacement	149
Flakes 예	154
무지개 빛의 비누 방울 쉐이더.....	157
Color Shaders를 사용한 모션 그래픽 효과	159
Emissive 쉐이더를 사용한 굴절 커스틱 효과	162
Remap an Image Using UV Coords	166
Rim 쉐이더.....	169
지구 쉐이딩 하기	171
Earth Shader	172
Cloud Shader	174
대기에 Volume 사용하기	175
장미 쉐이딩 하기	176
Specular BRDF(Bidirectional Reflectance Distribution Function: 양방향 반사율 분포 함수).....	180
Facing Ratio 쉐이더를 이용한 Toon Shading.....	185
User Data 튜토리얼	189
Ai User Data Float 튜토리얼	190
Ai User Data String	192
Ai Utility - Object ID Mode	195
Ai Write Color	197
물병 만들기	206
Curvature 쉐이더로 마모 효과 만들기	211
노이즈 제거.....	213
샘플링이란 무엇인가요?	223
노이즈 제거 워크플로우	230
Fireflies (반딧불 노이즈) - 보트 장면	231
Volume - UFO 장면	235
렌더링 튜토리얼.....	243

Stand-in 소개	244
Physical Sky를 사용한 구름 유체	252
Depth of Field	260
머리카락 렌더링	261
만델불브(Mandelbulb) 렌더링 방법	265
MakeTX: 나무 바닥	270
파트 1 – Scene 셋업	271
파트 2 – Render & Conversion	272
HDRI 및 Backplate를 사용하여 자동차 렌더링하기	275
HDRI 준비하기	276
Shadow Matte 생성하기	277
최종 렌더링	279
Curve 렌더링하기	282
깃털처럼 Curve 렌더링하기	283
Override Sets을 사용하여 Paint Effect Curves 렌더링하기	288
Render Curve를 사용하여 추상적인 머리 렌더링	293
Arnold로 유리 표면 렌더링하기	298
유리와 액체 모델링하기	300
유리와 액체 쉐이딩하기	303
렌더링 설정	304
페인트 효과와 함께 불투명도 맵 렌더링하기	305
Subdivision Displacement 예	308
안개 낀 수중 장면	310
기술적인 사항	316
Maya에서 Arnold Shader 만들기	317
쉐이더 생성하기	318
쉐이더 로더 생성하기	322
파라미터 추가하기	323
조명 필터 쉐이더 만들기	336
익스텐션(Extensions) 쓰기	340
기본 익스텐션 만들기	341
노드에 특정 Arnold 속성 추가하기	345
동일한 노드에 대한 여러 변환기	349
사용자 정의 노드의 변환기	351
익스텐션을 MtoA 1.4.0으로 포팅	357
Oculus Rift Camera Node 만들기	359
기본 구체 카메라 만들기	360

MtoA에 카메라 통합하기	363
입체 파노라마 카메라 만들기	364
Oculus Rift에서 결과 시각화하기	369
Volume 튜토리얼	370
Polymesh를 Volume으로 변환	371
Volume Shader를 사용하여 구름 렌더링하기	374
Volume - Displacement	376
Volume Sample RGB 예	378
XGen 튜토리얼	382
XGen 및 Arnold와 렌더링 가이드	383
아카이브 (Archives)	386
XGen에서 속성 패턴 애니메이트하기	387
텍스처 맵으로 프리미티브 제어하기	396
장난감 병사 만들기	403
아카이브 파일 만들기	403
XGen 설명 작성하기	404
Generator Attributes	406
수식을 사용하여 병사에 임의 색상 지정하기	407
수식을 마스크로 사용하기	409
마스크로 텍스처 맵 사용하기	410
워크플로우 문제 - Primitive Bound	412
수식으로 도미노 배치하기	414
색상 지정하기	422
프리미티브에 임의 색상 지정하기	423
프리미티브 아카이브에 색상 지정하기	428
구체 프리미티브에 색상 지정하기	434
스플라인에 색상 지정하기	439
스플라인 (Splines)	443
스플라인으로 잔디 만들기	444
점프하는 거미	451
XGen 스플라인 패턴	461
수식을 사용하여 구체의 크기를 균일하게 조절하기	467

Arnold Tutorials (Maya)



이 매뉴얼에는 Arnold와 MtoA의 실제 사용에 대한 특정 요점을 보다 잘 설명할 수 있도록 작성된 자습서가 포함되어 있습니다. 이 자습서의 주제를 최대한 활용하려면 Maya 및 MtoA를 사용하여 컴퓨터에서 작업하는 것이 좋습니다.

Maya와 MtoA가 이들 자습서에서 사용되었지만 많은 자료는 일반적으로 Arnold에 관한 것으로 다른 Arnold 소프트웨어 플러그인 사용자와도 관련이 있습니다.

이들 자습서와 함께 제공되는 장면 파일 상당 부분은 [Learning scenes](#) 페이지에서도 찾을 수 있습니다.

자습서의 주제는 다음과 같습니다.

- [Arnold for Maya 소개](#)
- [Learning scenes](#)
- [머트리얼 라이브러리](#)
- [Arnold에서 물리적 기반의 렌더링 이해하기](#)
- [감마 보정 및 리니어 워크플로우](#)
- [AOV 자습서](#)
- [Kick](#)
- [조명](#)
- [쉐이딩](#)
- [노이즈 제거](#)
- [렌더링 튜토리얼](#)
- [기술적인 사항](#)
- [Volume 튜토리얼](#)
- [XGen 튜토리얼](#)

Arnold for Maya 소개



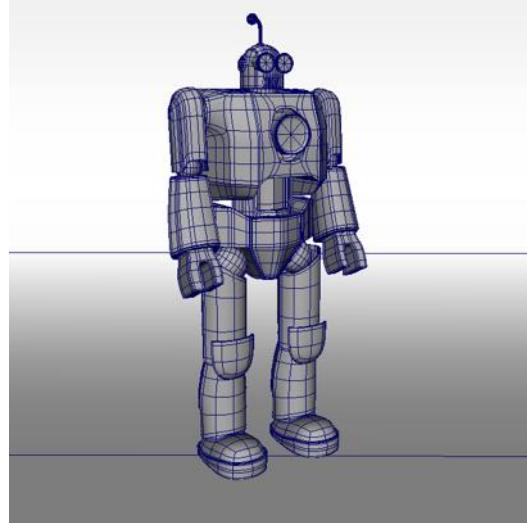
Arnold는 장편 애니메이션 및 시각 효과 영화의 요구에 부합하도록 개발된 고급 Monte Carlo 추적 렌더러입니다. 이것은 Autodesk Maya에서 Arnold 렌더러를 직접 사용할 수 있게 해주는 플러그인인 MtoA를 소개하는 초보자 안내서입니다. 이 튜토리얼에서는 모든 종류의 오브젝트를 조명 처리 및 렌더링하는 데 사용할 수 있는 간단한 사진 조명 스튜디오 설정 방법을 설명합니다. Arnold 렌더러를 사용하여 장난감 로봇 모델을 조명 처리, 음영 처리 및 렌더링하는 단계를 수행합니다. 우리는 물리적으로 정확하고 사실적인 조명 설정을 위해 Arnold의 독점적인 조명을 사용합니다. Arnold의 조명을 사용할 뿐만 아니라 Arnold로 렌더링할 때도 표준 Maya 조명을 사용할 수 있습니다. 조명을 선택한 다음 Maya 속성 편집기와 일반 조명 속성을 검사하면 해당 조명에 대한 새로운 Arnold 속성 그룹이 표시됩니다. 여기에서 Arnold가 사용하는 추가 설정에 액세스할 수 있습니다.

Standard Surface 쉐이더를 사용하여 로봇을 음영 처리하고 금속 마무리를 해보겠습니다. Standard Surface 쉐이더는 간단한 플라스틱에서부터 차량 페인트 또는 스키니에 이르는 모든 유형의 재료를 생산할 수 있는 다목적 쉐이더입니다. Standard Surface 쉐이더는 매우 강력하며, 다양한 종류의 재료를 생산하도록 지원하지만 처음에는 다소 부담스러울 수 있습니다. 많은 수의 컨트롤로 인해, Standard 쉐이더는 Diffuse, Specular, Sub-Surface scattering 등과 같은 여러 그룹으로 나뉩니다. 믿을 수 있는 브러싱 금속 효과를 얻으려면 Diffuse 및 Glossy 속성을 조정해야 합니다.

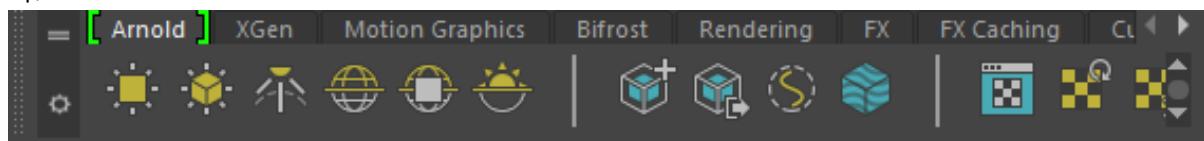
우리는 현실감 있고 물리적으로 정확한 피사계 심도를 만들기 위해 Arnold의 카메라 렌즈 옵션 중 일부를 살펴볼 것입니다. 마지막으로 렌더링 설정을 최적화하고 렌더링에 나타날 수 있는 노이즈를 제거하는 방법을 살펴보겠습니다. 이 튜토리얼에서는 Maya와 MtoA가 사용되었지만 대부분의 자료는 Arnold에 관한 것이므로 3ds Max to Arnold (MAXtoA) 및 Houdini to Arnold (HtoA) 등 다른 Arnold 구현 사용자와도 관련이 있습니다. 이 자습서를 완료하는 데 걸리는 예상 시간은 30분을 넘지 않아야 합니다.

장면 설정

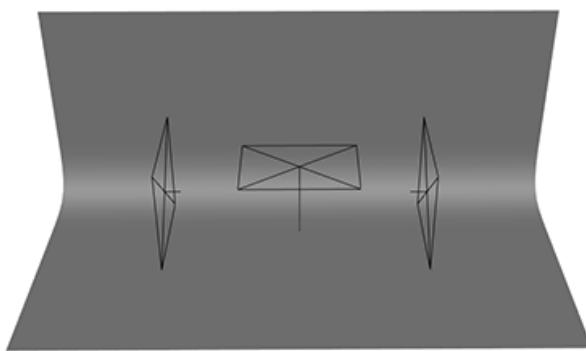
- 먼저 **Arnold-robot.ma** 장면을 엽니다. 그러면 장면에 로봇 모델(Maya 격자에서 0,0,0에 위치)과 간단한 스튜디오 배경 모델이 표시됩니다.



- 이제 스튜디오 장면에 조명을 켜보겠습니다. 먼저 3개의 **Ai Area 조명**을 생성합니다. 이것은 Arnold 셀프의 조명 아이콘을 클릭하거나 Arnold 메뉴: *Arnold-> Lights-> Area Light*를 선택하여 실행할 수 있습니다.

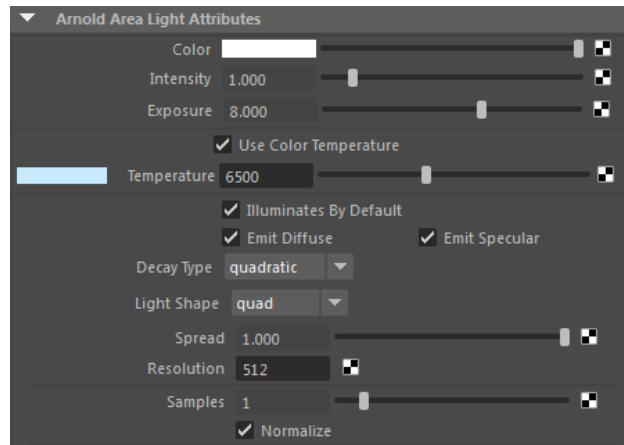


- 조명 1개는 로봇의 양쪽 중 하나에, 다른 1개는 중앙에 놓습니다. 아래 스크린그랩에서처럼 조명들이 로봇을 향해 안쪽을 가리키도록 회전합니다.



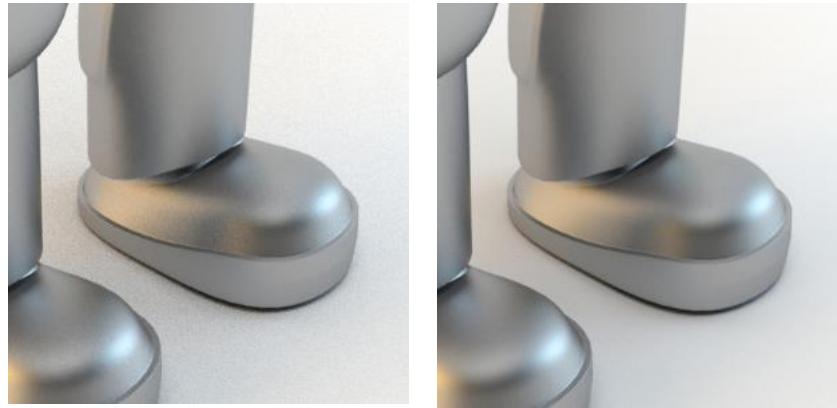
조명 설정

- 장면이 렌더링되면 어둡게 나타날 수도 있습니다. 따라서 조명의 노출을 높여야 합니다. Arnold 속성에서 **Exposure**를 각 조명에 대해 8 정도로 높입니다.
- Color Temperature**를 활성화하여 조명의 색상을 변경할 수 있습니다. Arnold 속성 내에서 올바른 조명을 선택하고 **Color Temperature**를 활성화합니다. 기본 온도 값인 **6500**으로 두면 조명에 시원한 파란색 느낌이 납니다. 왼쪽에 있는 조명을 선택하고 동일한 작업을 수행합니다. 단, **Color Temperature**를 따뜻한 색상인 **4000** 정도로 변경합니다.



조명 샘플

- Ai Area 조명의 음영에서 노이즈가 발견될 수도 있습니다. 이것은 조명 샘플이 기본적으로 1로 설정되어 있기 때문입니다. 노이즈를 줄이려면 조명 샘플을 3으로 높여야 합니다. 이 설정은 각 조명 내에서 Arnold 속성에 있습니다. 조명 샘플은 부드러운 음영의 노이즈 품질과 직접 반사 하이라이트를 제어합니다. 샘플 수치가 높을수록 노이즈가 적어지며 렌더링 시간이 오래 걸립니다.

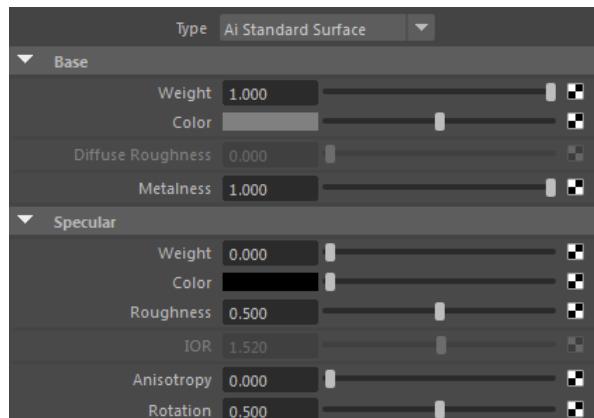


1

3

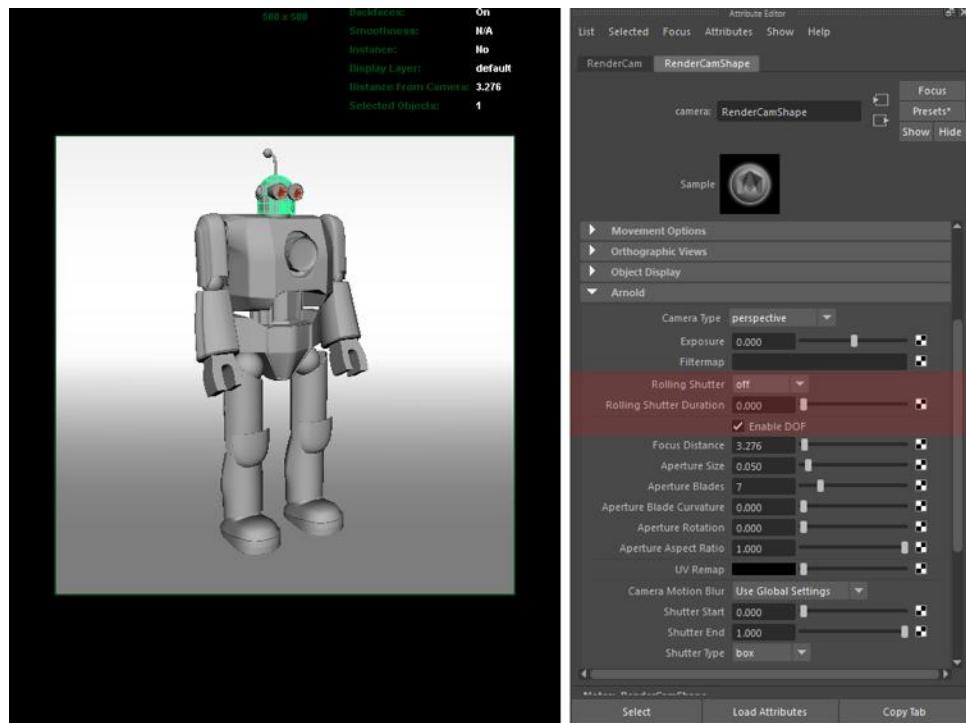
쉐이딩

- **Standard Surface** 쉐이더를 로봇에 할당합니다. 브러싱 금속 재료로 바꾸기 위해 다음 설정을 변경해 보겠습니다. **Metalness**를 1 정도로 높이고 **Base Color**를 중간 회색 색상으로 변경하고 **Specular Weight**를 0으로 줄이십시오. **Specular Roughness**는 정반사의 광택을 제어합니다. 이 값이 작을 수록 반사가 선명합니다. 이 수치를 0.5 정도로 높입니다.



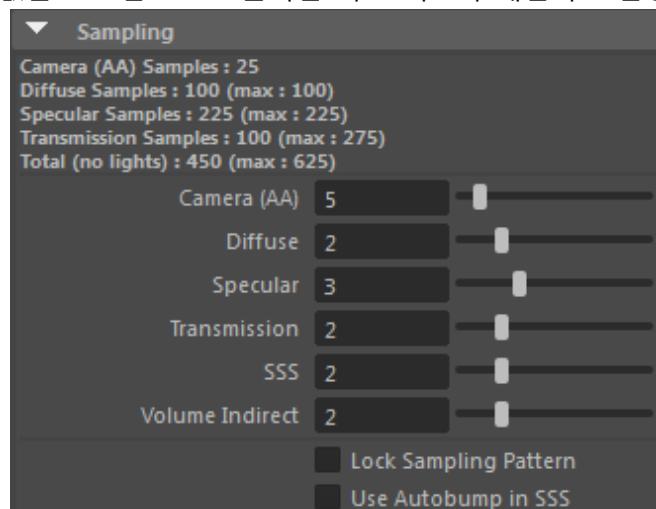
Camera Depth of Field

- Maya의 고유한 카메라를 사용하여 피사계 심도로 장면을 렌더링할 수 있습니다. 렌더링 카메라를 선택하고 속성 편집기에서 해당 Arnold 속성을 엽니다. 'Enable DOF'를 클릭합니다. 'Aperture Size'를 늘리지 않으면 렌더링에 차이가 없습니다. 하지만 먼저 무엇인가에 카메라의 초점을 맞추어야 합니다. 로봇의 머리를 선택하고 **Display>Heads up display>Object Details**로 이동합니다. 그러면 헤드 모델의 카메라로부터의 거리가 표시됩니다. 이 값을 'Focus Distance' 속성에 입력합니다. 이제 'Aperture Size'를 높이면 카메라의 초점에 로봇의 머리가 나타납니다. 0.05와 같은 낮은 값을 입력합니다(이 값은 장면 크기에 따라 다릅니다).



렌더링

- 기본 카메라(AA) 설정 3을 사용하여 장면을 렌더링하면 테스트 렌더링에 충분합니다. 그러나 최종 렌더링의 경우 장면에서 설정한 피사계 심도의 양에 따라 이 값을 5 이상으로 늘려야 합니다.
- 로봇 쉐이더의 간접 반사 성분을 샘플링하지 못하여 로봇 표면에 약간의 반사 노이즈가 있을 수도 있습니다. 전역 **Specular** 값을 2 또는 3으로 높이면 이 노이즈가 해결되고 선명한 렌더링이 지원됩니다.



샘플링과 노이즈

노이즈는 거의 항상 불충분한 샘플링에서 유래하지만, 잘못된 광선에 대한 샘플링을 증가시키면 노이즈를 제거하지 않고 렌더링 시간을 늘릴 수 있습니다. 목표는 광선을 가능한 효율적으로 할당하여 가장 효율적인 방식으로 잡음을 최소화하는 것입니다. 따라서 DOF 노이즈를 제거하기 위해 카메라 샘플을 늘려야 하는 경우 렌더링 시간을 관리하기 위해 다른 설정을 낮춰야 합니다. 그러나 DOF 또는 모션 블러가 문제가 되지 않는 경우 Camera 샘플을 늘리면 다른 곳의 모든 노이즈가 해결될 수 있지만 불필요한 광선으로부터 렌더링 시간이 느려집니다.

Learning scenes

이 페이지의 장면들은 Arnold를 구성하는 다양한 구성요소의 학습 안내서로 사용됩니다. 일부 장면은 여기에 있는 다른 튜토리얼의 일부이므로 이들 장면의 내용에 대한 상세정보는 해당 자습서를 방문해야 합니다. 이 페이지에서 자유롭게 장면을 탐색하고 일부 렌더링을 다시 만드십시오.

이들 장면은 학습 목적을 위한 것이므로 MtoA를 학습하는 데 필요한 기본 지식을 포함하고 있습니다. 일부 장면에는 사용자 고유의 텍스처 맵을 사용해야 할 수도 있습니다.

이 튜토리얼에 필요한 모든 Maya 예제 파일은 하단 링크의 사이트에서 다운로드 하실 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Learning+Scenes>

머트리얼 라이브러리

이 재료 라이브러리는 Standard Surface 쉐이더 사용 시 가능한 재료 효과 유형의 초보자 가이드입니다. 여기에 사용된 값들은 물리적인 실제 값을 기반으로 하지 않으며, 초보자가 탐색할 수 있는 출발점으로만 사용됩니다. 이들 재료는 또한 Maya의 Standard Surface 속성 편집기에서 프리셋으로 사용할 수 있습니다.



재료 라이브러리는 여기에 있습니다.

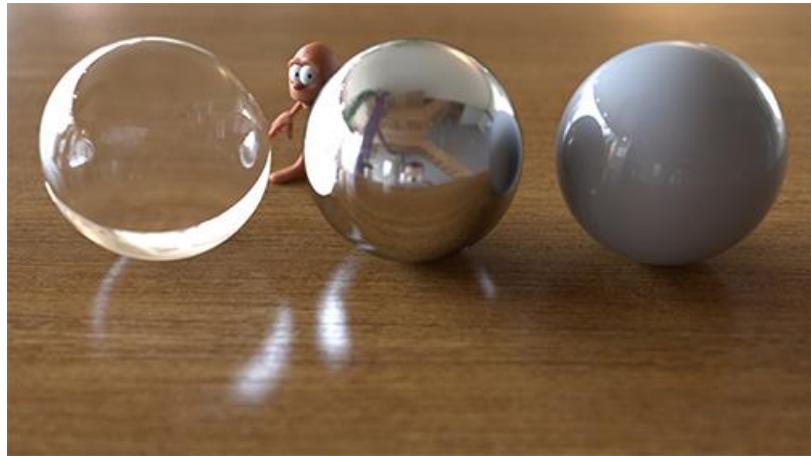
<https://support.solidangle.com/display/A5AFMUG/Material+Library>

Arnold에서 물리적 기반의 렌더링 이해하기



물리적 법칙을 기반으로 재료를 설계하면 특별히 사실주의나 물리적인 정확성을 위해 노력하지 않아도 음영 처리 및 조명 효과가 매우 간단해 집니다. 몇 가지 원칙을 이해 및 적용함으로써 보다 실감나는 이미지를 연출하고 다양한 조명 설정에서 더 예측 가능하게 작용하는 재료를 만들 수 있습니다.

현대 렌더링 장치에서 물리적 기반 렌더링은 일반적으로 에너지 절약, 물리적으로 현실적인 산란, 재료 및 선형 색 공간에서의 레이어링과 같은 개념을 의미합니다. Arnold는 물리적인 렌더링 도구이지만, 원하는 경우 규칙을 깨고 물리 법칙을 따르지 않는 재료와 조명을 만들 수도 있습니다. 이 문서에서는 기본 이론과 이러한 원리를 따르기 위해 쉐이더를 설정하는 방법에 대해 설명합니다.



Arnold는 Substance Painter 등과 같은 다양한 타사 애플리케이션을 지원합니다.

<https://www.allegorithmic.com/products/substance-painter>

Photons 및 Scattering

렌더링에서는 빛에서 방출되어 공기를 통해 이동하며 표면과 체적에 맞고 튀어 나와 결국 카메라 센서에 도달하는 광자를 시뮬레이션합니다. 그러면 카메라 센서에서 수백만 개의 광자가 결합되어 렌더링된 이미지가 형성됩니다.

이것은 물리학적인 관점에서 표면 쉐이더가 표면이 광자와 어떻게 상호 작용하는지를 보여줍니다. 물체에 부딪히는 광자는 흡수되거나, 표면에서 반사되거나, 표면을 통해 굴절하거나, 물체 내부에서 산란될 수 있습니다. 이러한 구성요소의 조합으로 다양한 재료가 탄생할 수 있습니다.

Energy Conservation

물체가 광자를 방출하는 광원이 아니라면 들어오는 빛에 의해 발생하는 것보다 더 많은 에너지를 반환할 수 없습니다. 재료가 에너지를 보존하기 위해서는 표면을 떠나는 광자의 수가 들어오는 광자의 수보다 작

거나 같아야 합니다. 재료가 에너지를 보존하지 못하면 재료가 지나치게 밝게 표현되고 특히 전역 조명을 사용할 때 노이즈가 증가합니다.

재료의 에너지를 보존하려면 재료 구성요소의 무게와 색상이 1을 초과해서는 안 됩니다. 또한 모든 구성요소의 결합이 에너지를 절약하도록 보장해야 합니다. 자세한 내용은 나중에 다시 설명하겠습니다.



Materials

현미경 수준에서 물체 표면을 보면 매우 복잡하고 세밀합니다. 렌더링 시 파라미터의 쉬운 이해를 위해 기하학이 아닌 통계적인 모델을 사용하여 이 모든 디테일을 표현합니다.

Arnold의 **Standard Surface** 쉐이더 모델 개체는 한 두 개의 반사 층, 확산 또는 투명한 내부를 갖습니다. 이 모델은 다양한 종류의 재료를 표현할 수 있습니다. 이제, 개별 구성요소를 살펴보도록 하겠습니다.

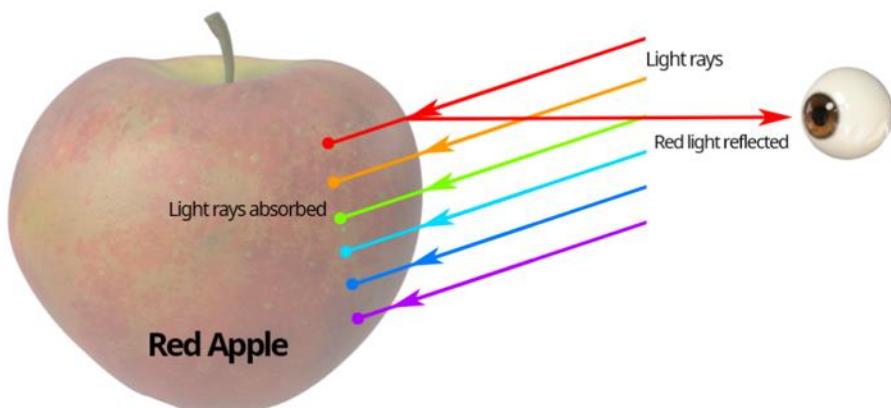
Diffuse 및 Subsurface Scattering

먼저 확산 내부를 고려합니다. 들어오는 광자가 물체에 들어가고 안쪽으로 흩어져 흡수되거나 다른 위치에서 물체를 떠납니다.

광자가 여러 번 산란되면 여러 다양한 위치와 방향에서 표면을 떠나는 광자로 인해 확산 모양을 갖게 됩니다. 피부와 같은 재료의 경우에는 광자가 표면 아래로 상대적으로 멀리 흩어져서 매우 부드러운 외관이 표현되며, 이는 표면 아래 산란으로 렌더링됩니다. 미완성 목재와 같은 재료의 경우에는 광자가 멀리 산란되지 않으므로 더 딱딱하게 표현되며 이들을 확산으로 렌더링합니다. 나뭇잎처럼 얇은 물체의 경우에는 광자가 물체의 반대편으로 끝까지 흩어질 수 있습니다. 이들은 `thin_wall`이 활성화된 상태로 확산 SSS로 렌더링됩니다.

근본적으로 이러한 모든 유형의 재료는 쉐이더에서 별도의 컨트롤을 제공하더라도 기본적으로 동일한 물리적 메커니즘을 가지고 있습니다.

확산 내부는 또한 일반적으로 재료의 전반적인 색상에 가장 큰 영향을 미칩니다. 각각의 광자가 관련된 파장을 가지며, 재료의 특성에 따라 일부 파장을 가진 일부 광자가 다른 광자보다 쉽게 흡수됩니다. 이것은 반대로 어떤 파장을 가진 광자들은 표면을 떠날 가능성이 높고 결국 색상이 있는 외관을 생성하게 된다는 의미입니다.



적색 사과의 스킨은 대부분 적색 조명을 반사합니다. 적색 파장만 사과 피부 바깥으로 흩어지며 다른 파장은 흡수됩니다.

Energy Conservation

단일 광자는 확산, 하위 표면 산란 및 배경 조명 구성 요소 중 하나만 참여할 수 있습니다. 물리적인 사실성을 위해 들어오는 광자가 표면을 떠나는 광자보다 많아야 합니다. Standard Surface의 경우에는 이들 구성 요소의 합이 1보다 크지 않도록 자동 보장됩니다.

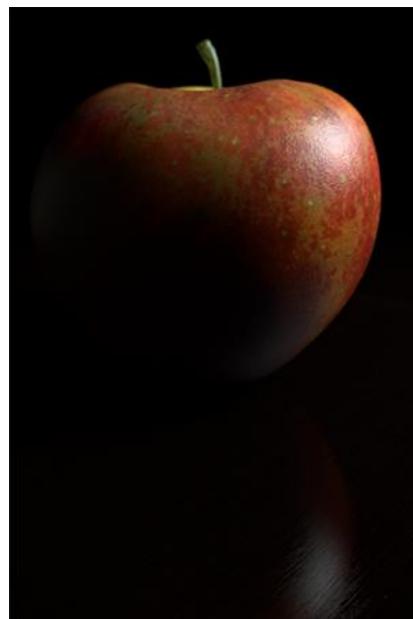
Specular Scattering (반사 산란)



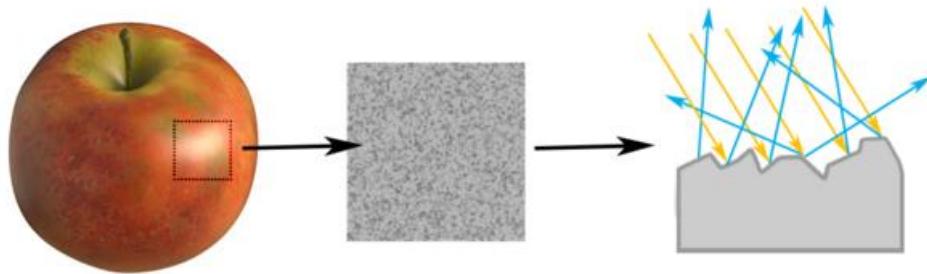
Specular Roughness 0에서 1

Roughness (거칠기)

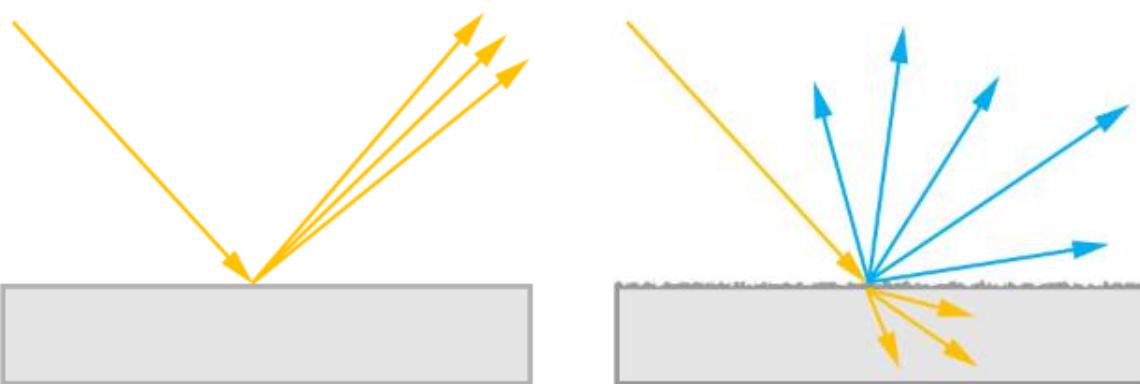
반사층은 미세면의 분산을 사용하여 모델링됩니다. 표면이 무작위 방향의 미세한 면으로 이루어져 있다고 가정합니다. 거울처럼 거칠기가 약한 표면은 면 사이의 변화가 거의 없으므로 날카로운 반사가 유발됩니다. 거칠기가 강하면 변화가 커서 더 부드럽고 광택이 있는 반사가 유발됩니다.



사과에 강력한 Specular 하이라이트가 표현됩니다. 테이블의 반사는 범위가 넓고 흐릿한 것을 볼 수 있습니다(높은 Specular Roughness 값).



거친 반사는 산란된 광선에 의해 유발됩니다.



광택 있는 표면. 입사각과 반사각이 동일합니다. 확산 표면. 광선 방향은 무작위로 다양합니다.

Roughness Map

표면의 하이라이트를 다양하게 변화시키려면 맵을 Specular Roughness에 연결해야 합니다. 이것은 하이라이트의 밝기뿐만 아니라 환경적인 반사의 크기 및 선명도에도 영향을 미칩니다.



Low Specular Roughness



High Specular Roughness



Specular Roughness에 연결된 '스크래치' 텍스처

Transmission

광자는 표면에서 반사될 뿐만 아니라 굴절될 수도 있습니다. 광자는 반사 층을 통과하면서 일반적으로 층의 다른 면에서 나올 때 방향을 바꾸며 굴절률(IOR)에 의해 제어됩니다.

투명 유리와 같이 표면의 내부가 투명하면, 광자는 물체를 통과하여 다른 면에서 빠져 나올 수 있습니다. 확산 내부가 있는 경우, 광자는 물체 내부로 흘어지고 물체에서 흡수되거나 물체를 다시 벗어날 수 있습니다. 반사 층의 굴절이 심할 수록 기저의 확산 내부가 더 많이 보입니다. 금속과 같은 물질의 경우에는 반사 층을 통해 굴절하는 광자가 일반적으로 즉시 흡수되기 때문에 확산 내부가 보이지 않습니다.

Fresnel (프레넬)

반사 층에 의해 반사되거나 굴절된 광자의 비율은 보는 각도에 따라 다릅니다. 표면을 위에서 보면 대부분의 빛은 굴절되고 지표각에서는 대부분의 빛이 반사됩니다. 이것을 프레넬 효과라고 합니다. 굴절률은 보기 각도에 따라 이 효과가 어떻게 달라지는지를 정확하게 제어합니다.



보기 방향에 따른 Specular BRDF의 변화

Opacity와 Transmission

불투명도는 텍스처를 사용하여 곡면 지오메트리를 모델링하는 방법입니다. 이것은 광자가 표면과 상호 작용하는 방식에는 영향을 미치지 않으며, 표면의 기하학이 없고 광자가 바로 통과할 수 있음을 나타냅니다.



Opacity에 연결된 램프 텍스처

불투명도의 일반적인 용도는 다각형 카드에서 나뭇잎의 모양을 잘라내거나 머리카락의 끝을 투명하게 만드는 것과 같은 스프라이트 유형의 효과입니다. 하지만 많은 불투명도 스프라이트(예: 나뭇잎)가 포함된 장면은 렌더링 속도를 상당히 저하시킬 수 있다는 점을 기억하십시오.



나뭇잎 불투명도: 활성화됨



나뭇잎 불투명도: 비활성화됨



불투명도에 연결된 알파 맵

전달 깊이는 비슷하지만 표면보다는 물체 내부의 밀도를 제어합니다. 밀도가 높으면 내부를 통과할 때 더 많은 광자를 흡수하여 더 두꺼운 곳을 더 어둡게 만듭니다.



Transmission Color: 흰색

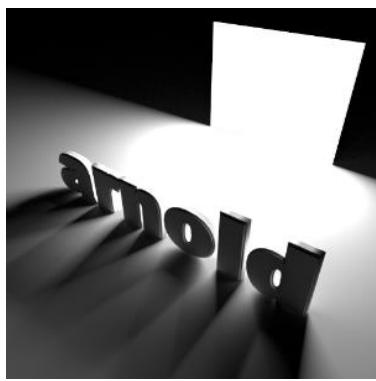


Transmission Color: 하늘색

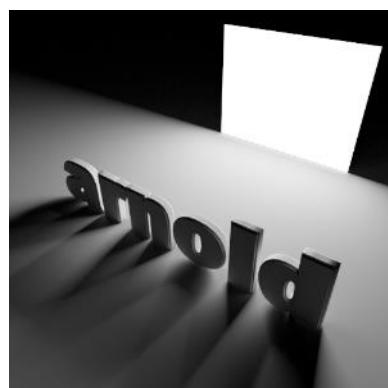
감마 보정 및 리니어 워크플로우

Gamma Correction 속성(attributes)은 지원이 중단되어 Maya 2017에 더 이상 나타나지 않습니다. 그렇기 때문에 Maya의 고유한 Color Management 설정을 사용해야 합니다. Maya 2017에서 MtoA를 통해 색상 관리(Color Management)를 사용하는 것에 대한 추가 정보는 여기에 있습니다.

감마 보정 컨트롤 (Gamma correction controls)은 사용자가 방사형의 선형 (Linear) 공간에서 작업하고 있는지 확인하는 데 유용합니다('리니어 워크플로우(linear workflow)'라고도 함). 리니어 워크플로우는 한 장면에서 상호 작용하는 모든 요소(빛의 색상, 텍스처 및 쉐이더의 색상)가 실제 빛의 양과 선형으로 일치하도록 하여, 렌더러가 물리적으로 정확한 빛의 계산을 올바르게 수행하도록 도와줍니다. 이것은 빛이 실제 선형으로(리니어) 작동하기 때문에 사실적인 렌더링 작업에 필요합니다(광자 양을 두 배로 하면 색 값도 두 배가 됩니다).



비선형 (Non-linear) 워크플로우: 부정확



선형 (Linear) 워크플로우: 정확

모든 계산은 리니어 컬러 스페이스에서 발생해야 하지만 모니터 등의 디스플레이 장치들의 입출력은 감마가 보정된 공간(sRGB 감마 등)에서 발생하게 됩니다.

적절한 리니어 워크플로우 없이 렌더링할 때의 주요 문제점은 빛이 과장되어 보이고 색상이 부자연스러워 보인다는 것입니다. 이때문에 아티스트들이 장면의 조명, 감도, 색상 등을 조절하려고 하지만 이러한 조절은 특정 범위 까지만 가능합니다. 아래 이미지는 다양한 감마 출력으로 이미지를 렌더링할 때 아티스트들이 범하는 흔한 실수입니다.



너무 밝습니다. 텍스처가 색상 관리(Color managed)되지 않으며 출력이 sRGB입니다 (부정확).



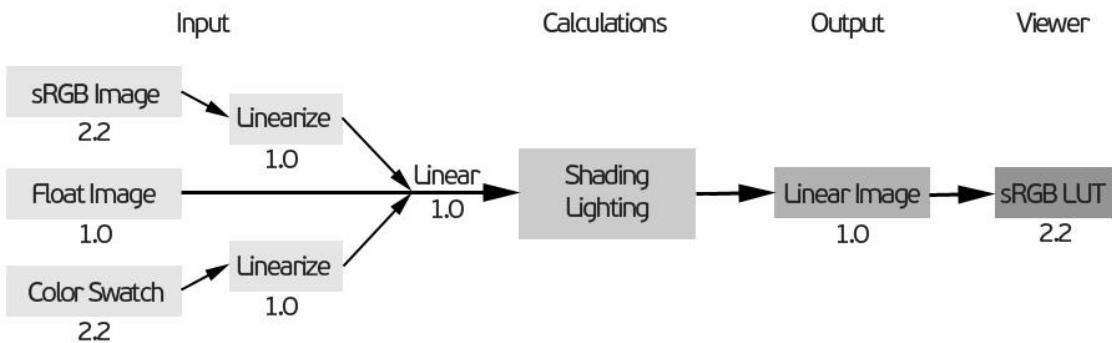
너무 어둡습니다. 텍스처가 색상 관리(Color managed)되지 않으며 색상 출력이 선형 (Linear)입니다 (부정확).



입력(Inputs)이 색상 관리(Color managed)되며 OUTPUT(출력) 디스플레이가 sRGB - 선형 워크플로우입니다. (정확)

선형 작업 흐름을 유지하려면 렌더링된 이미지를 EXR(선형 부동 소수점 형식(linear floating point format))로 출력하는 것이 좋습니다.

렌더링 파이프라인은 다음 단계로 분할될 수 있습니다.



선형으로 입력을 만드는 방법

디스크에 렌더링된 이미지(배치 렌더 사용 등)를 LUT 지원 이미지 뷰어로 확인해야 합니다. 일반적인 LUT 미지원 뷰어를 통해 이미지를 보면 렌더링된 이미지가 상대적으로 너무 어둡게 보입니다.

현재, MtoA는 파일 노드에서 새로운 Autodesk syncolor 색상 관리를 지원하지 않습니다. 즉, 렌더러 설정에서 Global 텍스처 감마를 설정하거나, Hypershade에서 감마 노드를 이용하여 직접 수동으로 감마를 처리하거나, maketx를 사용하여 텍스처를 사전 리니어화 해야 합니다. 마지막 방법이 렌더링 시 추가 계산을 피하기 때문에 가장 좋습니다. 작업자는 적절한 sRGB 보정 또는 보다 복잡한 공간 보정을 지정할 수도 있습니다(Arnold는 간단한 변환만 수행). 가장 중요한 사항은 수학적으로 볼 때 IOIO가 텍스처 필터링을 수행하기 전에 색상 보정이 렌더러의 텍스처 데이터 검색 단계에서 이루어져야만 한다는 것입니다.

Arnold는 텍스처의 내용을 스캔합니다(확장자에 의존하지 않음). Float(부동 소수점화) 되지 않은 모든 이미지는 이미지에 다른 색 공간이 포함되어 있어도 감마가 베이크된 것으로 간주해야 합니다.

Tx Texture Format

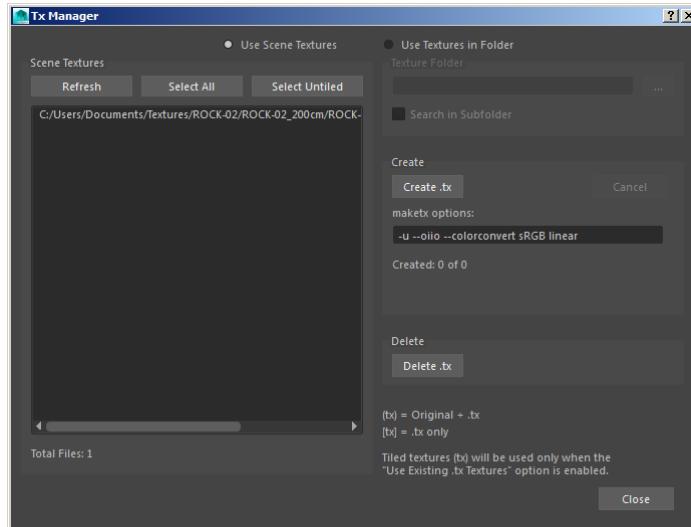
텍스처를 make tx 도구를 사용해 리니어 스페이스로 변환하면 렌더링하기 전에 모든 텍스처가 리니어가 되도록 변환 할 수 있습니다. 리니어, sRGB 및 Rec709 사이에서 선택 가능합니다. 예를 들어, 다음 명령을 참고 하십시오.

```
maketx --colorconvert sRGB linear C:\solidangle\mtoadeploy\2016\bin\crayons.jpg
```

Crayons.jpg 이미지의 Color space를 make tx 도구를 사용하여 변환하는 명령

Tx Manager

TX Manager를 사용하여 텍스처를 .tx 형식으로 쉽게 변환할 수 있습니다. 변환 프로세스에 파라메터를 추가할 수 있는 옵션 박스가 있습니다.

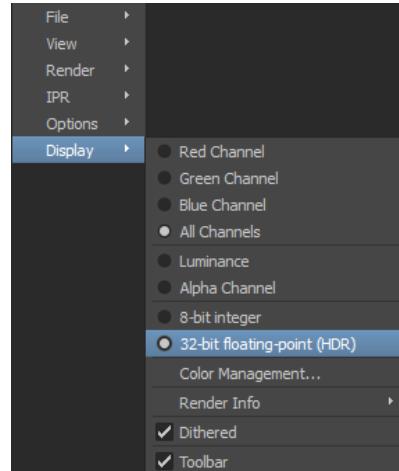


'maketx --colorconvert sRGB linear' 옵션이 있는 Tx Manager 창

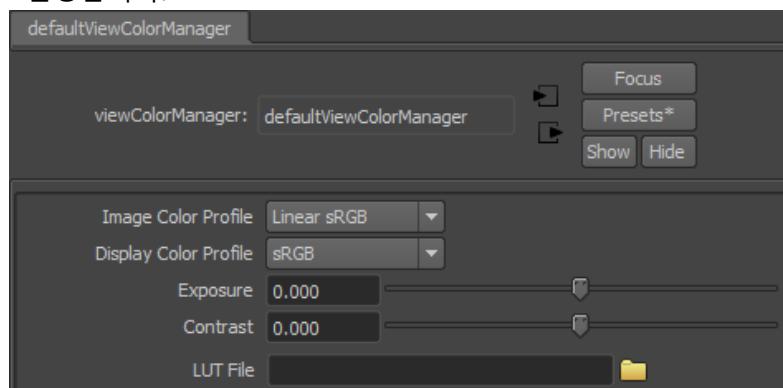
메뉴얼 리니어 워크플로우

또한 Arnold에 의존하지 않고도 다음 단계에 따라 리니어 워크플로우를 직접 관리할 수 있습니다.

- Render View > Display > 32-bit floating-point (HDR)로 설정합니다.



- Attribute Editor에서 ViewColorManager를 엽니다(Render View > Color Management). Image Color Profile을 Linear sRGB로 설정합니다.



'Linear sRGB'는 수신되는 이미지가 Linear임을 Render View 창에 알려줍니다.

- Open EXR로 렌더링합니다.
- Arnold 감마 설정을 다음과 같이 구성합니다.



Maya 2015 이상의 버전은 color swatch widgets의 색 보정을 지원하지 않으므로 쉐이더와 라이트 감마가 2.2로 설정됩니다.

색상 관리 (Color Management)

다음은 Arnold 및 보다 최신 Maya 릴리스에서 색상 관리의 기본 워크 플로우입니다.

- Maya에서 Color Management를 활성화합니다.
- Arnold 렌더 설정 내의 모든 감마 컨트롤을 1.0으로 설정합니다. 즉, Arnold는 모든 색상 및 이미지를 선형으로 예측합니다.
- MtoA Display Gamma는 렌더 뷰에만 적용되며, Maya가 디스플레이 색상 관리를 처리하게 두는 경우 1로 설정할 수 있습니다.
- 단색을 사용하는 경우에는 Color Picker에서 해당 색상을 선택하십시오. 특정 sRGB 값을 일치시키려면 "mixing color space"를 "Display Space"로 설정하고 sRGB 값을 입력하십시오. – 색 견본에 표시되는 대로 렌더링됩니다.
- 파일 텍스처를 사용한다면 파일을 먼저 리니어화하고 Maya에 로드합니다(.tx로 변환의 일부로 리니어화 할 수 있음).

파일 텍스처

MtoA는 파일 노드에서 새로운 Autodesk syncolor 색상 관리를 지원하지 않습니다. 즉, 다음 중 하나를 실행해야 합니다.

텍스처를 maketx로 사전에 리니어화 합니다(그다음 Arnold Textures 감마를 1로 설정)

또는

Arnold Renderer Settings에서 global Texture Gamma를 설정합니다.

또는

하이퍼쉐이드에서 감마 노드로 감마를 수동으로 조절 합니다(Arnold Textures 감마를 1로 설정).

렌더링 시간에서 모든 추가 계산을 피하기 때문에 텍스처를 미리 리니어화 하는 것이 최고의 옵션입니다. 적절한 sRGB 보정 또는 보다 복잡한 공간 보정을 지정할 수 있습니다(Arnold는 간단한 pow만 실행함). 마지막으로 가장 중요한 사항은 수학적으로 볼 때 OIIO가 텍스처 필터링을 수행하기 전에 색상 보정이 렌더러의 텍스처 데이터 검색 단계에서 이루어져야만 한다는 것입니다.

AOV 자습서

아래 자습서는 다양한 AOV를 사용하는 방법을 설명합니다.

- [Ai Write Color - 별도의 렌더링 AOV 생성](#)
- [Cryptomatte](#)
- [Motion Vector 쉐이더가 있는 사용자 정의 Motion Vector AOV](#)
- [Light Path Expression 소개](#)
- [Motion Vector AOV](#)
- [사용자 정의 AOV를 사용하여 매트 생성](#)
- [Z Depth AOV](#)

Ai Write Color - 별도의 렌더링 AOV 생성

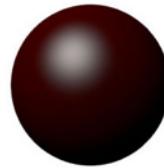
MtoA의 AOV를 사용할 뿐만 아니라, **Ai Write Color** 노드를 사용하여 사용자 정의 AOV를 생성 할 수도 있습니다. 이것은 MtoA의 내부 AOV와는 별도로 여러 개의 사용자 정의 렌더링 AOV를 작성하려는 경우 유용한 해결 방법입니다.

이 예에서는 두 개의 **Ai Write Color** 노드를 사용하여 동일한 세이더에 연결된 두 개의 다른 렌더링 AOV를 작성합니다.

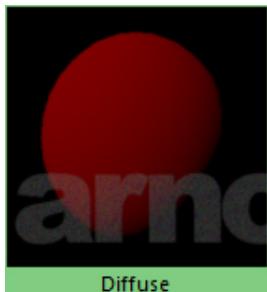
장면 파일은 여기에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/AFMUG/Ai+Write+Color+-+Creating+Separate+Render+AOVs>

- 먼저 구체와 빛을 만듭니다. Standard Surface 쉐이더를 구체에 할당합니다.



- Standard Surface 쉐이더를 생성하고 이름을 **Diffuse**로 변경합니다.
- 다른 Standard Surface 쉐이더를 생성하고 이름을 **Specular**로 변경합니다. **Specular Weight**를 1, **Base Weight**를 0으로 줍니다.



Diffuse 0.7

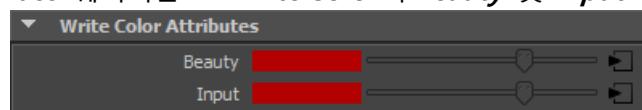


Specular 1



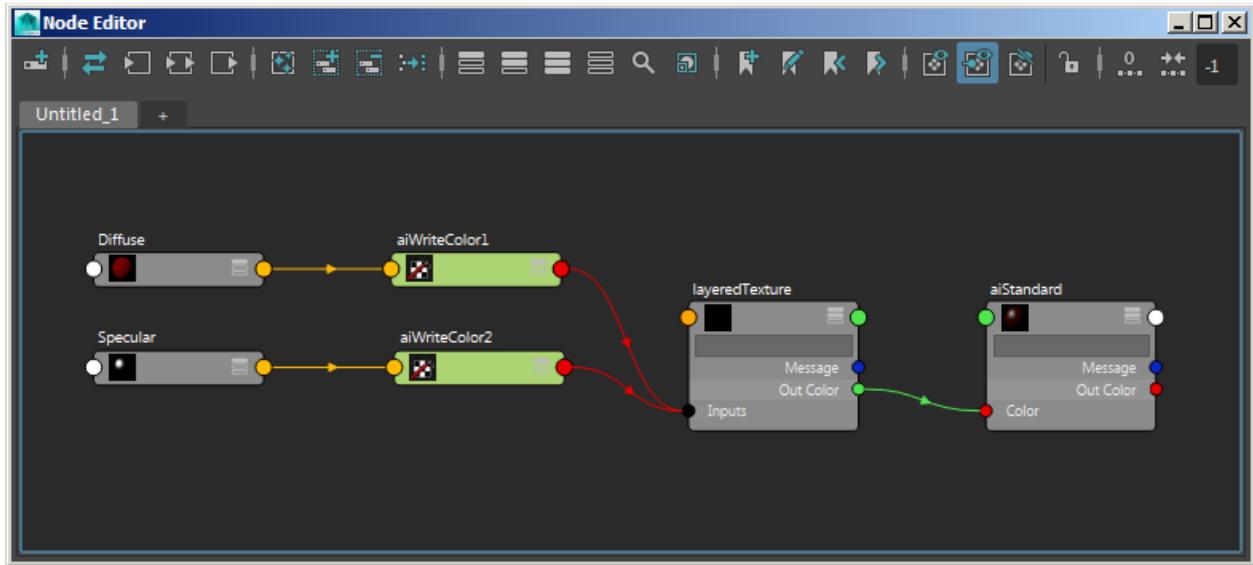
Beauty Ai Standard 쉐이더

- 두 개의 **Ai Write Color** 노드를 생성합니다. 하나는 'Diffuse' Standard Surface 쉐이더용이고 다른 하나는 'Specular' Standard Surface 쉐이더용입니다.
- 'Specular' Standard Surface 쉐이더를 **Ai Write Color**의 **Beauty** 및 **Input** 속성에 연결합니다.

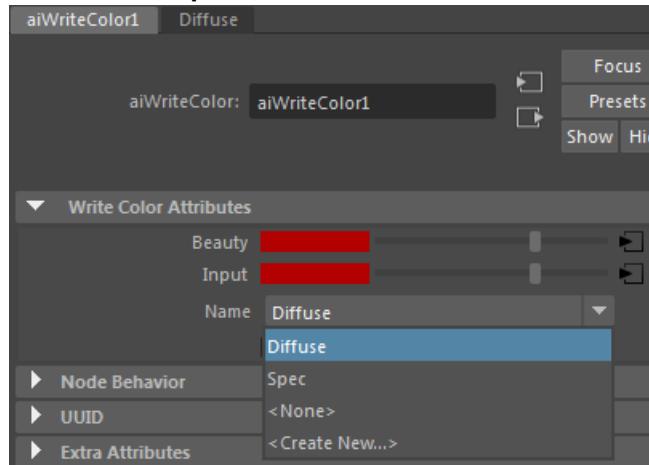


- Ai Write Color 노드 두 개 모두를 **Layered Texture**에 연결합니다. **Layered Texture**를 구체에 할당된 Standard Surface 쉐이더의 **Color** 속성에 연결합니다.

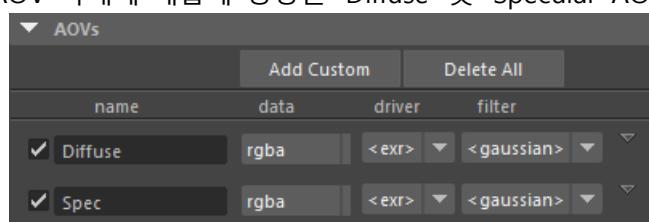
쉐이더 네트워크는 다음과 같은 모습입니다.



- Write Color Attributes: Diffuse 및 Specular에 새로운 이름을 지정합니다.



- Render Setting 창에 AOV 아래에 새롭게 생성된 'Diffuse' 및 'Specular' AOV가 나타납니다.



- 장면을 렌더링합니다. beauty 렌더 이외에 Diffuse 및 Specular AOV도 표시됩니다.



Beauty

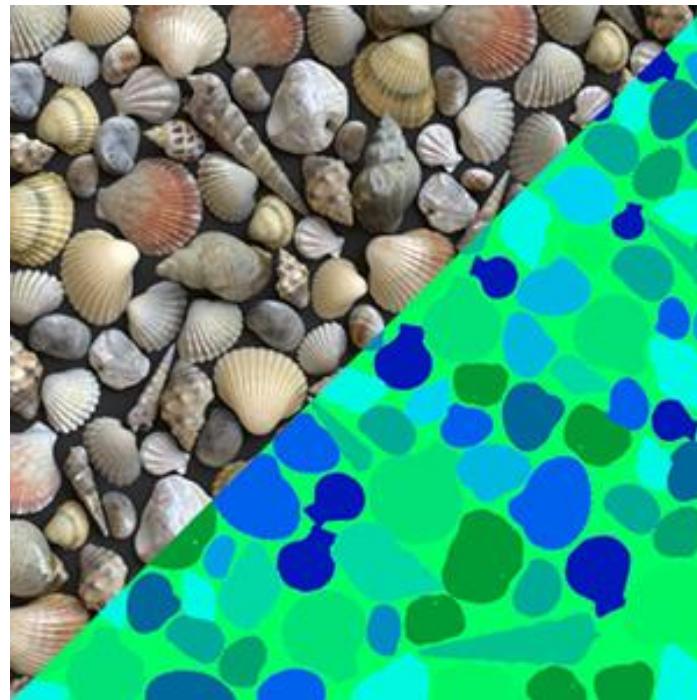


Specular AOV



Diffuse AOV

Cryptomatte



Beauty AOV/Cryptomatte 재료 AOV

Arnold는 **Psyop**에서 만든 ID 매트 생성 도구인 Cryptomatte를 지원합니다. 모션 블러, 투명성, 피사계 심도의 지원과 함께 ID 매트를 자동으로 생성합니다. 이를, 오브젝트 네임스페이스, 재료 이름을 사용하여 장면에서 ID 매트를 정리할 수 있습니다.

워크플로우

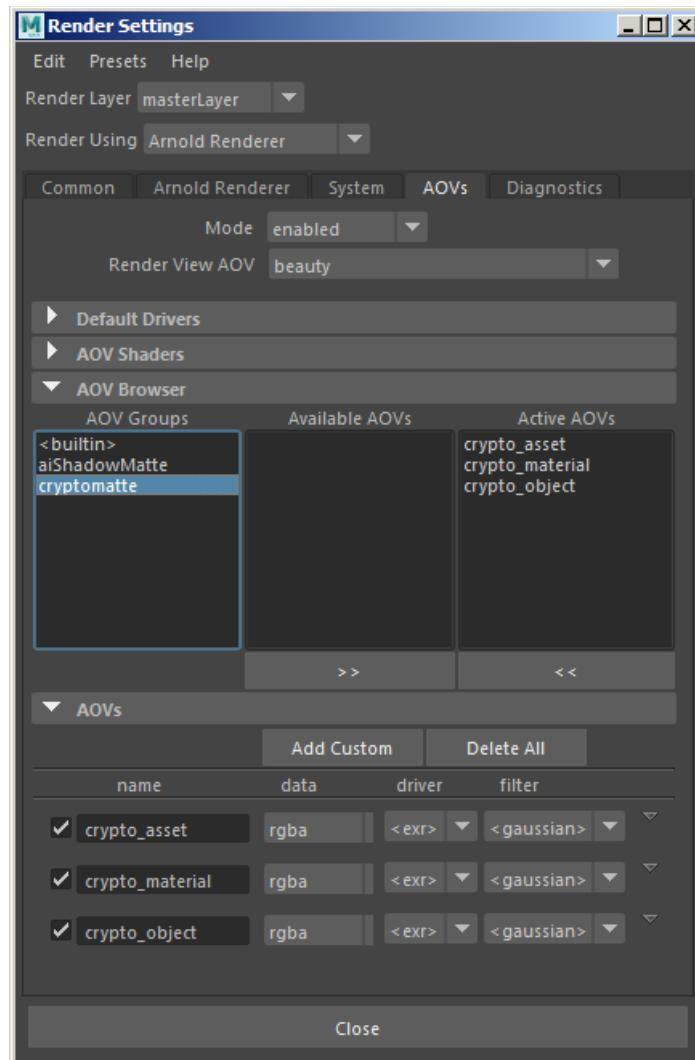
- 먼저 Cryptomatte 쉐이더를 다운로드해야 합니다.
(<https://github.com/anderslanglands/alShaders2/releases>)
- 이들을 MtoA 플러그인의 쉐이더 폴더에 복사합니다. 일반적으로 경로는 다음과 같습니다.

C:\solidangle\mtoadeploy\2017\shaders\cryptomatte.dll

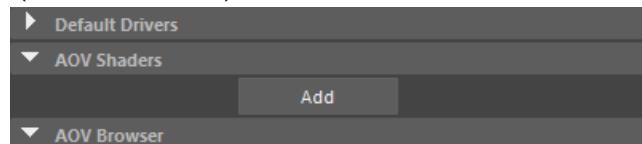
C:\solidangle\mtoadeploy\2017\shaders\cryptomatte.mtd

그러면 **Render Settings** 창의 **AOV** 탭에 다음 cryptomatte AOV가 나타납니다.

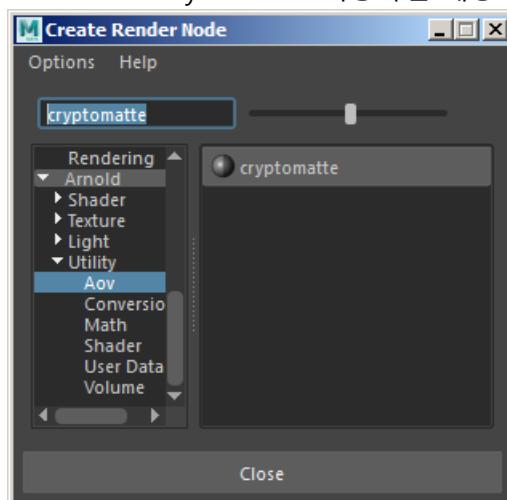
- crypto_asset**은 모든 자산에 대해 동일한 매트를 생성합니다.
- crypto_material**은 할당된 쉐이더를 기반으로 매트를 생성합니다.
- crypto_object**는 오브젝트 이름을 기반으로 매트를 생성합니다.
- Render Settings 창의 AOV 탭에 있는 AOV Browser에 필요한 cryptomatte AOV를 생성합니다.



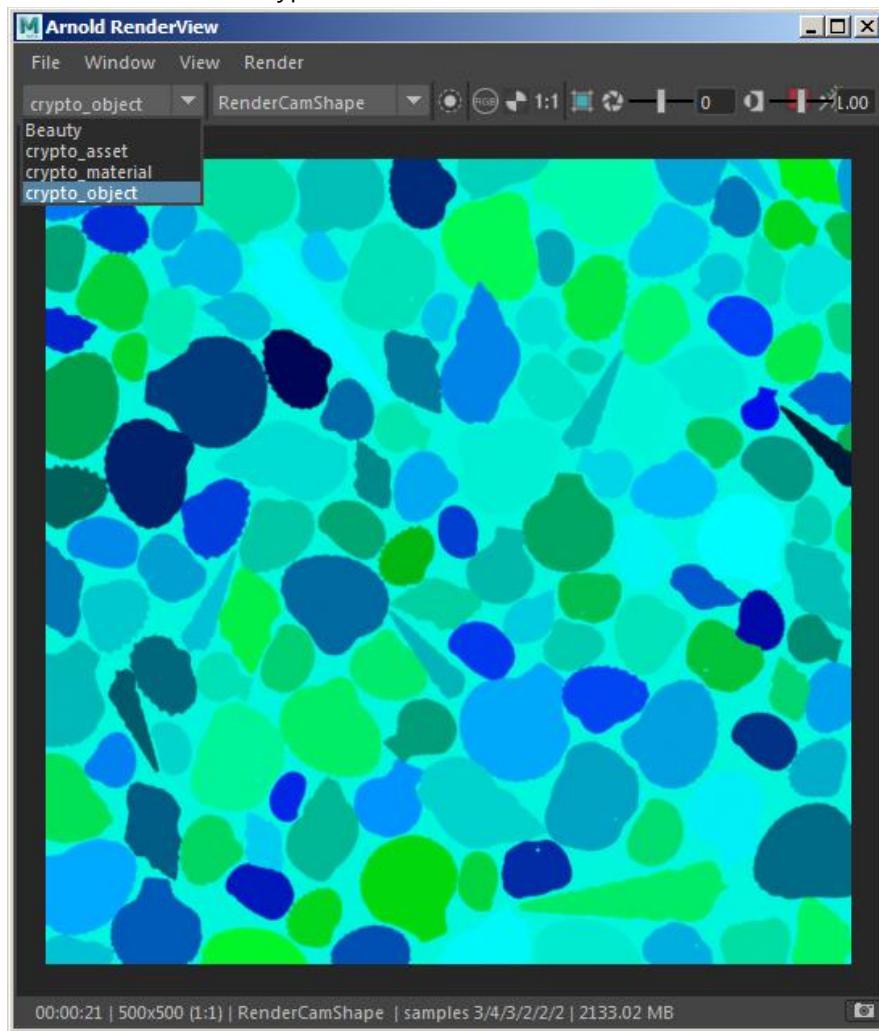
- AOV Shaders 부분에서(AOV Browser 위) **Add**를 선택합니다.



- Create Render Node 창에서 Arnold->Utility->AOV로 이동하면 해당 cryptomatte 쉐이더가 보입니다.



- 장면을 렌더링합니다. 이제 해당 cryptomatte AOV가 **Arnold RenderView** 창에 나타납니다.



Motion Vector 쉐이더가 있는 사용자 정의 Motion Vector AOV

모션 벡터 패스를 생성하는 또 다른 방법은 사용자 정의 AOV를 사용하고 Ai Motion Vector 쉐이더를 기본 'Shader' 속성에 할당하는 것입니다.

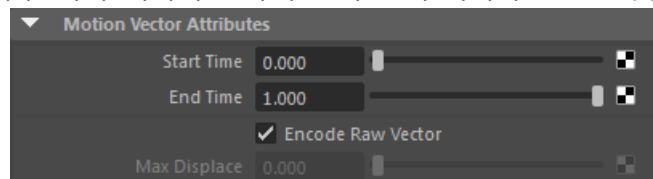
장면 파일은 여기에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Custom+Motion+Vector+AOV+with+Motion+Vector+shader>

- 먼저 Render Settings 창에서 AOV 탭을 선택합니다. **New AOV**를 추가하려면 **Add Custom**을 클릭합니다. 이 경우 우리는 '**mblur**'라고 이름을 지정했습니다.

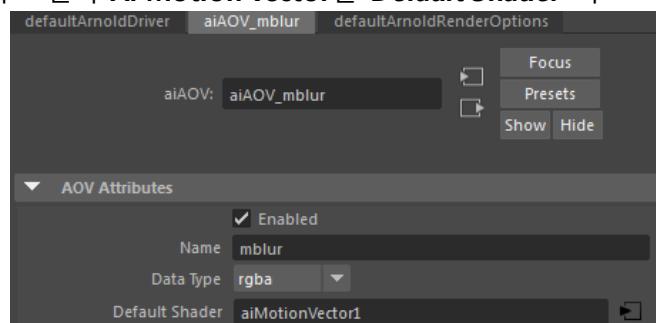
내장된 'motionvector' AOV와 충돌할 수도 있으므로 Custom AOV 'motionvector'로 이름을 지정하지 마십시오.

- Ai Motion Vector 쉐이더를 생성합니다. 벡터가 노멀화 되지 않도록 **Encode Raw Vector**를 활성화합니다. 모션 벡터 렌더링을 사후 처리 패키지로 가져올 때 절대 벡터가 필요할 것입니다.



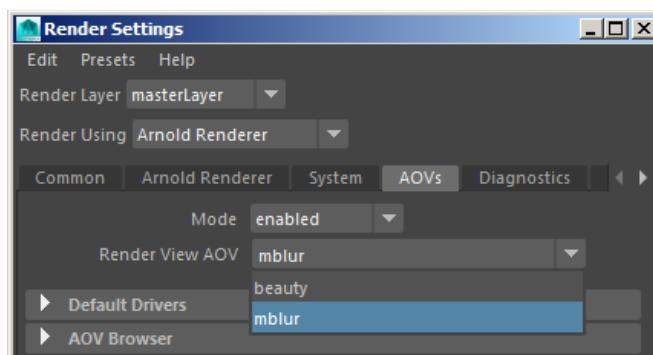
Ai Motion Vector 쉐이더. 'Encode Raw Vector' 활성화됨.

- 마우스 중간 버튼을 누르면서 Ai Motion Vector를 **Default Shader** 텍스트로 끌어옵니다.



마우스 중간 버튼을 누르면서 Ai Motion Vector를 Default Shader 텍스트로 끌어오기

- 렌더 뷰에서 사용자 정의 AOV **mblur**를 보려면 먼저 **Render View AOV** 드롭다운 메뉴를 선택해야 합니다.

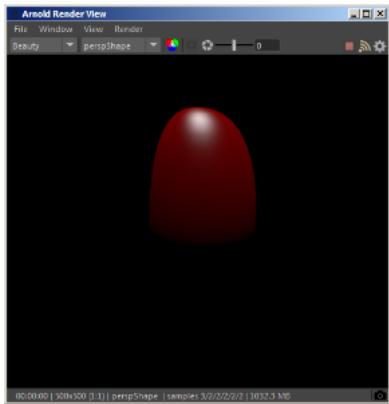


- 'Motion Blur' 설정에서 **shutter_start** 및 **shutter_end** 속성을 동일한 값으로 설정합니다. Motion blur Position이 **Start on Frame**에서 설정되면 양쪽 모두를 0으로 설정해야 합니다. **Center on Frame**인 경우에는 0.5로 설정합니다. **End on Frame**으로 설정된 경우에는 1.0으로 설정합니다.

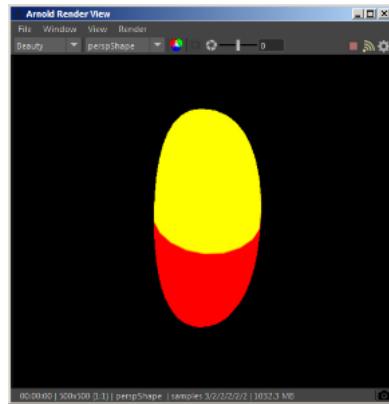
Motion Vectors에 대한 자세한 정보는 여기에 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Custom+Motion+Vector+AOV+with+Motion+Vector+shader>

- 장면을 렌더링합니다. RenderView 창에서 사용자 정의 AOV **mblur**가 렌더링됩니다.



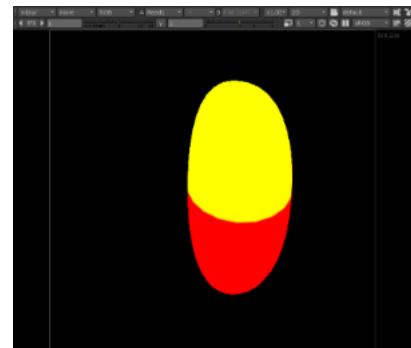
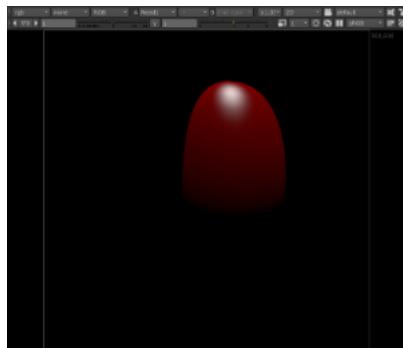
Beauty AOV



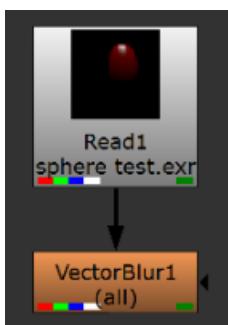
사용자 정의 Motion Vector AOV

Motion Vector 합성하기

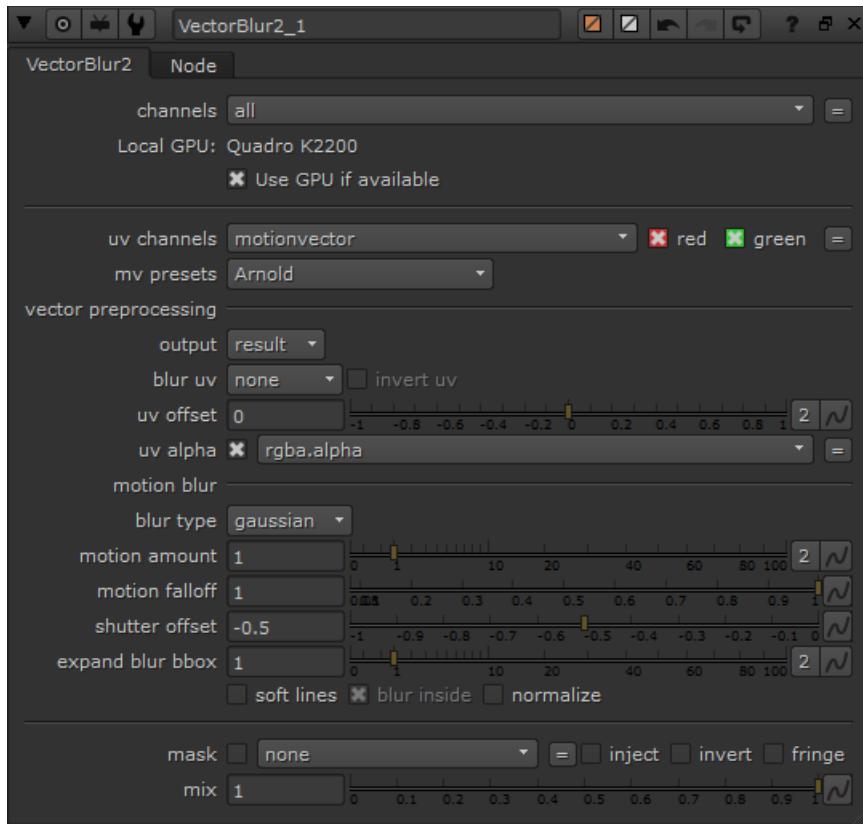
- Nuke와 같은 합성 패키지에서 렌더링한 EXR 파일을 엽니다. RGB와 모션 벡터 모두를 볼 수 있어야 합니다.



- EXR 렌더에 **VectorBlur**를 적용합니다.

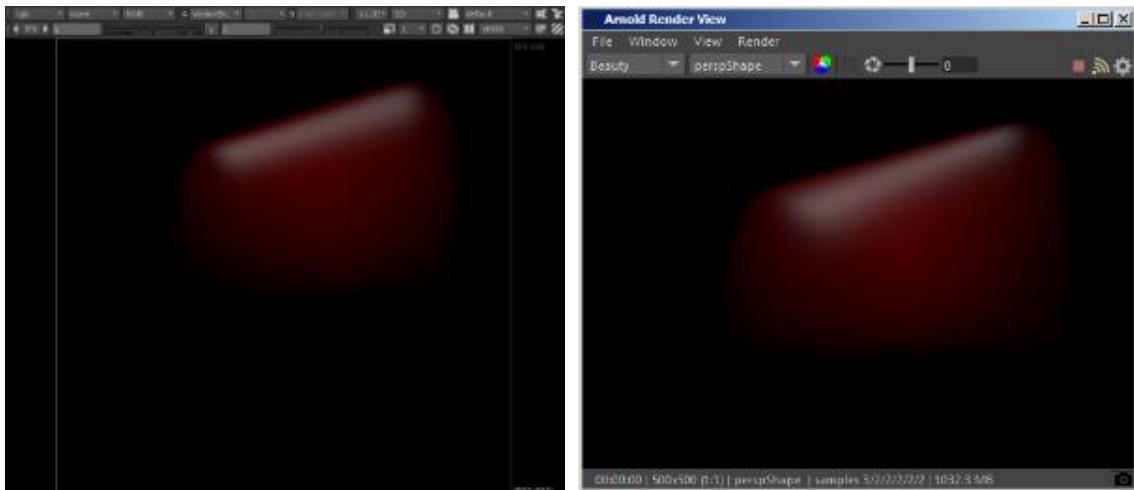


- Vector Blur 노드를 선택합니다. 'uv channels'를 'motionvector'로, 'mv presets'를 'Arnold'로 변경합니다. 'uv alpha'를 활성화하고 'rgba.alpha'를 선택하여 블러 효과가 Motion Vector AOV의 알파에 의해 잘리지 않도록 방지하십시오.



Nuke의 Vector Blur 노드 설정

최종 출력은 다음 왼쪽 이미지와 같습니다. true motion blur로 렌더링한 동일한 장면과 비교해서 매우 근접하게 보입니다. 하지만 2d 모션 벡터의 렌더링 한계 때문에 미세한 차이점이 존재합니다.



Light Path Expression 소개



Specular LPE AOV

LPE(Light Path Expression)는 빛을 특정 AOV로 출력하는 데 유용합니다. LPE는 장면을 통해 빛의 이동을 설명합니다. 광원에서 시작하여 물체들 사이로 튀어 오르고 마침내 카메라에서 끝납니다. LPE는 Arnold에 있는 특정 조명을 다양한 방법을 사용해 별도의 내장형 또는 맞춤형 AOV로 추출하는 데 사용할 수 있으며 이는 합성 패키지로 출력 및 재조합이 가능합니다.

먼저 'Light Path Expression AOV'에 익숙해지는 것이 좋습니다. 전체 목록은 여기에 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Introduction+to+Light+Path+Expressions>

최종 장면은 [Learning Scenes](#) 페이지에 있습니다.

내장 LPE AOV

LPE는 특히 아티스트 친화적이지 않으므로 Arnold에 LPE AOV 목록이 내장되어 있습니다. 다음은 이 장면에서 사용된 내장 LPE AOV의 몇 가지 예입니다.



Beauty: C.*



Diffuse Albedo: C<RD>A



Diffuse Direct: C<RD>L



Diffuse Indirect:
C<RD>[DSVOB].*



Diffuse: C<RD>.*



Direct: C[DSV]L



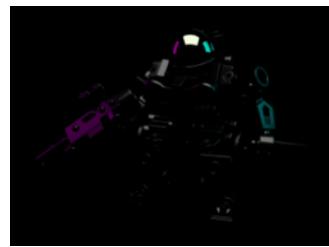
Emission: C[LO]



Indirect: C[DSV][DSVOB].*



Specular Direct: C<RS>L

Specular Indirect:
C<RS>[DSVOB].*

Specular: C<RS>.*



SSS Albedo: C<TD>A



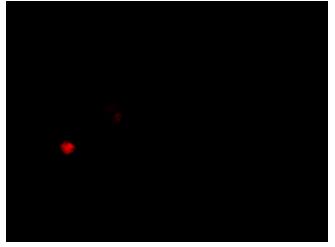
Direct: C[DSV]L



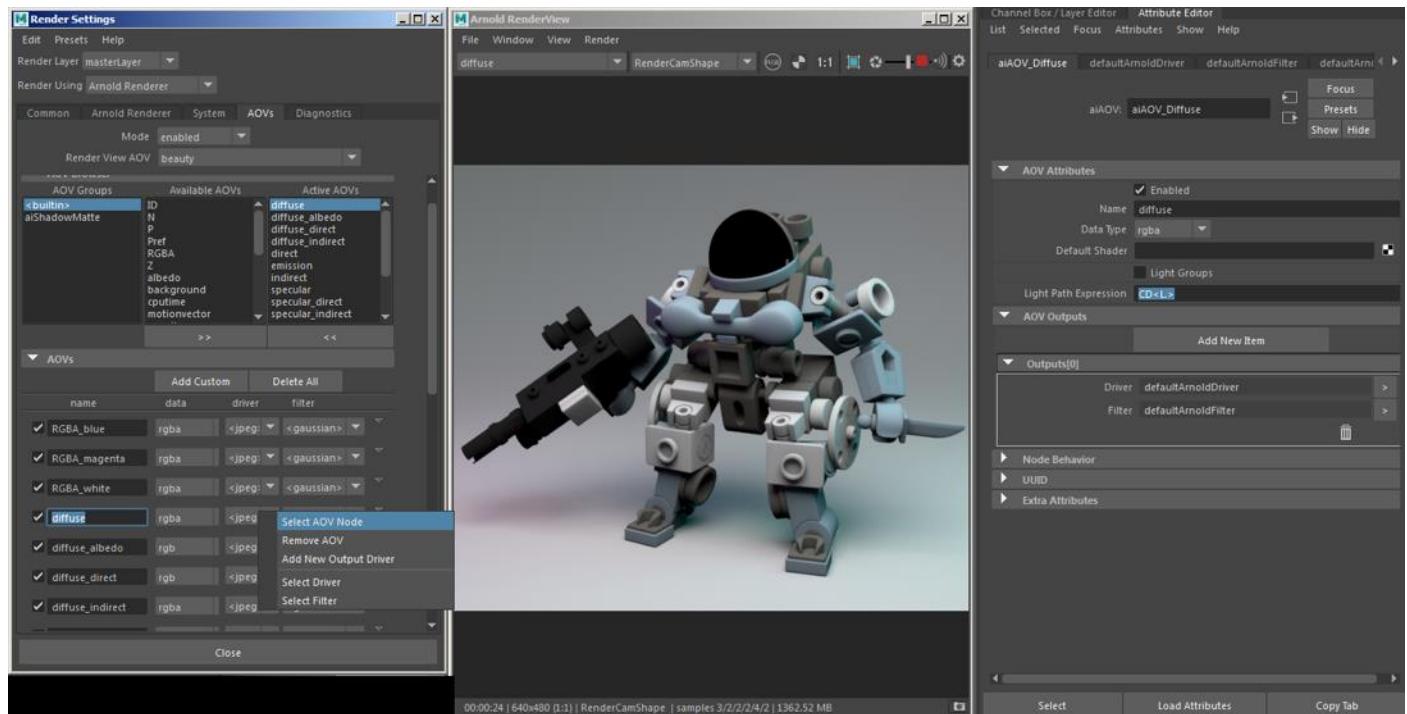
Indirect: C[DSV][DSVOB].*



SSS: C<TD>.*



Transmission: C<TS>.*



AOV 노드에 입력되고 Arnold RenderView에서 조회되는 LPE (diffuse) 'CD<L.>'

사용자 정의 AOV

내장된 AOV는 대부분의 작업에 적합하지만, 사용자 정의 AOV는 LPE를 사용할 때보다 더 많은 제어 기능을 제공할 수 있습니다. 예를 들어 장면을 렌더링할 때 사용할 LPE 조합을 선택할 수 있습니다.

예

아래 예에서 C[ST].*[LO]는 Specular 및 Transmission [ST]를 렌더링합니다. 이 경우에는 Diffuse (D)가 사용되지 않아 diffuse 구성요소만 가지고 있으므로 배경이 렌더링되지 않습니다.



Specular, Transmission: C[ST].*[LO]



Transmission: C[T].*[LO]

다음은 Custom AOV와 함께 사용되는 LPE의 추가적인 예입니다.



Direct + Specular + Volume Scattering: C[DSV]L



Diffuse: CD.*L



Emission: C[LO]



Emission + Emission Indirect: C.*O

Light Group 및 LPE

또한 빛 AOV별로 출력할 수도 있습니다. AOV를 합성할 때는 light group을 LPE로 합성하는 것이 강력한 도구일 수 있습니다. 기본적으로 LPE는 장면에서 모든 빛을 사용합니다. Light Group을 사용하여 단일 또는 다중 조명에서 다중 조명 기여를 제한할 수 있습니다.

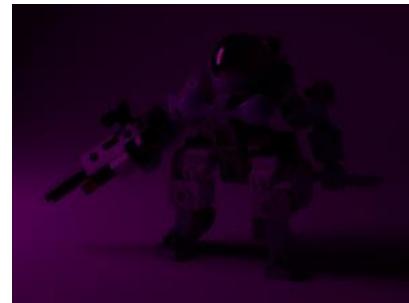
- 이 경우 우리는 각 조명에 다음 **AOV Light Group** 이름을 사용했습니다. key light (white), rim light1 (magenta), rim light2 (blue).
- 또한 "RGBA_" 접두사가 있는 사용자 정의 AOV를 추가해야 합니다. 예를 들어, 파란색 림 라이트에 대한

라이트 그룹 이름은 "RGBA_blue"여야 합니다.

이 예에서는 RGBA를 사용을 했으며 다른 LPE를 사용할 수도 있습니다.



Key light AOV: 'RGBA_white'



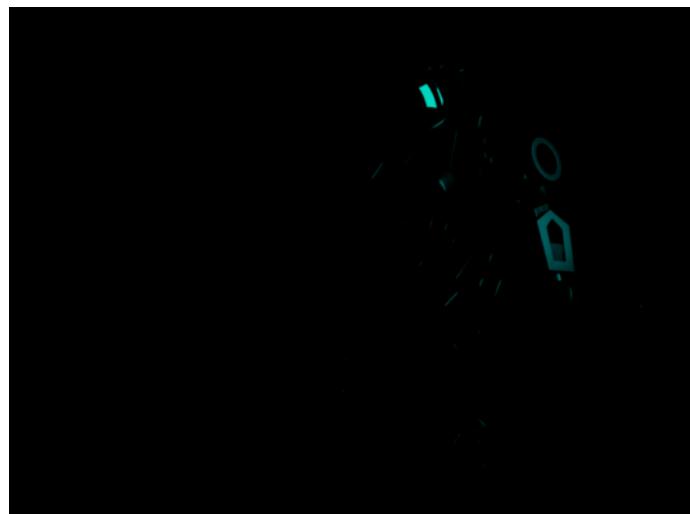
Rim light1 AOV: 'RGBA_magenta'



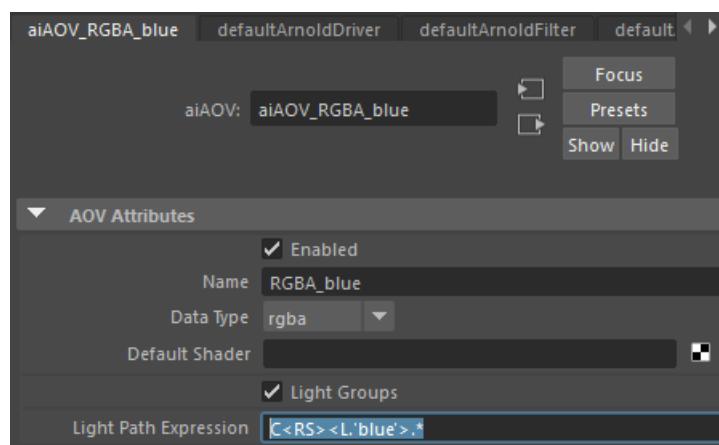
Rim light2 AOV: 'RGBA_blue'

- 이제 <L.'**groupname**'> 구문을 통해 다음 표현식 C<RS><L.'**blue**'>.*를 사용하여 블루 라이트로부터 정반사 기여를 격리할 수 있습니다.*

AOV 라이트 그룹의 경우에는 L이 아닌 <L.>을 사용해야 합니다.



블루 림 라이트로부터 Specular 반사: C<RS><L.'blue'>.*



라이트별 AOV에 대한 자세한 정보는 AOV(<https://support.solidangle.com/display/A5AFMUG/AOVs>) 및 라이트(<https://support.solidangle.com/display/A5AFMUG/Lights>) 페이지에 나와 있습니다.

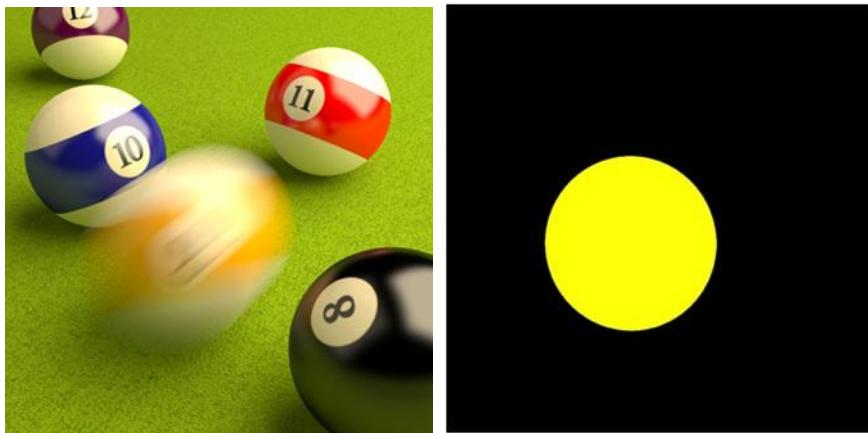
사용자 정의 쉐이더

표현을 특정 쉐이더 파라미터로 제한할 수 있습니다. 아래 예에서는 코팅 없이 장면을 렌더링하기 위해 **C<RS[^'coat']>.***가 사용되었습니다. 따라서 Coat가 1일 경우 렌더링된 이미지에 헬멧 쉐이더가 분명하지 않습니다. 하지만 Coat가 1보다 작을 때는 specular (**RS**)가 헬멧에 표시됩니다.



Coat 값이 1일 경우에는 헬멧이 보이지 않습니다

Motion Vector AOV



모션 벡터 AOV를 사용한 모션 블러

이 자습서에서는 당구공 장면에서 Motion Vector AOV를 사용하는 방법을 보여줍니다. 진정한 3D 모션 블러와 비교해서 이 기술을 사용할 때의 장단점에 대해 알아볼 것입니다. Motion Vector AOV는 장면 내에서 물체의 움직임을 나타내는 컬러 채널을 출력합니다. 이 AOV는 후 처리 소프트웨어에서 2D 모션 블러 효과를 계산하는 데 사용할 수 있습니다. 장점은 실제 3D 모션 블러와 비교해서 렌더링이 더 빠르다는 것입니다.

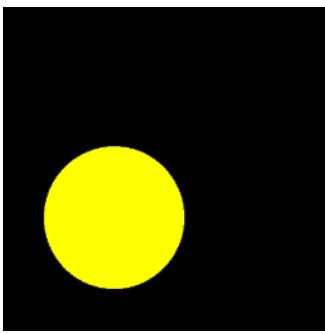
장면 파일은 여기에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Motion+Vector+AOV>

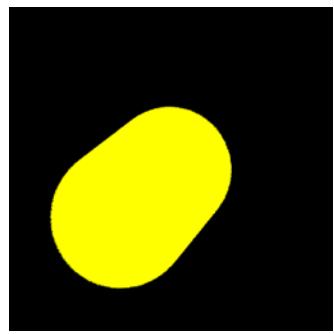
당구공 레이어 만들기

Motion Vector AOV를 렌더링하기 전에 먼저 풀 공을 배경과 분리해야 합니다. 당구공(Motion Vector AOV 포함)을 별도로 렌더링하고 누크(Nuke)와 같은 후 처리 프로그램에서 이들을 합성합니다.

- 분리된 당구공 층을 렌더링하기 전에 카메라의 순간 셔터를 설정해야 합니다. 그 이유는 렌더링에서는 모션 블러를 원하지 않지만 모션 벡터 AOV에서는 모션 속도 정보를 원하기 때문입니다. 이를 위해서는 해당 카메라를 선택하고 Attribute Editor의 Arnold 부분으로 이동합니다. 여기서 **shutter_start** 및 **shutter_end** 속성 값을 설정할 수 있습니다. 전역 렌더러 설정에서 모션 블러의 위치 오프셋(전체 장면의 셔터 시간 간격을 지정함)에 따라 즉석 값이 달라집니다. 기본적으로 위치는 Center on Frame으로 설정됩니다. 이것은 **shutter_start**의 값 및 0.5의 **shutter_end** 값을 의미합니다. Start on Frame을 설정하면 **shutter_start** 및 **shutter_end** 값이 0.0이 되어야 합니다. End of Frame을 설정하면 값이 1.0이 되어야 합니다.

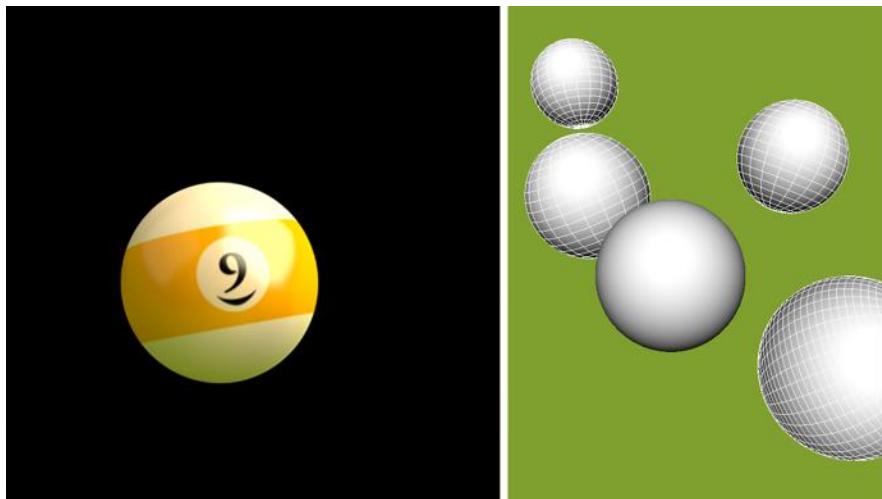


셔터 시작/끝이 0으로 설정됨(맞음)



셔터 시작/끝이 다른 값으로 설정됨(틀림)

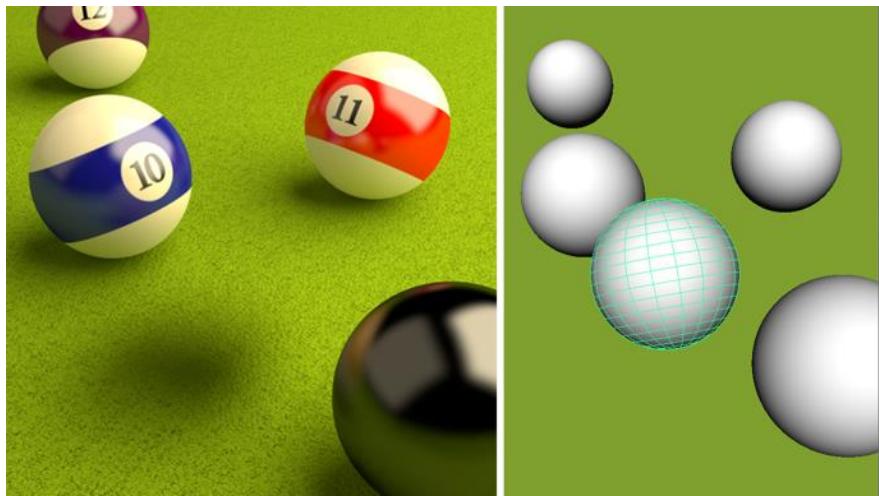
- 당구공을 위한 렌더링 층을 생성합니다. 장면에서 지오메트리를 선택하고(모션 블러를 적용하려는 당구공 제외) 속성 편집기에서 'Primary Visibility'를 비활성화합니다. 층을 렌더링할 때는 아래 이미지에서처럼 자체적으로 당구공을 가져와야 합니다.



당구공 렌더(나머지 장면에 대해 1차 가시성이 비활성화됨)

배경 레이어 만들기

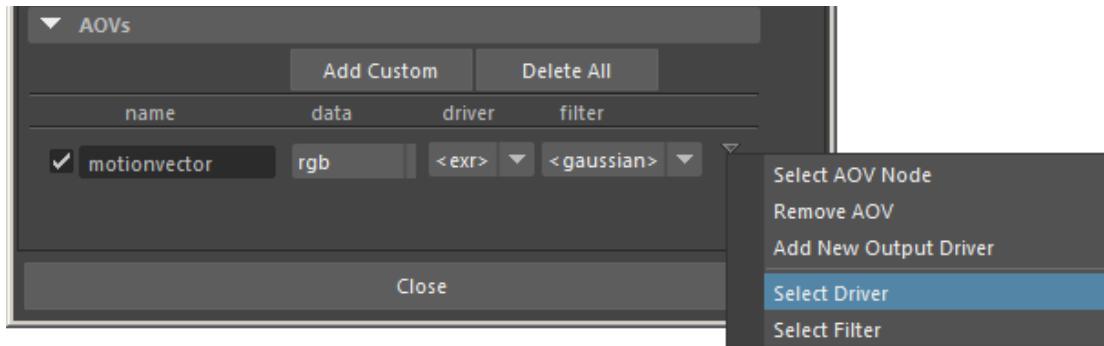
- 배경에 대해 새로운 렌더링 층을 만듭니다. 렌더링 층에 지오메트리를 할당하고 모션 블러를 적용하려는 풀 공에 대한 1차 가시성을 비활성화합니다. 렌더링되면 배경 오브젝트와 모션 블러가 적용된 당구공의 그림자만 볼 수 있습니다.



배경 렌더링(당구공에 대해 비활성화된 1차 가시성)

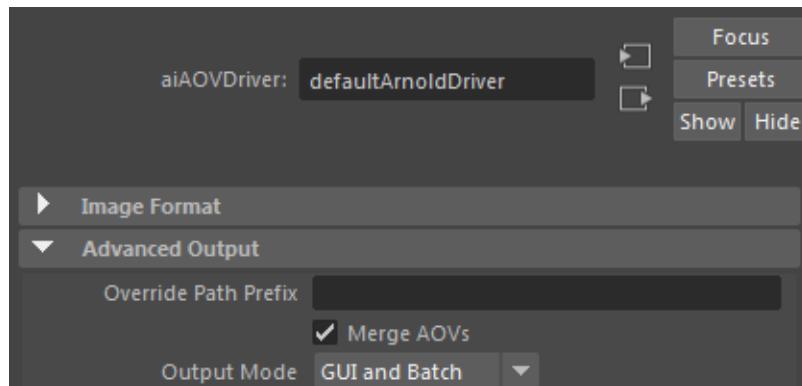
Motion Vector AOV 추가하기

- 당구공 렌더링 층이 활성화된 상태로 Render Settings 창에서 AOV 탭을 선택합니다. 'AOV Browser' 아래에서 '<builtin>'을 선택한 다음 Motion Vector AOV를 생성합니다. AOV 필터가 가장 가까운 항목에 설정되어 있어야 합니다. Active AOV에 추가된 내장 Motion Vector AOV
- AOV 아래에서 새롭게 생성된 Motion Vector AOV의 오른쪽에 아래쪽 화살표가 있습니다. 이 화살표를 클릭하고 메뉴에서 'Select Driver'를 선택합니다.



Motion Vector의 오른쪽에 있는 아래쪽 화살표를 클릭하여 드라이버 선택

- Arnold 드라이버가 속성 편집기에 나타나야 합니다. 'Merge AOVs'를 선택하면 Motion Vector AOV와 렌더링된 EXR이 하나의 파일로 병합됩니다.



'Merge AOVs'를 활성화하면 AOV들이 하나의 EXR 파일에 저장됨

- 두 렌더링 층(배경 및 당구공)을 모두 EXR 파일로 렌더링합니다.

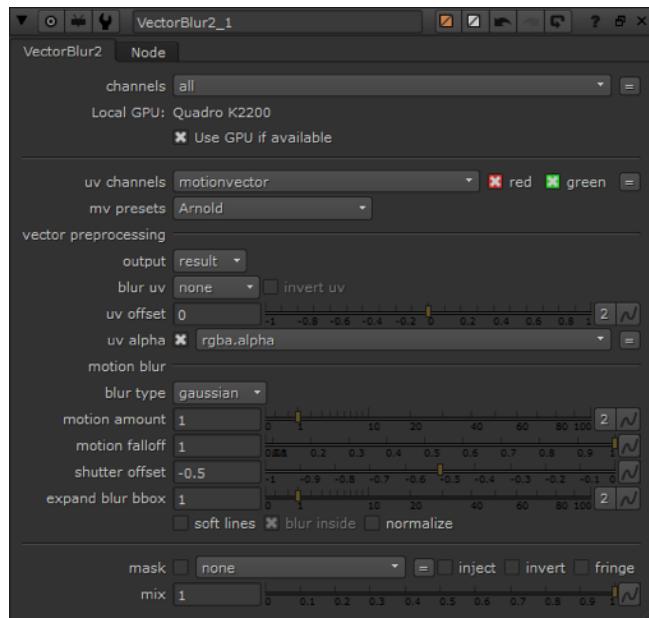
합성

- 두 개의 EXR 파일을 후 처리 패키지에서 엽니다(이 경우에는 Nuke).
- 'Vector Blur' 노드를 당구공 렌더링에 연결합니다. Vector Blur 노드를 사용하여 'Motion Vector AOV'를 사용하는 당구공을 블러 처리할 것입니다.

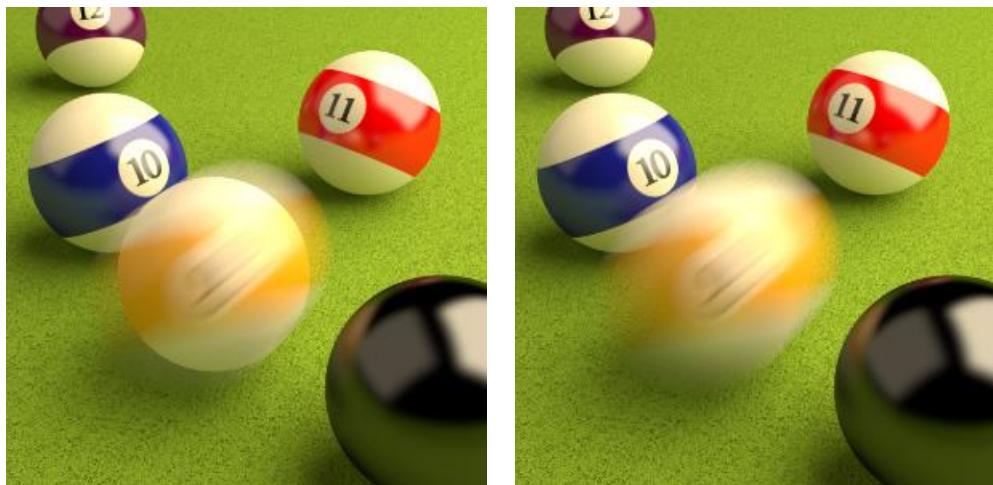


- Vector Blur 노드를 선택합니다. 'uv channels'를 'motionvector'로, 'mv presets'를 'Arnold'로 변경합니다. 'uv

alpha'를 활성화하고 'rgba.alpha'를 선택하여 블러 효과가 Motion Vector AOV의 알파에 의해 잘리지 않도록 방지하십시오.



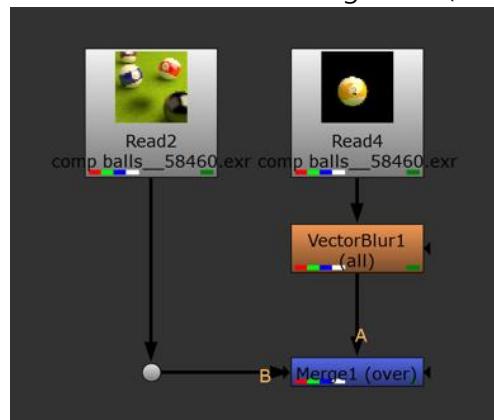
Nuke의 Vector Blur 노드 설정



VectorBlur 알파가 비활성화됨: 툴림

VectorBlur: 알파 활성화됨: 맞음

- A 입력은 Vector Blur이고, B 입력은 배경 렌더링인 'merge' 노드(작업이 'over'로 설정됨)를 사용합니다.



마지막으로 다음과 같이 당구공 렌더링이 배경 렌더링에 합성된 것을 볼 수 있습니다.



Motion Vector AOV로 렌더링된
당구공



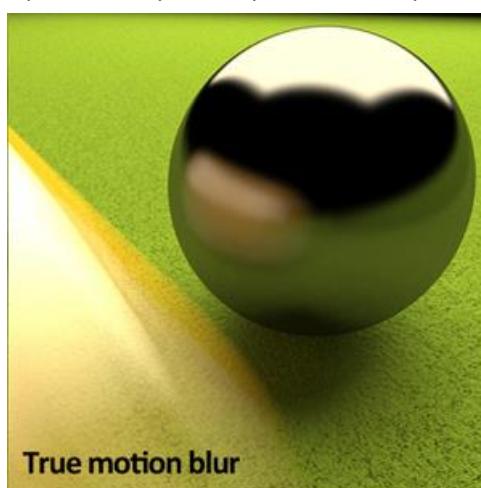
배경 렌더링



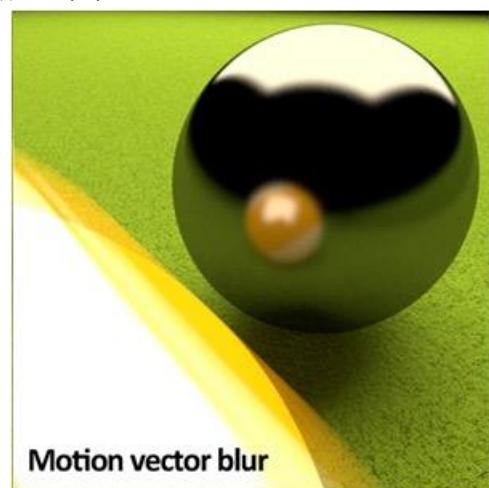
배경에 합성된 당구공

제한

Arnold의 고유한 3D 모션 블러 렌더링과 비교해서 3D 모션 벡터를 렌더링할 때는 일부 제한이 있습니다. 이러한 제한에는 반사면 내의 모션 블러, 변형 블러, 카메라를 마주보지 않는 바퀴 회전 등과 같은 비실제 3D 모션 블러 효과 및 모션 블러 광원을 사용한 타임 랩스 효과 등이 있습니다.



True motion blur



Motion vector blur

Motion Vector를 사용할 때는 반사된 모션 블러가 가능하지 않음

모션 벡터 AOV는 변형 모션 블러를 처리하는 데 문제가 있습니다. 변형 모션 블러를 위해 적절한 모션 벡터 AOV를 렌더링해야 한다면 motion_vector 쉐이더가 첨부된 사용자 정의 AOV를 생성해야 합니다. 이것의 실행 방법을 설명하는 자습서는 '[‘Motion Vector 쉐이더가 있는 사용자 정의 Motion Vector AOV’](#)에 있습니다.

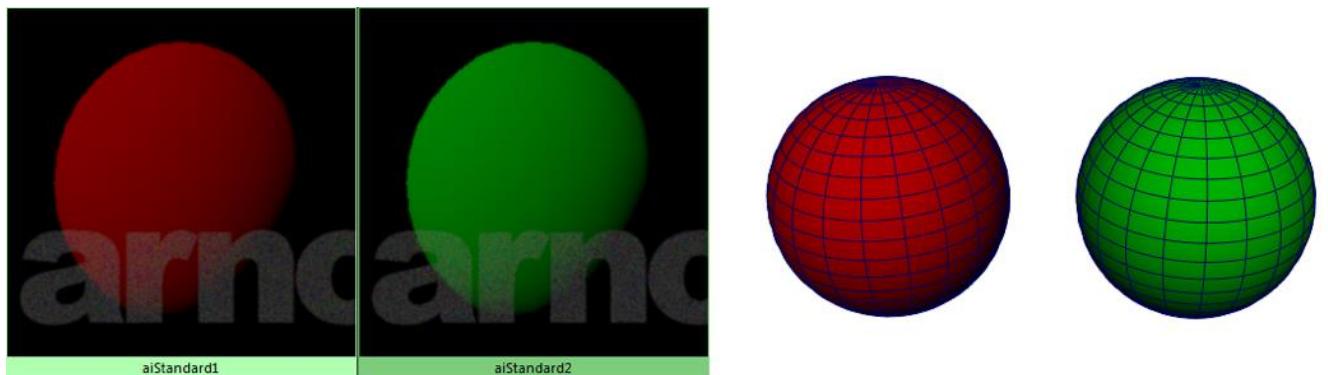
사용자 정의 AOV를 사용하여 매트 생성

이 자습서는 사용자 정의 로컬 AOV를 사용하여 매트를 생성 및 렌더링하는 매우 간단한 기술을 보여줍니다. 이것은 MtoA 내의 기존 AOV 목록에서 사용할 수 없는 사용자 정의 효과를 렌더링하려는 경우 특히 유용합니다. 이 사용자 정의 AOV는 Nuke와 같은 모든 합성 소프트웨어와 함께 사용할 수 있습니다.

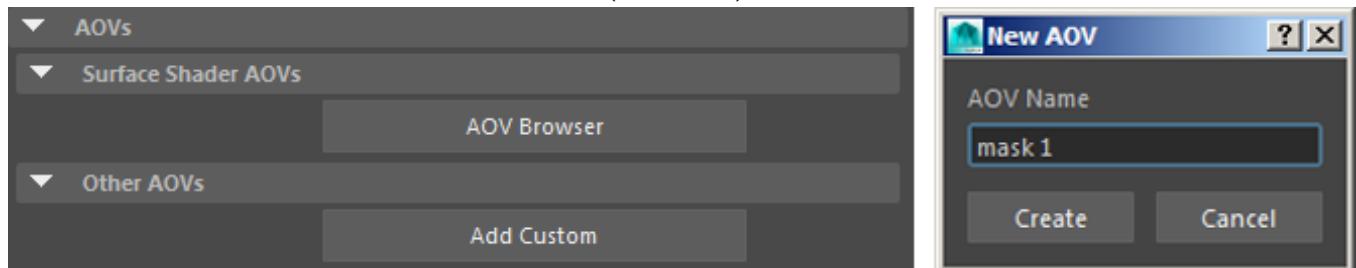
장면 파일은 여기에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/AFMUG/Using+Custom+AOVs+to+Create+Mattes>

- 먼저 두 개의 구체를 만들고 각각에 하나의 Standard Surface 쉐이더를 할당합니다. 각 Standard Surface 쉐이더에 대한 Base Color를 변경합니다. 사용자 정의 AOV를 추가하여 이러한 구체의 음영을 독립적으로 교체합니다.

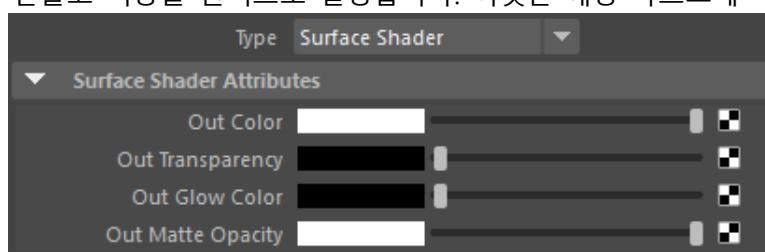


- 하나의 구체에 대해 Shading Group을 선택하고 하나의 Custom AOV를 추가합니다. 이름을 'mask1'라고 지정합니다. 이 마스크는 구체의 흰색 매트(알파 채널)를 만드는 데 사용됩니다.

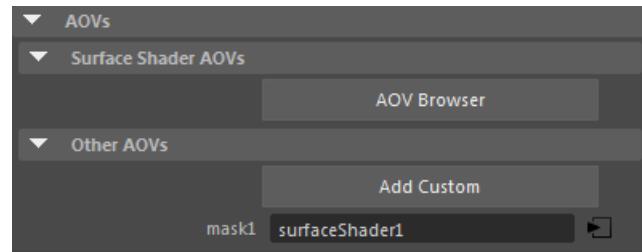


'Add Custom'을 클릭하고 새로운 AOV 'mask1'로 명명합니다.

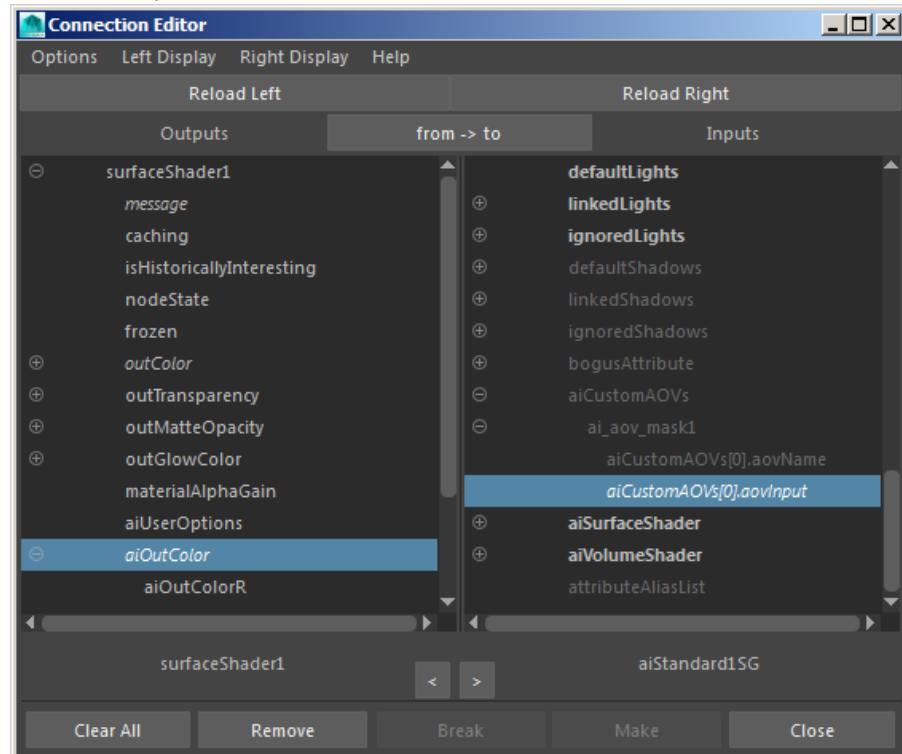
- Surface Shader를 만들고 색상을 흰색으로 설정합니다. 이것은 해당 마스크에 대해 흰색으로 사용됩니다.



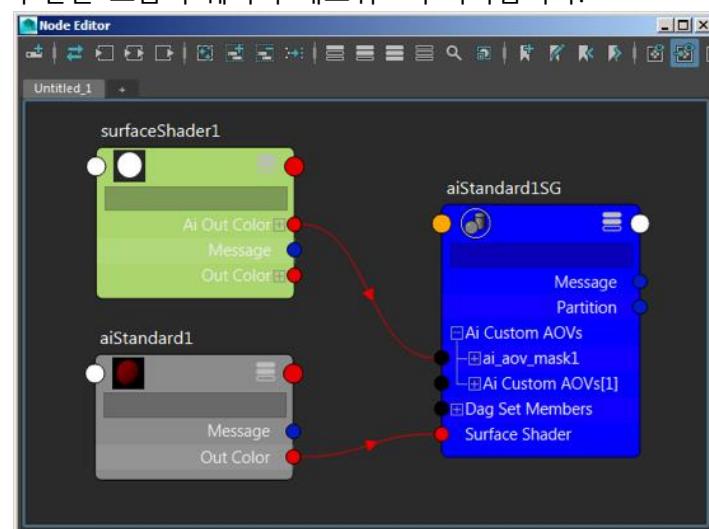
- Surface Shader를 마우스 중앙 버튼을 누른 채로 새롭게 생성된 'mask1' AOV로 끌어옵니다. 반드시 'mask1'이라는 텍스트로 끌어와야 합니다. 그렇지 않으면 연결이 적절하게 이루어지지 않습니다.



- 연결 편집기가 나타납니다. Surface Shader에 있는 'aiOutColor'를 'ai_aov_mask1'의 'aiCustomAOVs[0].aovInput'에 연결합니다.



일단 연결이 구축되면 다음과 같은 모습의 쉐이더 네트워크가 나타납니다.

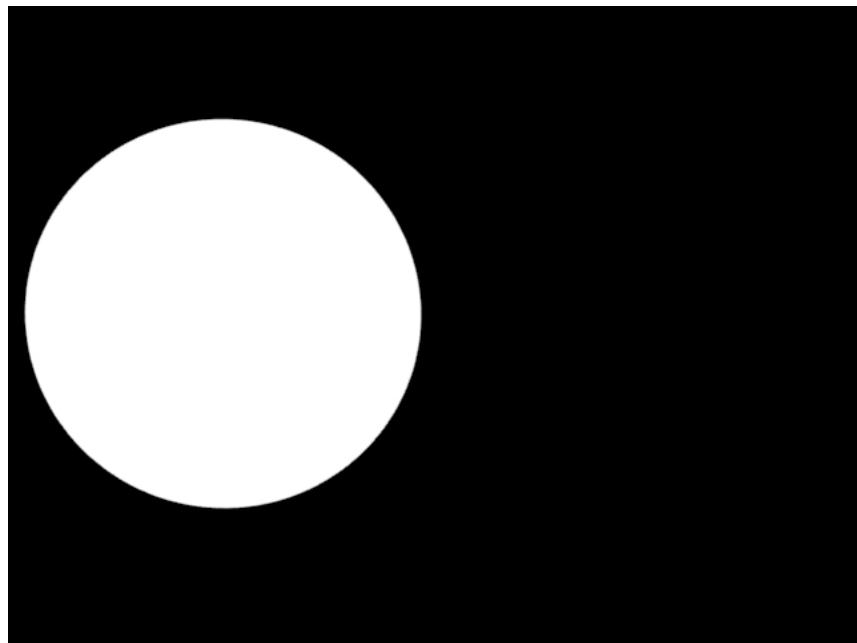


Standard Surface 쉐이더의 사용자 정의 AOV - 'mask1'에 연결된 White Surface Shader

- 다른 구체에 대해 위 절차를 반복합니다. 하지만 이번에는 블랙 Surface Shader를 생성합니다.

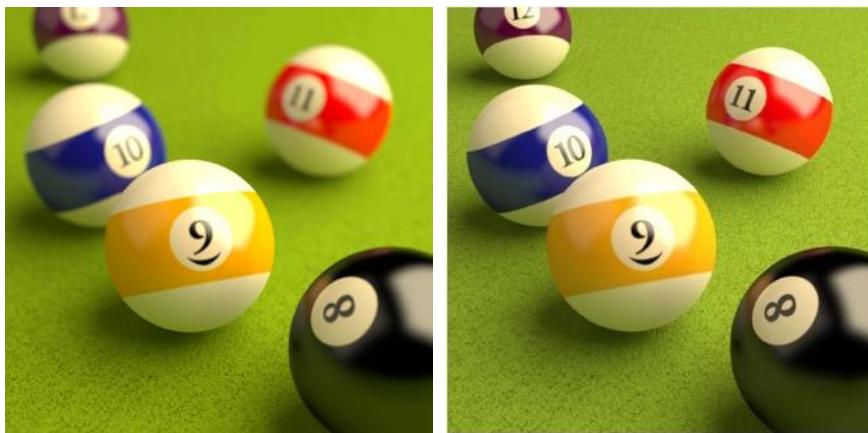
- 장면을 렌더링합니다. 사용자 정의 AOV '마스크'에 연결된 Surface Shader가 Standard Surface의 음영을 재정의한 것을 볼 수 있습니다.

완료된 렌더링은 다음과 같은 모습입니다.



적색 구체는 순백색으로, 녹색 구체는 검정색으로 렌더링됩니다.

Z Depth AOV



Z Depth AOV와 합성된 Beauty 렌더링

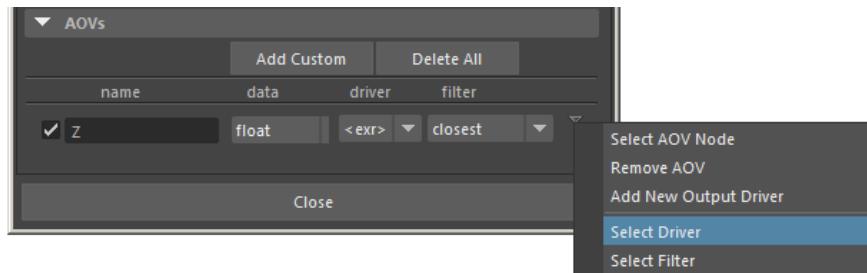
이 짧은 자습서에서는 MtoA로부터 'Z' 심도 AOV를 설정 및 렌더링하는 프로세스를 설명합니다. 이것은 현실감 있는 피사계 심도 효과를 얻기 위해 Nuke에서 뷰티 렌더링과 합성할 수 있습니다. 'Z' 심도 AOV에는 쉐이딩 점들의 깊이 정보가 포함되어 있습니다.

장면 파일은 여기에서 다운로드할 수 있습니다.

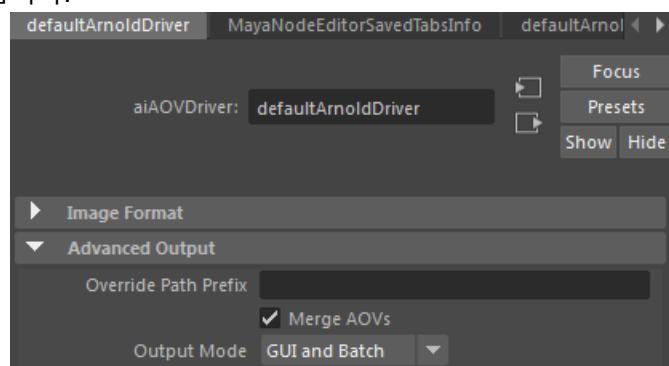
<https://support.solidangle.com/display/A5AFMUG/Z+Depth+AOV>

'Z' depth AOV를 렌더링 및 합성하려면 다음을 실행해야 합니다.

- 먼저 Render Settings 창을 열고 'AOVs' 탭으로 이동합니다. Z DEPTH AOV를 생성합니다.
- AOV 아래에서 새롭게 생성된 'Z' AOV의 오른쪽에 아래쪽 화살표가 있습니다. 이 화살표를 클릭하고 메뉴에서 'Select Driver'를 선택합니다.



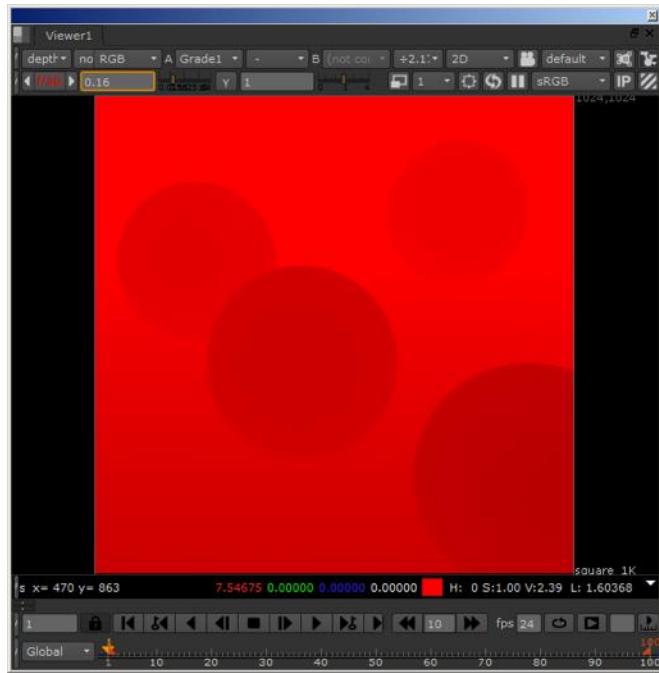
- Arnold 드라이버가 속성 편집기에 나타나야 합니다. 'Merge AOVs'를 선택하면 'Z' AOV와 렌더링된 EXR가 하나의 파일로 병합됩니다.



'Merge AOVs'를 활성화하면 AOV들이 하나의 EXR 파일에 저장됨

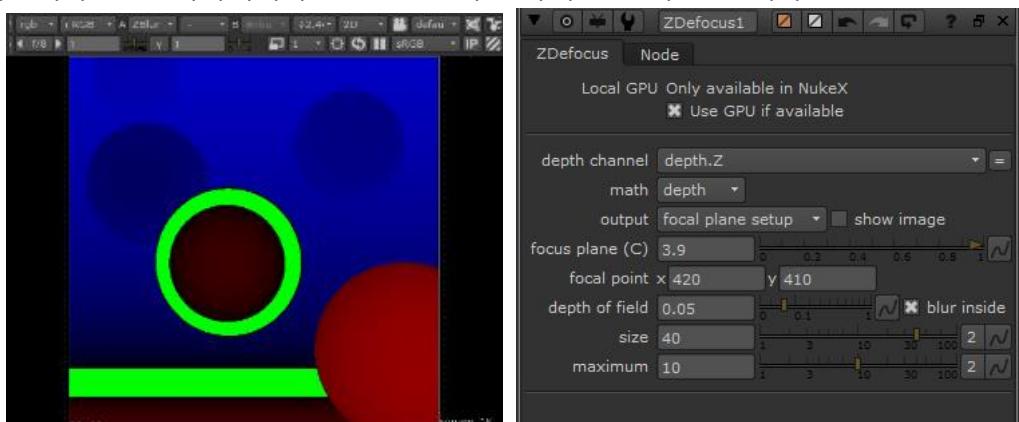
- 장면을 렌더링하고 Nuke에서 EXR 파일을 엽니다.

Nuke에서 심도 AOV를 볼 때 적색만 표시될 수 있습니다. Depth 채널은 알파 채널에서 비안티앨리어싱된, 정규화되지 않은 심도 데이터를 제공합니다. 비 표준화는 Z DEPTH 값이 0과 1 사이(카메라 앞의 거리)가 아님을 의미하므로 Z 값을 먼 클리핑 평면 값으로 나누어 0.1 값을 얻어야 합니다. 이 리매핑은 뷰어에서 심도 채널을 볼 때만 필요하며 zdefocus 노드를 작동시키는 데 필요하지 않습니다.



Nuke 내에서 확인한 Z DEPTH 채널

- 'Zblur'(Nuke 7에서 'ZDefocus')를 EXR 렌더링에 연결하고 'focal-plane setup'을 활성화합니다. Zblur 노드의 기본 설정이 작동해야 합니다. Z는 'depth.z'로 설정해야 하며, 'math'는 'depth'로 설정해야 합니다(심도 채널 내 Z 값은 카메라 앞의 거리).
- 녹색 영역이 원하는 초점 영역이 될 때까지 'focus plane'을 조절합니다. 적색은 낮은 피사계 심도의 영역을 보여줍니다. 파란색은 피사계 심도 바깥에 속하는 영역을 나타냅니다. 녹색은 피사계 심도 영역 내부를 나타냅니다. 심도 파라미터가 0으로 설정되면 녹색을 볼 수 없습니다.

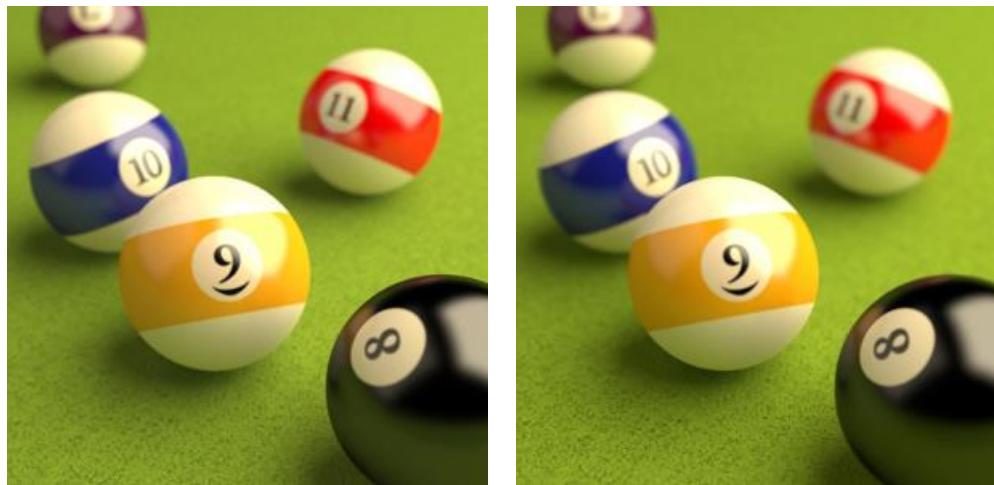


초점면 설정 활성화됨(녹색 영역이 초점 내)

- 'focus plane'이 마음에 들면 'focal-plane setup'을 비활성화합니다. 'depth-of-field'를 1로 높이면 RGB 이미

지가 'Zblur' 노드에 의해 블러 처리되는 것을 볼 수 있습니다.

이제 이 짧은 자습서를 마무리하겠습니다. Z Depth AOV를 사용해 얻을 수 있는 이점은 속도와 유연성입니다. 합성 도구에서 피사계 심도를 실시간으로 변경할 수 있다는 것이 큰 장점입니다. 진정한 피사계 심도로 렌더링하는 것은 대부분의 경우 더 좋은 결과를 가져오지만 노이즈와 관련된 문제를 해결하기 위해 많은 수의 AA 샘플을 요구할 수 있습니다. 다음은 실제 피사계 심도 렌더링과 DOF가 없는 장면의 렌더링을 비교한 것입니다(합성된 'Z' 심도 AOV 사용). 8개의 AA 샘플을 사용해도 실제 피사계 심도로 렌더링된 이미지의 심한 피사계 심도 부분에는 여전히 약간의 노이즈가 있습니다. 이 경우 실제 DOF를 사용한 렌더링은 Z Depth AOV를 사용하는 이미지 렌더링에 비해 약 두 배의 시간이 소요됩니다.

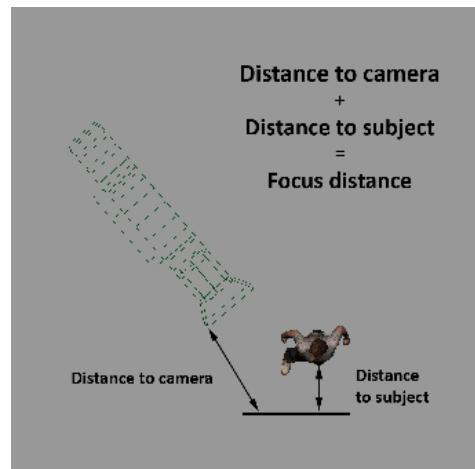
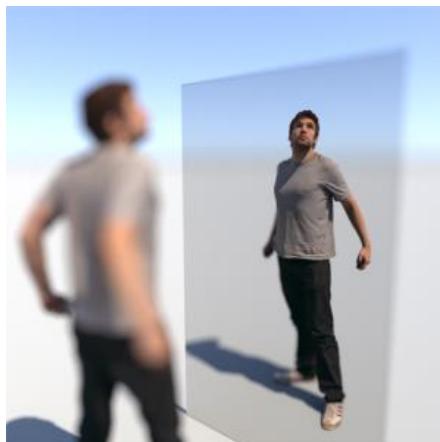


실제 DOF(8개의 AA 샘플)

Z Depth AOV를 사용하여 합성된 DOF

제한

'Z' 깊이 AOV 사용에는 제한이 있습니다. 예를 들어, 거울에 반사된 물체에 초점을 맞추는 것이 불가능합니다(아래 참조).



거울 안의 물체에 초점을 맞추는 것은 2차원 'Z' 심도 AOV(실제 피사계 심도로 렌더링된 이미지 위에)로는 불가능합니다.

Kick

다음 자습서에서는 Kick을 사용하여 **.ass** 파일을 렌더링하는 방법을 보여줍니다.

- [Kick 시작하기](#)
- [렌더 로그를 읽는 방법](#)

Kick 시작하기

소개

Arnold의 명령행 렌더러를 kick이라고 합니다. 이것은 **.ass** 파일을 읽고 Arnold를 사용하여 장면을 렌더링한 다음 이미지 파일을 출력합니다. Kick은 파라미터 및 기본 값에 대한 Arnold 노드를 조회하는 데 사용할 수 있습니다. 또한 장면 디버깅에 사용할 수도 있습니다.

Kick은 플러그인 다운로드에 포함되어 있습니다. 설치가 끝나면 Arnold 배포판의 bin 하위 디렉토리에서 찾을 수 있습니다.

Kick 실행하기

kick을 실행하려면 먼저 셸 하나를 엽니다(터미널). 디렉토리를 Arnold 배포판의 bin 하위 디렉토리로 변경하고 "kick"을 입력합니다. 다음은 Windows OS에서 예제입니다.

Kick은 현재 디렉토리에서 쉐이더와 프로시저를 항상 로드하므로 다른 DLL / SO / DYLIB가 많은 폴더에서 kick을 실행하지 마십시오. Kick은 각 DLL / SO / DYLIB를 로드하려고 시도할 것이며, 쉐이더나 프로시저를 포함하고 있는지 검사할 것입니다.

```
$ cd /solidangle/Arnold-5.0.0.0/bin
$ ./kick
Arnold 5.0.0.0 [2cfbe09c] linux clang-3.9.1 oiiio-1.7.12 osl-1.8.2 vdb-4.0.0 rlm-12.2.2 2017/04/10 16:48:44
No arguments. Try kick --help for a command summary
```

유용한 명령

다음 명령을 시도하려면 [예제 ass 파일](#) 페이지에서 **cornell.ass** 등과 같은 파일 중 하나를 다운로드하십시오. 가장 유용한 명령 중 하나는 "-h" 또는 "--help"이며 kick에서 사용 가능한 모든 옵션 목록이 나타납니다.

```
kick -h
```

"-i" 옵션을 사용하여 .ass 파일을 읽고 렌더링합니다.

```
kick -i path/to/cornell.ass
```

"-i" 옵션은 꼭 필요한 것은 아니며, kick은 .ass로 끝나는 인수를 자동으로 인식하므로 이것 역시 작동합니다.

```
kick path/to/cornell.ass
```

기본적으로 렌더링되는 이미지가 표시되는 창이 나타납니다. '-dw' 옵션으로 디스플레이 창을 끌 수 있습니다.

```
kick cornell.ass -dw
```

지오메트리가 분홍색으로 표시되면 쉐이더로의 경로가 필요하기 때문일 수 있습니다. 이것은 "-l" 플래그를 사용하여 추가할 수 있습니다.

```
kick cornell.ass -l /path/to/plugin/shaders/
```

로그 정보가 stdout로 전송됩니다. "-v <n>" 옵션으로 로그 상세 정보 수준을 늘리거나 줄일 수 있습니다(기본 상세 정보 수준은 1). 가장 높은 수준의 옵션은 "-v 6"입니다.

```
kick cornell.ass -v 2
```

"-v 0"으로 로그 출력을 끕니다:

```
kick cornell.ass -v 0
```

렌더링된 이미지를 출력 파일로 저장하려면 "-o" 플래그를 사용합니다.

```
kick cornell.ass -o cornell.exr
```

"-r <width> <height>" 옵션으로 렌더링 표시 크기를 변경합니다.

```
kick cornell.ass -r 1024 720
```

Arnold 버전 번호 또는 전체 버전 문자열을 인쇄합니다.

```
kick -av
kick --version
```

라이센스 서버에 대한 진단 정보를 인쇄하고 사용 가능하고 사용 중인 설치된 라이센스를 나열합니다.

```
kick -licensecheck
```

앤티앨리어싱 샘플을 재정의하려면:

```
kick cornell.ass -as 3
```

확산 GI 샘플을 재정의하려면:

```
kick cornell.ass -ds 3
```

점진적 세분화 모드를 비활성화하려면:

```
kick cornell.ass -dp
```

디버깅을 위해 텍스처, 라이트 및 쉐이더, 모션 블러, 서브 디비전, 변위 또는 SSS와 같은 여러 기능을 전역적으로 비활성화할 수 있습니다.

```
kick cornell.ass -it
kick cornell.ass -il
kick cornell.ass -is
kick cornell.ass -imb -isd -idisp -isss
```

다음과 같이 동적 라이브러리(.dll 또는 .so)에서 로드하여 사용자 정의 Arnold 노드를 설치할 수 있습니다.

```
kick cornell.ass -l path\to\plugin -l path\to\more\plugins
```

다음을 사용하여 설치된 모든 노드(내장 및 동적 로드 모두)의 목록을 가져올 수도 있습니다.

```
kick -nodes
kick -l path\to\plugins -nodes
```

"-info"로 노드를 검사할 수 있습니다.

```
kick -info polymesh
kick -info options
kick -l path\to\plugins -info custom_plugin_node
```

또는 주어진 파라미터에 대해 추가 정보를 얻습니다.

```
kick -info polymesh.sidedness
kick -info options.bucket_scanning
```

"-set" 명령을 사용하여 모든 노드에 대한 파라미터를 모두 재정의합니다.

```
kick cornell.ass -set options.AA_samples 3
```

특정 유형의 모든 노드에 대한 파라미터를 모두 재정의합니다.

```
kick cornell.ass -set curves.mode thick
```

워터마크를 사용하여 렌더링하지 않고 유효한 라이센스가 없는 경우 kick에서 렌더링을 중단하게 합니다.

```
kick cornell.ass -set options.abort_on_license_fail true
```

몇 가지 파라미터를 재정의합니다.

```
kick first.ass -set options.AA_samples 3 -set options.bucket_size 16
```

Interactive Mode

대화식 모드에서 렌더링하려면 "-ipr q" 옵션을 사용합니다. 그러면 장면을 탐색하고 평면/부드러운 법선, UV, 와이어프레임 등과 같은 다양한 디버그 쉐이딩 모드로 전환할 수 있습니다.

```
kick cornell.ass -ipr q
```

렌더링 창을 클릭하고 소문자 "i"를 누르면 기존 쉐이더는 무시됩니다. 그런 다음 숫자 키(0,1,2,3,4,5,6,7,8,9)를 사용하여 다양한 디버그 쉐이딩 모드를 전환합니다. (대문자 "i"를 눌러 장면 쉐이더를 복원할 수 있음).

선택한 상호 작용 모드로 탐색합니다(패닝/줌). Quake 컨트롤은 "q"(WASD), Maya 컨트롤은 "m"입니다(Alt + Mouse). "[" 및 "]" 키로 이미지 노출을 높이거나 줄입니다.

사용 가능한 모든 색 공간 인쇄

```
$ kick -lcs scene.ass
```

Available color spaces from color manager "defaultColorMgtGlobals" of type "color_manager_syncolor":

```
ARRI LogC
camera Rec 709
Sony SLog2
Log film scan (ADX)
Log-to-Lin (cineon)
Log-to-Lin (jzp)
Raw
ACES2065-1
ACEScg
scene-linear CIE XYZ
scene-linear DCI-P3
scene-linear Rec 2020
scene-linear Rec 709/sRGB
gamma 1.8 Rec 709
gamma 2.2 Rec 709
gamma 2.4 Rec 709 (video)
sRGB
ACES RRT v0.7
ACES RRT v1.0
```

```
Log
1.8 gamma
2.2 gamma
Rec 709 gamma
sRGB gamma
Raw
Stingray tone-map
```

특정 색 공간을 Kick 표시 창에 지정하는 방법(기본적으로 sRGB 감마)

```
$ kick -ocs "Log" scene.ass
```

색상 관리 통합 후 .ass 파일은 색상 관리가 활성화된 경우 Autodesk 색상 관리 카탈로그에 대한 액세스에 의존합니다. 이 경로는 color_manager_syncolor 노드에 있습니다. 그러나 더 많은 유연성을 갖기 위해 환경 변수 SYNCOLOR를 사용하여 경로를 재정의할 수 있습니다.

```
$ export SYNCOLOR=/path/to/synColorConfig.xml
```

렌더 로그를 읽는 방법

렌더링 로그는 Kick에서 이미지를 렌더링할 때 콘솔이나 파일에 진행 메시지, 경고 및 오류를 인쇄합니다. 완료율을 보여줄 수 있을 뿐만 아니라 이러한 상세한 통계를 사용하여 장면을 최적화하고 디버깅 할 수 있습니다.

Arnold 로그는 렌더링 디버깅, 최적화 및 벤치마킹에 유용한 상세 통계를 제공합니다. 이것은 오류가 발생하면 제일 먼저 검사해야 하는 것이며, 일반적으로는 지원에 보낼 첫 번째 정보입니다.

렌더러에 문제가 발생하면 로그 파일로 수행해야 할 첫 번째 작업은 "ERROR"라는 단어를 찾는 것입니다. 대부분 문제의 원인은 ERROR 메시지에서 찾아볼 수 있습니다.

상세 정보 수준은 kick에서 플러그인까지 다양합니다.

Kick	플러그인
0 - 출력 없음	오류
1 - 진행 상태	경고 + 정보
2	디버그
3	
4	
5 - 상세 정보	
6 - 디버그	

아래의 로그에는 가장 일반적인 출력이 포함되어 있습니다. 로그의 첫 번째 열인 **00:00:00**은 경과한 시간을 **hh:mm:ss**로 보여줍니다. 두 번째 열인 773MB는 이 단계에서 메모리 사용량을 보여줍니다.

초기화(정보)

이 부분에서는 사용 중인 Arnold의 버전과 빌드 상세 정보를 보여줍니다. 또한 렌더링되는 장치의 하드웨어 사양과 운영 체제들을 나열합니다.

00:00:00	773MB	log started Tue Jul 21 15:26:25 2015
00:00:00	773MB	Arnold 4.2.7.4 windows icc-14.0.2 olio-1.5.15 rlm-11.2.2 2015/06/15 09:39:31
00:00:00	773MB	host application: MtoA 1.2.3.1 03a85380bec8 (Master) MtoA-1.2.3.1 Maya 2016
00:00:00	773MB	running on PC, pid=12188
00:00:00	773MB	1 x Intel(R) Xeon(R) CPU E5-1650 v2 @ 3.50GHz (6 cores, 12 logical) with 32712MB
00:00:00	773MB	Windows 7 Professional Service Pack 1 (version 6.1, build 7601)

첫 번째 렌더링에서만 전체 렌더링 세션에 대해 단일 초기화 프로세스가 있으며, 모든 다음 렌더링에 대해 업데이트된 프로세스가 있습니다.

00:00:00	773MB	[mtoa.session]	Initializing at frame 1.000000
----------	-------	----------------	--------------------------------

```

00:00:00 773MB | [mtoa] Exporting Arnold options 'defaultArnoldRenderOptions'
00:00:00 773MB | [mtoa.extensions] aiOptions Using translator <built-in>, provided by <built-in>(<built-in>).
00:00:00 773MB | [mtoa.session] defaultArnoldRenderOptions | Exporting plug
defaultArnoldRenderOptions.message for type aiOptions

```

로딩/플러그인

이 부분에서는 쉐이더 또는 절차의 가져오기 결과를 표시합니다.

kick을 통해 자체적으로 렌더링하면 플러그인 및 ass 위치가 나타나지 않습니다.

```

00:00:00 773MB | loading plugins from . ...
00:00:00 773MB | no plugins loaded
00:00:00 773MB | [ass] loading project/scenes/simplescene.ass ...
00:00:00 773MB | [ass] read 11386 bytes, 11 nodes in 0:00.00

```

플러그인이 로딩되는 경우(그리고 상세 정보 수준이 충분히 높은 경우) 쉐이더들이 나열됩니다.

```

00:00:00 773MB | loading plugin: C:/solidangle/mtoadeploy/2016/shaders/mtoa_shaders.dll ...
00:00:00 773MB | mtoa_shaders.dll: MayaMultiplyDivide uses Arnold 4.2.7.4
00:00:00 773MB | mtoa_shaders.dll: MayaClamp uses Arnold 4.2.7.4
00:00:00 773MB | mtoa_shaders.dll: MayaGammaCorrect uses Arnold 4.2.7.4
00:00:00 773MB | mtoa_shaders.dll: MayaCondition uses Arnold 4.2.7.4
...
00:00:00 774MB | mtoa_shaders.dll: volume_sample_float uses Arnold 4.2.7.4
00:00:00 774MB | mtoa_shaders.dll: volume_sample_rgb uses Arnold 4.2.7.4
00:00:00 774MB | mtoa_shaders.dll: driver_mplay uses Arnold 4.2.7.4
00:00:00 774MB | loaded 90 plugins from 1 lib(s) in 0:00.00

```

라이센스 확인

라이센스가 유효하다면 로그에 다음이 표시됩니다.

```

00:00:00 810MB | [rlm] checkout of "arnold 20150615" in progress ...
00:00:00 810MB | [rlm] checkout of "arnold 20150615" from server PC in 0:00.01
00:00:00 810MB | [rlm] expiration date: 31-dec-2015 (164 days left)

```

하지만 RLM 서버가 실행되고 있지 않다면 경고가 표시됩니다.

```

00:00:00 810MB WARNING | [rlm] could not connect to license server on 5053@localhost

```

Color Management

```

00:00:00 56MB |
00:00:00 63MB | [color_manager_syncolor] Using syncolor_color_manager Version 2018.0.80
00:00:00 63MB | with the native catalog directory from
/path/to/synColor
00:00:00 63MB | and the optional custom catalog directory from
/other/path/to/Shared/

```

SYNCOLOR 환경 변수를 사용하는 경우 kick은 ass 색상 관리 경로를 재정의하므로 출력은 다음과 같습니다.

```

00:00:00 63MB | [color_manager_syncolor] Using syncolor_color_manager Version 2018.0.80
00:00:00 63MB | from the preference file /path/to/synColorConfig.xml
00:00:00 63MB | and the optional custom catalog directory from
/other/path/to/Shared/

```

장면 내용

여기서는 장면 내의 조명 및 오브젝트(그리고 유형)의 수가 나열됩니다.

```

00:00:00 818MB | there are 1 light and 2 objects:
00:00:00 818MB | 1 persp_camera
00:00:00 818MB | 1 distant_light
00:00:00 818MB | 1 utility
00:00:00 818MB | 1 lambert
00:00:00 818MB | 1 driver_exr
00:00:00 818MB | 1 gaussian_filter
00:00:00 818MB | 1 polymesh
00:00:00 818MB | 1 list_aggregate
00:00:00 818MB | 1 MayaShadingEngine
00:00:00 818MB | 1 renderview_display
00:00:00 818MB | 1 color_manager_syncolor

```

해상도/샘플

이 부분에서는 출력 해상도 및 샘플의 개수에 대한 자세한 정보가 표시됩니다.

```

00:00:00 818MB | rendering image at 640 x 480, 3 AA samples
00:00:00 818MB | AA sample clamp <disabled>
00:00:00 818MB | diffuse samples 3 / depth 1
00:00:00 818MB | specular samples 3 / depth 2
00:00:00 818MB | transmission samples 1 / depth 2
00:00:00 818MB | volume indirect <disabled by depth>
00:00:00 818MB | total depth 10

```

00:00:00	818MB	bssrdf	<disabled>
00:00:00	818MB	transparency	depth 10 / fast opacity off

노드

렌더링이 시작되기 전에 노드가 초기화됩니다. 여기에 있는 통계는 장면의 오브젝트와 조명에 대한 정보를 제공하며, 잘못된 구성이 있으면 경고를 표시할 수 있습니다. 일부 데이터는 즉시 초기화되지 않고 일부 계산은 첫 번째 광선이 오브젝트에 닿을 때까지 지연되므로 렌더링 중에 추가 정보가 표시될 수 있습니다.

00:00:00	818MB	initializing 11 nodes ...	
00:00:00	818MB	creating root object list ...	
00:00:00	818MB	scene bounds: (-0.995465517 -1.01209259 -0.995465755) -> (1.00453472 0.98790741 1.00453484)	
00:00:00	818MB	node initialization done in 0:00.00 (single-threaded)	
00:00:00	818MB	updating 12 nodes ...	
00:00:00	818MB	directionalLightShape1: distant_light using 1 sample	
00:00:00	818MB	node update done in 0:00.00 (single-threaded)	

드라이버/AOV

다음은 장면에서 사용되는 드라이버와 AOV 및 잘못 설정된 경우의 경고입니다..

00:00:00	818MB	[aov] parsing output statement: "RGBA RGBA defaultArnoldFilter@gaussian_filter defaultArnoldDisplayDriver@renderview_display"	
00:00:00	818MB	[aov] parsing output statement: "RGBA RGBA defaultArnoldFilter@gaussian_filter defaultArnoldDriver@driver_exr.RGBA"	
00:00:00	818MB	[aov] registered driver: "defaultArnoldDisplayDriver@renderview_display" (renderview_display)	
00:00:00	818MB	[aov] * "RGBA" of type RGBA filtered by "defaultArnoldFilter@gaussian_filter" (gaussian_filter)	
00:00:00	818MB	[aov] registered driver: "defaultArnoldDriver@driver_exr.RGBA" (driver_exr)	
00:00:00	818MB	[aov] * "RGBA" of type RGBA filtered by "defaultArnoldFilter@gaussian_filter" (gaussian_filter)	
00:00:00	818MB	[aov] done preparing 1 AOV for 2 outputs to 2 drivers (0 deep AOVs)	

진행 상태

렌더링이 시작되었으므로 로그에 사용 중인 버킷의 수와 크기가 표시됩니다. 중요도 표를 생성해야 하는 경우에는(이 경우에는 skydome에 대해) 지금 수행되어 평균 에너지 값을 보여줍니다. 완료된 백분율은 픽셀당 평균 광선과 함께 5%씩 증가합니다. 샘플링 설정이 너무 높으면 이 값이 한 눈에 표시됩니다.

00:00:00	825MB	starting 12 bucket workers of size 64x64 ...	
00:00:00	835MB	0% done - 9 rays/pixel	

00:00:00	835MB	[accel] bvh4 done - 0:00.00 - 400 prims, 1 key
00:00:00	836MB	5% done - 9 rays/pixel
00:00:00	836MB	10% done - 9 rays/pixel
00:00:00	836MB	15% done - 9 rays/pixel
00:00:00	837MB	20% done - 9 rays/pixel
00:00:00	837MB	25% done - 9 rays/pixel
00:00:00	837MB	30% done - 9 rays/pixel
00:00:00	837MB	35% done - 10 rays/pixel
00:00:00	837MB	40% done - 10 rays/pixel
00:00:00	838MB	45% done - 12 rays/pixel
00:00:00	838MB	50% done - 12 rays/pixel
00:00:00	838MB	55% done - 13 rays/pixel
00:00:00	838MB	60% done - 13 rays/pixel
00:00:00	838MB	65% done - 13 rays/pixel
00:00:00	838MB	70% done - 13 rays/pixel
00:00:00	838MB	75% done - 13 rays/pixel
00:00:00	838MB	80% done - 13 rays/pixel
00:00:00	838MB	85% done - 13 rays/pixel
00:00:00	838MB	90% done - 13 rays/pixel
00:00:00	838MB	95% done - 13 rays/pixel
00:00:00	838MB	100% done - 13 rays/pixel
00:00:00	824MB	bucket workers done in 0:00.38

출력

렌더링이 완료되면 선택한 출력 드라이버를 사용하여 파일(또는 화면)에 작성됩니다. 로그에서 관련 경로를 표시합니다.

00:00:00	824MB	[driver_exr] writing file 'C:/Users/Documents/example.exr'
00:00:00	821MB	render done

장면 생성 통계

이들 타이밍은 플러그인과 .ass 파일을 로드하는 데 걸린 시간을 보여줍니다.

00:00:00	821MB	scene creation time:
00:00:00	821MB	plugin loading 0:00.07
00:00:00	821MB	system/unaccounted 0:00.12
00:00:00	821MB	total 0:00.12 (1.05% machine utilization)

렌더링 시간 통계

이들 타이밍은 다양한 작업을 수행하는 데 걸린 시간을 보여줍니다. 일반적으로 퍽셀 렌더링은 대부분의 시간을

차지하며, 그렇지 않은 경우 노드 초기화를 살펴보면 지오메트리 통계가 비효율성을 나타낼 수도 있습니다. 컴퓨터 사용률이 낮으면 동일한 컴퓨터의 다른 프로세스가 렌더링 속도를 저하시킨다는 의미일 수도 있습니다. 그러나 노드 초기화, 절차 또는 메시 처리와 같이 단일 스레드 작업을 수행하는 데 많은 시간을 소비했음을 나타낼 수도 있습니다. 또 다른 가능성은 일반적으로 텍스처 또는 볼륨 등에 대해 많은 (느린) 파일 액세스가 있는 것일 수 있습니다. 텍스처 통계를 살펴보면 문제점이 파악될 수도 있습니다.

00:00:00	821MB	render time:
00:00:00	821MB	node init 0:00.00
00:00:00	821MB	sanity checks 0:00.00
00:00:00	821MB	bucket rendering 0:00.38
00:00:00	821MB	mesh processing 0:00.00
00:00:00	821MB	accel. building 0:00.00
00:00:00	821MB	pixel rendering 0:00.38 (multi-threaded render, this value may not be reliable)
00:00:00	821MB	system/unaccounted 0:00.17
00:00:00	821MB	total 0:00.55 (69.23% machine utilization)

메모리 통계

여기에는 다양한 유형의 데이터에 대한 시작 및 최대 메모리에 사용된 메모리가 나열됩니다.. 플러그인을 사용할 때 시작 메모리는 호스트 애플리케이션이 사용하고 있는 메모리 양을 보여 주며, 각에서 렌더링할 때 시작 메모리는 적은데 이것은 일반적으로 더 큰 장면을 렌더링할 수 있다는 의미입니다. 메모리 사용량이 너무 높을 경우 이러한 통계는 어느 데이터가 메모리 사용량을 줄이는 데 가장 도움이 되는지를 나타냅니다.

00:00:00	821MB	memory consumed in MB:
00:00:00	821MB	at startup 807.25
00:00:00	821MB	plugins 2.52
00:00:00	821MB	AOV samples 11.67
00:00:00	821MB	output buffers 5.32
00:00:00	821MB	node overhead 0.00
00:00:00	821MB	message passing 0.02
00:00:00	821MB	memory pools 13.54
00:00:00	821MB	geometry 0.01
00:00:00	821MB	polymesh 0.01
00:00:00	821MB	accel. structs 0.02
00:00:00	821MB	strings 0.30
00:00:00	821MB	texture cache 0.00
00:00:00	821MB	unaccounted 1.64
00:00:00	821MB	total peak 842.29

광선 통계

이것은 픽셀당, AA 샘플당 각 유형의 광선 수입니다. 렌더링 시간이 긴 경우 이러한 숫자들은 렌더링하는 데 가

장 비용이 많이 드는 샘플 유형을 파악하고 이를 줄이는 데 가장 도움이 될 것입니다.

00:00:00	821MB	ray counts:	(/pixel , /sample) (% total) (avg. hits) (max hits)
00:00:00	821MB	camera	2318256 (8.71, 1.00) (64.57%) (0.11) (1)
00:00:00	821MB	shadow	228076 (0.86, 0.10) (6.35%) (0.00) (0)
00:00:00	821MB	diffuse	1043914 (3.92, 0.45) (29.08%) (0.00) (0)
00:00:00	821MB	total	3590246 (13.48, 1.55) (100.00%) (0.07) (1)
00:00:00	821MB	max depth	1

쉐이더 통계

다양한 컨텍스트에서 쉐이더 호출 수가 여기에 표시되어 있습니다. 이것은 어떤 쉐이더 호출이 속도 향상에 가장 도움이 되는지 파악하는 데 유용합니다.

- Primary: 비-음영 광선의 표면 쉐이딩(기존 쉐이딩)
- Transparent_shadow: 그림자 광선이 투명한 표면을 통과할 때의 그림자 음영(광선이 차단되었거나 지속 가능한지 여부를 알기 위해 그림자를 적용해야 하고, 계속되면 색상이 변경됨).
- Autobump: autobump가 계산될 때마다 변위를 계산해야 합니다.
- Background: 배경 쉐이더.
- Importance: 중요도 테이블을 계산하는 동안 이루어진 호출
- Volume: 볼륨 쉐이더 호출.

00:00:00	821MB	shader calls:	(/pixel , /sample) (% total)
00:00:00	821MB	primary	523006 (1.96, 0.23) (100.00%)
00:00:00	821MB	total	523006 (1.96, 0.23) (100.00%)

이들 쉐이더 통계를 사용하여 렌더링 시간을 향상시킬 수 있습니다. 예를 들어, 많은 수의 transparent_shadow 가 보이면, 일반적으로 빛 샘플이 매우 높거나 투명한 오브젝트가 많이 있음을 나타냅니다. 또 다른 예는 autobump가 너무 높을 경우 autobump의 활용도가 높지 않은 메시에서 autobump를 비활성화하는 것입니다.

지오메트리 통계

이러한 통계를 통해 장면에서 지오메트리의 양을 알 수 있습니다. 메모리 사용량이 많거나 장면 설정이 오래 걸리는 경우 이들을 사용하여 가장 많이 기여하는 오브젝트를 찾을 수 있으며 단순화되거나 하위 반복을 줄여서 혜택을 얻을 수 있습니다.

00:00:00	821MB	geometry:	(% hit) (instances) (init mem, final mem)
00:00:00	821MB	lists	1 (100.0%) (0) (0.00, 0.00)

00:00:00	821MB	polymeshes	1 (100.0%) (0) (0.02,	0.01)
00:00:00	821MB	-----				
00:00:00	821MB	geometric elements: (min) (avg.) (max)	
00:00:00	821MB	objects (top level)	1 (1) (1.0) (1)
00:00:00	821MB	polygons	400 (400) (400.0) (400)
00:00:00	821MB	-----				
00:00:00	821MB	triangle tessellation: (min) (avg.) (max) (/ element)	(% total)
00:00:00	821MB	polymeshes	760 (760) (760.0) (760) (
00:00:00	821MB	unique triangles:	760			
00:00:00	821MB	memory use (in MB)	0.01			
00:00:00	821MB	vertices	0.00			
00:00:00	821MB	vertex indices	0.00			
00:00:00	821MB	packed normals	0.00			
00:00:00	821MB	normal indices	0.00			
00:00:00	821MB	uv coords	0.00			
00:00:00	821MB	uv coords idxs	0.00			
00:00:00	821MB	uniform indices	0.00			
00:00:00	821MB	userdata	0.00			
00:00:00	821MB	largest polymeshes by triangle count:				
00:00:00	821MB	760 tris -- pSphereShape1				

텍스처 통계

이미지 텍스처를 위한 자세한 통계로, 어떤 텍스처가 대부분의 메모리를 사용하는지, 그리고 어떤 텍스처가 .tx 파일로 변환되어 혜택을 얻는지 보여줍니다. 좋은 성능을 위해서는 메인 캐시 누락 비율이 매우 낮아야 합니다. 이것이 높으면 렌더링 시간이 크게 증가할 수 있습니다. 메인 캐시 누락 비율이 너무 높다면 .tx 파일만 사용도록 하고, 텍스처 수를 줄이거나 텍스처 설정을 조정하십시오.

00:00:00	841MB	OpenImageIO Texture statistics				
00:00:00	841MB	Queries/batches :				
00:00:00	841MB	texture : 261503 queries in 261503 batches				
00:00:00	841MB	texture 3d : 0 queries in 0 batches				
00:00:00	841MB	shadow : 0 queries in 0 batches				
00:00:00	841MB	environment : 0 queries in 0 batches				
00:00:00	841MB	Interpolations :				
00:00:00	841MB	closest : 0				
00:00:00	841MB	bilinear : 398256				
00:00:00	841MB	bicubic : 701789				
00:00:00	841MB	Average anisotropic probes : 2.8				
00:00:00	841MB	Max anisotropy in the wild : 671				
00:00:00	841MB					
00:00:00	841MB	OpenImageIO ImageCache statistics (000000003D574040) ver 1.5.15				

```

00:00:00 841MB | Images : 1 unique
00:00:00 841MB | ImageInputs : 1 created, 1 current, 1 peak
00:00:00 841MB | Total size of all images referenced : 1.7 MB
00:00:00 841MB | Read from disk : 1.7 MB
00:00:00 841MB | File I/O time : 0.2s (0.0s average per thread)
00:00:00 841MB | File open time only : 0.0s
00:00:00 841MB | Tiles: 25 created, 24 current, 24 peak
00:00:00 841MB | total tile requests : 1454328
00:00:00 841MB | micro-cache misses : 35383 (2.43294%)
00:00:00 841MB | main cache misses : 25 (0.00171901%)
00:00:00 841MB | Peak cache memory : 2.3 MB
00:00:00 841MB | Image file statistics:
00:00:00 841MB |     opens   tiles   MB read   I/O time   res           File
00:00:00 841MB |           1       1        12        1.7      0.2s    768x 768x3.u8
C:/Users/Documents/example.tif UNTILED UNMIPPED MIP-COUNT [12,7,3,2,1,0,0,0,0,0]
00:00:00 841MB |
00:00:00 841MB | Tot:     1       12        1.7      0.2s
00:00:00 841MB | 1 not tiled, 1 not MIP-mapped

```

종료

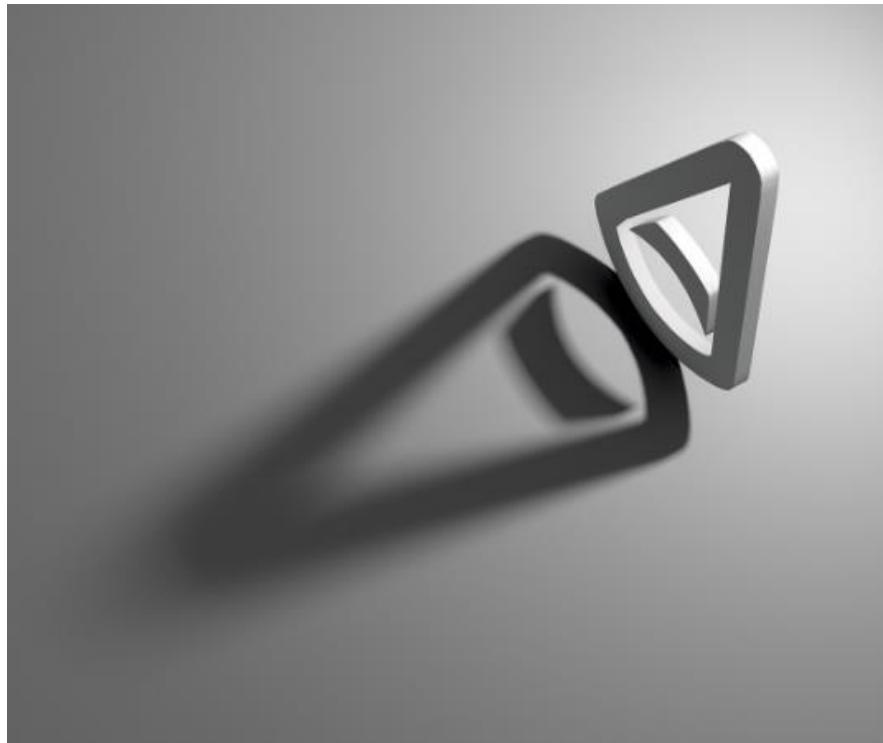
Arnold는 모든 리소스, 메모리/스레드가 릴리스된 다음 종료됩니다.

```

00:00:00 841MB | releasing resources
00:00:00 831MB | unloading 27 plugins
00:00:00 829MB | unloading plugins done
00:00:00 828MB | Arnold shutdown

```

조명

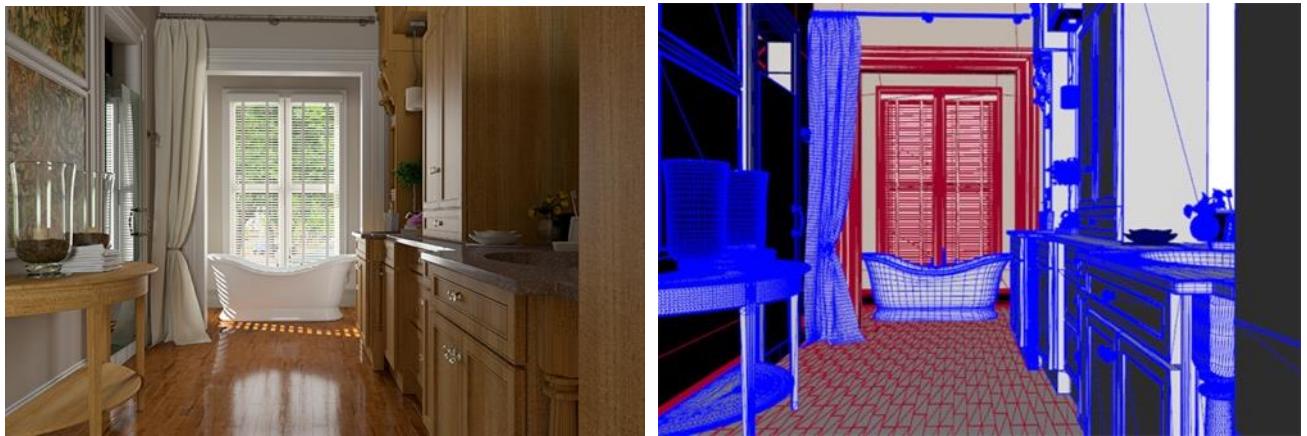


MtoA에서 조명 관련 자습서는 다음 페이지에 있습니다.

- [인테리어 장면을 MtoA로 변환](#)
- [플래시 사진 효과](#)
- [IES 조명](#)
- [방 조명](#)
- [자연사 박물관 조명 효과](#)
- [스튜디오 자동차 렌더링](#)
- [스튜디오 조명](#)
- [Sky Dome Light 사용](#)

이들 자습서와 함께 제공되는 많은 장면 파일은 [Learning scenes](#) 페이지에서도 찾을 수 있습니다.

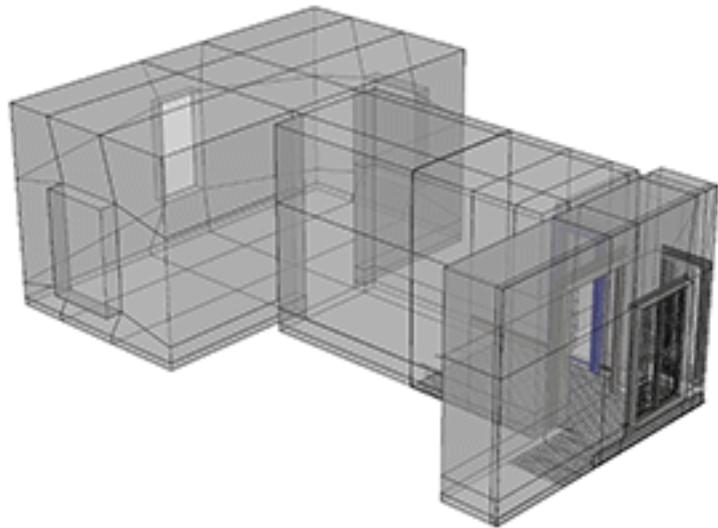
인테리어 장면을 MtoA로 변환



이 자습서는 MtoA를 사용하여 렌더링할 다른 렌더러에 대해 설정한 내부 장면을 변환하는 데 필요한 워크플로우를 분석한 것입니다. 장면 설정 및 렌더링은 1시간을 넘지 않아야 합니다.

방 조명에 접근하는 여러 가지 방법을 보여주는 또 다른 자습서가 [여기\(방 조명\)](#)에 있습니다.

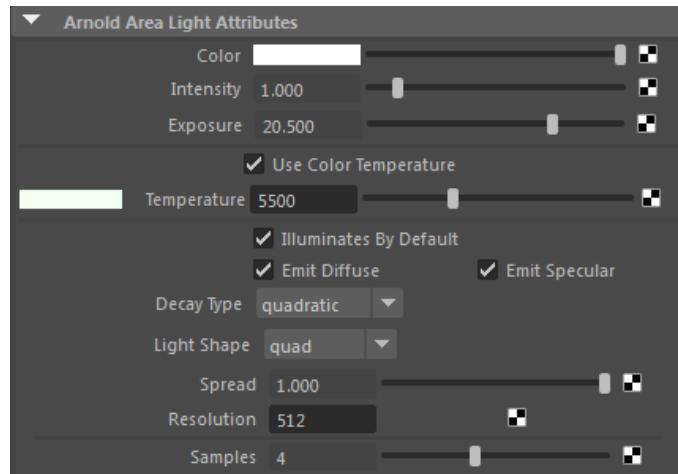
이 장면은 3ds Max에서 .fbx 파일로 내보내고 FBX 플러그인을 사용하여 Maya로 가져왔습니다.



- 장면을 가져올 때 Outliner 및 Hypershade 창이 상당히 복잡하다는 것을 알 수 있습니다. 시작하기 전에 File> Optimize Scene으로 이동하여 빈 그룹 노드를 삭제하고 그룹 노드에 지오메트리를 추가합니다.

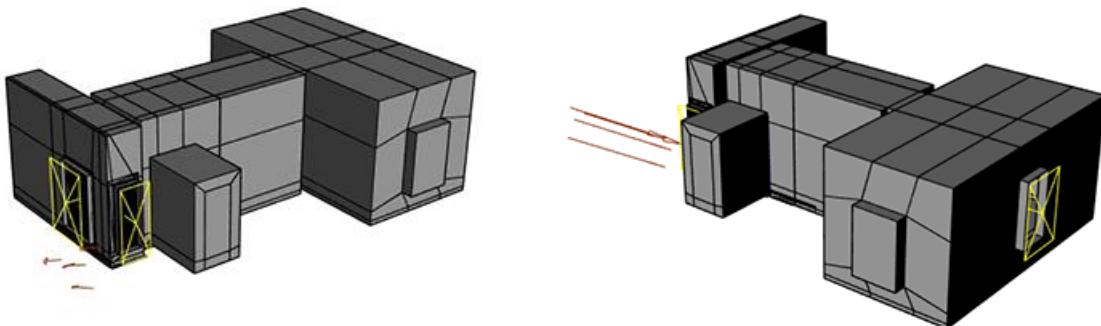
Lighting

- 방의 창 바깥에 위치한 **Ai Area lights**으로 간단하게 장면에 빛을 낼 수 있습니다. **Color Temperature**가 일광의 색 온도로 설정되었습니다(**5500**). 기본 조명 **Samples**를 **1**로 설정합니다. 최종 렌더링의 경우에는 이 값을 **4**로 높여서 그림자 내의 모든 노이즈를 줄입니다. 빛의 **Exposure**를 **20** 정도로 높입니다.



Ai Area Light 속성

- **Directional light**를 사용하여 먼 창을 통해 들어오는 햇빛을 시뮬레이션합니다. **Angle**을 **0.2** 정도로 살짝 높이면 바닥에서 그림자의 가장자리가 더 부드러워 집니다. 이 경우 **Directional light**의 **Samples**를 높여야 할 수도 있습니다(3이면 충분함). **Color Temperature**를 활성화하고 **5000**과 같이 약간 따뜻한 온도를 선택합니다.



장면이 창문 외부의 **Ai Area Light**과 햇빛을 표현하는 **Directional Light**으로 밝혀집니다.

조명이 장면에 어떻게 적용되는지 미리 보려면 장면에 기본 **Standard Surface** 쉐이더 오버라이드를 적용하십시오. 이렇게 하면 렌더링의 '초크 미리보기'가 효과적으로 생성되며 쉐이딩에 신경 쓰지 않고 조명에만 집중할 수 있습니다. 또는 **Arnold RenderView** 창에는 Debug Shading에 'Lighting' 모드가 있습니다.



Diffuse samples을 0으로 줄이면 간접 조명이 효과적으로 비활성화됩니다. 이것은 장면에서 직접 조명을 테스트하고자 할 때 유용하며 렌더링이 더 신속해집니다.



Diffuse 샘플: 0 (간접 조명 비활성화)

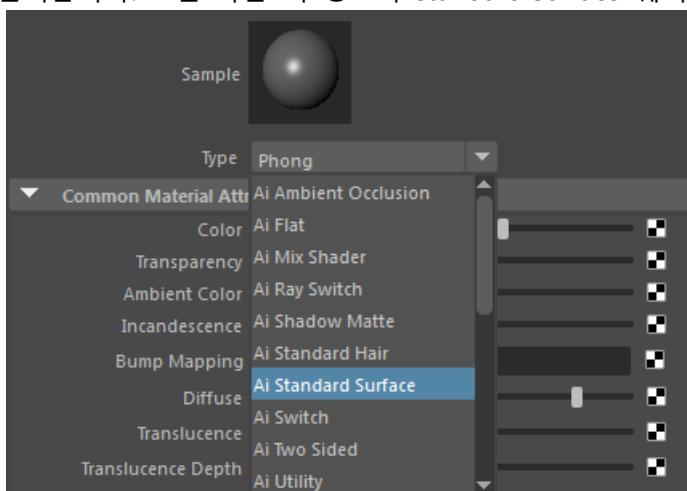
쉐이딩

이 장면은 Maya의 기본 Phong 쉐이더와 함께 제공되며 Hypershade 창에 표시됩니다. 이들을 **Standard Surface** 쉐이더로 변환해야 합니다.



Hypershade 창에 표시되는 Maya Phong 쉐이더가 있는 기본 Maya 장면

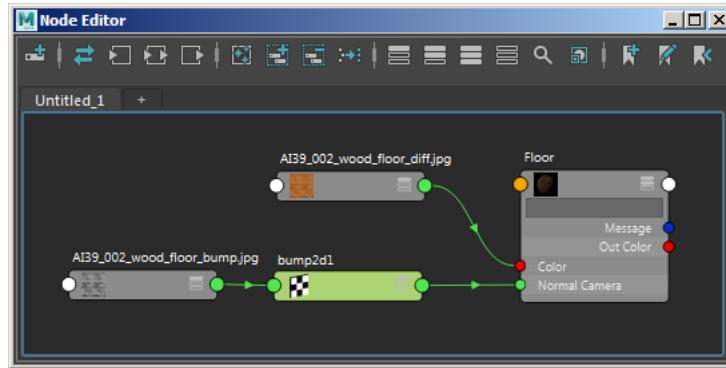
- Phong 쉐이더를 선택하고 Attribute Editor를 엽니다. 'Type' 스크롤다운 메뉴에서 Phong을 선택하고 Standard Surface를 선택합니다. 모든 확산 색 정보가 Standard Surface 쉐이더로 전송됩니다.



Phong 쉐이더가 Color에 할당된 텍스처를 가지고 있는 경우, 이들은 Standard Surface 쉐이더의 **Base Color**에 자동으로 연결되어 추가 작업이 필요하지 않습니다.

여러 쉐이더를 Ai Standard로 쉽게 변환할 수 있도록 쉐이더 변환 스크립트가 MtoA와 함께 제공됩니다.
(<https://support.solidangle.com/display/A5AFMUG/Shader+Conversion+Script>)

바닥 쉐이더



- 확산 색상 맵을 Standard Surface 쉐이더의 **Base Color**에 연결합니다.
- 범프 맵을 **Bump** 속성에 연결합니다. **Bump Depth**를 *0.03*과 같은 적은 양으로 줄입니다.
- Specular Weight**를 *0.3*으로 늘리고 **Specular Roughness**를 *0.1*로 줄여 바닥에 광택 있는 느낌을 연출합니다.

거울

- Standard Surface 쉐이더를 거울에 할당하고 이름을 'Mirror'로 변경합니다.
- Base Weight**를 *1*으로 줄입니다.
- Metalness**를 *1*로 늘립니다.

커튼

여기에서는 커튼이 뒤에서 비춰지는 효과를 주기 위해 약간의 반투명도 추가할 것입니다.

- Standard Surface 쉐이더를 커튼에 할당하고 이름을 'Curtain'으로 변경합니다.
- SSS**를 *0.5* 정도로 높이고 **Thin Wall**을 활성화합니다.



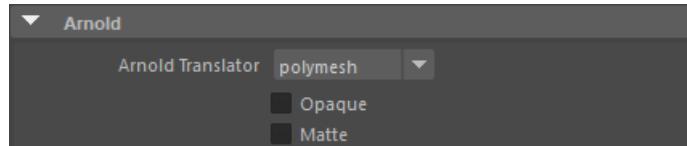
Thin Wall 및 SSS의 커튼에 대한

유리 문

모든 유리 표면은 '**Opaque**'을 비활성화해야 합니다. 그렇지 않으면 오브젝트에 의해 투영된 그림자가 항상 단색이 되며 쉐이더의 굴절 색상이나 밀도가 적용되지 않습니다.

폴리곤면의 법선이 모두 올바른 방향을 향하고 있는지 확인하십시오(특히 Arnold로 유리 표면을 렌더링 할 때 중요).

- 창 지오메트리를 선택하고 Attribute Editor의 Arnold 탭에서 'Opaque'을 비활성화하십시오.



유리와 같이 불투명하지 않은 표면에 대해 Opaque 스위치가

- Standard Surface를 유리 문에 할당하고 이름을 'Glass'로 변경합니다.
- Base Weight**를 **0**으로 줄입니다.
- Specular Weight**를 **1**로 높입니다.
- Transmission Weight**를 **1**로 높입니다(유리가 투명해짐).
- Index of Refraction**(IOR)을 **1.5**(유리)로 높입니다.
- 또한 **Transmission** 색상에 미세한 색조를 추가하여 유리에 색조를 매우 쉽게 추가할 수 있습니다.

렌더링 설정

샘플링

- 최종 렌더링을 위해 **Camera (AA)** 설정을 **6**으로 높였습니다.
- 간접 조명이 있는 공간의 노이즈를 줄이기 위해 **Diffuse Samples**도 **6개**로 증가시켰습니다. 아래 이미지는 **2개**의 (기본값) Diffuse 샘플과 **6개**의 샘플을 렌더링하는 것의 차이를 보여줍니다.



Diffuse Sample 2개와 6개 사이의 차이점을 볼 수 있습니다.

렌더링 시간이 크게 증가하므로 이 값을 늘릴 때는 주의해야 합니다.

노이즈 제거에 대한 자세한 정보 및 자습서는 [여기\(노이즈 제거\)](#)에 있습니다.

Ray Depth

Diffuse Ray Depth

아래 이미지들은 Diffuse Ray Depth 1(기본값) 및 4(롤오버 이미지)를 사용하여 렌더링되었습니다. 예를 들어, 커튼 주변의 반사된 빛의 양에 명확한 차이가 있음을 알 수 있습니다.



Diffuse Ray Depth 1(기본값)과 4 사이의 차이

광선 확산 반사 횟수와 관련하여 렌더링 시간이 선형적으로 증가하므로 이 값을 늘릴 때는 주의해야 합니다.

Transmission Ray Depth

아래 이미지에서 Transmission Ray Depth를 늘리면 유리 꽃병의 차이를 '분명히' 볼 수 있습니다.



2

4

8

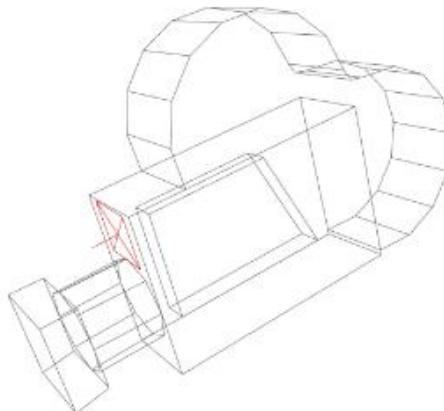
플래시 사진 효과



이 짧은 자습서에서는 좀비 공격의 놀라운 렌더링을 개선하는 데 사용되는 플래시 사진 효과를 에뮬레이트하는 방법에 대해 보여줍니다!

Lighting

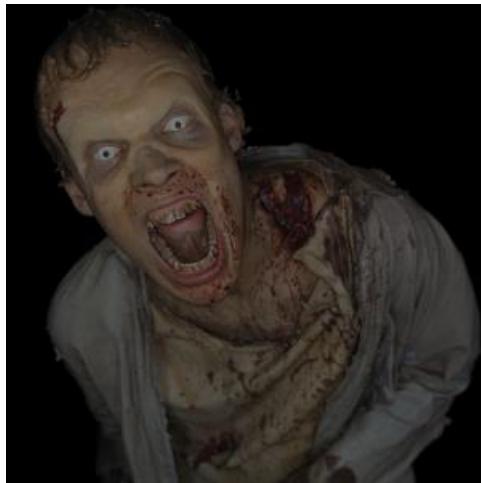
- **Area** 조명을 생성합니다(**Quad**로 설정). 렌더링 카메라 앞에 오도록 배치하고 회전합니다. 이 Area 조명을 마우스 중앙 버튼을 누른 채로 아웃라이너 내의 카메라로 끌어옵니다. 이렇게 하면 카메라가 그룹화되어 카메라를 움직이면 조명이 켜집니다. 조명의 **Exposure**를 **4** 정도로 높입니다.



카메라 앞에 위치한 Quad 조명

Spread

- Quad 빛에 **Spread**를 사용하여 좀비의 얼굴에 빛을 더 집중시킬 수 있습니다. 이 경우에는 **0.2**로 줄였습니다.



Spread: 1 (기본값)



Spread: 0.2

낮은 spread 값은 높은 기본 spread보다 노이즈가 많을 수 있으므로 사용 시 주의하십시오. 이 값이 낮을수록 그림자에서 노이즈를 줄이려면 더 많은 빛 샘플이 필요합니다.

장면에서 빛이 떨어지는지 방식을 제어하는 또 다른 방법은 빛에 **Light Decay** 필터를 추가하는 것입니다.

Atmosphere Volume

Atmosphere Volume을 사용하여 장면에 오싹한 느낌을 줄 수 있습니다.

- Render Settings를 열고 **Arnold Renderer** 탭에서 **Environment**로 이동합니다. **Atmosphere**의 오른쪽에 있는 아이콘에서 마우스 오른쪽 버튼을 클릭하고 **Create aiAtmosphereVolume**을 선택합니다.
- Atmosphere Volume**의 **Density** 속성을 **0.005** 정도로 살짝 높입니다.

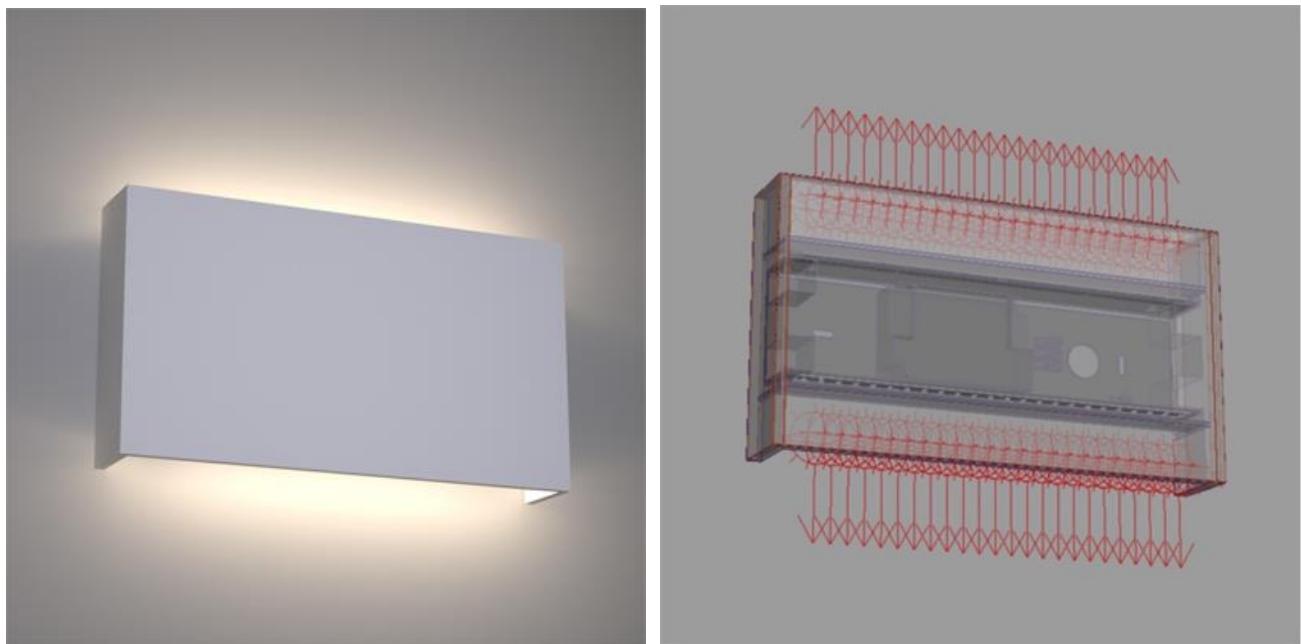


Volume Scattering 없음



Volume Scattering 적용

IES 조명



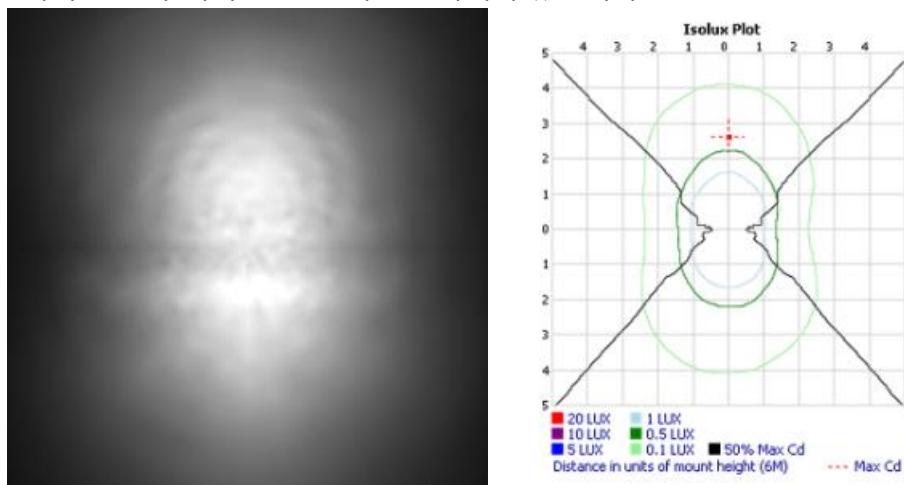
이 자습서에서는 **Photometric Light**을 사용하여 조명 회사로부터 실제 조명을 정확하게 조명하는 방법을 보여줍니다. 이 특별한 조명은 상단 및 하단 LED 보드에서 24개의 점 광원(14w LED)으로 구성되어 있습니다.

장면 파일 및 IES 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다. 각 Photometric 조명에 대해 IES 파일을 열어야 합니다.

(<https://support.solidangle.com/display/A5AFMUG/IES+Lighting>)

IES 파일

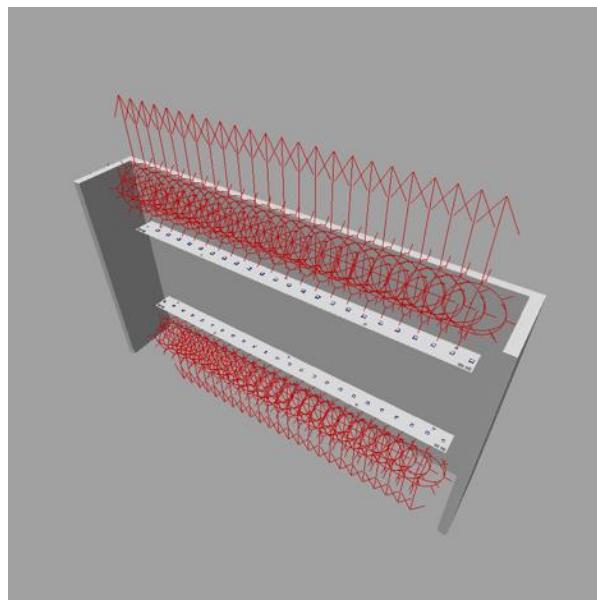
예를 들어 <http://www.visual-3d.com>에 있는 Photometric 조명 뷰어를 사용하여 위 링크에 제공된 IES 파일을 볼 수 있습니다. 쉬운 이해를 돋기 위해 도표 형태로 묘사되어 있습니다.



MtoA로 렌더링되고(왼쪽) Photometric 조명 뷰어로 표시된(오른쪽) IES 조명 프로필.

- 먼저 Photometric 조명을 생성한 다음 왼쪽 끝 LCD 구멍 위에 있는 LCD 보드 위에 놓습니다.

- 조명들을 복제하여 연속으로 24개를 만드십시오. 행을 복제하고 아래 그림과 같이 아래쪽을 가리키도록 회전시키십시오.

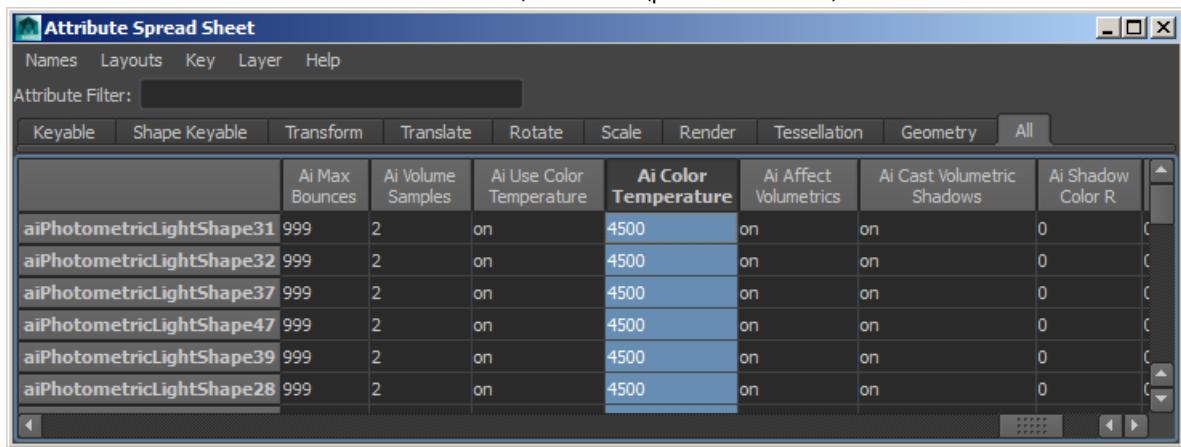


행당 24개의 Photometric 조명

Color Temperature

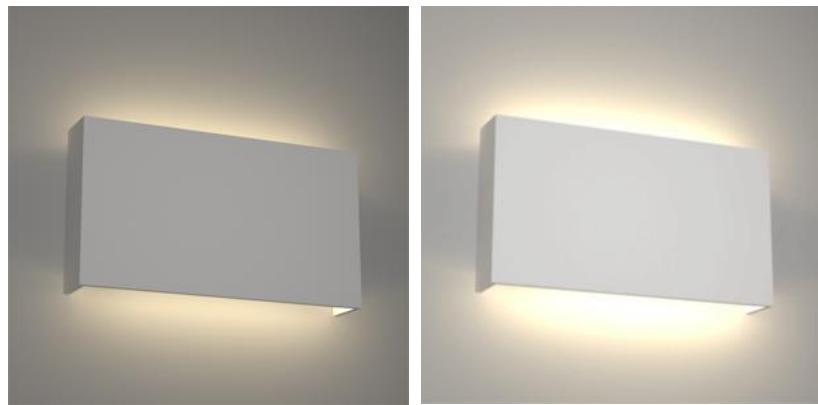
램프 색상의 한 가지 특징은 램프의 빛이 '따뜻'하거나 '차가운' 것과 관계없이 상관 색온도(CCT)로 측정된 색상 외관입니다. CCT가 2700-3000K인 따뜻한 외관의 램프는 일반적으로 국내 환경에 적절한 것으로 간주됩니다. 4000K 이상의 램프는 '시원'하게 느껴지기 때문에 사무실 및 일부 소매업계의 응용 분야에 더 적합합니다. 조명 제조업체가 표시한 이 특별한 조명의 색 온도는 4500K입니다.

- 'Attribute Spread Sheet' 창을 사용하여 모든 조명에 대한 MtoA 속성을 동시에 변경합니다. MtoA 속성을 보려면 해당 형상 노드를 선택해야 합니다(아래쪽 키(pickWalkDown)를 사용하여 형상 노드 선택).



Lighting

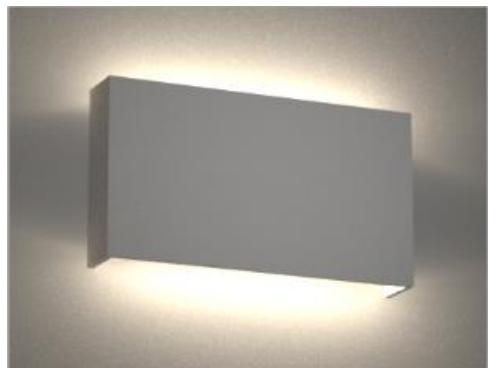
조명의 노출 변경은 광원을 통해 가능합니다. 그러나 장면이 32비트 EXR 이미지로 렌더링되면 후 처리 소프트웨어를 사용하여 노출을 조정하는 것이 더 쉽습니다.



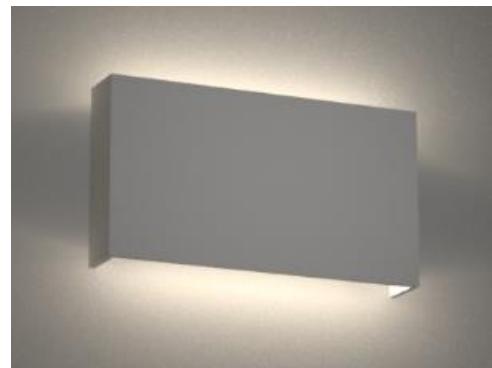
Light exposure 수정

이미지를 톤 매핑해 벽의 과다 노출 부분을 줄일 수 있습니다. 또한 벽에서 밝은 조명 부분의 노이즈를 줄이는 데 유용합니다.

- 장면을 32비트 EXR로 렌더링합니다.



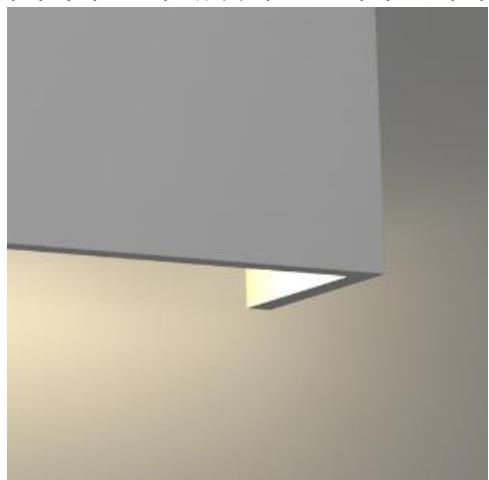
톤 매핑 전



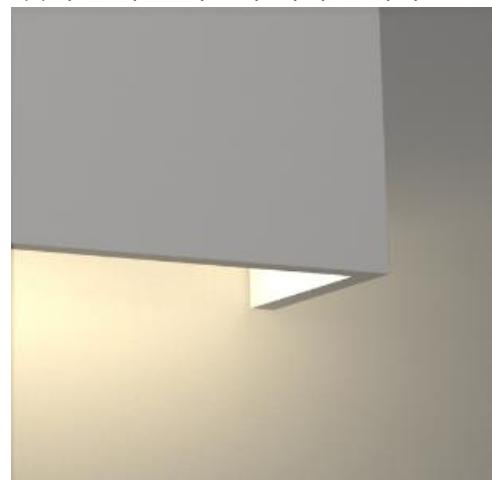
톤 매핑 후

간접 조명

노이즈와 렌더링 시간을 개선하는 한 가지 방법은 Photometric 조명에 대해 간접 조명의 양을 줄이는 것입니다. 그러나 아래 이미지에서 볼 수 있듯이 장면에서 반사되는 빛의 양이 줄어들어 미세한 디테일을 잃게 됩니다.



간접 0



간접 1

Diffuse Samples

Photometric 조명은 빛과 벽에 상당히 가까우며 밝게 빛나는 영역에서 약간의 노이즈가 발생합니다. Photometric 조명 Sample을 높여도 노이즈는 개선되지 않습니다. GI Diffuse Samples를 높이면 노이즈가 줄어들지만 렌더링 시간은 늘어납니다.



Diffuse Samples 4



Diffuse Samples 2

지금까지 Photometric 조명과 함께 IES 파일을 사용하는 방법을 살펴봤습니다. 또한 **Erco, Lamp, Osram, Philips** 등 기타 조명 제조업체의 IES 파일을 테스트해 볼 수 있습니다.



방 조명



Arnold는 멋진 실내 렌더링을 생성해 줍니다. 이 자습서는 바깥에서부터 실내에 조명을 비출 수 있는 일련의 대체 방법을 보여줍니다. 또한, 렌더링 시간과 노이즈를 줄이기 위한 대체 조명 설정 방법에 대해서도 살펴봅니다.

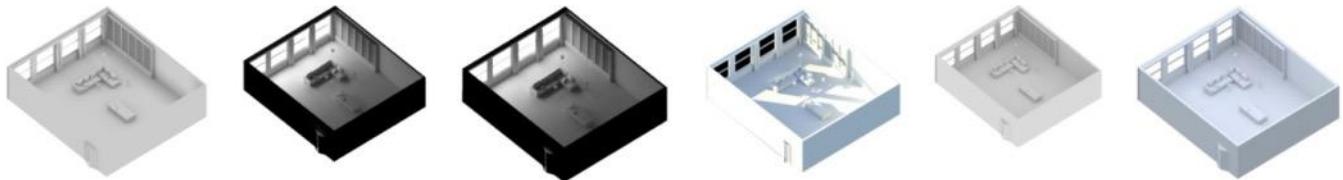
실내 장면에는 `skydome_light`의 사용을 권장하지 않습니다. 이 조명은 야외 장면용이며 배경에서 구형 돔으로 표현됩니다. 여러 중요도 조명 샘플링이 광선을 이 돔의 특정 방향으로 추적합니다. 하지만 실내 장면에서는 이러한 광선이 대부분 물체에 닿기 때문에 조명으로부터의 효과가 전혀 없어서 노이즈가 발생합니다. 이러한 상황에서는 `quad` 조명 및/또는 멀리 있는 조명이 더 나은 옵션이 될 수 있습니다

Maya scene 하단 링크 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Lighting+a+Room>)

이 자습서에 사용된 모든 HDR 맵은 여기서 다운로드할 수 있습니다.

(<http://www.hdrlabs.com/sibl/archive.html>)



Ambient Occlusion

Area Lights

Emissive Geometry

Physical Sky

Sky

SkyDome Light

다음과 같이 다양한 조명 기술을 살펴보겠습니다.

- [방 쉐이딩](#)
- [Ambient Occlusion으로 렌더링](#)
- [Ai Area Lights](#)
- [Emissive Geometry](#)
- [Physical Sky 조명 적용](#)
- [Ai Sky Dome Light 조명 적용](#)

방 쉐이딩

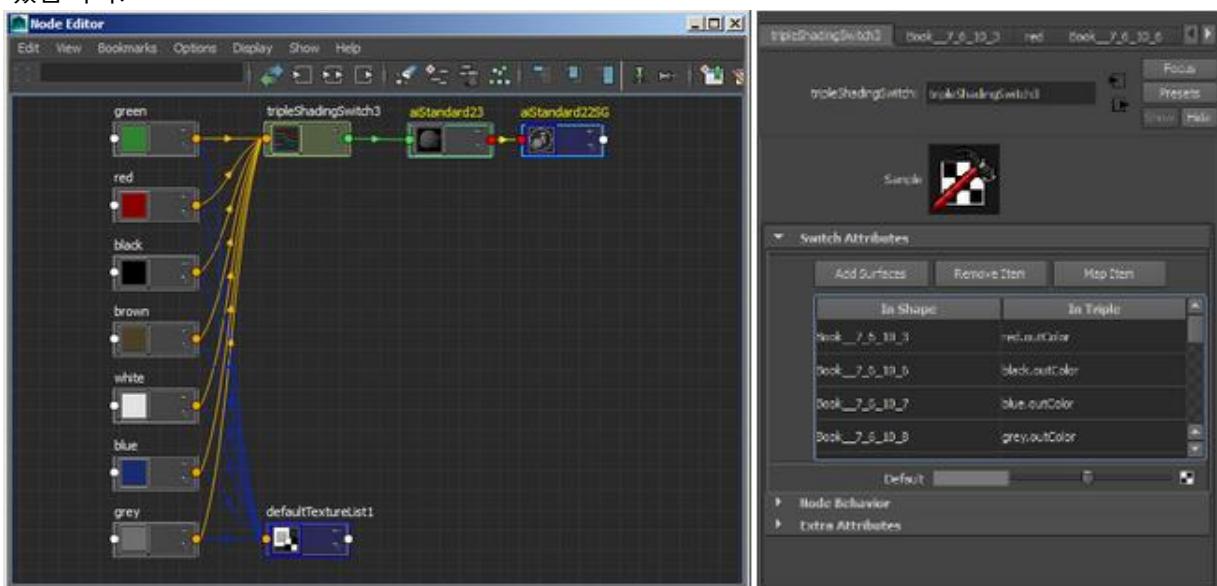
책장 - Shading Switch

Maya의 **ShadingSwitch** 노드는 책으로 가득 찬 책장처럼 여러 질감을 지정해야 하는 많은 물체를 가진 장면이 있는 경우에 매우 편리합니다. 하나의 쉐이더에 여러 텍스처를 할당할 수 있어 동일한 쉐이더가 모든 객체에 지정되므로 쉐이더 속성을 변경할 때 시간을 절약할 수 있습니다.

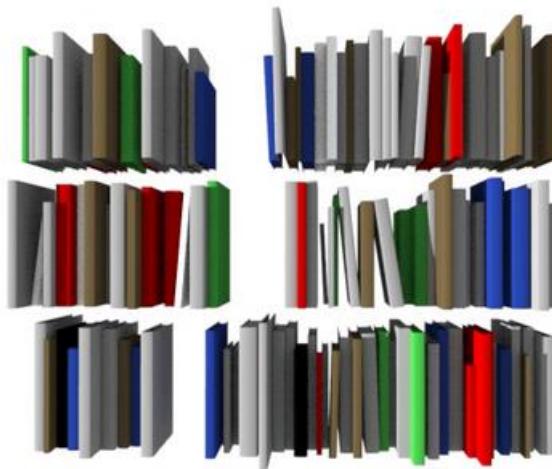


수백만 권의 책이 꽂혀 있는 책장. Maya Shading Switch 노드의 이상적인 상황

- 책 지오메트리를 선택하고 **Standard Surface** 쉐이더를 할당합니다. Maya 'Triple Switch' 노드를 Standard Surface 쉐이더의 **Color** 속성에 연결합니다.
- 해당 책 지오메트리를 선택하고 Triple Switch 노드에 있는 '**Add Surfaces**'를 클릭합니다(이를 위해서는 Attribute Editor 탭을 복사해야 할 수 있음).
- 몇 가지 색상의 램프를 만들고 이들을 각각 **Triple Switch** 위의 '**In Shape**' 열 아래에 있는 책들 위로 끌어옵니다. 그러면 여러 책 모델에 여러 색상이 할당됩니다. 또 다른 방법은 '**In Triple**' 아래에 있는 필드 하나를 클릭하고 사용하려는 컬러 램프의 이름을 입력하는 것입니다. 먼저 해당 램프의 색상에 따라 램프의 이름을 지정하십시오. 뷔포트에 있는 Shading>Hardware Texturing을 사용하여 진행 상태를 확인할 수 있습니다.



Maya TripleShadingSwitch 노드를 사용하여 여러 책 모델에 할당된 컬러 램프



동일한 Standard Surface 쉐이더 및 Triple Shading Switch 노드에 연결된 여러 컬러 램프 텍스처

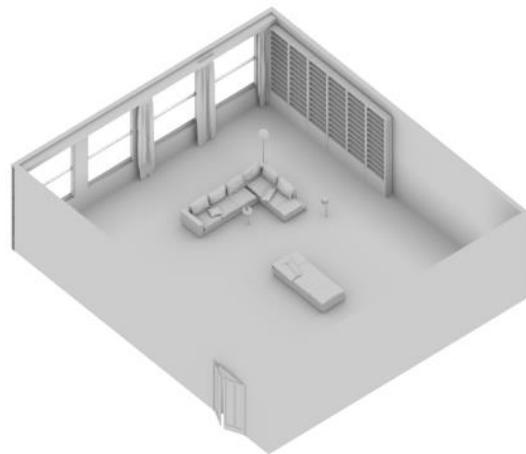
여러 책에 여러 색상을 할당하는 대체 방법은 [User Data](#) 노드를 사용하는 것입니다.

Ambient Occlusion으로 렌더링



후처리 소프트웨어를 사용하여 통합된 Ambient occlusion 및 직접 조명 렌더링

방에 빛을 비추는 대체 기술은 **Ambient Occlusion**을 사용하는 것입니다. 일반적인 워크플로우는 Ambient Occlusion 렌더링을 **GI Diffuse ray**이 없는 Diffuse 렌더링된 이미지로 합성하는 것입니다. 이 방법의 장점은 일반적으로 렌더링이 더 빨라지고 합성할 때 창의적인 통제력이 높아지는 것입니다. 또한 장면에 어떠한 GI Diffuse나 GI Glossy 광선을 추가하지 않아도 됩니다. 하지만 이것은 실질적으로 정확한 방법은 아니며, 최종 모양을 세부적으로 조정하려면 합성 경험이 필요합니다.



Ai Ambient Occlusion 쉐이더

- 먼저 **Ai Ambient Occlusion** 쉐이더를 생성하고 실내 지오메트리에 할당합니다. Ambient Occlusion shader의 **Samples**을 **6** 정도로 높이면 이 쉐이더와 관련된 노이즈가 감소합니다.

- Ambient Occlusion 쉐이더를 사용하여 장면을 렌더링합니다. 다음과 같은 이미지입니다.



Ai Ambient Occlusion 쉐이더만 사용하여 렌더링됨

Direct lighting render

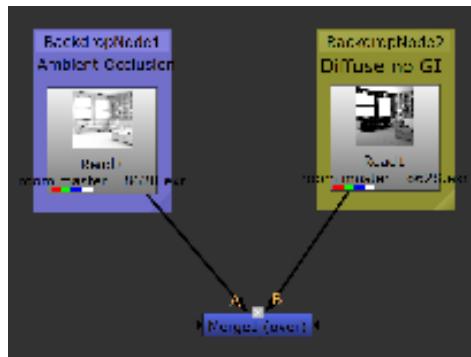
- 이제 직접 조명만 사용하여 장면을 렌더링합니다(이 경우, **Area Light**이 창문 바깥에 위치하고 있지만 **Ai SkyDome** 조명과 같은 기타 조명을 시도할 수도 있음). 하지만 이 경우 GI Diffuse 샘플이 0으로 설정되어 있어야 합니다. 장면을 렌더링하면 반사되는 빛이 없으며 뒷벽이 검은색이 되는 것을 볼 수 있습니다. Nuke 안에 앰비언트 오클루전을 사용하여 반사되는 GI Diffuse 빛이 없는 영역을 '채워'보겠습니다.



Direct Light (Indirect 0)

합성

- Nuke에서 두 렌더링을 모두 엽니다. **merge** 노드를 만들고 **Ambient Occlusion** 렌더링을 '**A**' 입력에 연결하고, **Direct light**를 '**merge**' 노드의 '**B**'에 연결합니다. '**Operation**'이 '**Over**'로 설정되어야 합니다. '**mix**'를 **0.5** 정도로 조절합니다. Ambient Occlusion이 'Direct light' 이미지의 검은 부분에 나타나는 것을 볼 수 있습니다. 실제로는 Ambient Occlusion을 사용하여 반사된 빛의 사용을 '속이고' 있는 것입니다.



확산 렌더링과 병합된 앰비언트 오클루전(GI 없음)

아래 이미지들은 '앰비언트 오클루전' 렌더링을 확산 조명(GI 없음) 렌더링과 병합할 때의 결과를 보여줍니다. 이 때 장점은 렌더링 시간이 빨라지며 GI 노이즈가 발생하지 않는 것입니다.



Direct Light (Indirect 0)



Ambient occlusion과 direct light 렌더링의 병합



Ambient occlusion 렌더링

아래 이미지들은 앰비언트 오클루전 AOV와 병합되어 **flat** 색으로 설정된 **Ai Utility** 쉐이더를 사용할 때의 렌더링 결과를 보여줍니다. 이것은 매우 빠르게 렌더링되지만, 그 결과는 사실적인 것과는 거리가 멍니다.



Ambient Occlusion



Flat색으로 설정된 Ai Utility 쉐이더



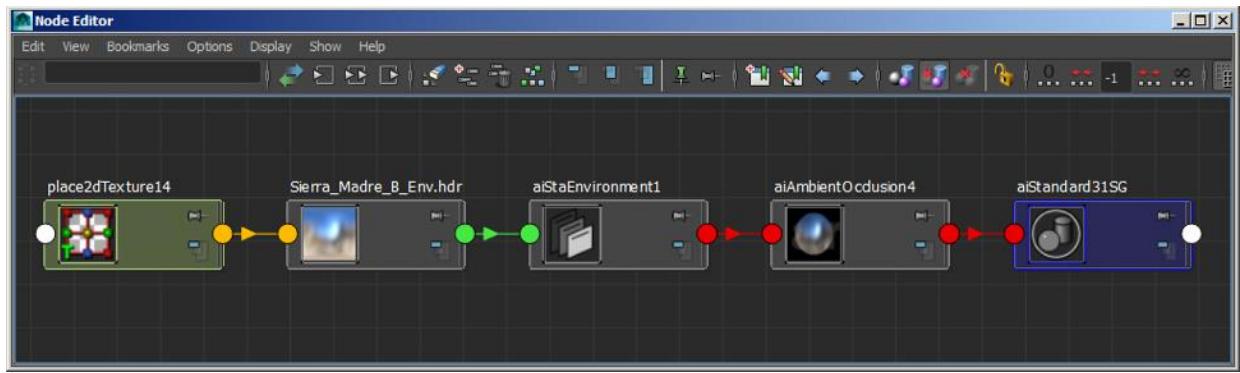
Ambient Occlusion과 Utility 쉐이더 합성

Ambient Occlusion – Environment 쉐이더

StaEnvironment 쉐이더를 사용하려면 SitoA 플러그인을 다운로드해야 합니다. 그런 다음 `sitoa_shaders.dll`(또는 `.so`) 파일을 MtoA 쉐이더 폴더에 복사할 수 있습니다.

Ai Ambient Occlusion 쉐이더의 'White' 속성에 연결되어 있을 때 **Ai StaEnvironment** 쉐이더를 환경 맵으로 사용할 수 있습니다. 그러면 샘플이 표면 법선 대신 반사 방향 주위로 분산되므로 반사 오클루전이 생성됩니다. 이를 설정하려면 다음을 수행해야 합니다.

- **Ai StaEnvironment**를 하나 만들고 **Ai Ambient Occlusion** 쉐이더의 **White** 속성에 연결합니다.
- HDRI 맵을 **Ai StaEnvironment** 노드의 **Tex** 속성에 연결합니다.



Ai Ambient Occlusion 쉐이더에 연결되어 'Ai StaEnvironment' 쉐이더에 연결된 HDRI

Mode

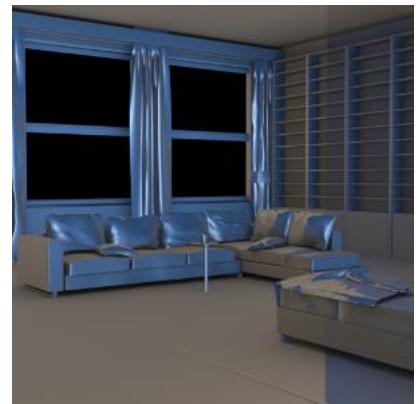
'Tex' 속성에 할당된 텍스처 맵에 대해 여러 매핑 방법을 나타내는 다양한 모드가 있습니다. 이러한 모드는 다음과 같습니다.



Mode 0: Mirrored Ball



Mode 1: Angular map



Mode 2: Cube Cross map



Mode 3: Lat long map



Mode 4: UV 맵은 환경 조회를 수행하는 데 사용

합성

아래 이미지들은 직접 조명 렌더링과 함께 앰비언트 오클루전 '모드' 렌더링을 합성할 때의 결과를 보여줍니다. 적절하게 사용하면 장면에 흥미로운 컬러 조명을 추가할 수 있습니다. 물리적으로는 올바르게 보이지 않아도 최종 결과를 매우 쉽게 렌더링할 수 있습니다.



앰비언트 오클루전과 직접 조명 렌더링의 병합



Mode 1: Angular map



직접 조명, 앰비언트 오클루전, Mode 1 병합

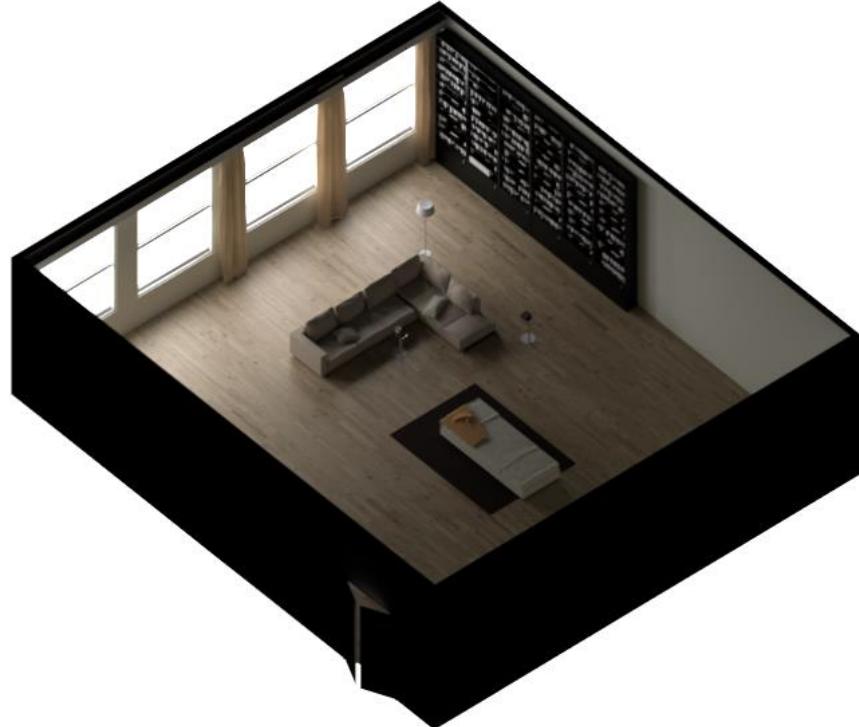
Ai Area Lights



창문을 통해 들어오는 4개의 Ai Area Light

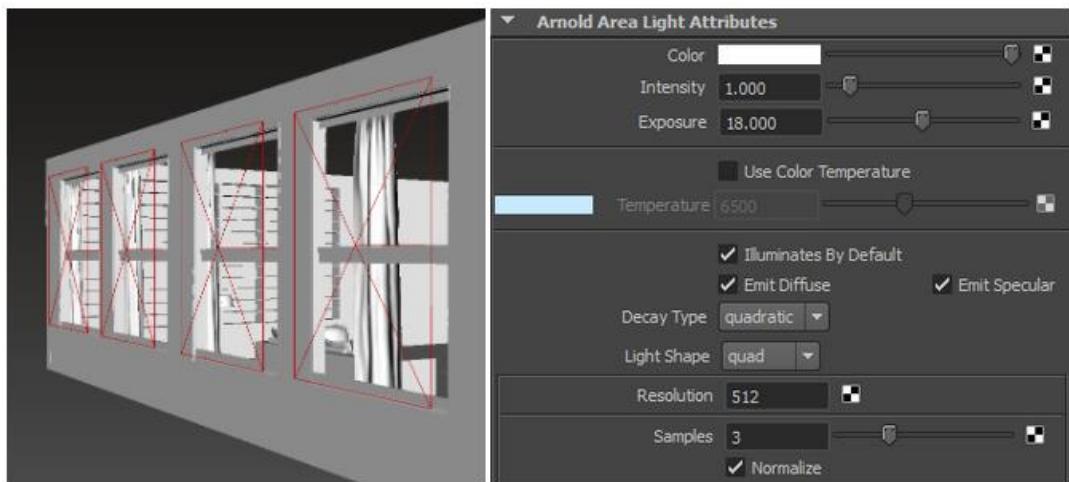
일반적인 실내 조명 방법은 창 바깥의 Area light가 방 안으로 향하도록 배치하는 것입니다. 빛의 크기와 위치에 유의하십시오. 빛이 창 프레임과 너무 가까운 경우 창 근처 영역의 색이 바랜 것처럼 보일 수 있습니다(아래의 톤 매핑 참조). 모든 샘플 이미지에 가벼운 샘플 값인 3이 사용되었습니다.

쿼드 창 영역 조명과 **Directional Light**을 결합하여 skydome light와 비슷한 효과를 얻을 수 있음을 기억하십시오.



- 네 개의 **Ai Area Lights**을 만들고 아래 왼쪽 이미지에 표시된 대로 창 외부에 배치합니다. 장면의 네 개

영역 조명에 사용된 설정은 다음과 같습니다. 최종 장면 렌더링을 위해 **Exposure**는 **18**로 증가되었으며 **Light Shape**은 **quad**로, **Samples**는 **3**으로 설정되었습니다.



Ai Area light shape을 'Quad'로 설정

유리가 있는 창문을 사용하면 Arnold에서 유리 교차점을 처리할 필요가 없도록 유리 앞면에 영역 조명을 배치해야 합니다. 그러나, 창의 모양이 이상하다면 창 바로 뒤에 놓으십시오. 그러면 창 전체가 quad 조명으로 덮이므로 다음 번 옵션이 간단해집니다.

Indirect Diffuse Ray Depth

실제로는 빛이 외부 장면에서보다 실내 장면에서 훨씬 더 많이 산란될 것입니다. 따라서 더 많은 2차 산란 조명을 장면에 적용하려면 이 값을 늘려야 합니다.

아래 이미지는 Render Settings에서 **Indirect Diffuse ray depth**를 증가시킬 때의 효과를 보여줍니다. 렌더링 시간이 빛의 산란 횟수에 따라 선형적으로 증가하는 것을 알 수 있습니다.



Indirect Diffuse Ray Depth 1



Indirect Diffuse Ray Depth 2



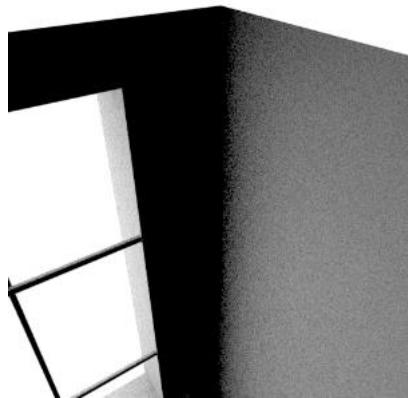
Indirect Diffuse Ray Depth 3

Sampling Noise

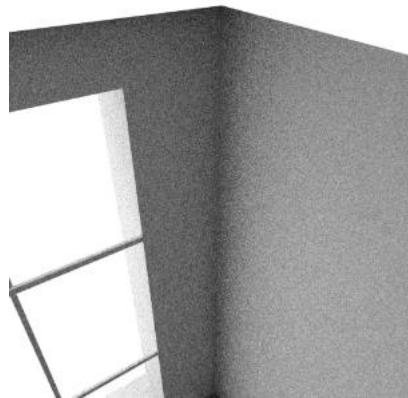
실내를 렌더링할 때는 노이즈가 생기는 것이 흔합니다. 노이즈가 생기는 이유는 빛에 샘플이 부족하거나 **Gi Diffuse Rays**가 충분히 높지 않기 때문입니다.

Gi Diffuse Rays를 **0**으로 설정하면 조명 샘플에 주의를 집중할 수 있습니다. 아래 이미지를 보면 방 구석 그림자의 노이즈가 선명한 것을 볼 수 있습니다. 조명 샘플을 점차 늘리면 렌더링 시간을 짧게 유지하면서 그림자 품질

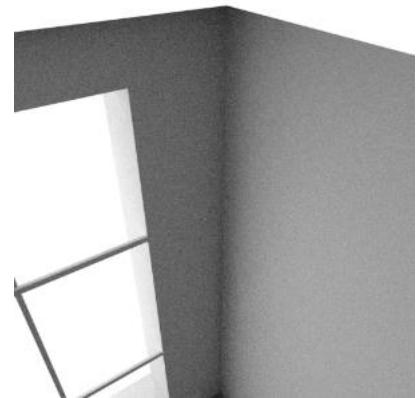
을 높일 수 있습니다. **Lights Samples**를 3으로 증가하기만 해도 그림자 노이즈의 양을 크게 줄일 수 있습니다.



0으로 설정된 **Gi Diffuse**(비활성). 빛의 노이즈를 더 쉽게 식별할 수 있습니다.



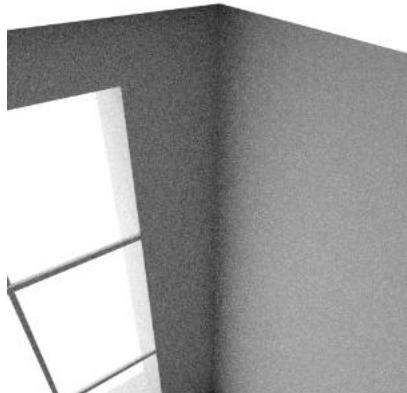
Shadow 노이즈와 **Gi Diffuse** 노이즈(구분이 힘듦).



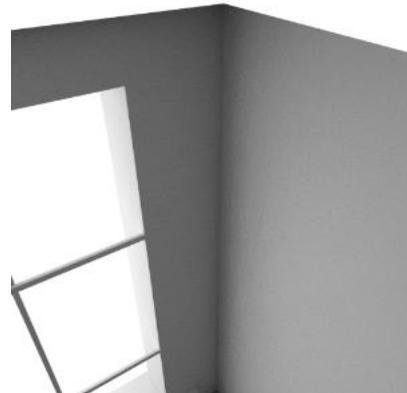
area light samples를 3으로 높이면 노이즈가 없어짐

Diffuse Samples

빛의 샘플을 미세하게 조정했으므로 이제는 **Gi Diffuse** 노이즈를 집중적으로 살펴보겠습니다. 이 경우 **Gi Diffuse** 샘플을 5로 높이면 깨끗해집니다.



Diffuse 샘플 1



Diffuse 샘플 5

Area Light Samples

아래 이미지에서 1개의 큰 영역 조명, 2개의 영역 조명, 4개의 영역 조명을 렌더링하는 것의 차이점을 비교하십시오.



1개의 Area Light: 6분 38초

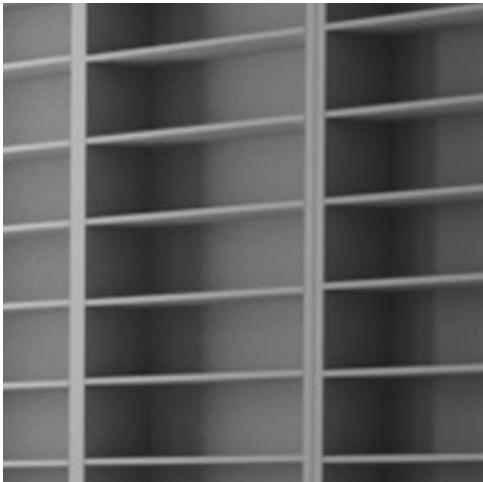


2개의 Area Light: 8분 41초



4개의 Are Light: 12분 30초

각 조명에 대해 Arnold는 9개의 광선을 추적합니다(4개의 조명에는 총 36개). 따라서 작은 조명이 많을수록 1개의 큰 조명을 렌더링하는 것보다 더 오래 걸립니다. 하지만 조명 샘플의 수가 동일하다고 해도(이 경우에는 3개) 1개의 대형 영역 조명만 사용하는 것이 노이즈가 더 심합니다. 이것은 책장 주변의 영역에서 보다 뚜렷합니다.



4개의 영역 조명



1개의 영역 조명(그림자에 노이즈가 더 심함)

Light Decay 필터

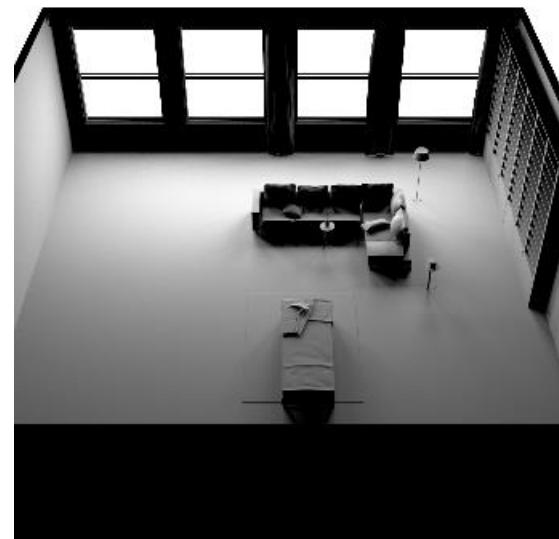
Near Attenuation

창문 주변이 매우 밝고 번져 보입니다. 이것을 수정하는 방법 중 하나는 톤 매핑을 사용하는 것입니다(아래 참조). 이외에 Light Decay 필터를 사용하는 방법도 있습니다.

- 1개의 Light Decay 필터를 4개의 모든 **Ai Area Light**에 연결합니다. 이 필터를 통해 **Use Near Attenuation** 속성을 사용하여 조명의 감소를 제어할 수 있습니다. 아래 조명 이미지를 보면 **Near Start**가 40으로, **Near End**가 200으로 설정되어 있습니다. 바닥 위에서 조명에 가까운 부분이 'Light Decay' 필터가 없는 렌더링과 비교할 때 번짐 현상이 훨씬 적은 것을 알 수 있습니다(효과가 더 잘 보이게 하기 위해 GI Diffuse를 0으로 설정함).



Light Decay 필터 없음(창 주변이 번져 보임)



Light Decay 필터를 사용하여 이런 현상이 향상됨

Far Attenuation

'Far Attenuation' 속성은 일반적인 decay와 같은 효과가 있지만 시작 및 종료 거리를 지정할 수 있습니다. 이렇게 하면 '종료' 거리를 설정하여 조명이 방을 채우는 방법을 좀 더 조절할 수 있습니다. 아래 간단한 실내 장면은 Light Decay 필터의 "Far End" 속성을 변경할 때의 효과를 보여줍니다. 이 효과가 바닥의 정반사에 어떤 영향을 주는지 살펴보십시오.



Far End 0.05



Far End 1



Far End 2

톤 매핑

창문 주변의 번진 부분을 수정하는 쉬운 방법은 후처리 소프트웨어를 사용하여 이미지를 '톤 매핑'하는 것입니다.

- 장면을 32비트 EXR 파일로 렌더링하십시오. 그러면 여러 시도를 위한 충분한 톤 범위가 제공됩니다.



톤 매핑 전(창문 주변이 번짐)



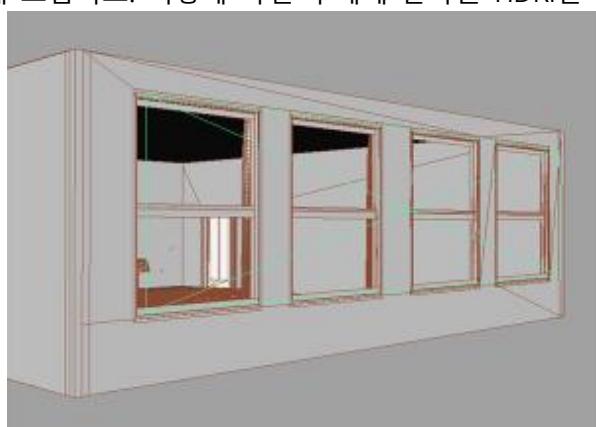
톤 매핑 후

HDRI와 Area Lights

방 안에 보다 자연스러운 색상을 추가해볼 수 있습니다. HDRI와 Area light를 함께 사용하여 가능합니다.

- HDRI를 Ai Area 조명의 **Color** 속성에 연결합니다.

정상 범위 내까지 조명의 강도를 줄이면 HDRI의 효과를 충분히 볼 수 있습니다. 하지만 텍스처 맵이 조명에 올바르게 매핑되었는지 확인하기 힘들 수 있습니다. 이 효과의 시각화를 향상시키려면 반사 구체를 만들고 창 앞에 배치해 보십시오. 이렇게 하면 구체에 반사된 HDRI를 볼 수 있습니다.

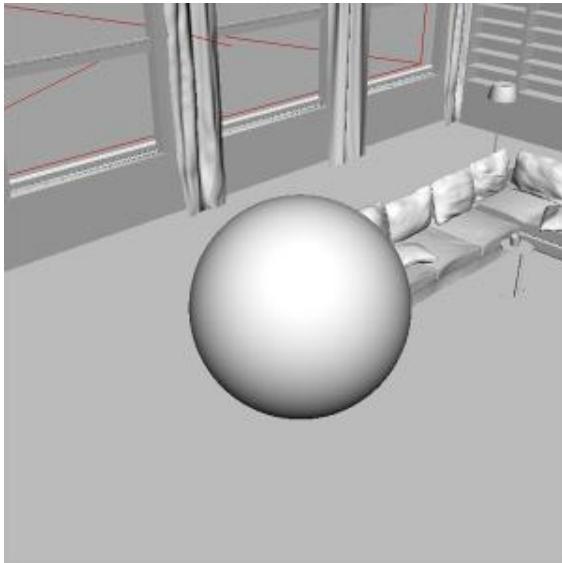


모든 창에 맞게 크기가 조절된 단일 Ai Area 조명

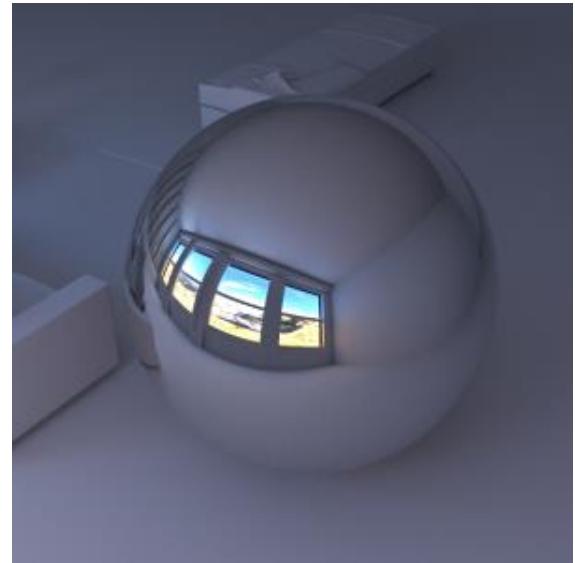


Ai Area 조명에 연결되어 있는 HDRI로 조명된 방

- 구체를 만들고 Ai Standard 재료를 할당합니다. **Diffuse Weight**를 0으로 줄이고 **Specular Weight**를 1로 높이고 **Specular Roughness**를 0으로 줄입니다. 장면 내에서 구체를 렌더링합니다. 아래 오른쪽 이미지에서처럼 구체에 HDR이 반사된 것을 볼 수 있습니다.

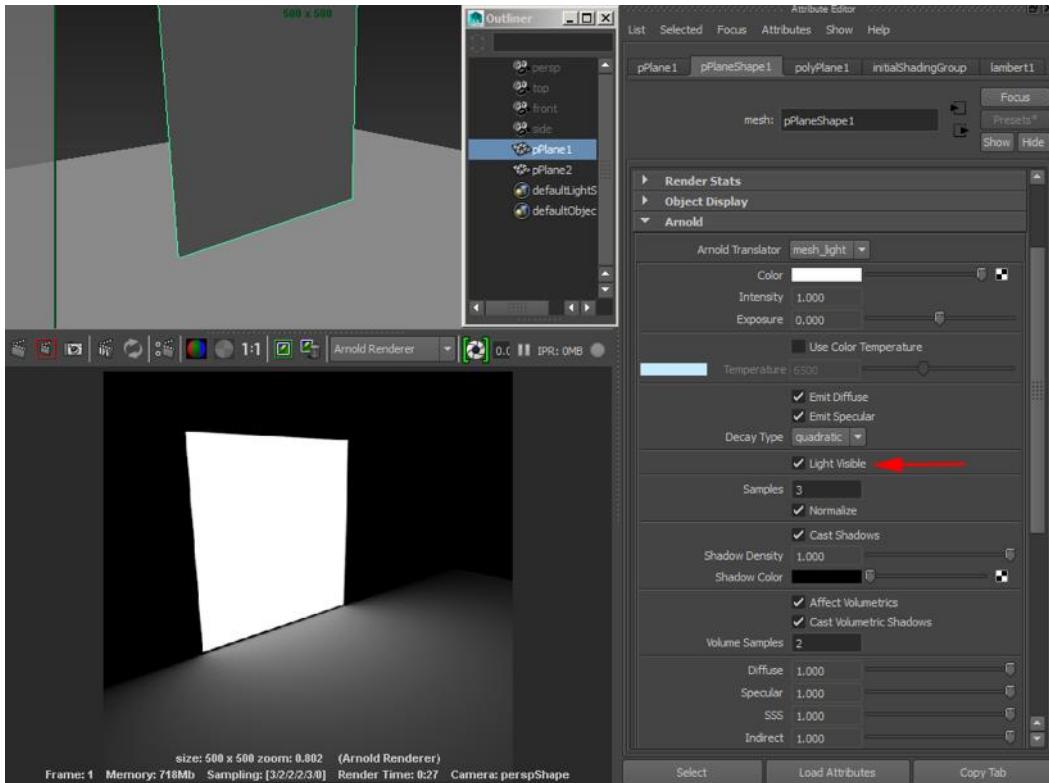


창문 앞에 배치된 반사 구체



구체 내에서 영역 조명에 매핑된 HDRI 반사를 볼 수 있습니다.

Arnold 조명을 카메라에 표시되게 하는 것은 불가능합니다. 하지만 **Ai Mesh Light**에는 '**Light Visible**' 옵션이 있습니다.



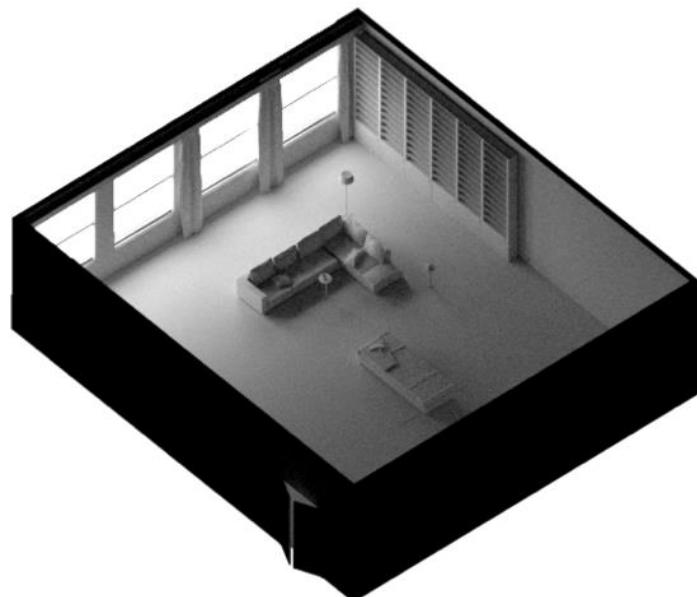
'Light Visible'이 활성화된 Mesh Light(다면)

Emissive Geometry



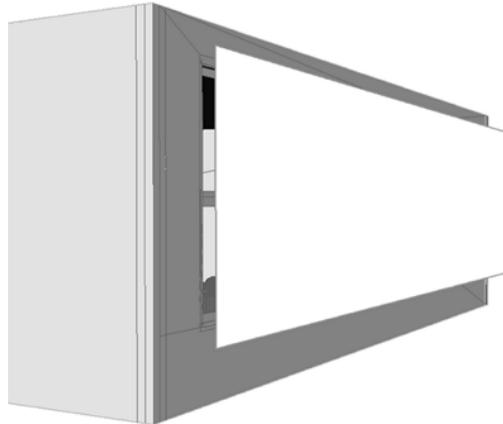
Standard Surface 쉐이더의 방출 속성에 연결된 HDRI

Standard Surface 쉐이더에 **Emission**을 추가하여 지오메트리가 '빛을 방출'하도록 만들 수 있습니다. 따라서 창문 바깥에 방출 지오메트리를 배치하여 방을 비추게 할 수 있습니다. 하지만 이 방법은 권장하지 않습니다. 평면에 적용된 Standard Surface 쉐이더에서 방출되는 빛은 확산 광선의 빛만 포착하므로 충분하지 않아 노이즈가 매우 심합니다. 이런 면에서 텍스처가 있는 영역 조명(quad로 설정됨)은 언제나 탁월합니다.



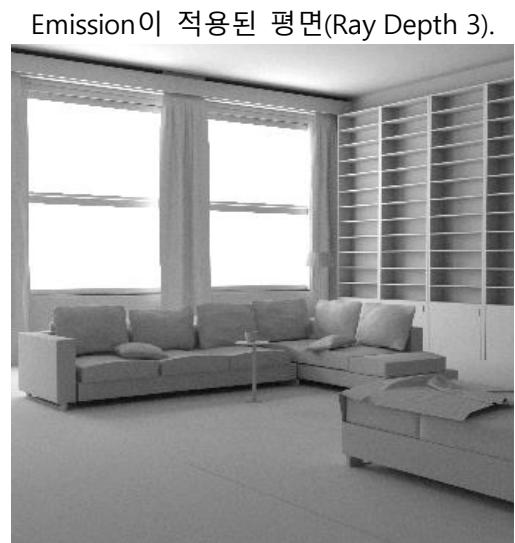
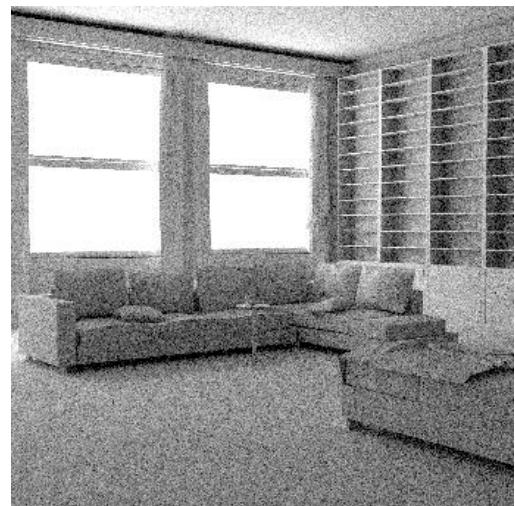
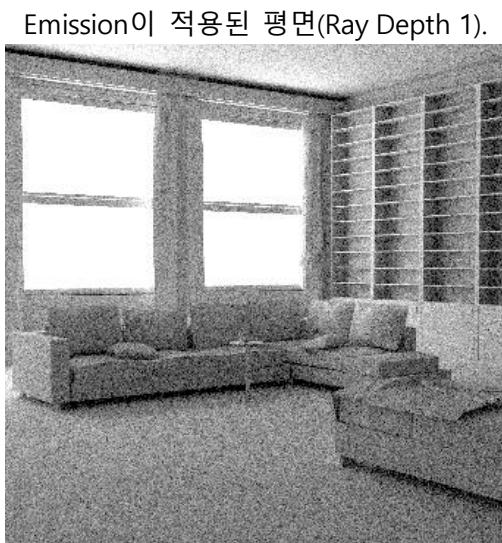
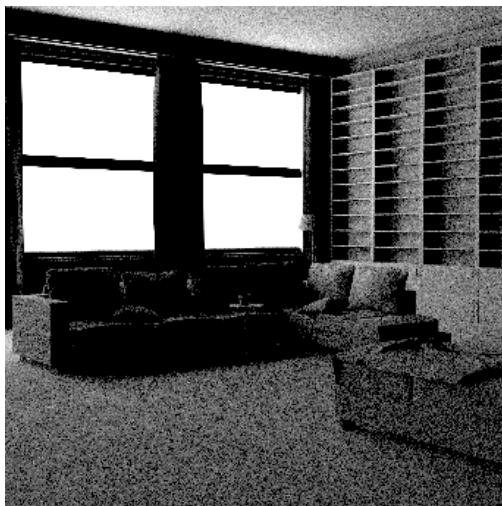
Emission Scale이 1로 설정된 평면 바깥 창

- 먼저 다각형면을 만들고 방 바깥의 창문을 덮도록 배치합니다.



- Standard Surface** 쉐이더를 평면에 할당합니다. **Emission Scale**을 **8**로 높입니다. 순수하게 방출만 하면 되므로 **Base Weight** 값을 **0**으로 줄이는 것이 좋습니다.

아래 이미지에서 볼 수 있듯이 이 방법을 사용하면 많은 노이즈가 생길 수 있습니다. Gi Diffuse Samples의 수를 크게 높이면 노이즈는 줄어들지만 렌더링 시간은 늘어납니다.



HDRI가 적용된 Emissive Plane

Standard Surface 쉐이더의 **Emission Color**에 HDRI 매핑을 추가하는 것은 장면을 조명할 수 있는 또 다른 옵션입니다. 아래 이미지에서는 창문을 통한 뷰에 HDRI가 사용되었습니다.



바깥 창문 평면에 할당된 Emission Color of Standard에 할당된 HDRI



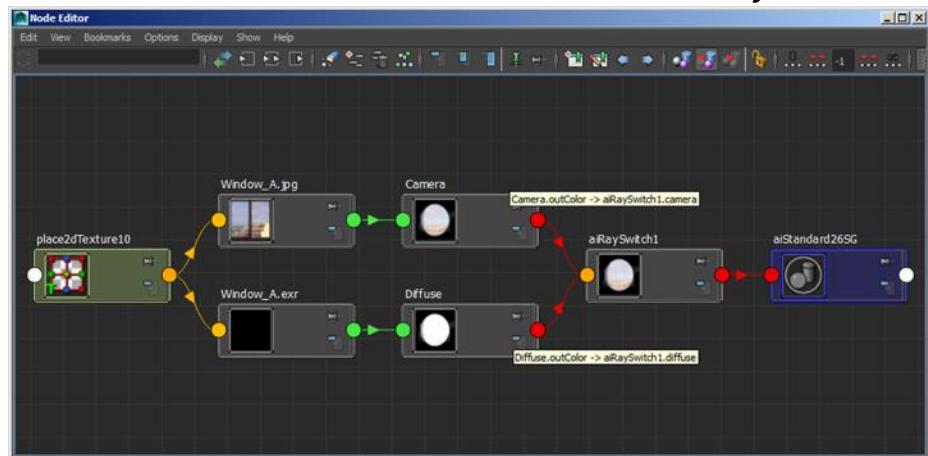
창문 내의 HDRI가 뭉개짐

방출면에 할당하면 창문의 HDRI가 뭉개져 보입니다. 이를 해결하는 방법은 **Rayswitch** 쉐이더를 사용하는 것입니다. HDRI 이미지와 함께 **Standard Surface** 쉐이더를 사용하여 **Diffuse GI Rays**을 제어할 수 있으며 JPEG와 동일한 이미지가 포함된 다른 **Standard Surface**를 *camera rays*에 대해 사용할 수 있습니다.



방출면 지오메트리에 할당된 Rayswitch 쉐이더. 창문의 JPEG 텍스처는 카메라 광선에 보이며, HDRI는 확산 광선에서 보입니다.

방출면에 할당하면 창문의 HDRI가 뭉개져 보입니다. 이를 해결하는 방법은 **Ai Rayswitch** 쉐이더를 사용하는 것입니다. HDRI 이미지와 함께 **Ai Standard Surface** 쉐이더를 사용하여 **Diffuse GI rays**을 제어할 수 있으며 JPEG와 동일한 이미지가 포함된 다른 Standard Surface를 **camera rays**에 대해 사용할 수 있습니다.



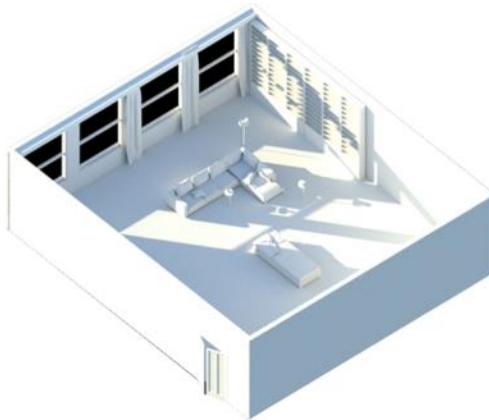
방출면 지오메트리에 할당된 Rayswitch 쉐이더. 창문의 JPEG 텍스처는 카메라 광선에 보이며, HDRI는 확산 광선에서 보입니다.

Physical Sky 조명 적용



Physical Sky는 장면에서 정확한 조명을 얻을 수 있는 매우 쉬운 방법입니다. 'Elevation' 및 'Azimuth' 설정만 조절하면 흥미로운 'golden hour' 조명 효과를 생성할 수 있습니다. 이것은 다음 단계를 통해 매우 간단하게 실행할 수 있습니다.

실내 장면에는 `skydome_light`의 사용을 권장하지 않습니다. 이 조명은 야외 장면용이며 배경에서 구형 돔으로 표현됩니다. 여러 중요도 조명 샘플링이 광선을 이 돔의 특정 방향으로 추적합니다. 하지만 실내 장면에서는 이러한 광선이 대부분 물체에 닿기 때문에 조명으로부터의 효과가 전혀 없어 노이즈가 발생합니다. 이러한 상황에서는 **quad light** 및/또는 **directional lighting**이 더 나은 옵션이 될 수 있습니다.



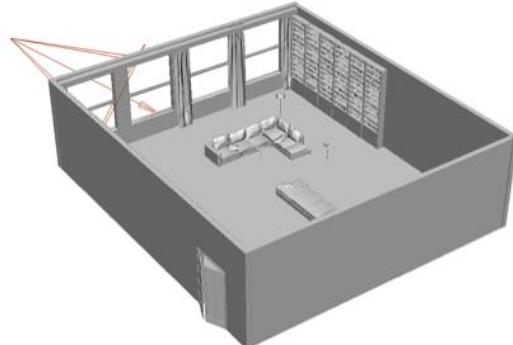
Sky Dome light에 연결된 Physical Sky 노드로 조명이 적용된 장면

- 먼저 **Physical Sky** 쉐이더를 생성합니다.
- SkyDomeLight**을 만들고 **Physical Sky**를 **SkyDomeLight**의 색상 속성에 연결합니다.
- IPR은 장면을 렌더링하고 Elevation 및 Azimuth 설정을 조절하여 태양빛이 창문을 통해 빛나고 오른쪽 맨 끝의 벽과 바닥에 닿게 합니다. 이 경우에는 **Elevation**이 상당히 낮으며(15) 태양빛이 먼 벽에 닿도록 **Azimuth**의 각도를 조절했습니다(143).

Atmosphere Volume

SkyDome 조명은 체적/대기 효과에 영향을 주지 않습니다. 하지만 장면에 스포트라이트를 추가하여 이것을 속일 수 있습니다.

- 창문을 관통하도록 스포트라이트를 배치합니다. Physical Sky 햇빛의 각도와 일치하도록 회전합니다. 이제 장면에 햇빛을 표현하는 조명 2개가 있는 것을 볼 수 있습니다. 이것은 우리가 원하지 않는 2가지 유형의 그림자를 생기게 합니다(스포트라이트는 체적 산란 효과 생성에만 사용됨). 이 효과가 명백한 경우 physical_sky 셰이더에서 'enable_sun'을 활성화하고 스포트라이트를 사용하여 직사광선을 시뮬레이션할 수 있습니다.



창문을 통과하는 스포트라이트

신이 내린 빛' 효과를 연출하고 싶다면 스포트라이트에 **Gobo** 필터를 추가해 보십시오.

- 스포트라이트를 선택하고 Gobo 필터를 연결합니다.
- 해당 gobo를 선택하고 **Slide Map**에 **noise map**을 연결합니다.
- Atmosphere Volume**을 활성화하고 **Density**를 1로 높입니다.
- 마지막으로 스포트라이트의 **Color Temperature**를 약 4500으로 변경하면 따뜻한 햇살이 창을 통해 쏟아지는 것을 표현할 수 있습니다.



장면에 Atmosphere Volume이 추가된 스포트라이트



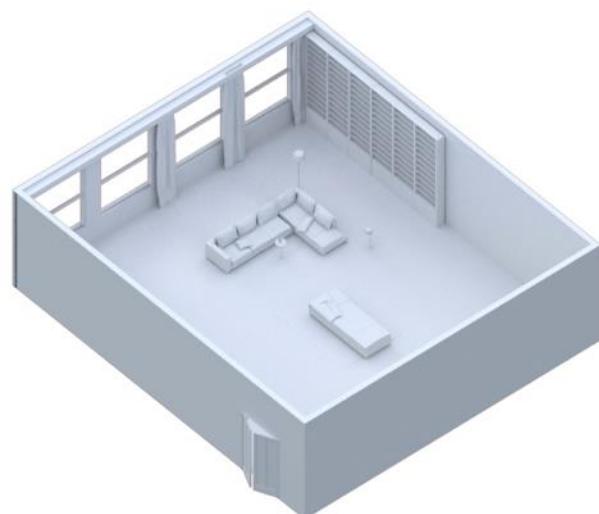
Sky Dome 조명에 연결되어 Physical Sky 조명이 적용된 실내의 또 다른 예

Ai Sky Dome Light 조명 적용



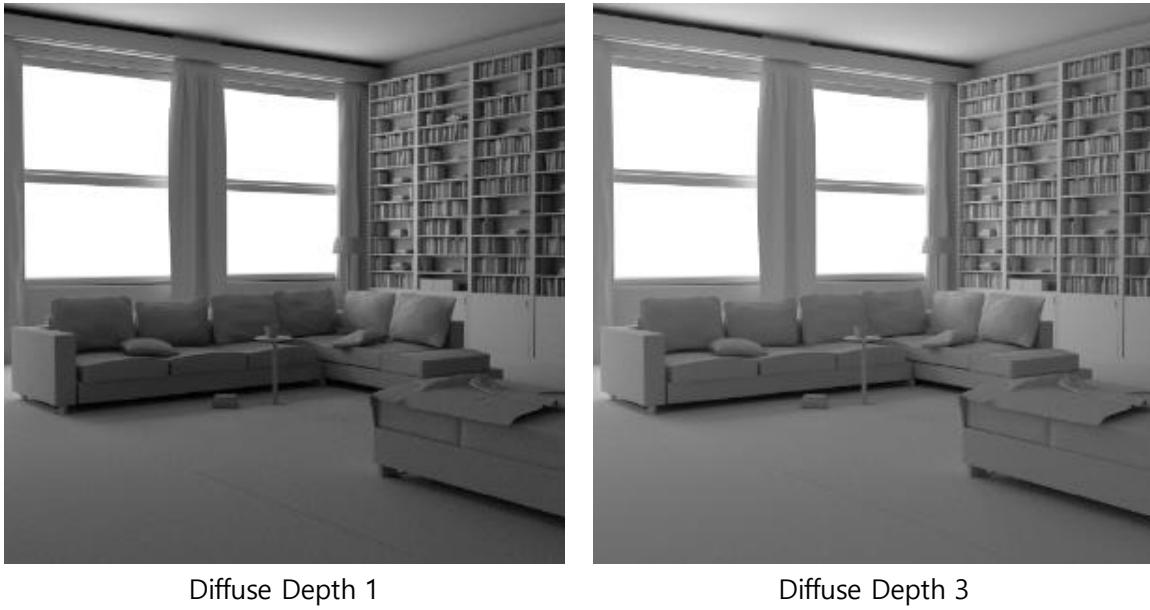
또한 **Ai Sky Dome Light**을 사용하여 실내의 조명을 표현할 수 있습니다. 하지만 이 조명은 여러 방향에서 조명을 필요로 하는 야외 장면을 위해 설계되었습니다. 장면의 수평선에 구형의 둑이 있다고 가정하며 물체들이 광선을 추적하여 빛을 샘플링합니다. 실내에서는 이러한 광선이 대부분 하나의 물체에 닿기 때문에 전혀 효과가 없습니다. 이것을 사용하여 실내 조명을 표현하는 경우 장면에 간접적인 노이즈가 너무 많이 발생하므로 대부분의 경우 권장하지 않습니다.

실내 장면에는 `skydome_light`의 사용을 권장하지 않습니다. 이 조명은 야외 장면용이며 배경에서 구형 둑으로 표현됩니다. 여러 중요도 조명 샘플링이 광선을 이 둑의 특정 방향으로 추적합니다. 하지만 실내 장면에서는 이러한 광선이 대부분 물체에 닿기 때문에 조명으로부터의 효과가 전혀 없어서 노이즈가 발생합니다. 이러한 상황에서는 **quad lights** 및/또는 **directional lighting**이 더 나은 옵션이 될 수 있습니다.



- Hypershade 창 또는 MtoA 쉘프에서 Ai Sky Dome Light을 만듭니다.

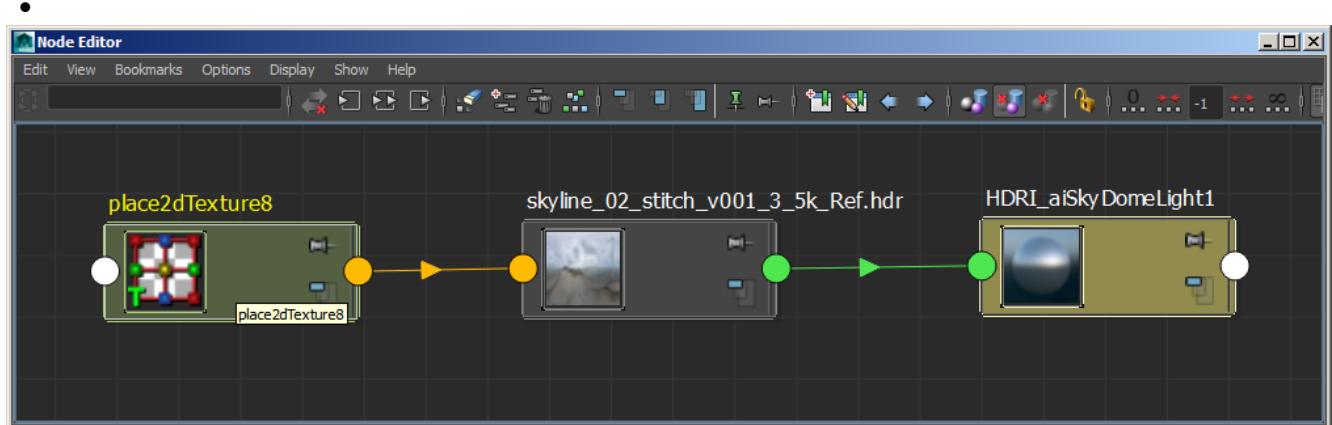
다음 이미지들은 Color가 white로 설정되어 있는 Ai Sky Dome Light을 보여줍니다. Ai Sky Dome 조명 샘플이 3으로 설정되고 GI Diffuse 샘플이 8로 설정되어 있으면 이미지가 깨끗해지는 것을 볼 수 있습니다.



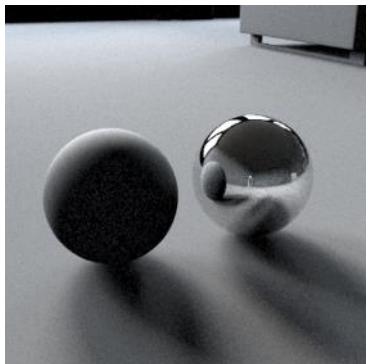
HDRI

또한 HDRI를 사용하여 방 조명을 표현할 수도 있습니다. 이것은 방 안에 자연스러운 조명 효과를 주며 일부 HDRI는 이러한 유형의 실내 조명 상황에서 다른 것보다 효과가 뛰어납니다. 창문을 통해 더 나은 방향성 빛이 들어오게 하려면 강력한 햇빛을 가진 HDRI를 사용하는 것이 좋습니다.

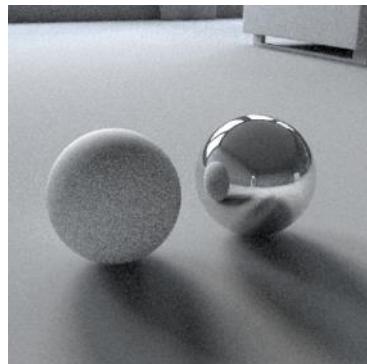
- Maya 파일 텍스처를 만들고 HDRI를 엽니다. 이 경우에는 강력한 방향성 일광을 가진 HDRI를 사용했습니다. 따라서 창문을 통해 보다 강력하고 날카로운 방향성 그림자가 생성됩니다. HDRI를 Ai Sky Dome 조명의 **Color** 속성에 연결합니다.



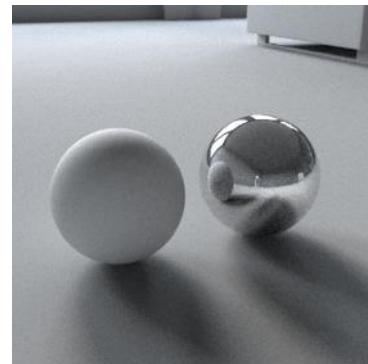
Diffuse 샘플을 높이면 이미지의 노이즈는 감소하지만 렌더링 시간이 늘어나며, 8개의 **Diffuse** 샘플의 경우에도 이미지에 노이즈가 있습니다.



Diffuse Samples 0



Diffuse Samples 3



Diffuse Samples 8

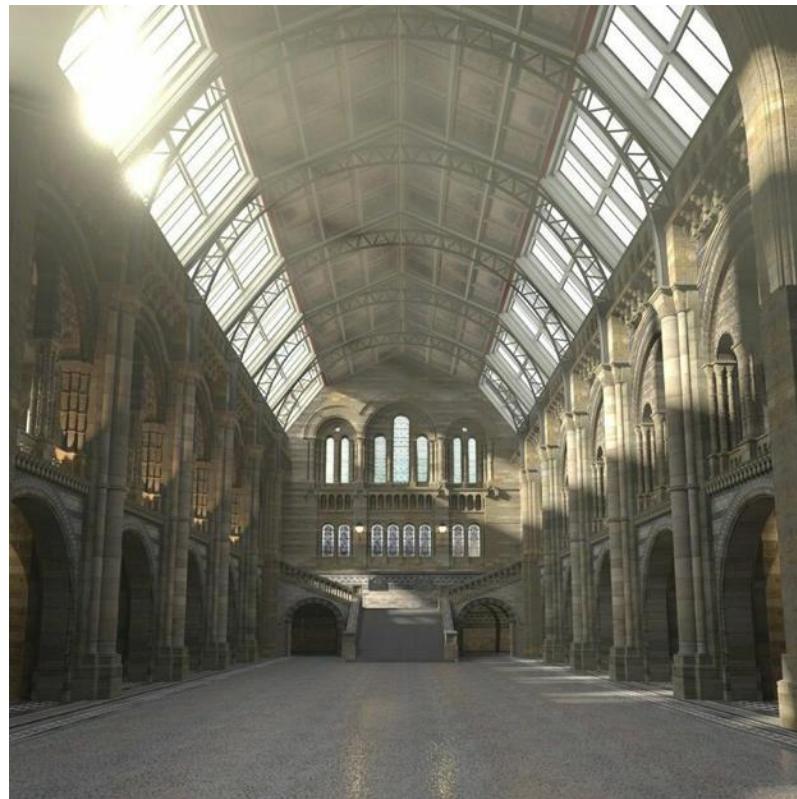


Diffuse Samples 3 3분



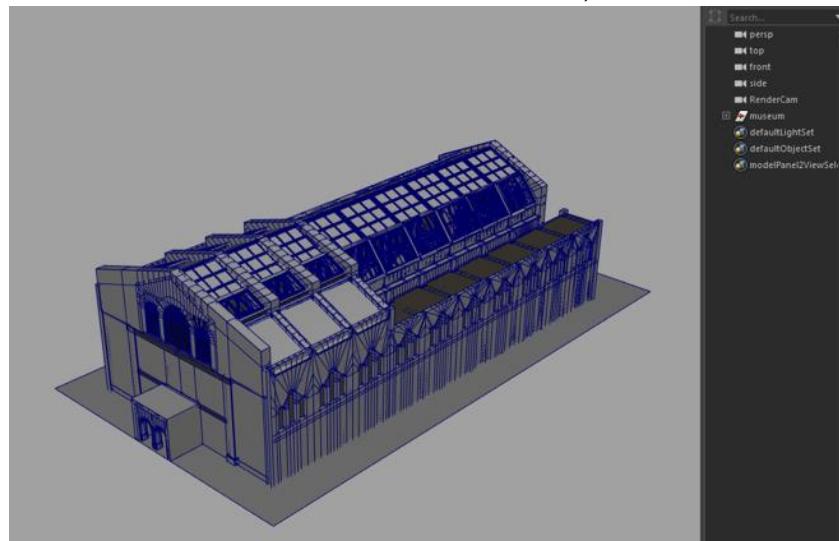
Diffuse Samples 8 14분

자연사 박물관 조명 효과



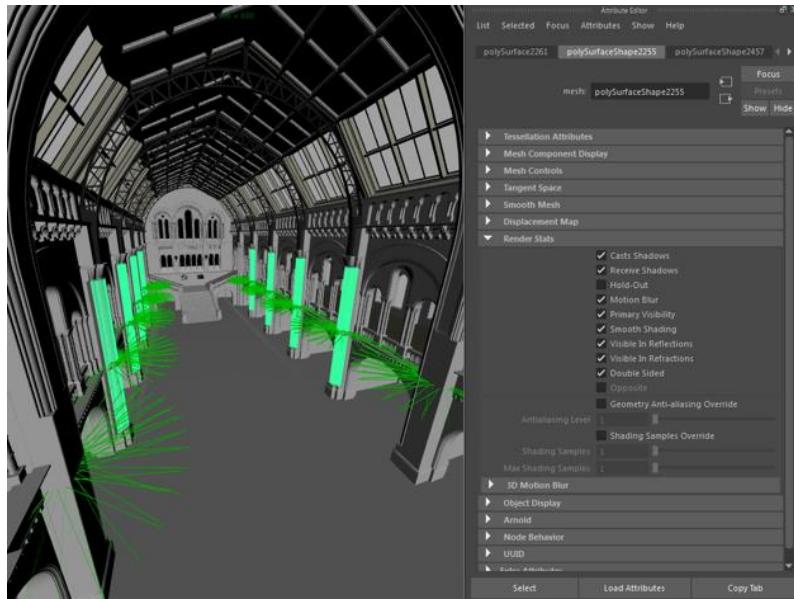
이 자습서에서는 Maya의 기본 조명과 Arnold의 독점 조명을 사용하여 사실적인 실내 장면의 조명을 표현하고 음영 처리하는 방법에 대해 설명합니다. 여기에는 Arnold로 **Volumetric Light** 효과를 설정하는 방법이 포함됩니다. 체적 조명이란 공기 중의 먼지 입자에 닿으면 산란되는 빛에 의해 발생하는 광선의 가시 효과입니다. 또한 발광체를 사용하는 방법을 살펴보고 Arnold의 조명 도구를 사용하는 방법과 비교해봅니다. 마지막으로 Arnold의 카메라 렌즈 옵션과 렌더링 설정을 최적화하는 방법을 살펴 보겠습니다.

- 먼저 Maya에서 실내 장면 모델을 엽니다. 이 자습서에서는 런던의 자연사 박물관 모델을 사용합니다. 이것은 <http://www.3drender.com/challenges/>에서 다운로드할 수 있습니다(이 웹 사이트는 Solid Angle과는 관련이 없지만 렌더링의 여러 측면에서 유용한 정보원입니다.).



- 이 모델은 다른 3D 소프트웨어 패키지에서 생성되었으므로 기하 법선이 모두 올바른 방향을 향하고 있는지 확인하십시오. 그렇지 않으면 텍스쳐링이 일관되지 않을 수 있습니다(범프 맵 값이 반전될 수 있음).

형상에 대한 **Render Stats**에서 '**Visible in Reflections**' 및 '**Visible in 'Refractions'**' 스위치가 켜져 있는지 확인하십시오. 때로는 다른 애플리케이션에서 형상을 가져올 때 이 기능을 중단할 수 있습니다.



- 처음에는 조명 프로세스에 집중하기 위해 이들 창을 숨깁니다. 나중에 쉐이딩 과정 중에 창을 추가할 수 있습니다(창에 대해 **Opaque**를 비활성화해야 함).
- 모든 형상을 선택하고 Hypershade 창에서 **Ai Standard** 쉐이더를 할당합니다. 이 쉐이더를 테스트 조명의 기반으로 사용할 것입니다.

Ai Skydome Light



- **Skydome light**을 생성합니다. Color를 mid-blue로 변경합니다. 장면에서 형상을 선택하는 데 문제가 있다면 **Viewport** 아래에 있는 **Sky Radius**를 줄이면 됩니다. 이 수치를 줄여도 카메라에서 볼 수 있습니다.
- 기본 렌더링 설정을 사용하여 장면을 렌더링합니다. 이 단계에서는 노이즈를 신경 쓰지 마십시오. 최종적으로 조명을 추가하고 렌더링 설정을 향상시킬 것입니다. 너무 어두운 것 같다면 skydome light의 강도를 높이십시오. 그러면 실내가 밝아지고 아래와 같이 렌더링됩니다.

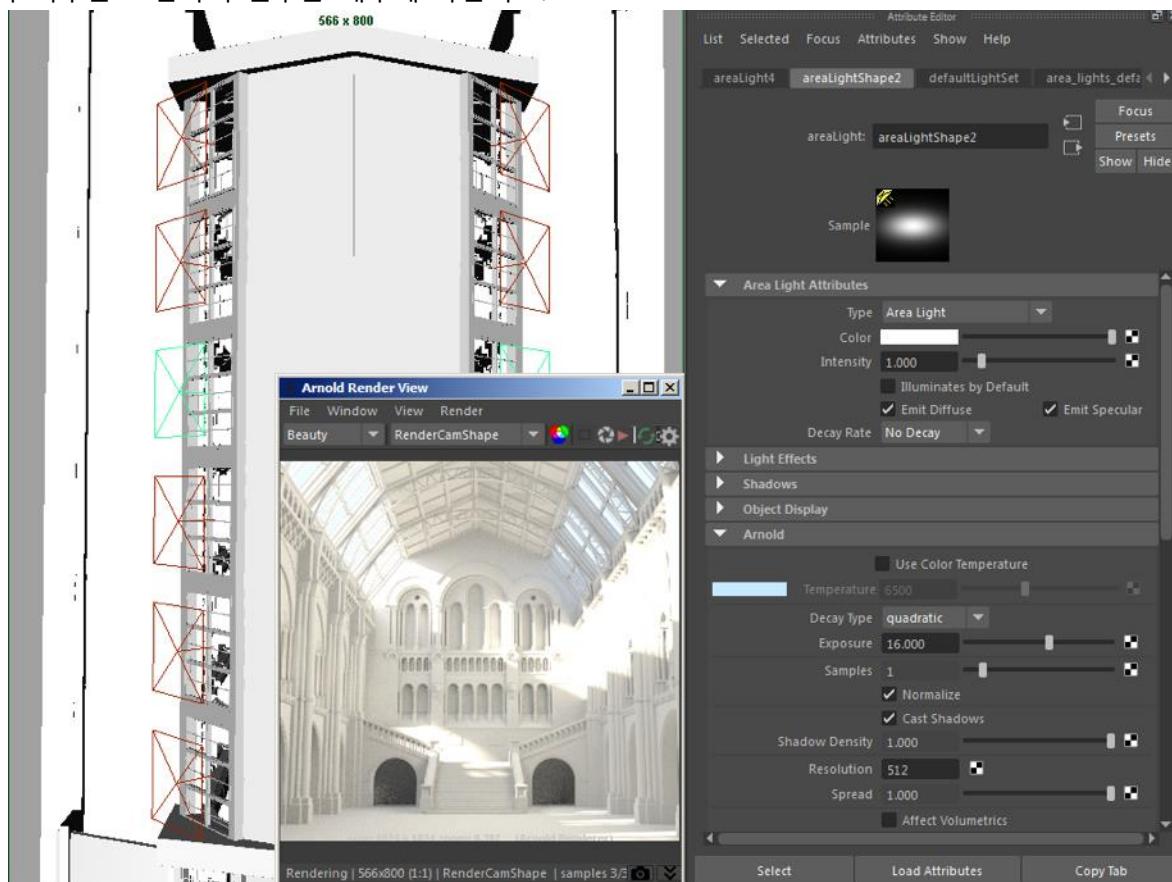


- **Directional Light**을 생성하고 지붕 창을 통해 들어오도록 회전 시킵니다. 색상을 따뜻하고 밝은 노란색으로 변경하고 **Exposure**을 5 또는 6으로 증가시키십시오. **Angle**를 1로 증가시키면 방향 조명 그림자의 가장자리가 부드럽게 표현됩니다. **Use Color Temperature**를 활성화하고 **Temperature**를 5500 정도로 변경합니다.
- 테스트 렌더링에 IPR을 사용합니다. 너무 투박해 보이더라도 걱정하지 마십시오. 재료와 질감을 추가한 후에 이러한 값을 수정해야 합니다.

Ai Area Lights

이제 창문 밖에 조명을 만들어서 내부의 어두운 영역에 빛을 비추고자 합니다. 이러한 조명은 하늘에서 비추는 빛을 의미합니다.

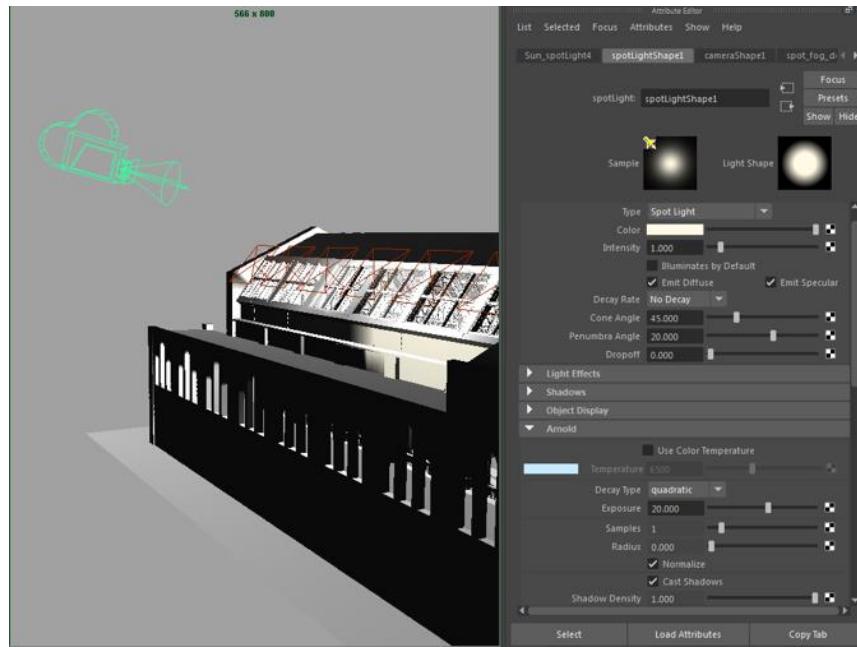
- **Ai Area Light**을 생성합니다. **Light Shape**을 **Quad**로 설정합니다. 조명을 지붕 창 중 하나의 바깥으로 확장합니다. 조명을 복사하여 모든 지붕 창을 덮을 수 있도록 배치합니다.(설정을 변경해야 할 경우 인스턴스화하면 시간이 절약됨). 색상을 하늘색으로 변경하고 **Intensity** 및 **Exposure**를 높여서 위와 같이 실내에서 어두운 그림자의 일부를 채우게 하십시오.



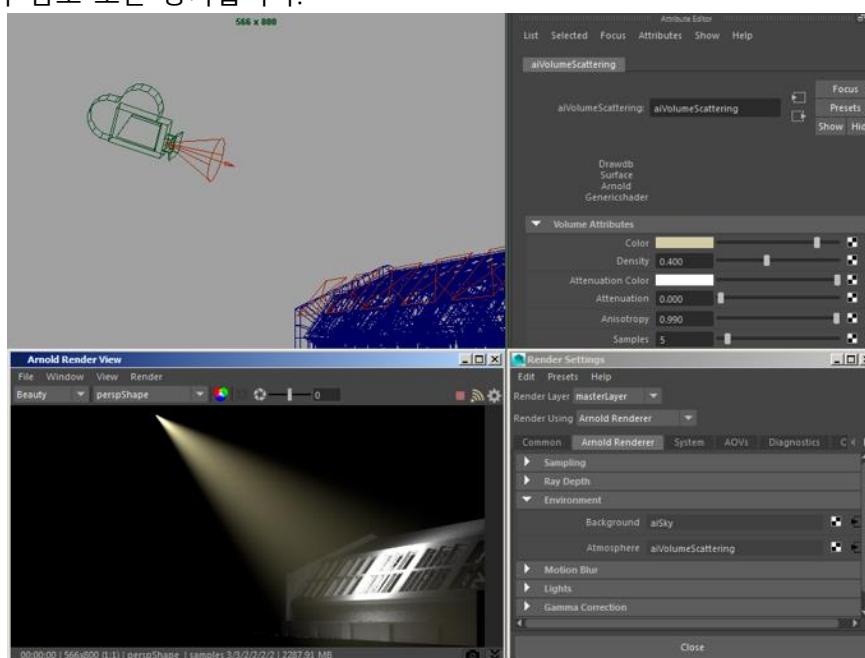
Atmosphere Volume

체적 조명 효과를 만들기 위해 스포트라이트를 사용하겠습니다.

- 스포트라이트를 만들고 지붕 창 바깥에 배치합니다. 앞서 방향 조명과 동일한 방향으로 배치합니다. 색상을 약간 어두운 노란색으로 변경합니다. 장면에서 Atmosphere Volume의 효과를 조절하려면 Cone Angle을 변경해야 합니다. 해당 스포트라이트의 Arnold 속성에서 **Exposure**를 20과 같이 높은 값으로 올립니다. Atmosphere Volume이 장면에서 올바르게 작동하는 경우 이 값을 낮출 수도 있습니다.

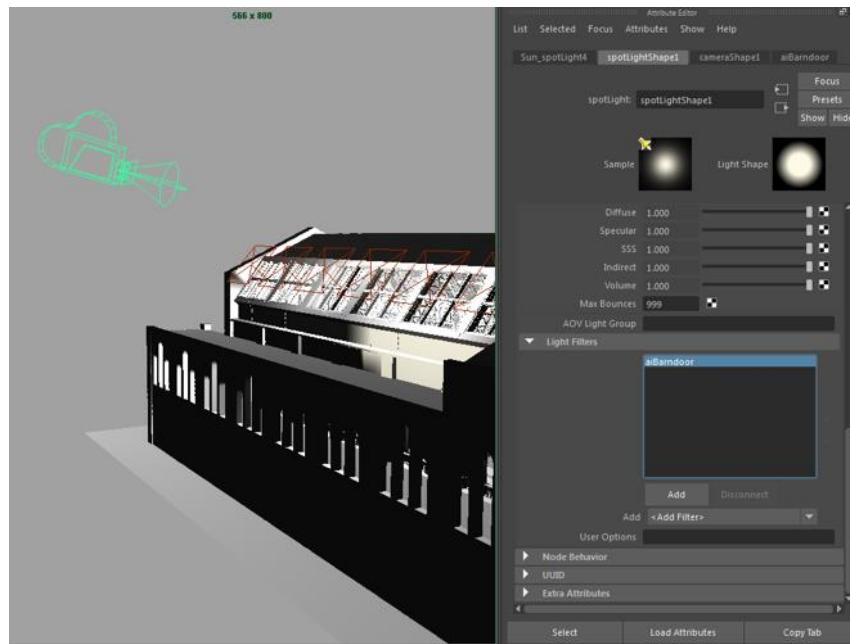


- Volumetric 안개를 켜려면 Render Settings로 가서 **Environment** 아래 **Atmosphere** 오른쪽에 있는 메뉴를 클릭합니다. **Atmosphere Volume**을 선택하면 속성 편집기에 나타납니다. 이제 IPR을 시작할 수 있습니다. 색상을 약간 짙은 노랑으로 변경하고 안개의 밀도 값을 낮춰서 좀더 미세하게 표현합니다. **Attenuation** 값을 살짝 낮추면 거리에 따라 안개가 점차적으로 감소합니다. 장면에서 체적 안개의 전반적인 효과를 줄이려면 기본 **Camera Contribution** 값을 낮춰보십시오. 이들 값을 변경하면 장면 내에서 안개의 밝기가 감소 또는 증가합니다.

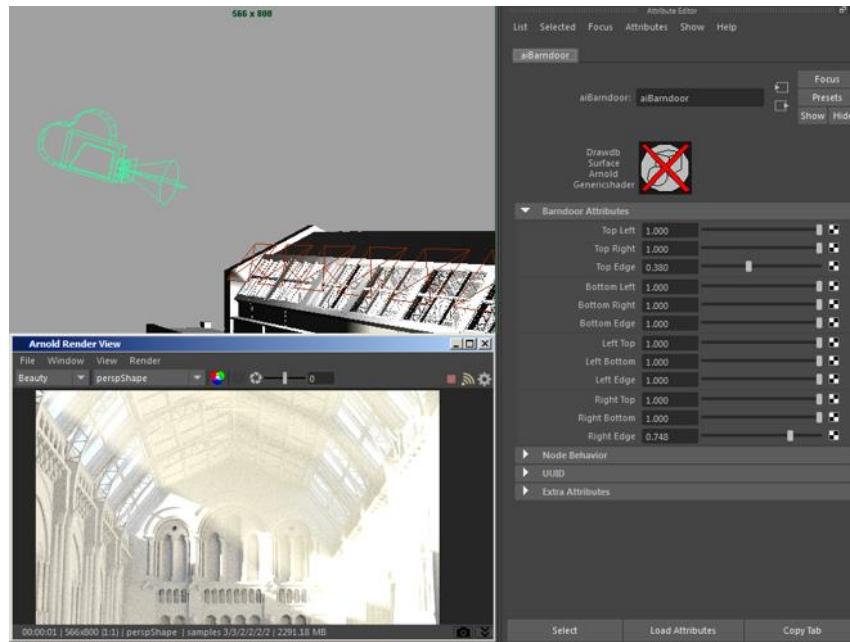


Barndoors

- 스포트라이트의 효과와 체적 안개가 장면에서 반응하는 방식을 쉽게 제어하기 위해 **Ai Barndoors**라는 이름의 조명 필터를 사용할 수 있습니다. 해당 스포트라이트의 속성 편집기에서 아래쪽의 **Light Filters**로 스크롤 이동하고 Add filter 메뉴에서 **Ai BarnDoor**를 선택합니다.

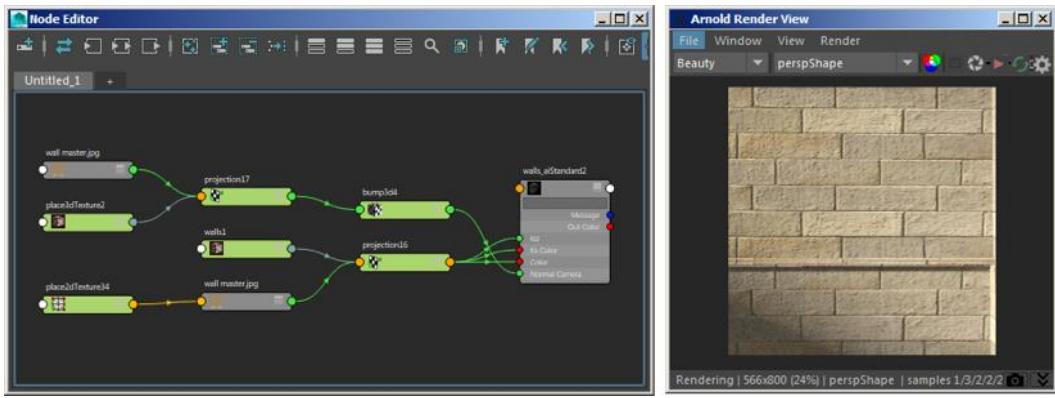


- 이 Barndoors 필터는 4개의 barndoors 플랩이 있습니다. 각 barndoors 플랩에는 3개의 파라미터가 있습니다. 처음 2개의 파라미터는 플랩의 양쪽 끝을 조명의 면을 가로 질러 배치합니다. 세 번째 파라미터(Edge)는 가장자리의 부드러움을 조절합니다. 이 경우에는 해당 barn door의 Right Bottom 및 Right Edge를 닫아서 체적 안개 광선의 가장자리를 단단하게 만들 수 있습니다.

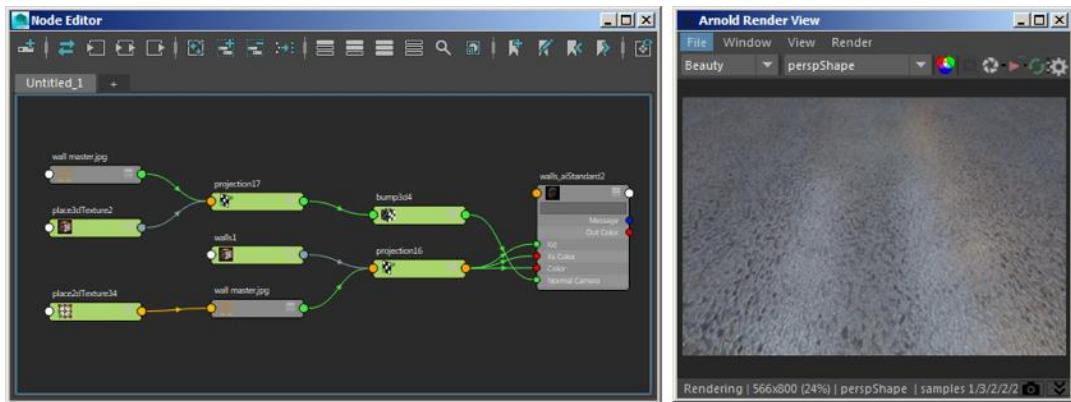


이제 일부 재료를 추가하면서 장면이 어떻게 보이는지 확인할 수 있습니다. Standard Surface 재료를 생성하고 장면 내 실내의 벽돌 벽에 할당합니다. 벽돌 파일 텍스처를 **Base Color** 및 **Diffuse Weight**에 할당해보십시오. 그러면 조명이 재료의 확산 구성요소와 반응하는 방법에 영향을 줍니다. **Roughness** 값을 1로 높이면 재료가 좀 더 부

드러워지고 분필 같아집니다. **Bump Mapping** 속성에 범프 맵 텍스처를 할당합니다. 이 경우 파일 텍스처가 삼각형 투영 방법을 사용하여 투영되었습니다. 이것은 텍스처를 X, Y, Z 방향으로 투영합니다.

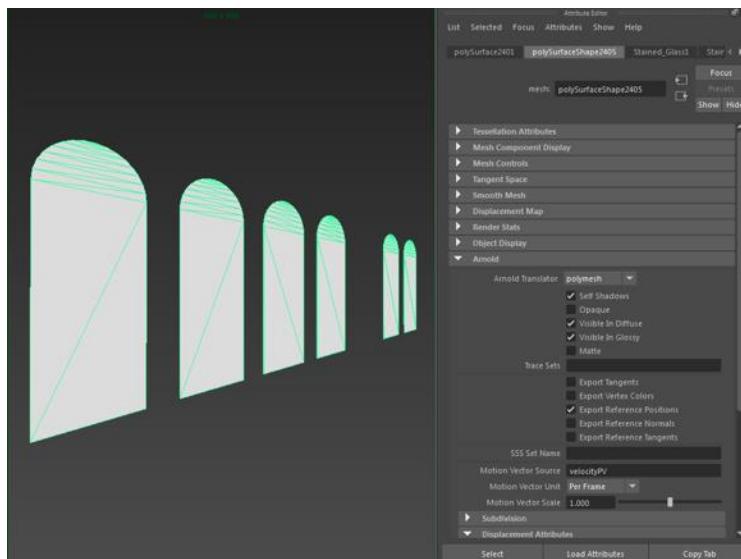


- 다른 Standard Surface 쉐이더를 생성하고 바닥에 할당합니다. 파일 맵을 Base Color, Specular 및 Bump map에 할당합니다. Specular Weight를 0.3 정도로 높이면 바닥이 반짝이는 효과가 나타납니다. Roughness를 높이면 바닥 내 반사가 흐려지며 보다 자연스럽게 연출됩니다. 0.2 정도의 낮은 값을 시도해 보십시오.



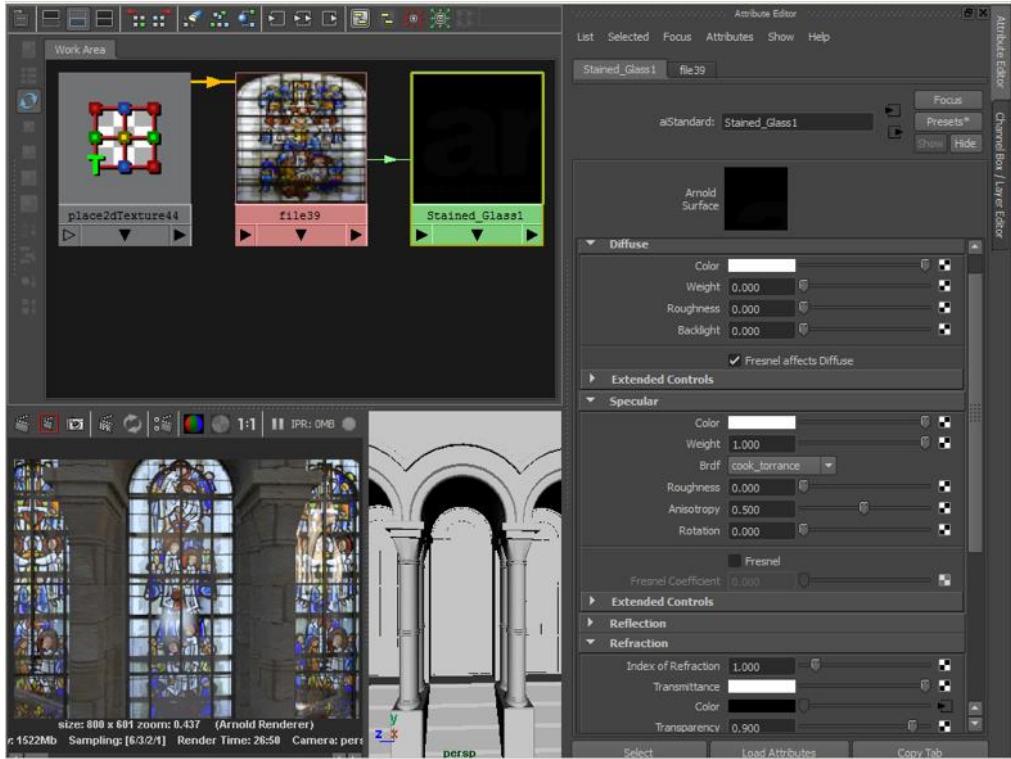
Stained Glass Window

- 이제 스테인드 글라스 창 효과를 만들어보겠습니다. Arnold와 같이 투명한 오브젝트를 렌더링할 때는 **Opaque**를 비활성화해야 합니다. 해당 창의 속성 편집기에 있는 Arnold 부분에서 **Opaque** 스위치를 비활성화합니다.

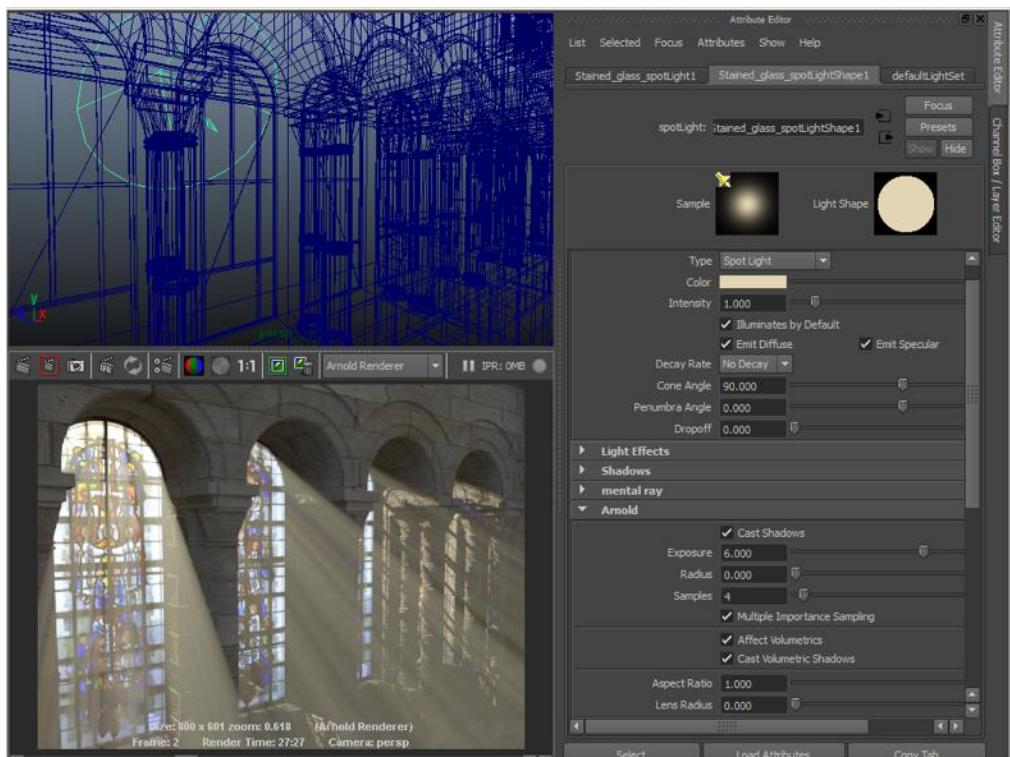


유리 형상에 대해 비활성화된 Opaque

- 창 유리를 선택하고 Standard Surface를 할당합니다. **Base Weight**를 0으로 낮추고 **Specular Weight**를 1로 높입니다. 스테인드 글래스 색상 맵을 **Transmission Color**로 할당하고 **Transmission Weight**가 0.9 등 높은 값으로 설정합니다.

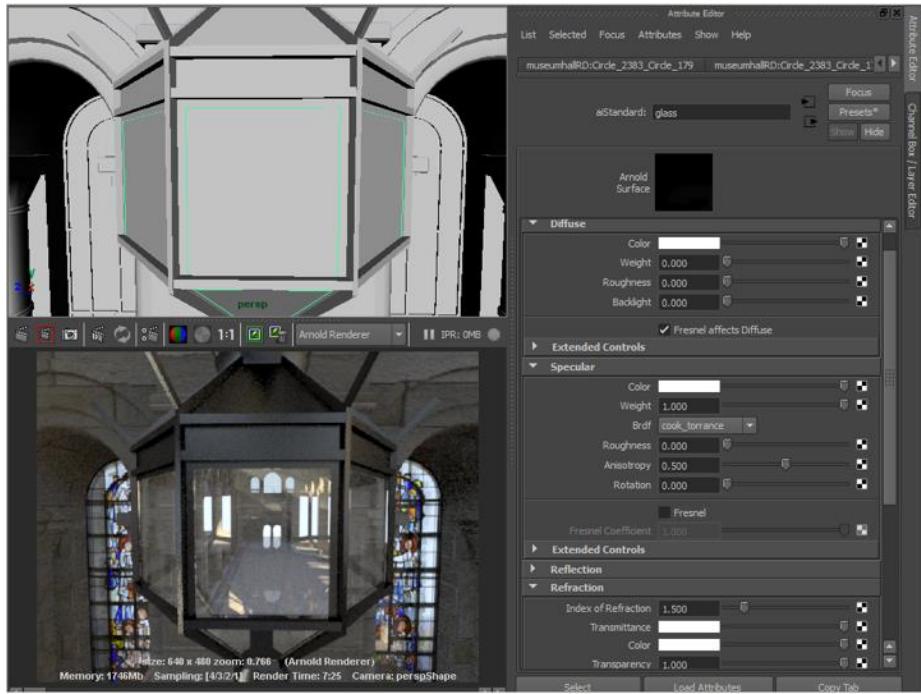


- 다른 **Spot Light**를 생성하고 스테인드 글래스 창 바깥에 배치합니다. 조명의 Intensity 및 Exposure가 충분히 높으면 체적 안개 효과가 나타납니다. 안개에 너무 노이즈가 많다면 **Spot Light**에서 **Sample**의 개수를 높여보십시오. 이 경우에는 4개의 샘플이 최적입니다. 안개의 색상이 창을 통해 반사된 색상에 영향받는 것을 알 수 있습니다.

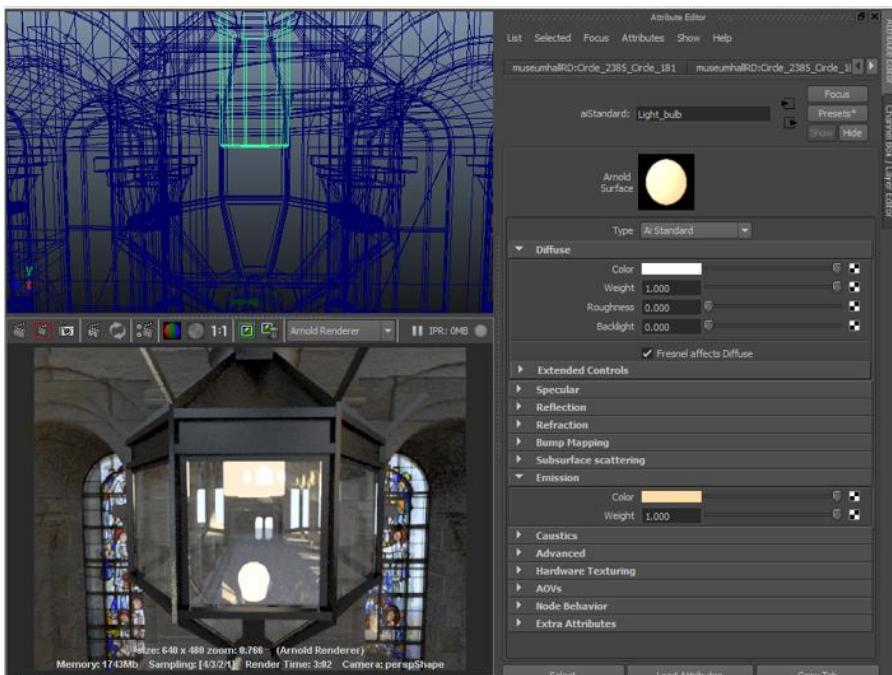


Interior Lighting

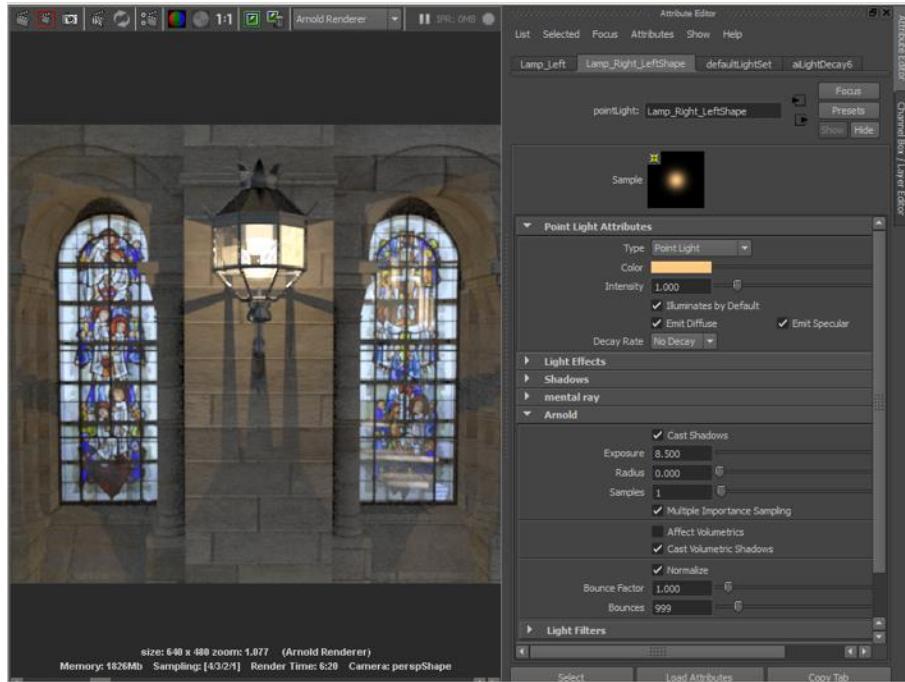
- 이제 내부 조명을 작업해보겠습니다. 스테인드 글래스 창 옆에는 2개의 램프가 있습니다. 유리를 선택하고 Standard Surface를 추가합니다. 유리 메시의 Arnold 부분에서 **Opaque**가 비활성화되어 있어야 합니다. 이번에는 유리를 굴절시킬 것입니다. Standard 재료의 Transmission 아래에서 **Index of Refraction**을 유리의 수치인 **1.5**로 변경합니다. **Base Weight**를 줄이고 **Transparency**를 **1.0**으로 설정해보십시오.



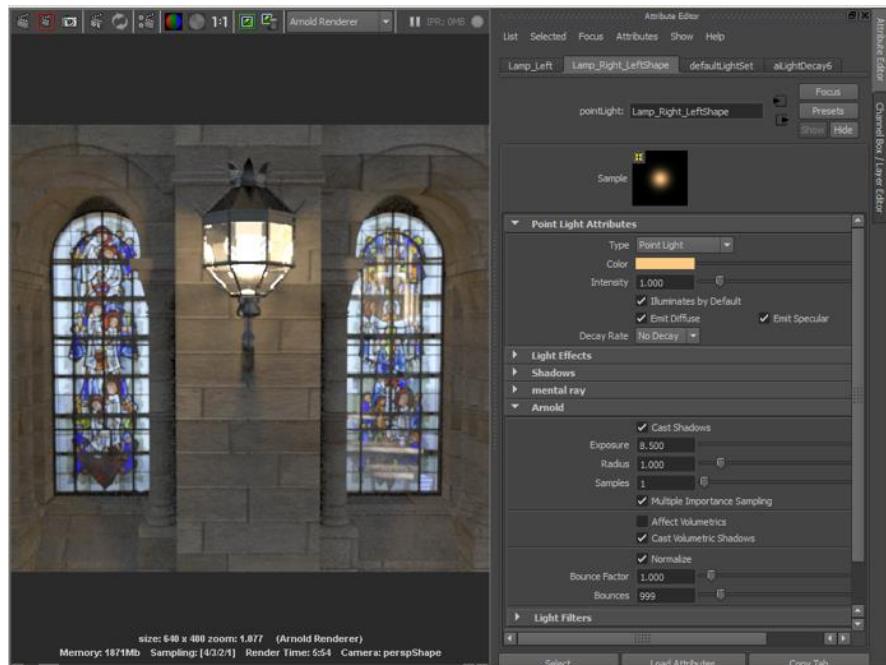
- Standard Surface 쉐이더를 전구에 할당합니다. **Base Weight**를 **0**으로 줄이고 **Emission Weight**를 **1.0**으로 높입니다. **Emission Color**를 밝고 따뜻한 노란색으로 변경합니다. Emission을 높이면 형상에서 빛이 발산되는 느낌이 납니다. 보다 쉬운 제어를 위해 그 자리에 조명을 생성하고자 하므로 이것은 적용하지 않겠습니다. 전구 형상의 Arnold 부분에서 **Diffuse Reflection** 및 **Specular Reflection**을 비활성화합니다. 점광원을 사용하여 장면에 조명을 추가할 것이기 때문입니다.



- Point Light을 생성하고 램프의 유리 안에 배치합니다. 조명을 전구 안에 배치할 수 있지만 전구 형상에 대해 Cast Shadow는 꺼야 합니다. 그렇지 않으면 빛이 벗어나지 못합니다. Point Light의 Color를 밝고 따뜻한 노란 색조로 변경하십시오. 이에 대해 체적 안개를 표시하지 않고자 하므로 해당 점 광원에 대한 Volume 가시성을 0으로 낮추십시오. Exposure를 높이면 장면 내에서 조명 효과를 볼 수 있습니다. 그림자가 너무 선명하고 거칠어 보입니다. 그림자를 부드럽게 만들어야 합니다.

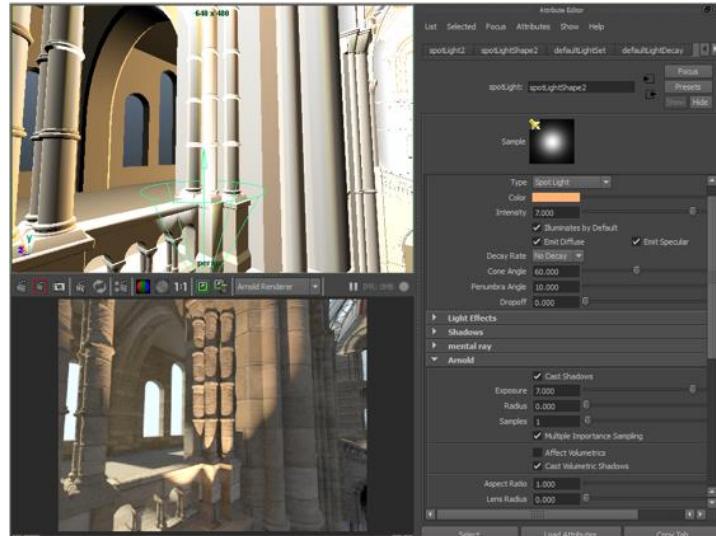


- 그림자를 부드럽게 만들기 위해서는 조명의 크기를 증가시켜야 합니다. 현실에서처럼 빛이 더 커질수록 그림자는 부드러워집니다. 점 광원의 Arnold 부분에서 Radius를 1.0 정도로 높입니다. 그림자가 더 부드럽고 훨씬 사실적으로 나타납니다.



- 그 다음에는 Spot Light를 만들어 1층의 기둥에서 위를 가리키도록 회전시키겠습니다. 이것이 상향등이 될 것입니다. 색상을 밝고 따뜻한 오렌지색으로 변경합니다. Cone Angle을 60 정도로 열고 Penumbra 각

도를 높여서 빛의 가장자리를 부드럽게 만들어야 합니다. Intensity 및 Exposure를 7.0 정도로 높입니다.



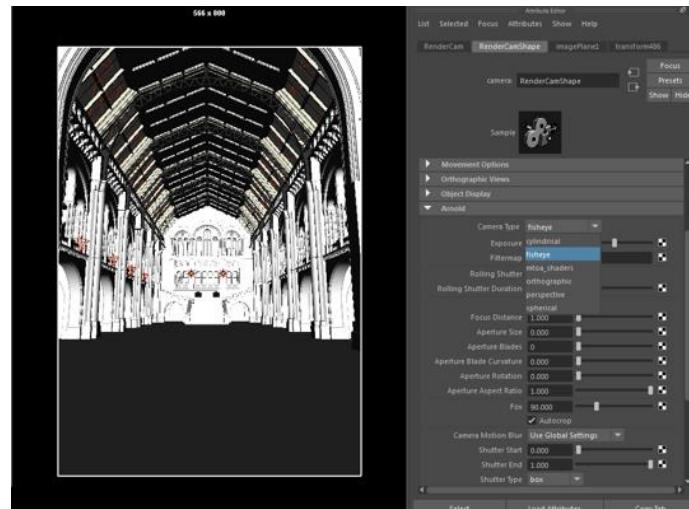
Emission

이 영역에 빛을 적용하는 또 다른 방법은 메시를 사용하여 장면을 '밝게' 만드는 것입니다. 이것은 **Standard Surface** 쉐이더에 있는 **Emission**이나 **Mesh Light**을 사용하여 실행할 수 있습니다.

- **Ai AreaLight**를 원통형 메시로 교체하고 비슷한 모양으로 크기를 조절합니다. **Standard Surface** 쉐이더를 할당하고 **Emission**을 1로 높입니다. **Emission Color**를 클릭하고 밝고 따뜻한 노란 색조를 선택합니다. Color 선택기에서 Hue 밑에 Value가 있습니다. 해당 장면에서 방출되는 빛의 밝기가 부족한 경우에는 1.0 이상으로 높여보십시오. 이 경우에는 4.0이 사용되었습니다. 하지만 이 기술은 노이즈를 유발할 수 있습니다. Arnold Render 설정에서 **Diffuse Samples**를 높이면 보다 선명한 결과를 얻을 수 있습니다. 이 값을 너무 높이지 않도록 주의하십시오. 렌더링 시간이 크게 늘어나기 때문입니다. 상기 렌더링의 경우에는 4.0이 사용되었습니다. 또한 앞서 램프에서처럼 Emission을 사용하여 조명과 오브젝트의 조합을 생성할 수 있습니다. 하지만 이 경우에는 실린더 형상에서 **cast shadow**를 꺼야 합니다.

Camera Type

- 렌더링 카메라에 대한 위치를 선택합니다. 카메라를 선택하고 Attribute Editor에서 아래로 스크롤하여 Arnold를 표시합니다. **Camera Type** 옆에 **perspective**라는 단어가 있습니다. 이것을 **fisheye**로 바꿔보십시오. 이렇게 하면 어안 렌즈가 카메라에 부착된 것처럼 보입니다.



아래에서 perspective 렌즈와 fisheye 렌즈 사이의 렌더링 차이점을 확인할 수 있습니다.



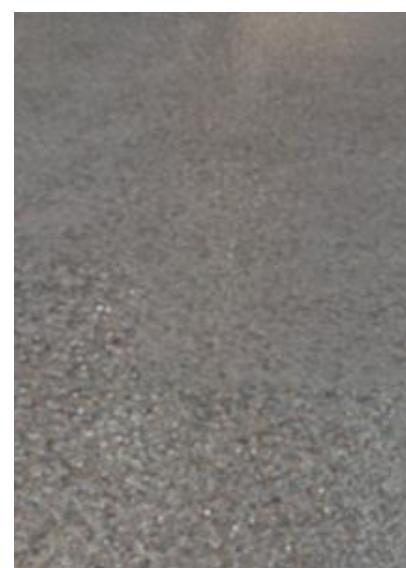
Clamp Sample Values

이제 렌더링의 품질을 높일 수 있는 방법을 살펴보겠습니다. 바닥 재료의 반사에 약간의 노이즈가 발생하는 것을 알 수 있습니다. 이것은 배경에 있는 램프 두 개에서 나오는 밝은 빛 때문입니다. 렌더링 설정에서 **Specular Samples**를 높일 수 있지만 전체 장면이 영향을 받고 렌더링 시간이 늘어날 수 있습니다. 그 대신 장면에서 픽셀의 밝기를 제한할 수 있습니다. 이렇게 하면 바닥의 밝은 반사가 감소되어 바닥에 반사되는 노이즈를 줄일 수 있습니다.

- Arnold Render 설정에서 **Clamp Sample Values**를 켭니다. 1.0 값을 시도해 보십시오. 그래도 소용없다면 이 값을 살짝 낮춰보십시오.



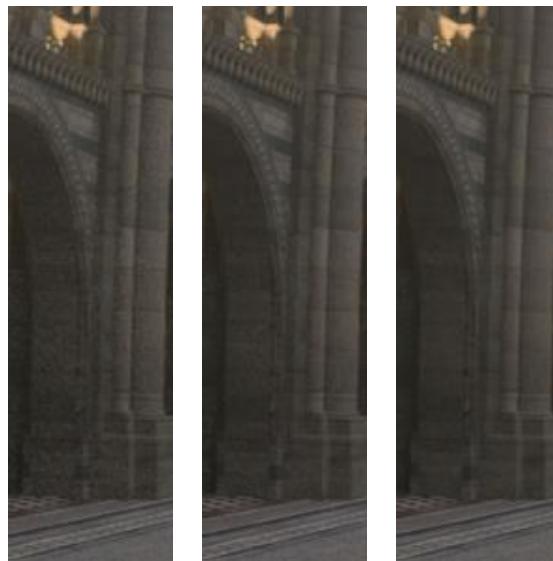
Clamp Sample Values Off



Clamp Sample Values: On

Diffuse Samples

다음은 다양한 **Diffuse Samples**를 사용하는 예제입니다. 기본적으로 Arnold는 두 개의 'Diffuse' 샘플을 사용합니다. 하지만 렌더링 예제에서는 전역 조명 'Indirect Diffuse' 샘플이 충분하지 않은 이미지의 더 어두운 영역에서 노이즈를 볼 수 있습니다. 장면에서 'Diffuse' 샘플의 개수를 늘리면 이 특별한 유형의 노이즈를 줄여 렌더링 품질이 향상됩니다. 하지만 렌더링 시간도 늘어나므로 이 값을 적절하게 사용해야 합니다. 최종적으로 렌더링된 이미지는 6개의 'Camera (AA)' 샘플과 3개의 'Diffuse' 샘플로 렌더링되었습니다.



Diffuse 2

Diffuse 3

Diffuse 4

스튜디오 자동차 렌더링



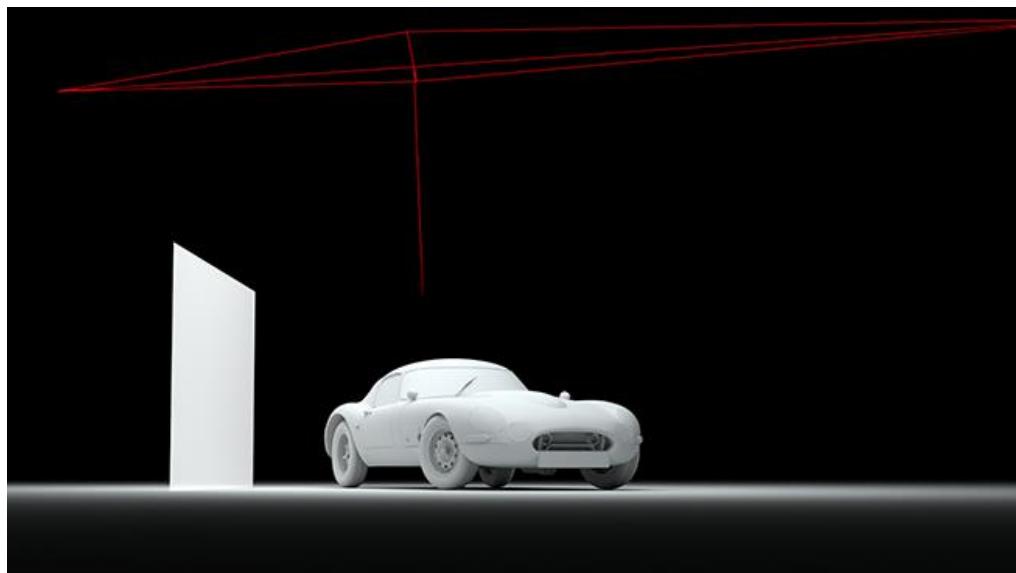
이 '만들기' 자습서에서는 자동차 스튜디오 스타일 장면을 복제하는 데 필요한 일부 조명, 쉐이딩 및 렌더링 팁을 다룹니다. 성공적인 자동차 렌더링을 위한 열쇠는 사실적인 '실제' 조명을 사용하는 것입니다. 스튜디오에서 자동차의 올바른 조명을 위해서는 많은 작업과 '미세 조정'이 필요합니다. 적절한 조명은 자동차의 미학을 강조하는 데 도움을 주는 흥미로운 거울 반사를 만드는 데 중요합니다. 여기에 제공된 팁들은 이러한 과정을 쉽게 만들어 줄 것입니다.

실제로 자동차의 조명을 설정하는 예는 하단 링크 사이트에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Studio+Automotive+Rendering>

Quad Area Lighting

장면에 대형 스튜디오 소프트 박스 조명을 나타내는 Quad Area 조명이 적용됩니다. 반사장치가 추가되어 차량 물체의 내부에서 반사를 표시합니다.



Color Texture

자동차 페인트와 유리창에서 사실적인 실제 색상과 반사를 표현하려면 빛의 색상에 HDR 맵을 추가하는 것이 가장 좋습니다. 여기서 이 장면에 사용되는 샘플 HDR 맵을 다운로드할 수 있습니다(Flat_2.hdr).

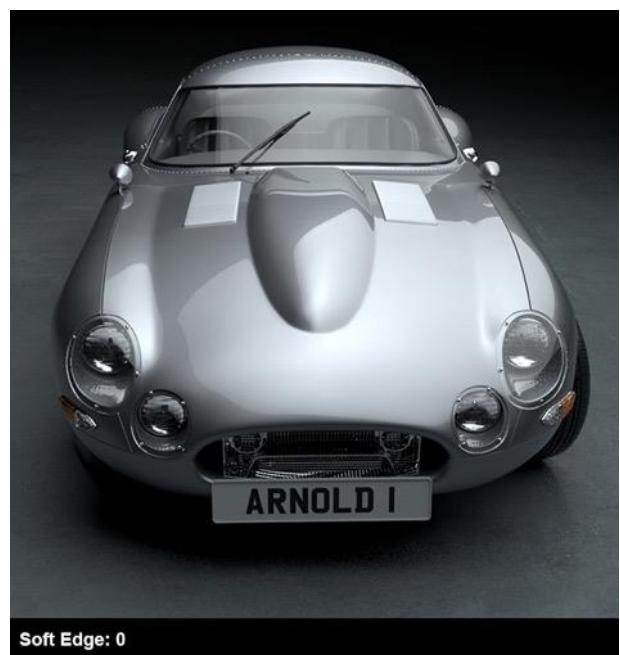
(<https://www.pingo.nu/vfxtools>)



빛의 반사(및 색상)는 HDR 맵이 없을 때는 현실감이 부족합니다.

Soft Edge

Quad 조명 내의 Soft Edge로 조명의 가장자리에 부드러운 감쇠를 지정하여 자동차 페인트와 유리의 빛 반사를 보다 부드럽게 만듭니다.



여기서 렌더링 카메라를 사용하여 *Soft Edge*를 0.5로 증가시킬 때의 효과를 볼 수 있습니다.



Specular 효과

자동차의 후드에서 일부 하이라이트 부분이 날아갔습니다. 빛의 노출을 줄이면 이런 현상에 도움이 되지만, 장면의 전반적인 조명까지 줄어들 수 있습니다. 이는 우리가 원하는 것이 아닙니다. 보다 간단하고 효과적인 방법은 Quad 조명의 Specular 효과를 줄이는 것입니다.



Quad 조명 'Specular' 1. Quad 조명 'Specular' 0.6을 볼 수 있음

Quad 조명에 **Light Blocker**를 추가하면 조명이 자동차에 반사되는 위치와 방법을 보다 예술적으로 제어할 수 있습니다. 자동차 오른쪽의 조명 및 반사 효과를 확인합니다.



Without Light Blocker



With Light Blocker

Emissive Plane

반사성의 흰색 평면이 배치되어 자동차 측면에 반사 및 조명을 추가했습니다. 이는 일반적으로 자동차 사진 작가가 보통 자동차에 직접 조명을 비추지 않고 자동차 페인트에 반사되는 표면에 빛을 비추기 때문입니다. 이 경우 Emission Weight은 0.5입니다. 물론 *Emission Color*에 텍스처를 매핑할 수도 있습니다. 하지만 이 경우에는 흰색 *Emission Color*의 효과도 좋습니다.



Without Emissive Plane



With Emissive Plane

몸체 및 바퀴에 조명 효과 확인

*Diagnostics-> Feature Overrides*에 있는 'Ignore Lights'를 활성화하면 방출면이 자동차의 반사에 미치는 영향을 정확하게 확인할 수 있습니다.



Feature Overrides 아래에서 'Ignore Lights'가 활성화되어 있으면
방출면의 효과를 보다 확실하게 볼 수 있습니다.

쉐이딩

Facing Ratio 쉐이더

Facing Ratio는 자동차 페인트에 진주 광택의 색상 코팅을 만드는 데 유용할 수 있습니다. 아래 예에서는 Coat Color가 Standard Surface 쉐이더에 추가되었습니다. 이 경우에는 Facing Ratio 쉐이더가 Coat Weight에 연결되었으며 Coat Color 효과를 과장하는 데 사용되었습니다.



Base Color



Metalness



Coat Color



Coat Weight에 Facing
Ratio



With Facing Ratio



Without Facing Ratio

이 효과는 전조등과 같이 카메라를 향하고 있는 영역 주변에서 보다 잘 보입니다.

Thin Film

또한 Standard Surface 쉐이더 내의 *Thin Film*을 사용하여 멀티톤의 카 페인트 재료를 만들 수 있습니다(Thickness: 380, IOR: 1.43).



With Thin Film



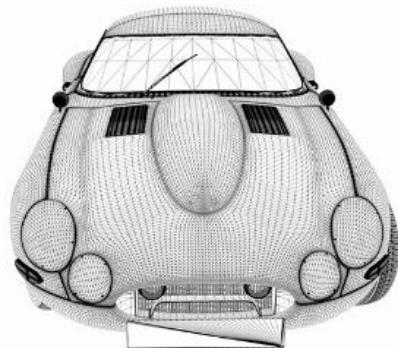
Without Thin Film

Utility 쉐이더

자동차 모델을 렌더링하는 데 유용한 쉐이더가 있습니다. **Utility** 및 **Wireframe** 쉐이더를 사용하여 모델을 평가할 때 등고선 및 점 연속성 문제를 진단할 수 있습니다. Utility 쉐이더의 Reflection Lines 모드는 실제 반사가 어떻게 작용하는지에 대한 개념을 제공할 수도 있습니다. 이런 종류의 쉐이딩은 표면 평가가 중요한 자동차를 모델링할 때 특히 유용합니다.



Reflection Lines (Utility 쉐이더)



Wireframe 쉐이더

Utility 쉐이더에는 또한 'Metal' Shade Mode가 있습니다. 이것은 자동차에서 반사 하이라이트를 테스트 및 검사하기 위한 간단한 쉐이딩 방법입니다.



조명의 정반사는 Utility 쉐이더로 보다 잘 볼 수 있습니다('Metal'로 설정된 Shade Mode).

스튜디오 조명



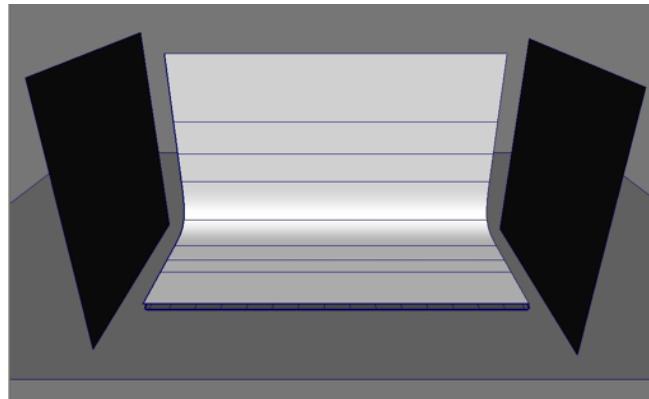
이 자습서에서는 모든 종류의 오브젝트를 조명 처리 및 렌더링하는 데 사용할 수 있는 간단한 사진 조명 스튜디오 설정의 재생성 방법을 설명합니다. Arnold 영역 quad 조명과 HDR 맵의 조합을 사용하여 장면에서 사실적인 정반사를 생성합니다. HDR 맵은 전면에 있는 모든 모델의 표면에 풍부한 반사 효과를 제공하고 최종 렌더링의 사실감을 더해줍니다.

장면 파일은 하단 링크 사이트에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Studio+Lighting>

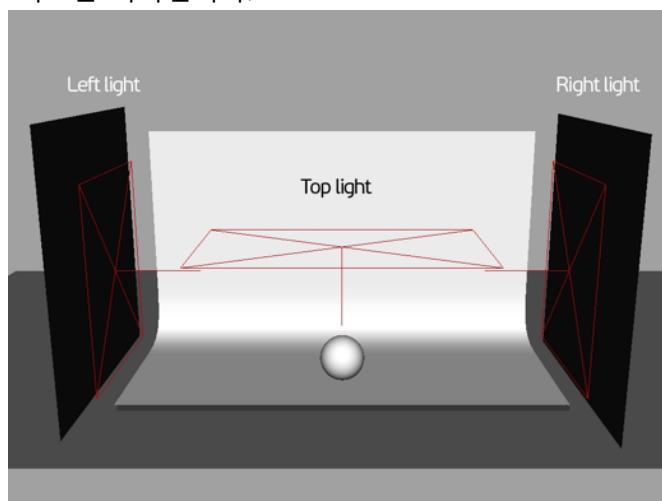
Set Model

장면은 양쪽에 바닥 평면과 하나의 평면이 있는 곡선형 배경으로 구성되어 있으며, 어두운 회색 **Standard Surface** 쉐이더가 양쪽에 할당되어 있습니다.



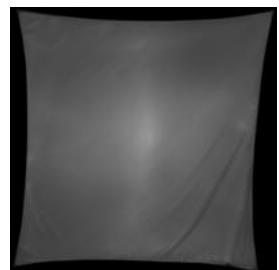
Lights

- 위쪽, 왼쪽 및 오른쪽 3가지 조명 각각에 대해 Arnold 영역 조명을 만듭니다. 이것은 사진 스튜디오에서 사용되는 대형 소프트 박스를 나타냅니다.



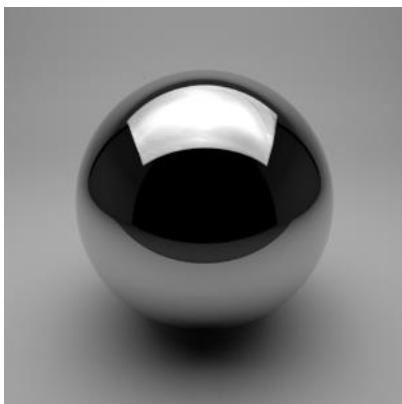
각 조명에 대해 조명 유형을 'Quad'로 변경합니다.

이 자습서에서는 HDR 맵이 사용되었습니다. 파일 텍스처를 사용하지 않으려는 경우 간단한 상자 램프 텍스처 맵을 사용할 수 있지만 모델의 반사는 사실적으로 나타나지 않습니다.

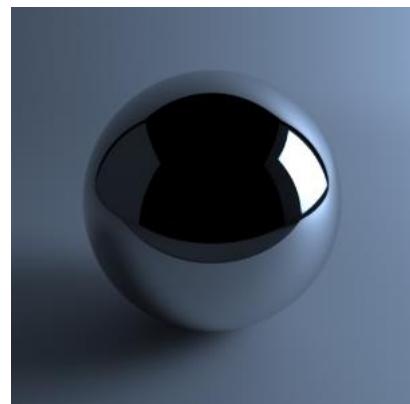


왼쪽에서 오른쪽으로: softbox_middle_wrm.HDR, softbox_middle_cold.HDR, softbox_square_gs.HDR.

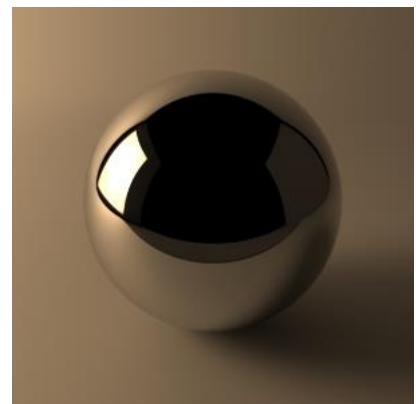
- 각 HDR 맵을 아래 이미지에 표시된 대로 각 영역 조명의 **Color** 속성에 연결합니다.



Arnold 영역 조명 '상단'의 Color
속성에 연결된
softbox_square_gs.hdr



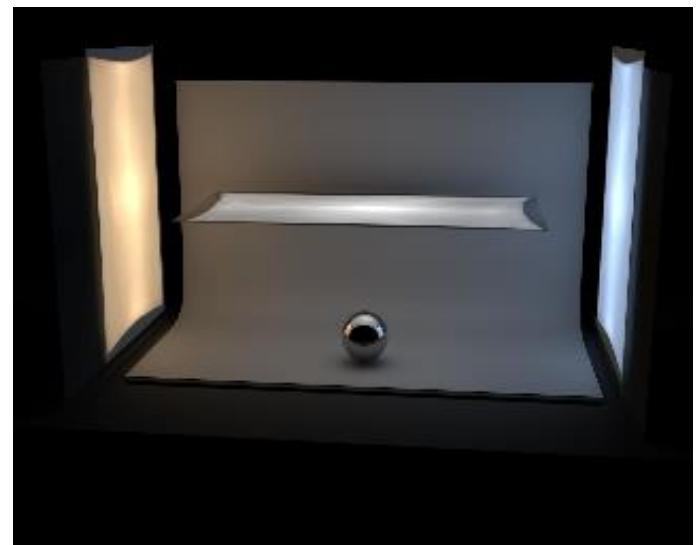
Arnold 영역 조명 '오른쪽'의 Color
속성에 연결된
softbox_middle_cold.hdr



Arnold 영역 조명 '왼쪽'의 Color
속성에 연결된
softbox_middle_wrm.hdr

Exposure

장면을 테스트 렌더링합니다. 아래와 같은 이미지가 나타납니다. 반사가 너무 밝은 경우에는 **렌더링 카메라**의 Arnold 속성에서 **exposure** 설정을 낮출 수 있습니다.



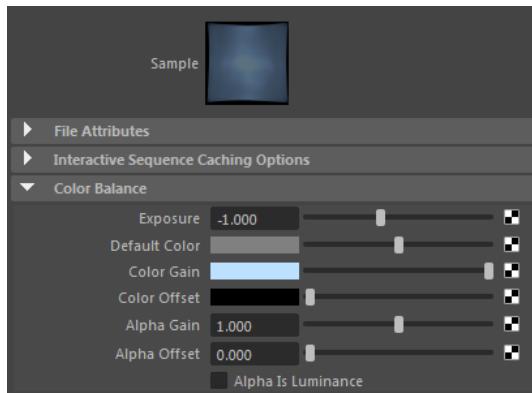
HDR 맵의 높은 동적 범위는 Arnold 카메라 노출 속성을 낮추면 표시됩니다.

Camera exposure을 낮추면 조명 HDR 맵 세부 정보가 표시됩니다. 장면의 **specular reflections** 내에서 보이는 높은 동적 범위 전체를 볼 수 있습니다.



또는 조명 강도를 변경하거나 원하는 효과를 얻기 위해 HDR 맵의 HSV 속성을 변경하면서 실험할 수 있습니다.

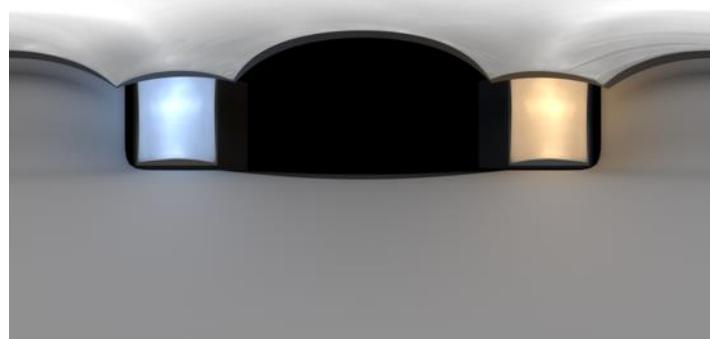
스튜디오에서 물건의 앞면에 약간 어둡게 보이는 조명이 있으면 카메라 앞에 반사면을 추가해 보십시오(평면에 대해 'Primary Visibility' 끄기).



조명의 파일 텍스처 'Color Gain' 및 'Exposure' 속성을 수정할 수 있습니다.

Spherical Camera Lens

이 자습서를 완료하면 Spherical 카메라 렌즈를 사용하여 장면을 HDR 맵으로 변환해보는 것이 어떨까요?



Spherical 카메라 렌즈로 렌더링된 장면

구체 카메라에 대한 추가 정보는 여기에 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Spherical+Camera>

Sky Dome Light 사용하기



이 자습서는 HDR 맵과 Arnold 렌더러를 사용하여 장면을 조명 처리 및 렌더링하는 과정을 설명합니다. 여기서는 SkyDomeLight을 사용하여 장난감 자동차 모델에 사실적으로 조명을 적용해 보겠습니다.

SkyDomeLight은 특히 사실적인 HDRI 기반 실외 조명이 필요한 장면을 렌더링할 때 유용합니다. 또한 **Standard Surface** 쉐이더를 기반으로 사용하여 모델에 대해 사실적인 재료를 만들어 보겠습니다. 마지막으로 Arnold의 카메라 설정을 사용하여 피사계 심도를 추가하고 최종 이미지를 렌더링하기 전에 어떤 렌더링 설정을 사용해야 하는지 알아보겠습니다.

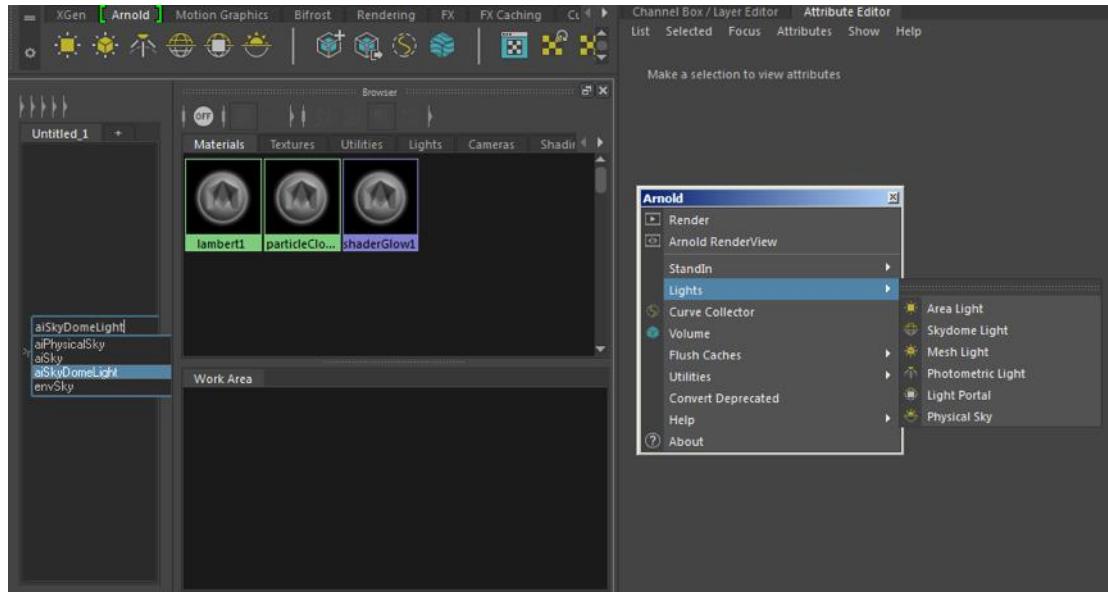
- [Skydome & The Table 설정하기](#)
- [Standard Surface를 사용하여 자동차 쉐이딩 하기](#)
- [최종 렌더링 설정하기](#)

이 투터리얼에 사용된 HDRI는 하단 링크의 사이트에서 다운로드할 수 있습니다.

<http://www.aversis.be/hdri/hdri-free-mirrored-ball-collection.htm>

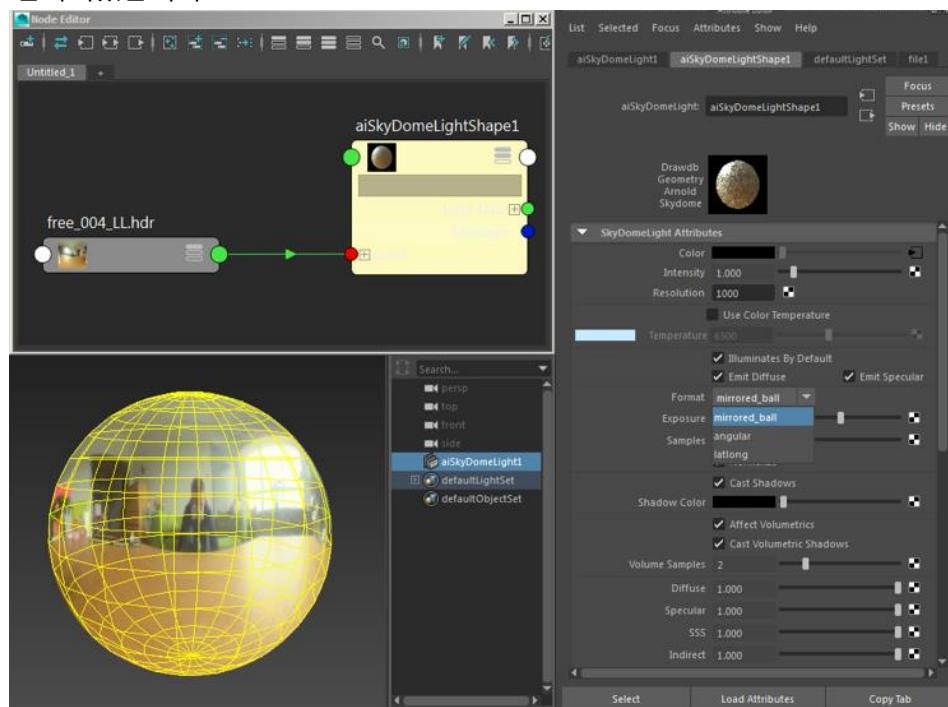
Skydome & The Table 설정하기

먼저 SkyDomeLight를 생성해보겠습니다. Hypershade 창을 엽니다. Hypershade 창의 왼쪽에 있는 Create Bar에 Arnold 노드 목록이 있습니다. 우리가 원하는 것은 **SkyDome Light**입니다. 클릭하면 생성됩니다. Hypershade 창의 Lights 탭과 Outliner에 새로운 노드가 생긴 것을 볼 수 있습니다. 이것은 뷰포트 안에서 노란색 와이어프레임 구체로 나타납니다.



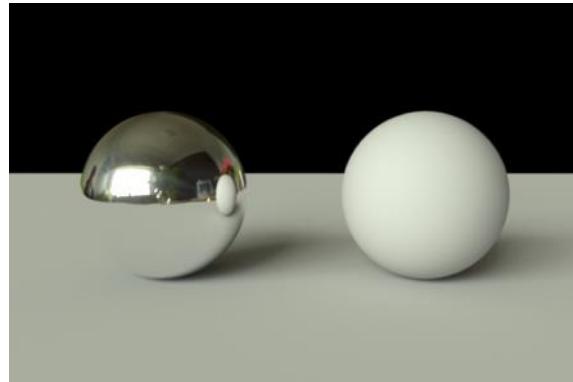
Ai SkyDomeLight 만들기

- Ai **SkyDomeLight**를 선택하고 해당 속성 편집기를 표시합니다. **2d file texture**를 Ai SkyDomeLight의 **Color** 속성에 연결합니다. hdr 맵을 사용한다면 파일 유형을 **All Types**로 전환해야 열려 있는 브라우저 창에 hdr 파일이 나타납니다. **Format**을 HDR 맵을 캡처한 것과 동일한 포맷으로 변경합니다. 이 경우에는 미러 볼로 캡처되었습니다.



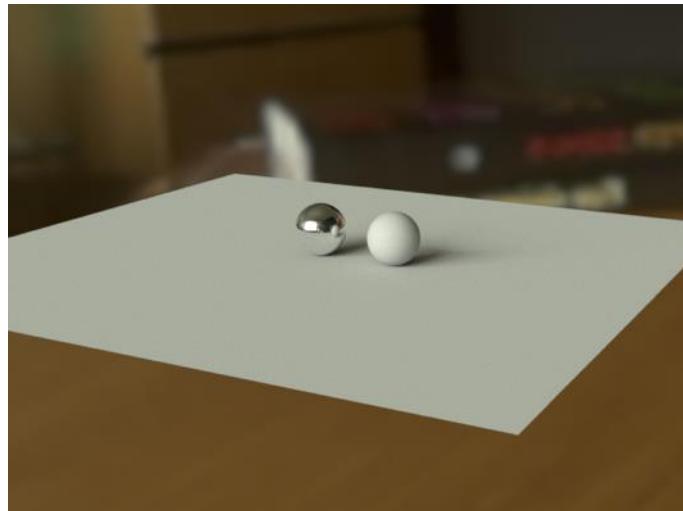
Ai SkyDomeLight에 연결된 HDR 맵. 포맷이 '미러 볼'로 설정됨

- Ai **SkyDomeLight**의 속성 편집기에서 **Resolution**으로 스크롤 다운합니다. HDR 맵의 해상도와 일치하는 값을 입력합니다. SkyDomeLight은 중요도 샘플링을 사용하여 중요한 방향으로 더 많은 광선을 보내며, HDR 맵에서 작은 디테일을 캡처할 수 있도록 중요도 테이블의 해상도를 충분히 높여야 하지만 HDR 맵 자체의 해상도보다 높아서는 안 됩니다.
- Ai SkyDomeLight의 효과를 테스트하기 위해 간단한 장면을 생성해야 합니다. 폴리곤 판을 만들고 격자 크기만큼 조절합니다. 두 개의 폴리곤 구체를 만들어 평면 위에 놓습니다.
- Hypershade 창에서 두 개의 **Standard Surface** 쉐이더를 만듭니다. 하나의 Standard Surface를 하나의 구체에, 다른 surface를 다른 구체에 할당합니다.
- 이제 두 번째 Standard Surface 쉐이더를 크롬 재질과 비슷하게 변경하려고 합니다. 이것을 선택하고 속성 편집기를 엽니다. 먼저 이름을 Chrome으로 변경합니다. 반사처럼 완벽하게 깨끗한 거울을 위해서 Base Weight는 1로, Metalness는 1로 높이고 Roughness는 0으로 줄입니다.



하나는 구체 크롬으로, 다른 하나는 무광 재료로 만듭니다.

- 이제 평면에서 구체를 테스트 렌더링합니다. Skydome 조명에 연결된 HDR 맵이 카메라에 표시됩니다. 나중에 조정할 것이므로 이 단계에서는 그림자에 노이즈가 있어도 걱정할 필요가 없습니다.

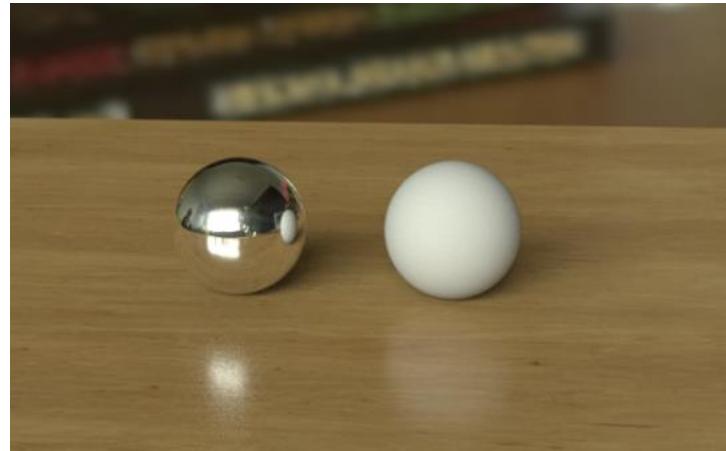


카메라의 배경에서 볼 수 있는 Skydome 조명

Table Wood

- 다른 Standard Surface 쉐이더를 생성하고 평면에 할당합니다. 이름을 '**Wood Table**'로 변경합니다. 나무 색 파일 텍스처를 속성 편집기의 Base Color 슬롯에 연결합니다.
- 반사 맵을 속성 편집기의 **Specular Color** 슬롯에 추가합니다. 이제는 나무 재질의 반사 설정을 미세 조정

하기 위해 IPR 렌더링을 시작하는 것이 좋습니다. Specular Weight 값을 0.3 정도로 변경합니다. 나무 재료의 구면 아래에 밝은 반사 하이라이트가 보일 것입니다. roughness 값을 조절해 보십시오. Roughness 값을 높이면 이 반사 하이라이트가 부드러워지고 낮추면 선명해 집니다.



다각형 면에서 Standard Surface에 연결된 나무 바닥 텍스처

- **Bump Mapping**을 열고 **2d bump file texture** 노드를 추가합니다. **bump** 값을 **0.01** 정도로 조절합니다. 다시 IPR을 사용하여 대화식으로 설정을 조정하십시오.



Wood 쉐이더에 범프 맵 텍스처 추가

Ai Standard 쉐이더를 사용하여 자동차 쉐이딩 하기

Car Paint

- Standard Surface 쉐이더를 생성하고 자동차의 바디 패널 지오메트리에 할당합니다. 이름을 **Car Paint**로 변경합니다. **Base Color**의 색상을 파란색으로 바꾸고 **Specular Weight**를 **0.3**으로 높입니다.



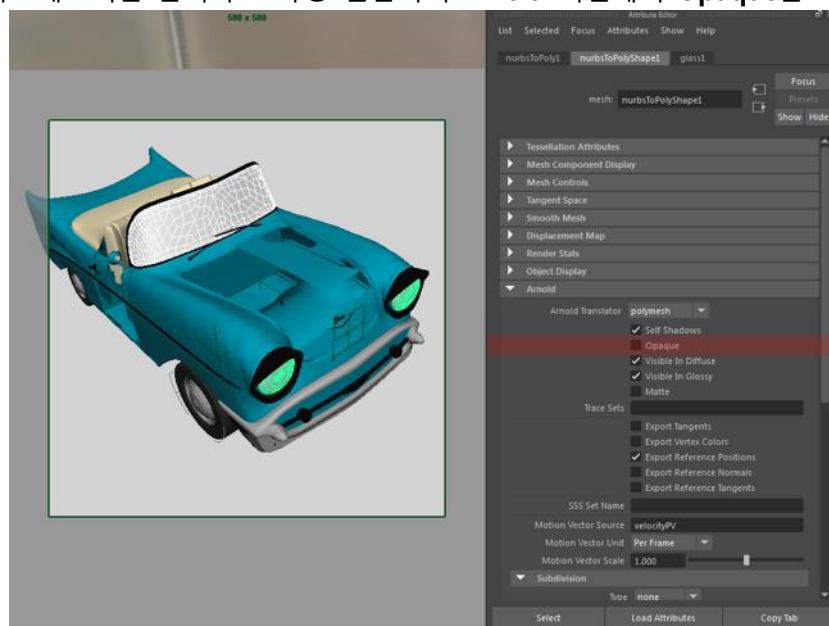
자동차 페인트



금속 자동차 페인트

Glass

- Arnold에서 투명성이 필요한 지오메트리에 대해 **Opaque**가 비활성화되어 있는지 확인해야 합니다. 유리를 만들려는 지오메트리를 선택하고 속성 편집기의 Arnold 섹션에서 **Opaque**를 비활성화하십시오.



모든 유리 표면에 대해 'Opaque'를 비활성화합니다.

장면 내 많은 오브젝트에 Arnold 속성을 적용할 때는 **Override Sets**를 사용하십시오.

- 다른 Standard Surface 쉐이더를 생성하고 이름을 **Glass Windscreen**으로 바꿉니다. 이것을 앞 유리 지오메트리에 적용합니다. **Base Weight** 및 **Specular Weights**를 **0**으로 낮춥니다. Index of Refraction을 1.5(실제 유리 값)로 변경하고 **Transmission Weight**를 **1.0**으로 높입니다. **Transmittance Color**를 연청색으로 변경하면 앞 유리가 연청색이 됩니다.



앞 유리 쉐이더 생성

- 이전에 구형에 대해 사용했던 크롬 재질을 헤드라이트 및 트림 표면에 할당합니다.



Rims

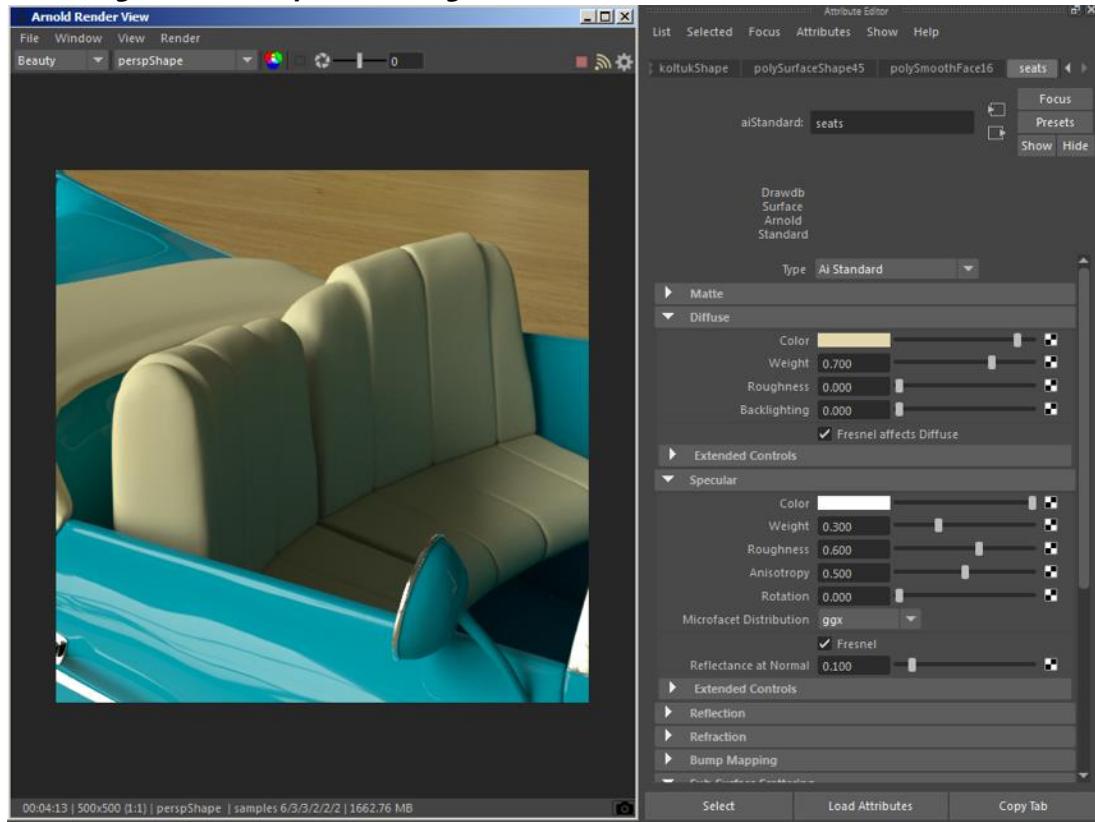
이제 부드러운 새틴 금속 소재를 만들어 보겠습니다. 새로운 Standard Surface 쉐이더를 만들고 바퀴 테두리와 범퍼에 할당합니다. 이름을 **Satin Metal**로 변경합니다. **Metalness**를 0.8 정도로 높이고 **Specular Weight**를 0.7로 낮춥니다. **Specular Roughness**를 0.3 정도로 높여서 반사를 흐리게 합니다.



바퀴 테두리에 대해 새틴 금속 쉐이더를 생성합니다.

플라스틱

- 또 다른 Standard Surface 쉐이더를 좌석에 할당합니다. 이 경우에는 좌석에 부드러운 플라스틱 느낌을 만들어 보겠습니다. 이름을 **Soft Plastic**으로 변경합니다. **Base Color**를 연한 베이지 색으로 변경합니다. **Specular Weight**를 0.2로, **Specular Roughness**를 0.6 정도로 설정합니다.



좌석을 위한 플라스틱 쉐이더

- 재료에 부드러운 느낌을 주려면 하위 표면 산란을 추가하십시오. 그러면 좌석의 재료가 보다 많은 빛을 흡수하는 느낌을 줄 수 있습니다. **Subsurface scattering**으로 스크롤 다운하고 **Weight** value를 0.1로 약간 높이십시오. **SSS Radius** 값을 높여 플라스틱에 부드러운 느낌을 주십시오.



플라스틱 좌석에 하위 표면 산란을 추가합니다.

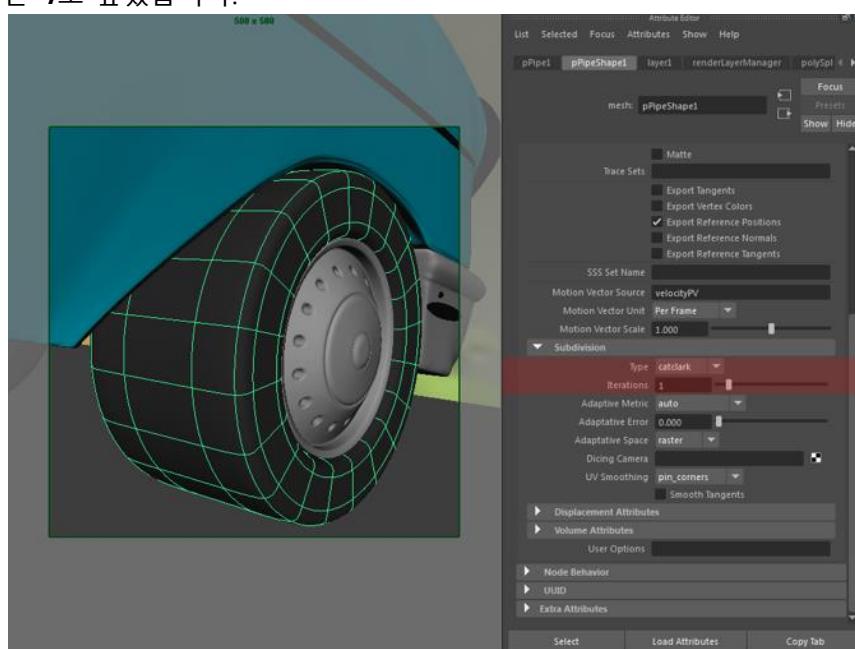
- 다른 Standard Shader를 만들고 앞 유리 와이퍼 및 운전대에 지정합니다. 이름을 **Glossy Plastic**으로 바꿉니다. **Base Color**를 짙은 회색으로 변경합니다. **Specular Color**를 중간 회색으로 변경하고 **Specular Weight**를 **0.3** 정도로 변경합니다.



앞 유리 와이퍼에 지정된 광택이 있는 플라스틱 재료

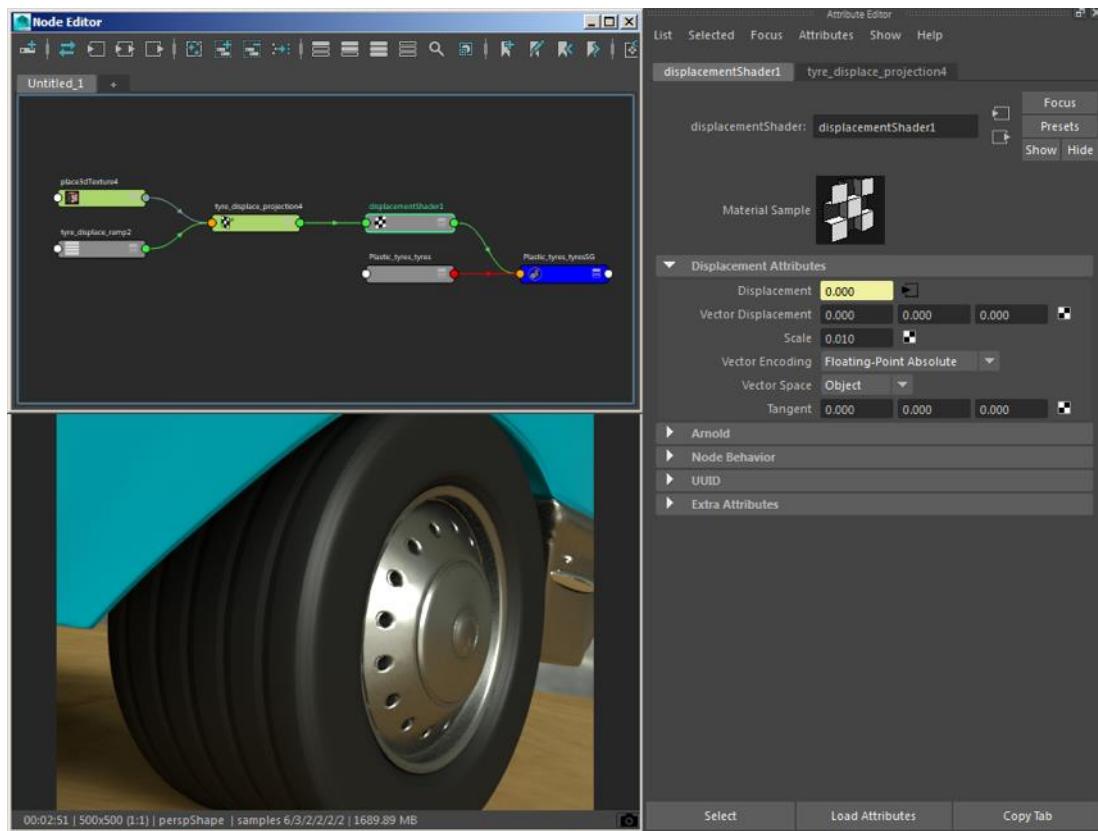
타이어: Displacement

- 타이어의 경우에는 Maya 변위 쉐이더를 Ai Standard 쉐이더와 함께 사용하게 됩니다. **Glossy Plastic** 재료를 복제하고 이름을 **Plastic Tire**로 변경합니다. 이것을 타이어 지오메트리에 지정합니다. **Plastic Tire** 노드에 대해 **Shading Group**을 선택하고 아래 그림에서처럼 **Displacement** 노드를 마우스 중간 버튼을 누른 채로 속성 편집기 내 Shading Group 노드의 **Displacement mat.** 슬롯으로 끌어 옵니다.
- 이제 Arnold에게 타이어 지오메트리를 변위시킬 때 어떤 **Subdivision Type**을 사용해야 하는지 지시해야 합니다. 해당 타이어 지오메트리를 선택하고 속성 편집기를 엽니다. Arnold 아래에 Subdivision Type이 있습니다. **None**에서 **Catclark**로 변경합니다. Subdivision Iterations를 높이면 변위 효과의 미세함이 증가합니다. 이 경우에는 **1**로 높였습니다.



바퀴 지오메트리의 Arnold 속성에서 subdivision 설정 변경

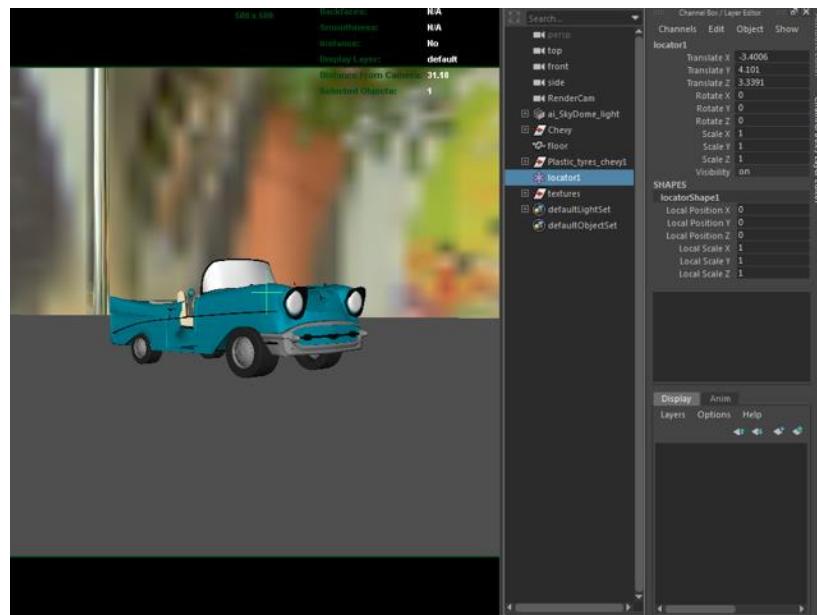
- 변위 노드를 선택하고 변위 노드의 **Map** 슬롯에 맵 텍스처를 지정합니다. 이 경우에는 **cylindrical projection** 노드가 있는 **ramp**가 사용되었습니다. **Displacement Scale**을 **0.01** 정도로 높이고 테스트 렌더링을 실행합니다. IPR 렌더링은 변위에 적용된 변경 사항을 업데이트하지 않습니다. Displacement 값은 변경할 때마다 다시 렌더링해야 합니다.



타이어에 지정된 Displacement 쉐이더

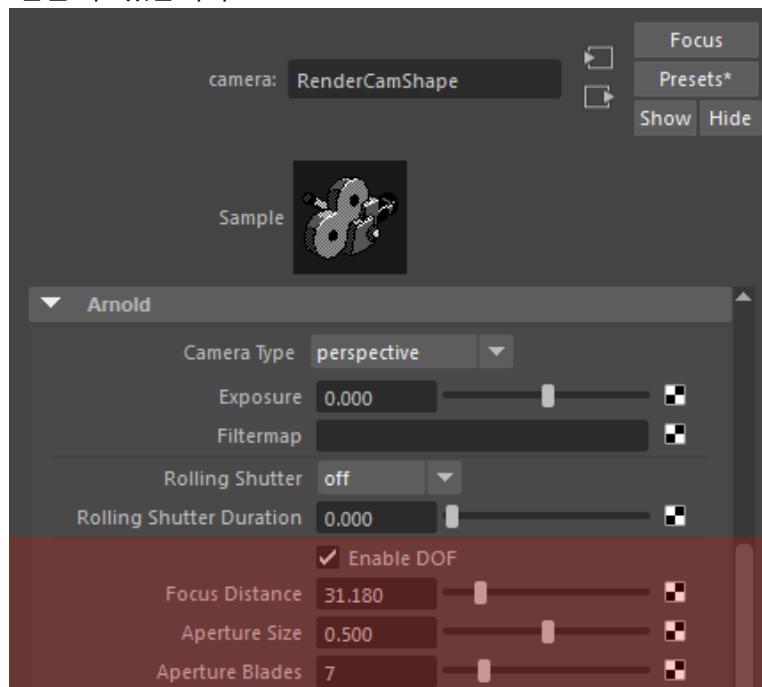
최종 렌더링 설정하기

- 이제 조명과 음영 처리가 끝났으므로 카메라를 렌더링이 준비된 위치로 옮길 수 있습니다. 먼저 카메라에 일부 피사계 심도를 추가해 보겠습니다. 로케이터를 만들고 **Display>Heads Up Display>Object Details**로 이동합니다. 이것은 로케이터가 카메라와 얼마나 멀리 떨어져 있는지를 알려줍니다. 로케이터를 관심의 초점으로 이동하고 카메라 뷰포트의 상단 오른쪽에 있는 **Distance From Camera** 값을 메모해 두십시오.



로케이터를 만들어서 초점 거리를 배치하려는 곳으로 이동합니다.

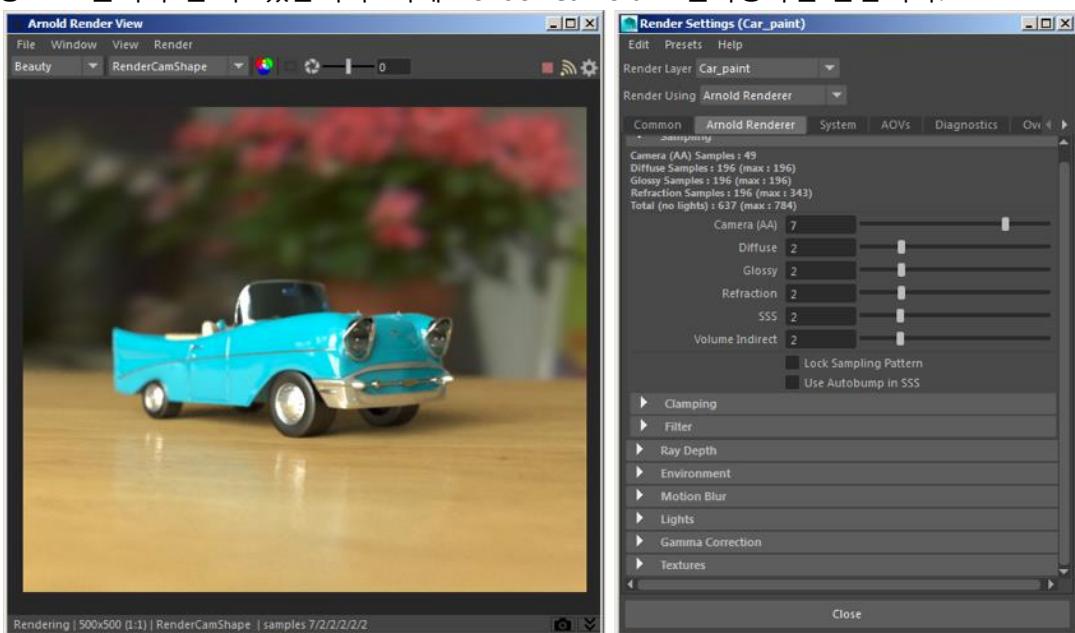
- 속성 편집기에서 해당 Render Camera를 Arnold 부분으로 스크롤 다운합니다. **Enable DOF**를 활성화하고 **Focus Distance**에 로케이터에 대한 카메라로부터의 거리 값을 입력합니다. **Aperture Size**를 높여서 카메라의 피사계 심도 효과를 볼 수 있도록 합니다. Arnold와 IPR에서는 대화식으로 피사계 심도를 조정하여 매우 빠른 결과를 얻을 수 있습니다.



Focus Distance를 로케이터에 대한 Distance from Camera와 동일하게 설정합니다.

최종 렌더링

- 이제 장면의 테스트 렌더링을 완료했습니다. 다음으로 품질 설정을 높여야 합니다. 먼저 **Ai SkyDomeLight**을 선택합니다. 속성 편집기에서 **Samples**를 3으로 올리면 장면에서 그림자 샘플 수가 증가하여 모든 그림자 노이즈가 효과적으로 제거됩니다. 이 숫자가 클수록 결과는 세밀합니다. 하지만 렌더링 시간이 크게 증가하기 때문에 이 값을 너무 높여서는 안 됩니다. 그림자가 아니라 노이즈가 증가한다면 Render 설정에서 **Diffuse Samples**의 수를 늘려야 할 수도 있습니다. 3까지 높여보십시오.
- 피사계 심도, 조명, 음영 처리가 마음에 든다면 최종 렌더링을 실행해야 합니다. Render Settings 창을 열고 Arnold 탭을 클릭합니다. 기본적으로 **AA Samples**는 3으로 설정되어 있습니다. 테스트 렌더링으로는 충분한 값이지만 최종 렌더링에서는 이 값을 높여야 합니다. 현재 장면의 피사계 심도 값에 따라 이 값을 7 이상으로 높여야 할 수 있습니다. 이제 Render Camera로 렌더링하면 끝납니다.



Render Settings에서 최종 AA Sample이 7로 설정됨

Shading



MtoA에서 Shading에 관련된 자습서들은 다음 페이지들에 있습니다.

- [Anisotropic Brushed Metal](#)
- [Normal과 Bump 맵 결합하기](#)
- [Displacement 튜토리얼](#)
- [Flakes 예](#)
- [무지개 빛의 비누 방울 쉐이더](#)
- [Color Shaders를 사용한 모션 그래픽 효과](#)
- [Emissive 쉐이더를 사용한 굴절 커스틱 효과](#)
- [Remap an Image Using UV Coords](#)
- [Rim 쉐이더](#)
- [지구 쉐이딩 하기](#)
- [장미 쉐이딩 하기](#)
- [Specular BRDF\(양방향 반사율 분포 함수\)](#)
- [Facing Ratio 쉐이더를 이용한 Toon Shading](#)
- [User Data 튜토리얼](#)
- [Barndoar Light Filter 사용하기\(AiBarndoar\)](#)
- [물병 만들기](#)
- [Curvature 쉐이더로 마모 효과 만들기](#)

이들 자습서와 함께 제공되는 많은 장면 파일은 하단 링크의 사이트에서 다운로드 할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Learning+Scenes>)

Anisotropic Brushed Metal



냄비 바닥의 이방성 브러시 금속 효과

이 짧은 자습서는 이방성(Anisotropic) 반사 특성을 가진 사실적인 강철 쉐이더를 만들어서 냄비 바닥에 브러시 금속 효과를 연출하는 방법을 보여줍니다. 이방성 반사는 반사 표면에 존재하는 작은 홈(돌출부, 섬유질 또는 스크래치)의 방향이 그 기준입니다. 이방성에 연결된 방사형 램프와 범프 맵 속성에 연결된 노이즈 텍스처의 조합을 사용하여 이러한 효과를 얻을 수 있습니다.

Maya 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

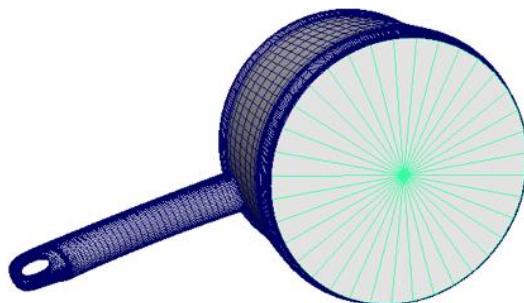
(<https://support.solidangle.com/display/A5AFMUG/Anisotropic+Brushed+Metal>)

이방성 반사에 대한 자세한 내용은 하단 링크에서 보실 수 있습니다.

<https://support.solidangle.com/pages/viewpage.action?pageId=40111476>

<https://support.solidangle.com/pages/viewpage.action?pageId=40111353>

- 냄비의 바닥을 선택하고 **Standard Surface** 쉐이더를 지정합니다. 이름을 'Base'로 변경합니다.



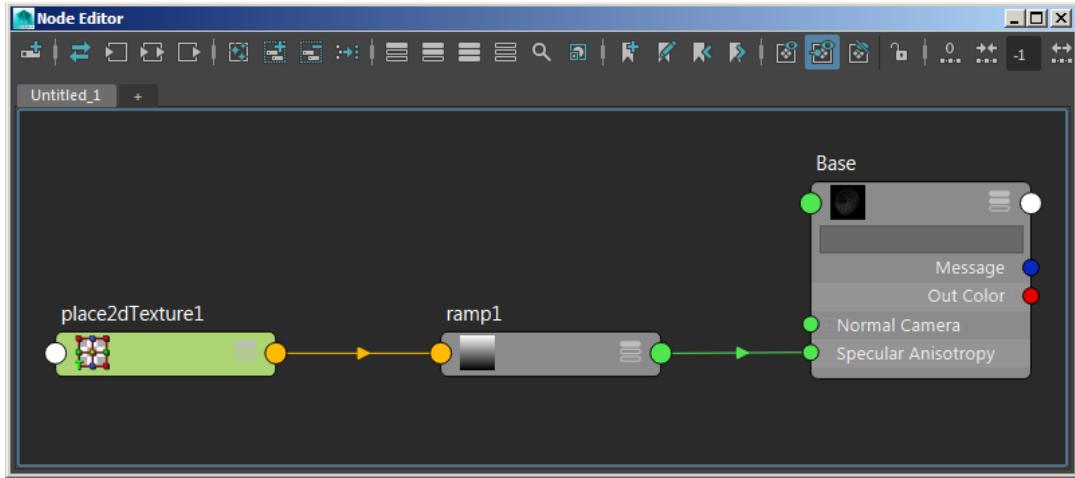
Standard Surface 쉐이더를 냄비의 바닥에 지정합니다.

Specular Shading

이제 브러시 금속 효과를 가진 강철 쉐이더를 만들어 보겠습니다.

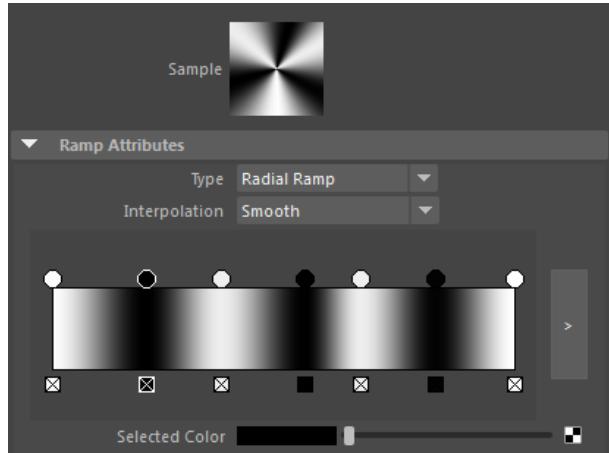
- **Metalness**를 1로 높입니다. **Specular Color**를 mid-grey로 줄이고 **Roughness**를 0.8 정도로 높입니다. 그러면 부드러운 이방성 효과가 적용됩니다.

- ramp를 *Specular Anisotropy* 속성에 연결합니다.

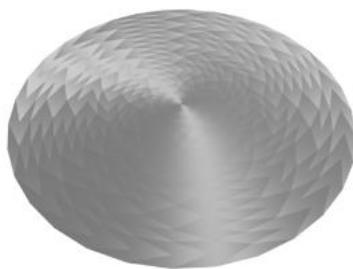


Standard 쉐이더의 Specular Anisotropy 속성에 램프가 연결됨

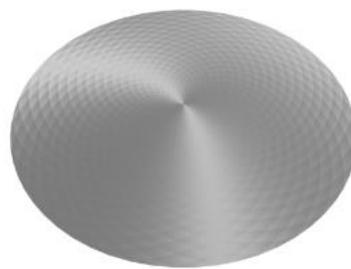
- Ramp Type을 *Radial Ramp*로 변경하고 Interpolation을 *Smooth*로 변경합니다. 아래 이미지와 같이 램프에 흑백의 색을 삽입합니다.



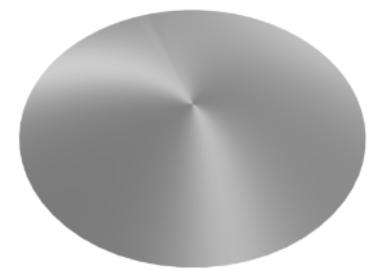
이방성을 사용할 때 반사 하이라이트에 각면이 나타날 수 있습니다. 해당 메시의 Arnold 속성에서 **Smooth Tangents**를 활성화하여 이러한 각면을 제거할 수 있습니다. 이를 위해서는 폴리곤 메시에서 최소 하나의 **Subdivision Iteration**이 필요함을 기억하십시오.



Subdivision Iteration 없음



Subdivision Iteration: 1



Subdivision Iteration: 1 (Smooth Tangents 활성화됨)

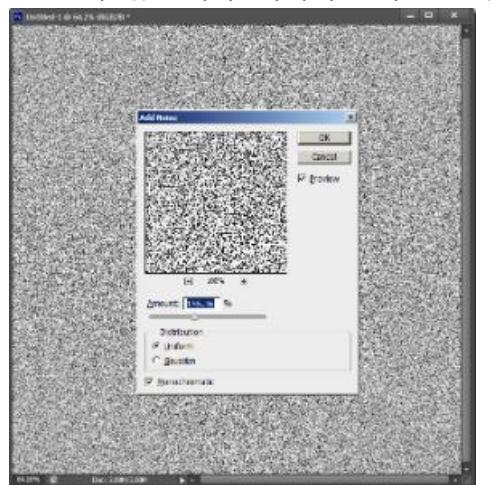
이방성 방향은 사용 중인 메시의 UV가 기준입니다. UV를 적절하게 생성하여 월드 공간에서 흙의 원형 방향을 따르도록 해야 합니다. 이 방법을 사용하면 램프가 필요하지 않으며, 물체나 카메라 위치가 변경될 때 정확한 이방성 반사를 얻을 수 있습니다.

이제 냄비 바닥에서 이방성 브러시 금속 효과를 볼 수 있습니다. 하지만 이 효과와 관련된 돌출된 부분은 부족합니다.

Bump Map

이제 스크래치 맵을 작성하고 범프 속성에 연결하여 세밀한 브러시 금속 효과를 만들어야 합니다. 이것은 Photoshop에서 가능합니다.

- **Filter->Noise->Add Noise**로 이동하여 이미지에 약간의 노이즈를 추가합니다.
- **Filter->Distort->Twirl**로 이동하여 노이즈에 원형 효과를 추가합니다. 이것을 두 세 차례 반복해야 원하는 결과를 얻을 수 있습니다. 이미지를 자르고 저장합니다.

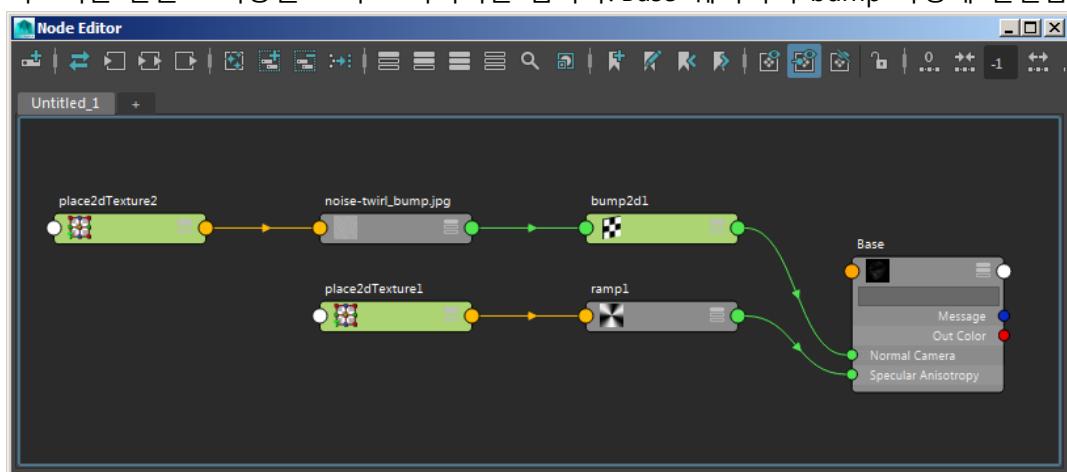


Photoshop의 Noise 필터



Photoshop의 Twirl 필터

- 파일 텍스처를 만들고 저장된 노이즈 이미지를 엽니다. Base 쉐이더의 bump 속성에 연결합니다.

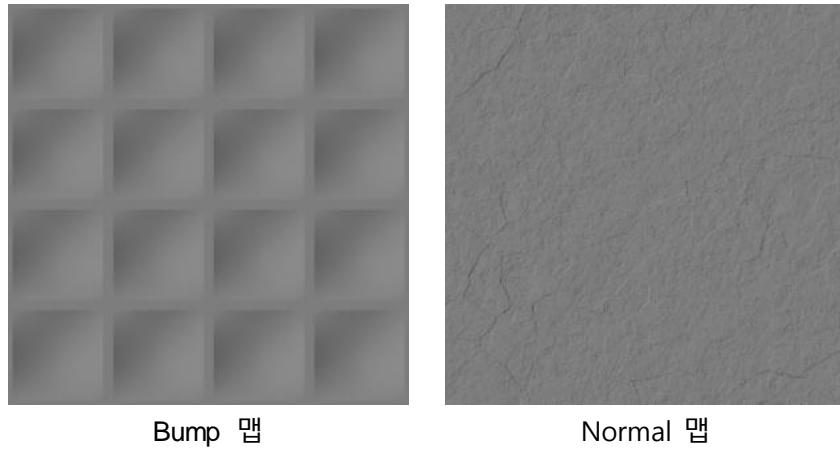


Normal과 Bump 맵 결합하기

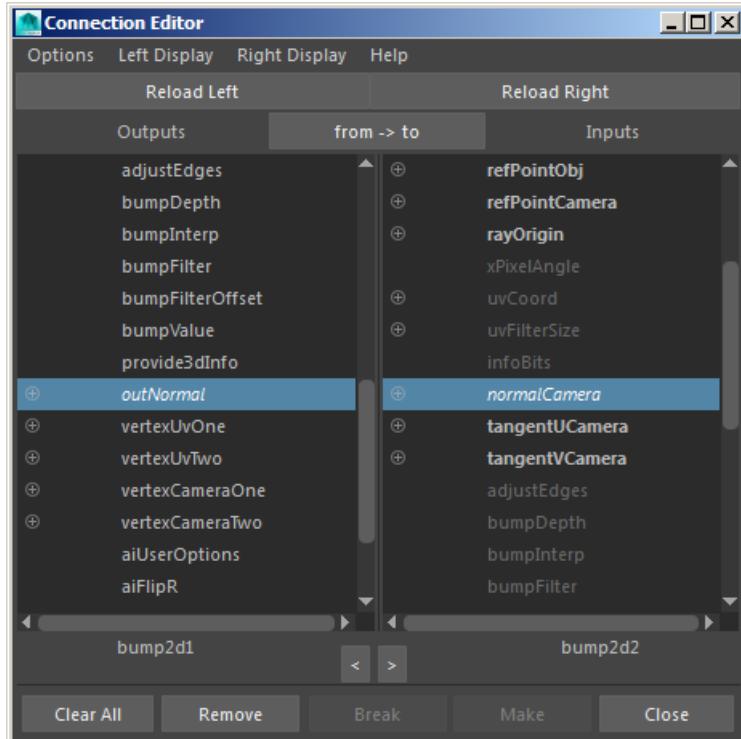
이 간단한 자습서는 하나의 쉐이더에서 Normal 맵과 Bump 맵을 결합하는 방법을 보여줍니다. Maya에서 쉐이더에 일반 맵을 추가하면 기본적으로 Bump 속성에 연결됩니다. 이 기술은 두 개의 Normal 맵을 결합하는 데에도 사용할 수 있습니다.

Maya 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

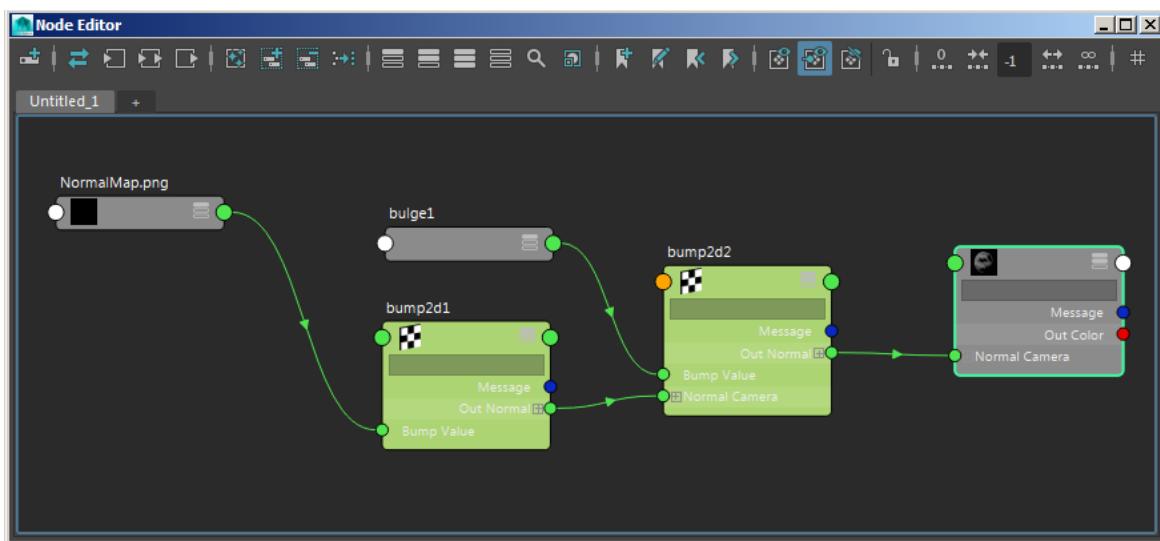
(<https://support.solidangle.com/display/A5AFMUG/Combining+Normal+and+Bump+Maps>)



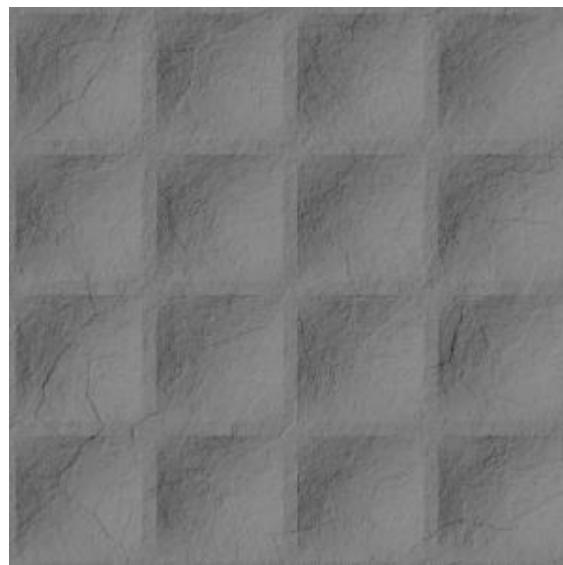
bump2d 노드를 결합하려면 **outNormal**을 첫 번째 **bump2d** 노드에서 두 번째 노드의 **normalCamera**에 연결해야 합니다. 이것은 Connection Editor를 통해 가능합니다('Show Hidden'을 활성화해야 **normalCamera** 속성을 볼 수 있음).



먼저 범프 노드를 Standard Surface 쉐이더에 연결한 다음 일반 맵 노드를 추가할 수 있습니다.



첫 번째 bump2d 노드의 OutNormal이 두 번째 노드의 normalCamera에 연결됨



Bump 맵과 Normal 맵의 결합

Displacement 튜토리얼

다음 튜토리얼들은 MtoA에서 Displacement 사용에 대해 설명합니다.

- [<udim> Token을 사용한 Displacement 맵핑](#)
- [Displacement로 바다 렌더링하기](#)
- [Mudbox에서 Arnold로 Vector Displacement](#)

<udim> Token을 사용한 Displacement 매핑



이 간단한 자습서의 목적은 Arnold와 함께 올바른 Displacement 워크플로우를 사용하여 Mari에서 내보낸 텍스처 맵을 렌더링하는 방법을 설명하는 것입니다. 이 예제는 크리쳐 모델과 Displacement 맵(Alex Huguet이 제공해주시신)을 사용합니다.

UDIM 텍스처 토큰에 대한 추가 정보는 하단 링크 사이트에 있습니다.

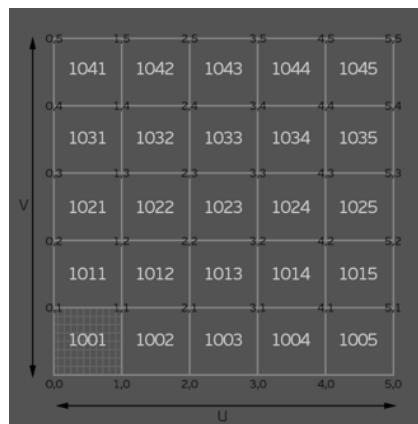
(<https://support.solidangle.com/display/A5AFMUG/Tokens>)

UV TILE - UDIM

UDIM 값은 UV 공간 내 왼쪽 하단 좌표로부터 텍스처의 정수 좌표를 표현하는 방법입니다. 이 방법을 사용하면 하나의 훨씬 더 큰 텍스처가 아닌 여러 텍스처를 사용하여 모델을 커버할 수 있습니다.

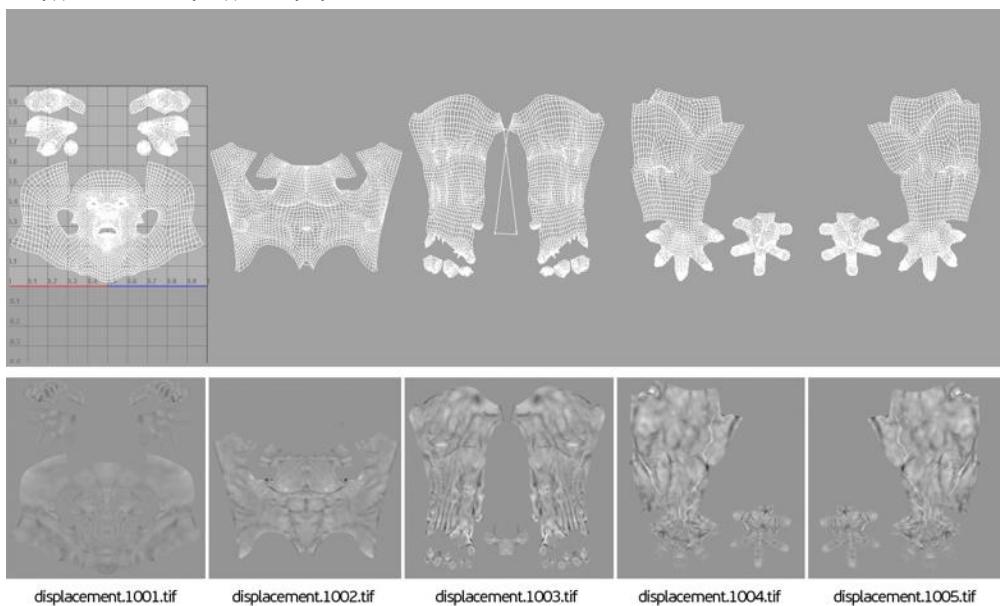
이것은 UV 텍스처 공간을 내보낼 때 Mari에서 사용하는 규칙입니다. 1001 (0,0)에서 시작해서 10개의 추가 오프셋에 대해 U에서 지속됩니다. 그 다음 행인 V로 계속 진행되고, U에서 10마다 V에서 1 이동됩니다.

1021 1022 1023 1024 1025 1026 1027 1028 1029 1030
1011 1012 1013 1014 1015 1016 1017 1018 1019 1020
1001 1002 1003 1004 1005 1006 1007 1008 1009 1010



옆의 UV 오프셋 격자는 이 관계를 보다 잘 보여줍니다(이 예제는 크리쳐 모델이 5개의 Displacement 맵을 사용함에 따라 5개의 범위만 보여줍니다).

UV Texture Editor에서 UV가 1001 (0,0) ~ 1005 (5,0)로 UV 공간에 배치되었고 그에 따라 Displacement 텍스처 파일의 이름이 지정되었음을 알 수 있습니다.



텍스처 경로

Displacement 파일 텍스처를 열 때 Arnold은 '상대적' 파일 경로 명명 규칙을 사용합니다. 텍스처 렌더링에 문제가 있는 경우 파일 경로 이름을 '절대' 경로로 변경하십시오(프로젝트 디렉토리를 변경하면 경로 이름이 자동으로 '절대'로 변경됩니다).

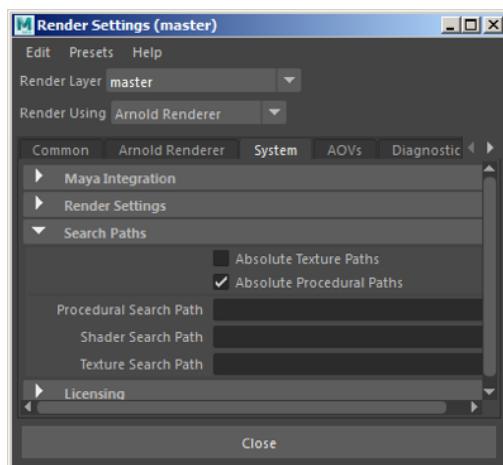
파일 입력에서 이 방법을 사용하려면 파일 이름에 <udim> 태그를 사용하고 적절한 번호로 대체합니다.

Relative file path - **Textures\displacement.<udim>.tif**

Absolute file path - **C:\Users\Documents\Project\Textures\displacement.<udim>.tif**

Name	Size	Modified	Type
displacement.1001.tif	4.861.444	Long ago	TIF File
displacement.1002.tif	3.868.220	Long ago	TIF File
displacement.1003.tif	5.581.792	Long ago	TIF File
displacement.1004.tif	4.149.408	Long ago	TIF File
displacement.1005.tif	20.926.864	Long ago	TIF File

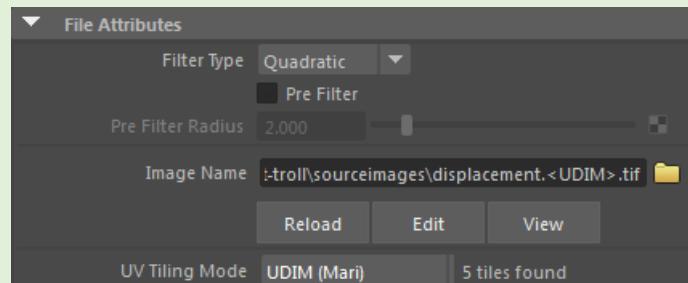
또 다른 옵션은 'Use Absolute Texture Paths'를 비활성화하는 것입니다. 이 설정은 Render Settings 창에서 System, Search Paths 아래에 있습니다.



'Absolute Texture Paths'는 Render Settings 창 안의 System>Search Paths에 있습니다.

또한 파일 2d 텍스처 노드를 음영 처리 네트워크에 연결할 수 있으며, 속성 편집기에서 Image Name 속성에 UV 타일 중 첫 번째를 로드합니다.

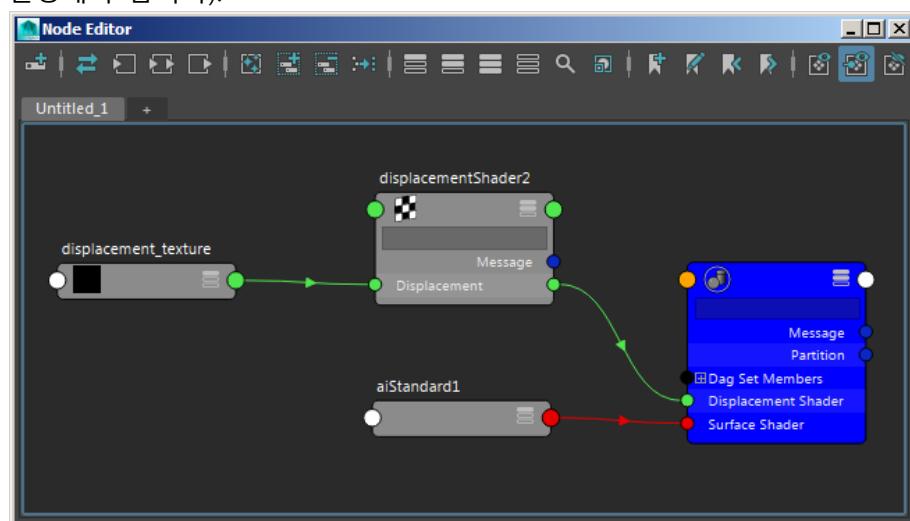
UDIM (Mari) UV 타일링 모드 선택 - UV 좌표가 수식 $1000 + (u + 1 + v * 10)$ 을 사용하여 4 자리 숫자로 표시되는 경우 이 옵션을 선택합니다.



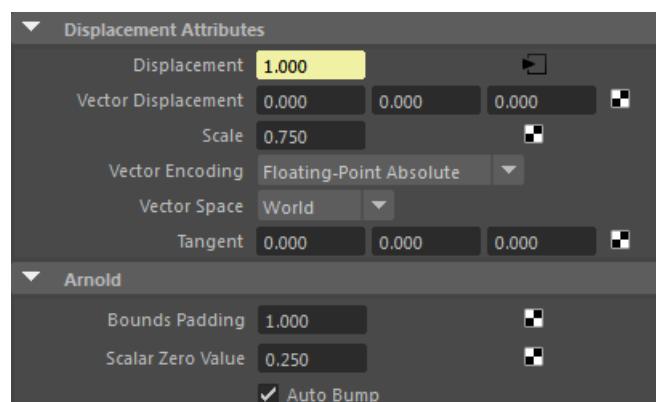
Displacement Attributes

(C:\Users\Documents\projects\Displacement-troll\sourceimages\displacement.<udim>.tif의 절대 경로 이름을 사용하는) 파일 텍스처는 해당 크리쳐에 할당된 쉐이더에 연결된 Displacement 노드에 연결됩니다.

Vector Space는 'World' 공간으로 설정되고 scale은 0.75로 설정되었습니다. 'Bounds Padding'은 1로 설정되고 'Scalar Zero Value'는 0.25로 설정되었습니다(이들 값은 개략적인 가이드이며 해당 Displacement 맵이 생성된 방법 및 위치에 따라 변경해야 합니다).



Displacement 쉐이더 네트워크

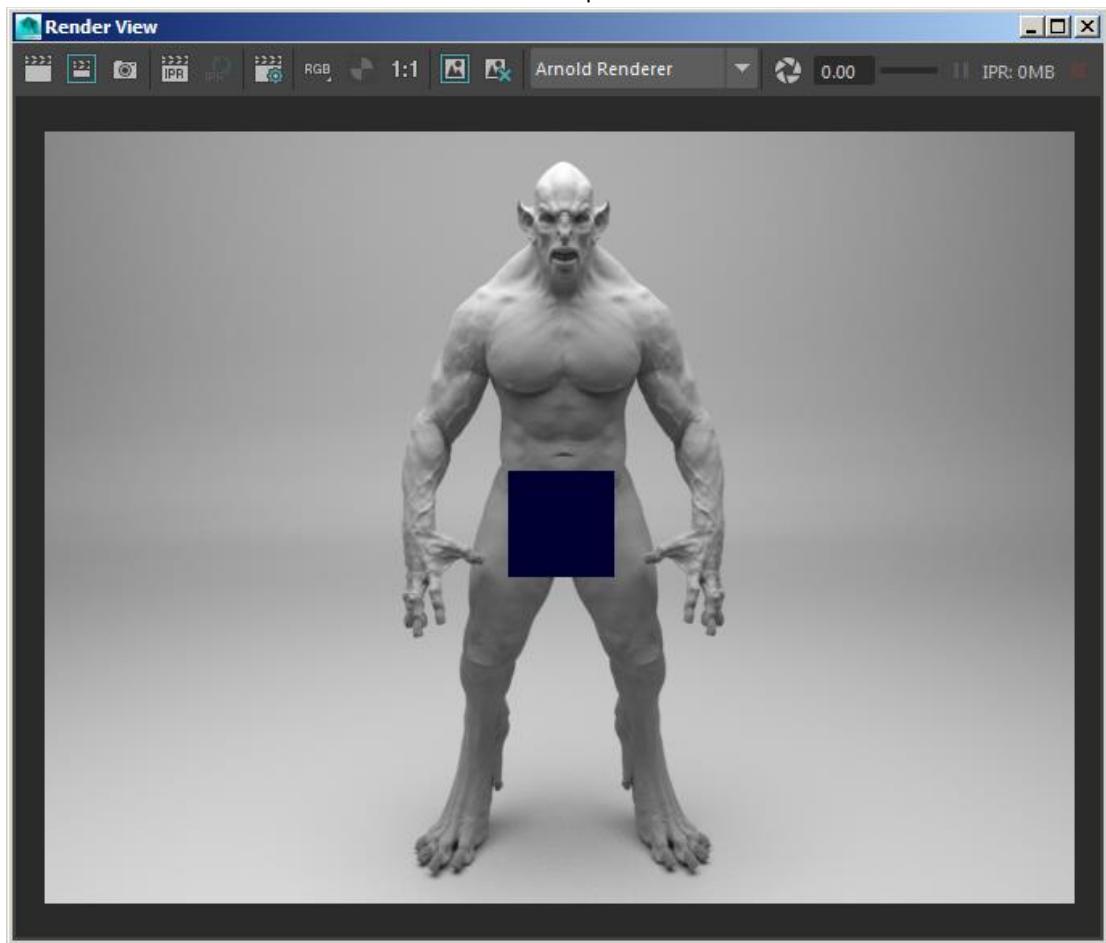


이 예제 사용된 Displacement 속성

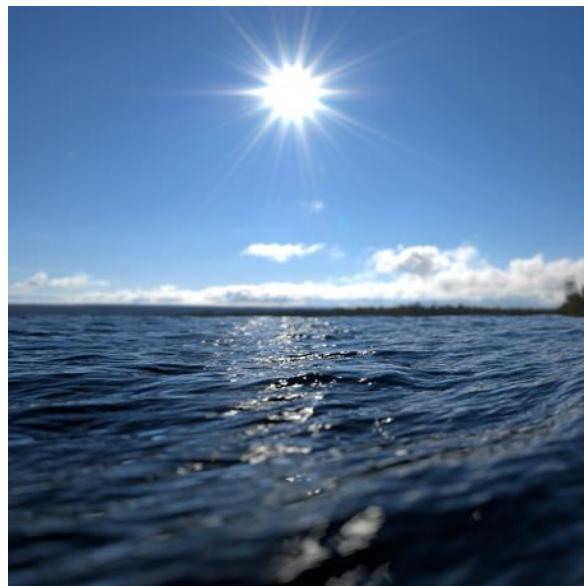
Subdivision

Displacement 맵에서 세부 사항을 보려면 모델의 구획 설정을 늘려야 합니다. 이 경우 구획 유형은 Catclark로 설정되어 있으며 반복이 2로 증가했습니다.

지금까지 <udim> 토큰을 사용하여 Mari로부터 저장된 Displacement 맵을 렌더링하는 방법을 알아봤습니다.



Displacement로 바다 렌더링하기

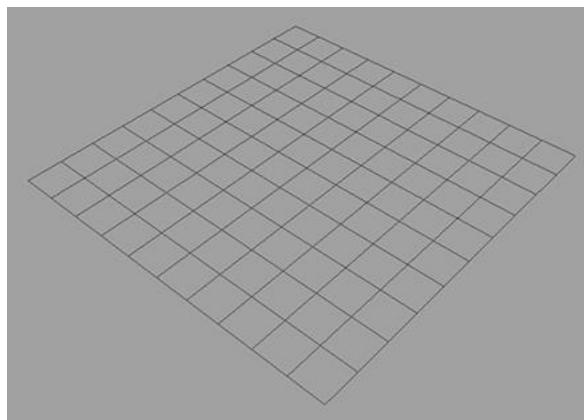


이 자습서에서는 폴리곤 평면을 사용하여 바다 장면을 렌더링하는 방법과 Skydome 조명에 연결된 HDR 이미지를 다루는 방법을 설명합니다. 바다 쉐이더의 세부 정보는 Vector Displacement 맵에서 가져옵니다. Displacement 매핑을 사용할 때는 다양한 속성을 설정해야 합니다. 그러나 올바른 설정을 사용하여 설정하고 설득력 있는 결과를 얻는 것은 매우 간단합니다.

바다를 위한 Displacement 맵은 하단 링크의 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Rendering+an+Ocean+with+Displacement>)

- 먼저 해수면으로 사용할 수 있는 폴리곤 평면을 만들겠습니다.



Scene 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Rendering+an+Ocean+with+Displacement>)

Ocean Material

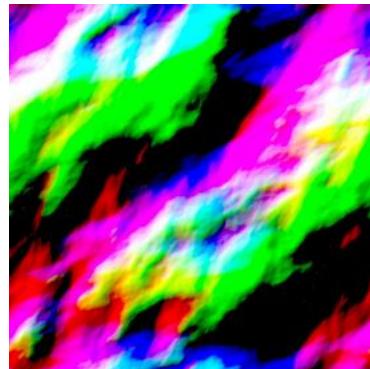
- Standard Surface 쉐이더를 생성하고 폴리곤 평면에 할당합니다. **Diffuse**를 **0**으로 설정합니다. **Specular Weight**를 **1** 정도로 높입니다. **Transmission Weight**를 **1**로 높입니다(물이 투명해짐). 마지막으로 **index of**

refraction을 물의 수치인 **1.33**으로 설정합니다.

- 해수면을 투명하게 할 것이기 때문에 폴리곤 평면에 대해 **Opaque**가 비활성화되어 있는지 확인합니다.

Vector Displacement Map

사용되는 Vector Displacement 맵은 32비트 EXR 이미지입니다. 따라서, Vector Displacement 효과에 최고 품질을 유지할 수 있습니다. 보다 현실적인 파도 효과를 얻기 위해 Vector Displacement 맵을 사용하여 바다의 파도를 대체합니다. 그러면 지오메트리를 한 방향으로만 옮기는 기존의 Displacement 맵과는 다르게 바다 지오메트리가 보통과 다른 방향으로 이동됩니다.



바다 Displacement를 생성하는 데 사용되는 Object Space Vector 맵

(다운로드: <https://support.solidangle.com/display/A5AFMUG/Rendering+an+Ocean+with+Displacement>)

파도의 효과가 너무 날카로워 보인다면 부드러운 파도를 표현하기 위해 Vector Displacement 맵을 흐리게 할 수 있습니다.

- 벡터 맵을 Maya Displacement 노드에 연결하고 바다 재질로 사용되는 Ai Standard의 음영 처리 그룹에 연결합니다. 쉐이더 네트워크는 바다 재료의 음영 처리 그룹 노드에 연결된 Maya Displacement 노드에 연결되어 있는 Vector Displacement 맵으로 구성되어야 합니다.

Ai Skydome light

Skydome 조명을 만들고 HDR 맵을 해당 Color 속성에 연결합니다.

Subdivision settings

- 장면을 렌더링합니다. Displacement가 없습니다! 그 이유는 바다 평면의 지오메트리에 충분한 구획이 없기 때문입니다.



- 해당 평면을 선택하고 Arnold 속성에서 **Subdivision type**을 **Catclark**로 변경하고 **iteration** 횟수를 늘립니다. 먼저 3과 같은 낮은 숫자부터 시작합니다.



Catclark, Subdivision Iterations 3으로 설정된 Subdivision Type

- 바다가 지금도 괜찮아 보이지만 좀 더 디테일이 필요합니다. **Subdivision Iterations**를 **8**로 높입니다. 한 번 높일 때마다 메모리 사용량과 렌더링 시간이 증가할 수 있으므로(각 반복이 지오메트리를 4배로 늘림) 이 값을 너무 높이지 않도록 주의하십시오.



Subdivision Iterations 8. 자세한 디테일은 바다의 파도에서 볼 수 있습니다.

Lighting

- 마지막으로 장면에 **Directional Light**을 추가합니다. 이를 통해 해수면에 태양의 효과를 과장하여 해수면의 하이라이트를 표현할 수 있습니다.



Directional Light으로 렌더링 됨

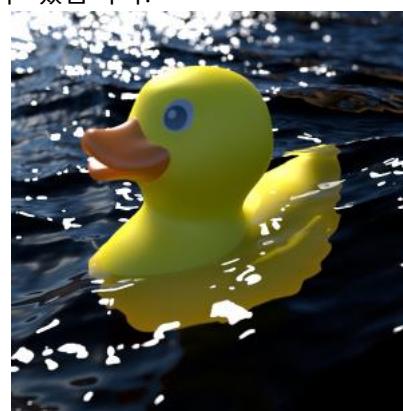
- 방향 조명의 **Angle** 속성을 높이면 조명의 각도 크기가 증가하여 해수면에 큰 정반사가 생성됩니다.



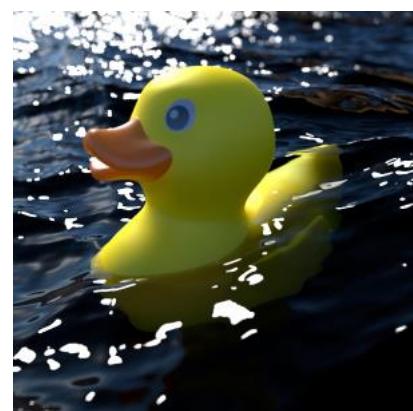
Angle 5

Transmission

해수면을 관통하는 빛은 물속에서보다 공기 속에서 더 빨리 이동하기 때문에 '굴절'됩니다. 빛이 물에 도달하면 바다의 입자들에 의해 흩어지거나 흡수됩니다. 바다가 깊어지면 물은 빛의 침투 깊이를 감소시킵니다. Standard Surface 쉐이더의 Transmission 속성을 사용하여 이 효과를 매우 쉽게 재현할 수 있습니다. 이 효과는 물체를 해수면 속에 배치할 때 볼 수 있습니다. 더 긴 빛이 메시 내부로 이동할수록 푸른 Transmission Color의 영향을 많이 받는다는 것을 알 수 있습니다.

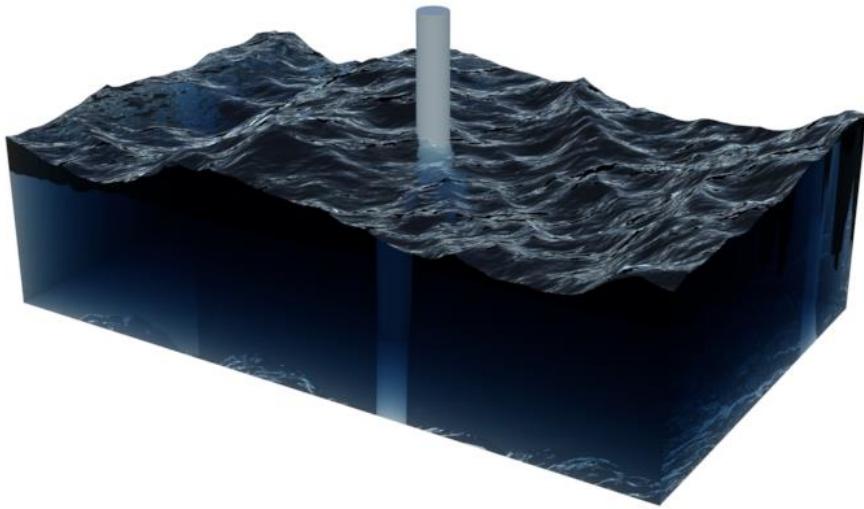


Transmission Color: 흰색



Transmission Color: 파란색

바다 쉐이더에 적용할 때 Transmission Color와 Depth가 아래의 닫힌 해수면 안의 흰색 실린더에 미치는 영향을 볼 수 있습니다.



Dark Patches

바다 쉐이더는 완전히 투명하지 않습니다. 즉, 반짝이지 않는 광선은 반사되지 않고 대신 굴절하여 물속을 통과합니다(실제와 같이). 하지만 이것은 투명하지 않으므로 광선을 중단하고 검게 만들겠습니다.

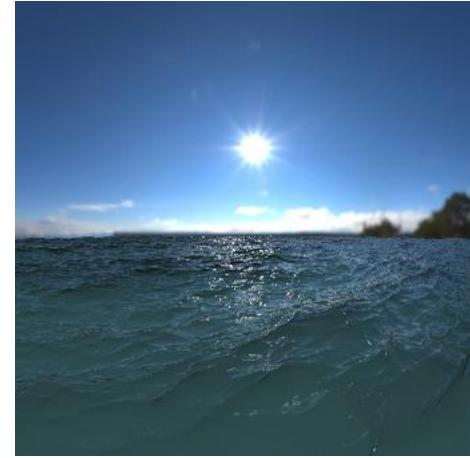
음영 처리 법선은 지오메트리 법선이 광선을 향하고 있어도 음영 처리 법선이 광선에서 멀리 떨어진 곳을 가리킬 때 문제를 일으킬 수 있습니다. 이 경우 음영 처리(Auto Bump) 법선은 물의 뒷면에 닿고 아래로 반사되어야 하지만 실제로 광선은 물 메시의 윗부분에 닿습니다.

지표각에서 실제 물은 거의 100% 반사되지만, 이 경우에도 빛이 물결 무늬 뒤의 골 부분에서 빠져나올 때까지 여러 번 반사될 수 있습니다. 이렇게 몇 번 반사된 후에도 주변의 물에 반사되는 것보다 수평선으로부터 더 높이 있는 물체에 반사될 수 있습니다.

보다 사실적으로 표현하려면 물을 투명하게 만들고 물 아래에 해수의 색깔을 가진 평면을 놓으면 됩니다. 아래 이미지들에서 해수면의 Standard 쉐이더는 1의 Emission Scale과 파란색/녹색 Emission Color를 가지고 있습니다. Subdivision Iterations을 증가시키면 Displacement된 바다의 파도에서 검은 부분을 줄이는 데도 도움이 됩니다.



Subdivision Iterations 3. 검은 부분이 보임



Subdivision Iterations를 8로 증가시키면 없어짐

물결 무늬(자체 반사) 뒤에 있는 어두운 부분은 또한 Render Settings에서 **Specular Ray Depth**를 높여서 없앨 수도 있습니다.



Specular Rays: 1

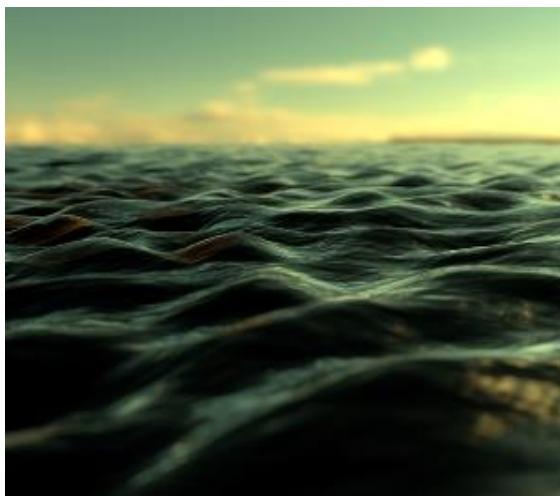


Specular Rays: 2

결론

지금까지 그럴 듯한 바다 장면을 만드는 방법에 대해 알아봤습니다. 올바른 Displacement 설정을 고품질 32비트 Vector Displacement와 함께 사용하고 올바른 Standard 쉐이더 설정을 사용하면 사실적인 표현을 얻을 수 있습니다. 피사계 심도를 추가하거나 카메라 렌즈를 어안 렌즈와 같은 것으로 변경해서 장면에 시각적 흥미를 더해보세요.

아래에서 Vector Displacement 맵을 사용하여 파도 효과를 생성하면 일반 지오메트리를 사용할 때와 비교하여 훨씬 더 세부적인 결과를 얻게 된다는 것을 알 수 있습니다.



폴리곤 지오메트리를 가진 파도



Vector displacement 맵은 지오메트리를 사용하는 것보다 훨씬 더 많은 디테일을 표현해줄 수 있습니다.

Mudbox에서 Arnold로 Vector Displacement

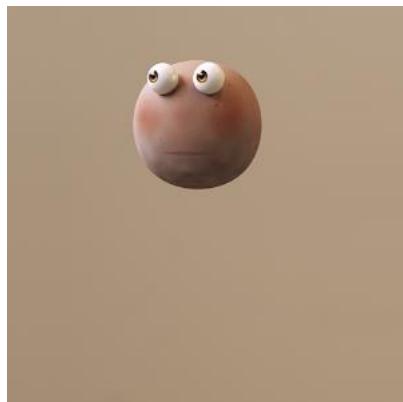


이 자습서에서는 구형 및 Vector Displacement 맵만으로 헤드 모델을 렌더링하는 과정을 설명합니다. Vector Displacement 맵은 복잡한 지오메트리를 관리해야 하는 부담 없이 장면에 디테일을 추가할 수 있는 강력하고 효율적인 방법입니다. 이들은 3D 조각 애플리케이션의 모델과 Arnold로 작업할 때 매우 효율적인 워크플로우를 제공합니다. 정점 법선에 대해 단순히 긴 선을 사용하는 전통적인 그레이 스케일 Displacement 맵과 달리 Vector Displacement 맵에는 XYZ Displacement 방향과 크기를 나타내는 RGB 값이 있습니다. 즉, 고정된 표면 법선을 따르지 않고 서로 중첩되는 등 모든 방향으로 이동할 수 있다는 의미입니다.

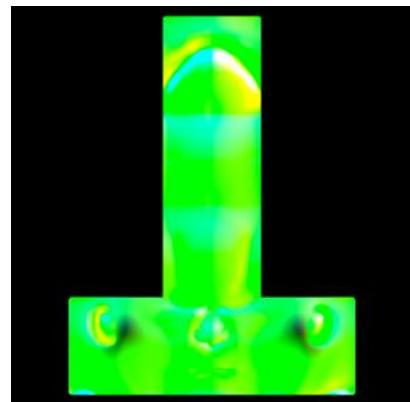
그러나 Vector Displacement 맵은 일반 Displacement 맵보다 많은 정보를 저장해야 하므로 파일 크기가 커집니다. 물론 Vector Displacement 맵은 일반적인 Displacement 맵과 마찬가지로 모든 상황에 완벽하지는 않습니다. 예를 들어, Vector Displacement 맵을 사용할 때는 UV seam이 문제가 될 수 있습니다.

sphere.obj, Vector Displacement 맵 및 Diffuse 색상 맵은 하단 링크의 사이트에서 다운로드 할 수 있습니다.
 (<https://support.solidangle.com/display/A5AFMUG/Vector+Displacement+from+Mudbox+to+Arnold>)

sphere.obj를 임포트할 때 *Texture Coords*를 활성화 하세요, 그렇지 않으면 UVs가 나오지 않고 정확하지 않은 Displacement 결과를 얻게 됩니다.



Vector Displacement가 없는 구체



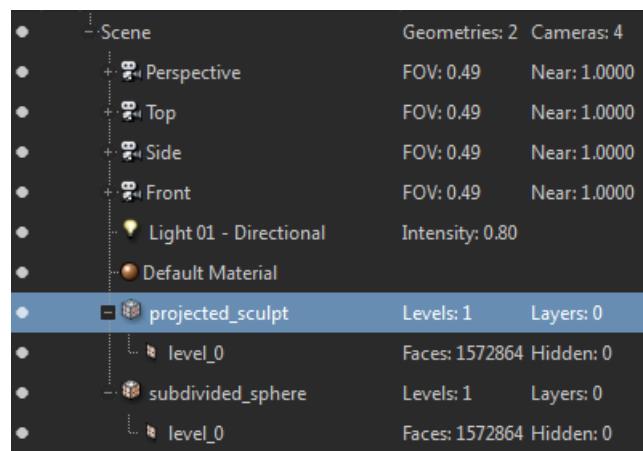
Vector Displacement 맵



Vector Displacement가 있는 구체

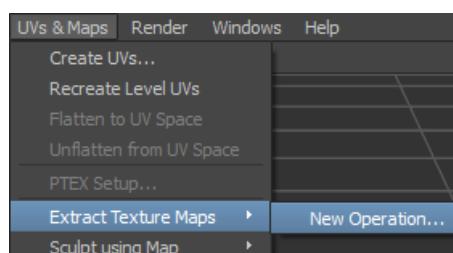
Mudbox

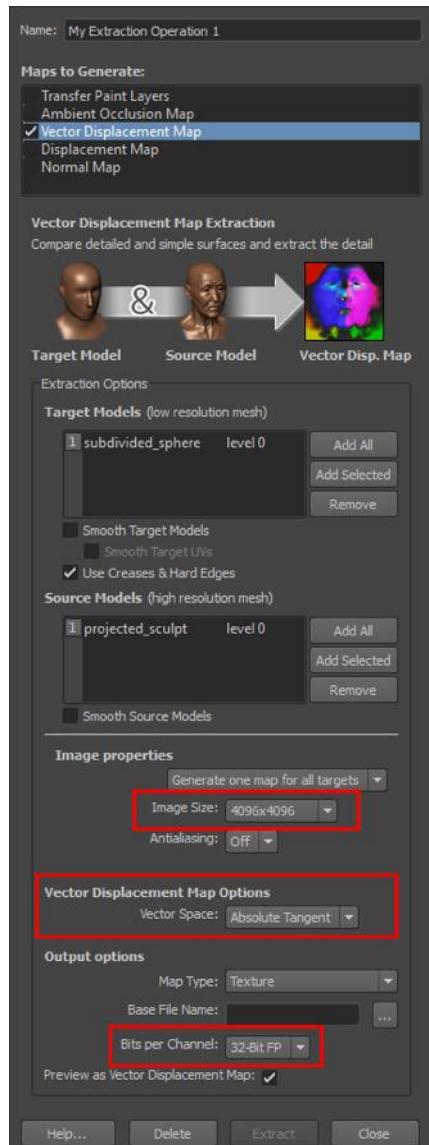
원래 헤드 조각은 Mudbox에서 다음 설정을 사용하여 추출되었습니다.



Mudbox sculpt 세부사항

- Vector Displacement는 Mudbox에서 **UVs & Maps > Extract Texture Maps > New Operation** 메뉴를 통해 원래 헤드 모델에서 추출했습니다.



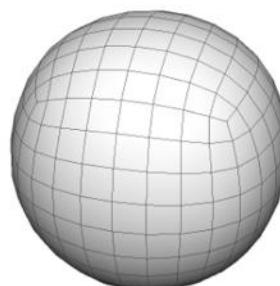


- Mudbox는 모델로부터 텍스처를 추출하기 위해 다음 설정을 제공합니다. 이 경우 **Vector Displacement Map**을 선택하고 **Vector Displacement**을 **Absolute Tangent**로 변경했습니다. 맵이 **32비트 FP** 깊이로 저장되고 해상도가 충분히 높아야 합니다. 그러면 Arnold에서 정확하게 Displacement된 렌더링을 얻을 수 있습니다.

32비트 또는 16비트 부동 소수점 형식을 사용하여 정수 형식이 아닌 이미지를 저장해야 합니다. 정수 형식은 올바르게 작동하지 않습니다. 이는 정수 형식이 부동 소수점 Displacement 맵에 의해 사용되는 음수 픽셀 값을 지원하지 않기 때문입니다.

Maya

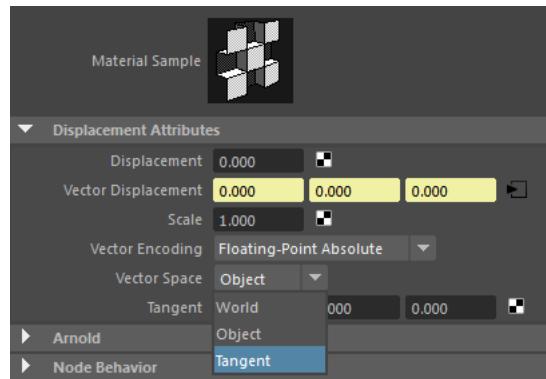
- 모델 - **sphere.obj**를 엽니다. 그러면 **obj_base**라는 구획이 나눈 큐브가 나타납니다.



obj_base

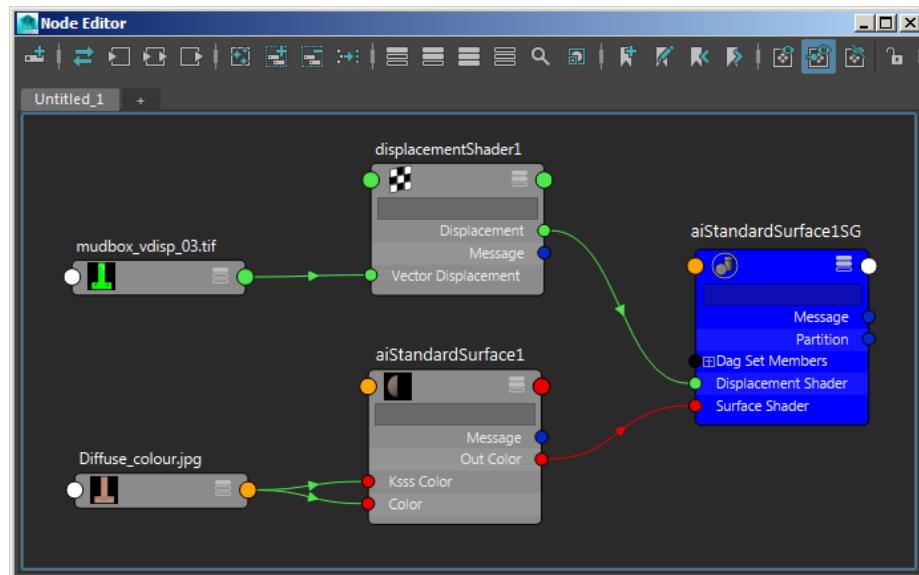
Vector Displacement 쉐이더

- **Standard Surface** 쉐이더를 구체 모델에 할당합니다. 이미지 파일 텍스쳐를 만들고 *Diffuse_colour.jpg* 파일을 엽니다. 이것을 **Base Color** 및 **SSS Color** 속성에 연결합니다. **SSS Weight**를 **0.5** 정도로 높입니다.
- **Displacement** 쉐이더를 생성하고 이것을 **Standard** 쉐이더의 음영 처리 그룹의 **Displacement mat.** 속성에 연결합니다.
- **Image file** 텍스쳐를 만들고 **Vector Displacement map**을 엽니다. - *mudbox_vdisp.tif*. 이것을 아래 이미지처럼 **Displacement** 쉐이더의 **Vector Displacement** 속성에 연결합니다. **Scale**을 **1**로 높이고 **Vector Space**가 **Tangent**로 설정되어 있는지 확인합니다.



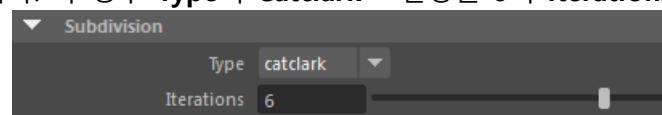
Vector Space가 Tangent로 설정되어 있습니다.

최종 쉐이딩 네트워크는 다음과 같은 모습입니다.

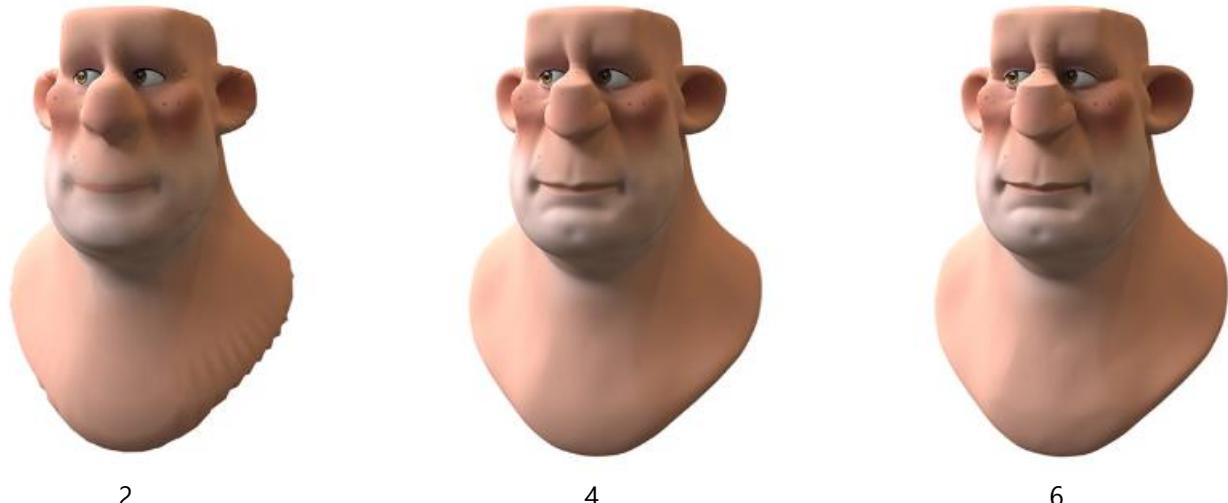


Subdivisions

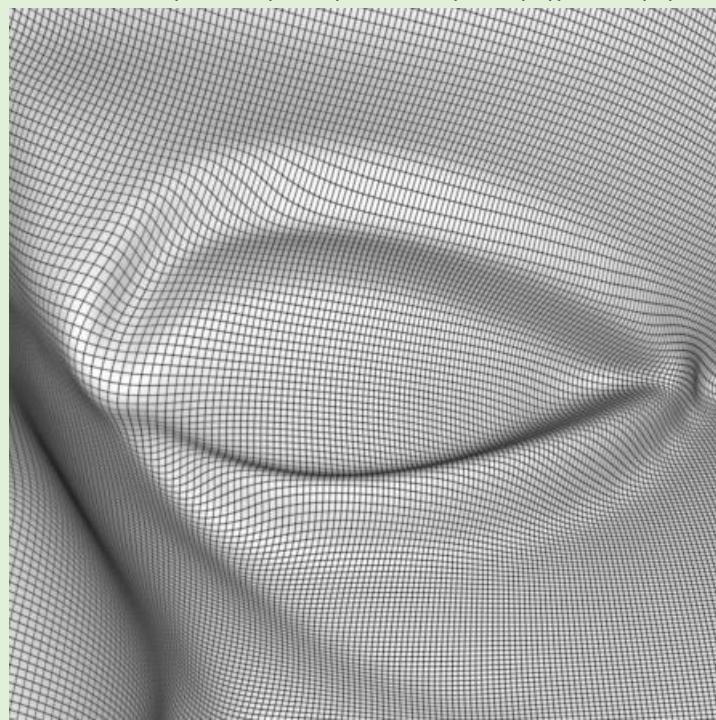
- 구체 지오메트리에 대해 충분한 **subdivisions**이 있는지 확인하십시오. 그렇지 않으면 Displacement가 올바르게 보이지 않습니다. 이 경우 **Type**이 **Catclark**로 설정된 **6**의 **iterations**면 충분합니다.



아래 이미지에서 4~6 subdivision iterations 사이의 미묘한 품질 차이를 볼 수 있습니다.



Utility 쉐이더는 Displacement 맵을 렌더링할 때 문제를 진단하는 데 유용합니다.



Utility 쉐이더. 'ndoteye'로 설정된 Shade 모드와 'polywire'로 설정된 Overlay Mode.

Flakes 예



Flakes 쉐이더를 사용하는 방법은 여러 가지가 있으며, 이 자습서에서는 그 중 일부를 설명하고자 합니다. 이 간단한 자습서에서는 Flakes 쉐이더(Standard Surface와 함께)를 사용하여 반짝이 효과를 만들어 보겠습니다.

Specular Color

- 다음은 *Specular Color*에 연결된 색상 텍스처 맵이 있는 기본 Standard Surface 쉐이더입니다.



Normal

- Flake 처리된 반사의 색상을 변형시키려면 하위 레이어에 Flake를 적용해야 합니다(*Geometry* 섹션의 *Normal - Bump Mapping*). 이 효과는 금속 자동차 페인트와 같은 재료에서 확실합니다.



Normal과 Coating (기본 White Color)

- *Coating Weight*를 높이면 흰색 *Coating Color*이 적용됩니다(기본). 이 효과는 자동차 페인트 또는 플라스틱에 유용하지만 반짝이는 화장에는 적합하지 않습니다. *Flake*를 *Coat Normal*에도 추가해야 합니다.



*Coating*에 지정하는 색상은 하위 *Base*, *Metal* 및 *Specular* 레이어에만 적용됩니다. *Coat* 레이어는 반투명 유전체 유형의 레이어로, 어떤 색상을 지정하든 상관 없이 늘 흰색을 반사합니다. *Coat* 레이어의 색상은 기본 *Base* 및 *Specular* 레이어에 적용됩니다. 예를 들어, 약간은 반 직관적으로, 녹색 유리는 녹색 반사를 갖지 않습니다. 녹색 유리 아래에 있는 것이 굴절됩니다. 이것은 유색 플라스틱에도 마찬가지입니다.

Normal 및 Coat Normal

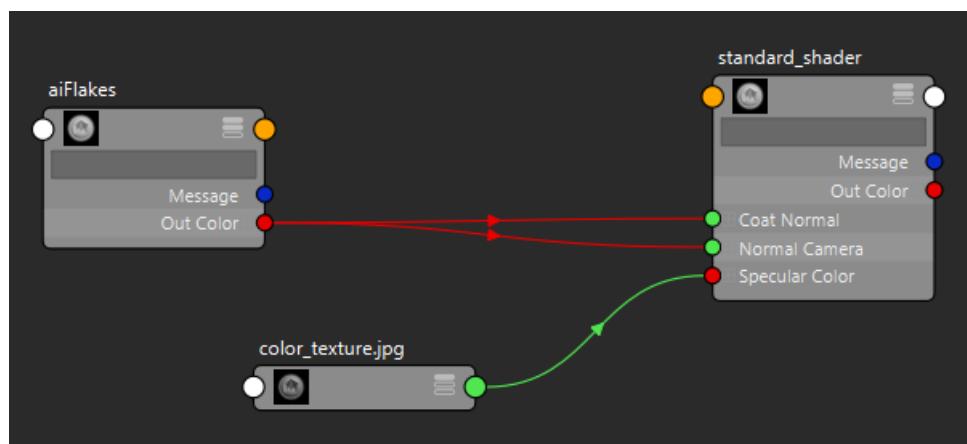
- *Flake*를 *Normal* 속성 및 *Coat Normal* 모두에 연결하면 우리가 원하는 최종 결과를 얻을 수 있습니다. 마지막으로 *Specular* 및 *Coat* 레이어 모두에서 *Normal*을 사용했습니다. 이 방법은 타당하지만, 외관은 메이크업에 에나멜/오일 등 반짝이는 것이 얼마나 섞여 있는지에 따라 달라집니다.



Flakes 쉐이더의 Output Space가 *World*로 설정되게 합니다.

자동차 페인트를 사용하면 *Coat* 레이어가 매우 두껍기 때문에 재현하기 쉬운 부드럽고 세련된 외관을 표현할 수 있습니다. 그러나, 이 메이크업 예제에서는 *Coat* 레이어가 매우 얇고 광택이 덜합니다. *Coat*의 *Normal* 레이어는 기저 피부의 유통불통함, 반짝이의 방향, 두껍고 얇은 부분이 있는 메이크업 자체의 분포, 그리고 얇은 액체 층의 표면 장력이 거의 모든 거친 면에 제공하는 부드러운 표면의 혼합이 됩니다. 메이크업이 말라서 맨 얼굴이 나타

나면 Coat 레이어의 표면에도 금이 가고 칙칙해집니다.



Flakes 쉐이더가 Standard Surface의 Normal Camera에 연결.

Flakes shader Coord Space (object)와 Output Space (World)

무지개 빛의 비누 방울 쉐이더

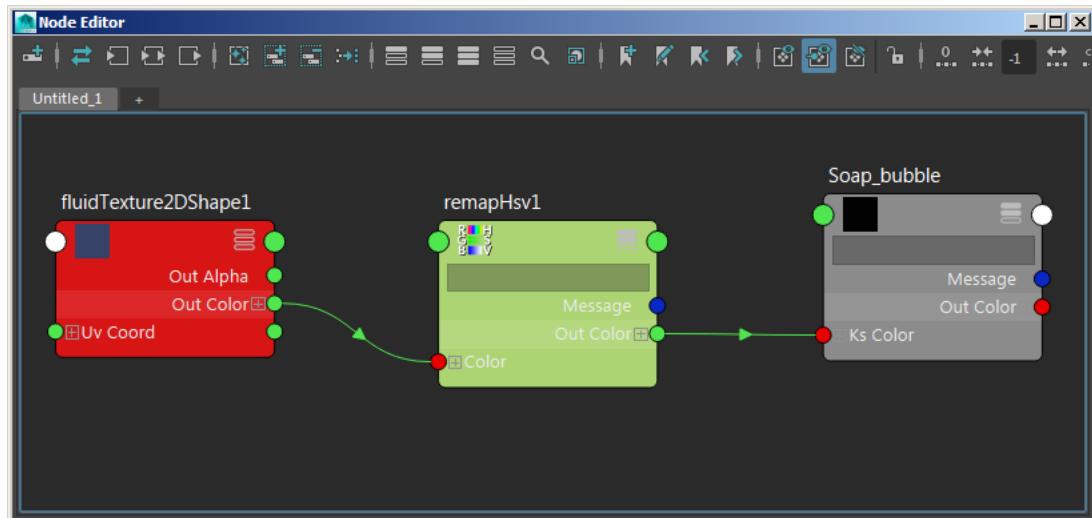
비누방울 쉐이더를 만드는 더 좋은 방법은 **Thin Film**을 사용하는 것입니다.



이 자습서에서는 비누방울에 소용돌이치는 무지개 색 효과를 쉐이딩하는 방법을 보여줍니다. **fluidTexture2DShape**을 사용하면 됩니다. 이 장면 파일은 **Standard Surface** 쉐이더의 *Specular Color*에 연결된 **fluidTexture2DShape**으로 구성되어 있습니다.

장면은 하단 링크의 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Iridescent+Soap+Bubble+Shader>)

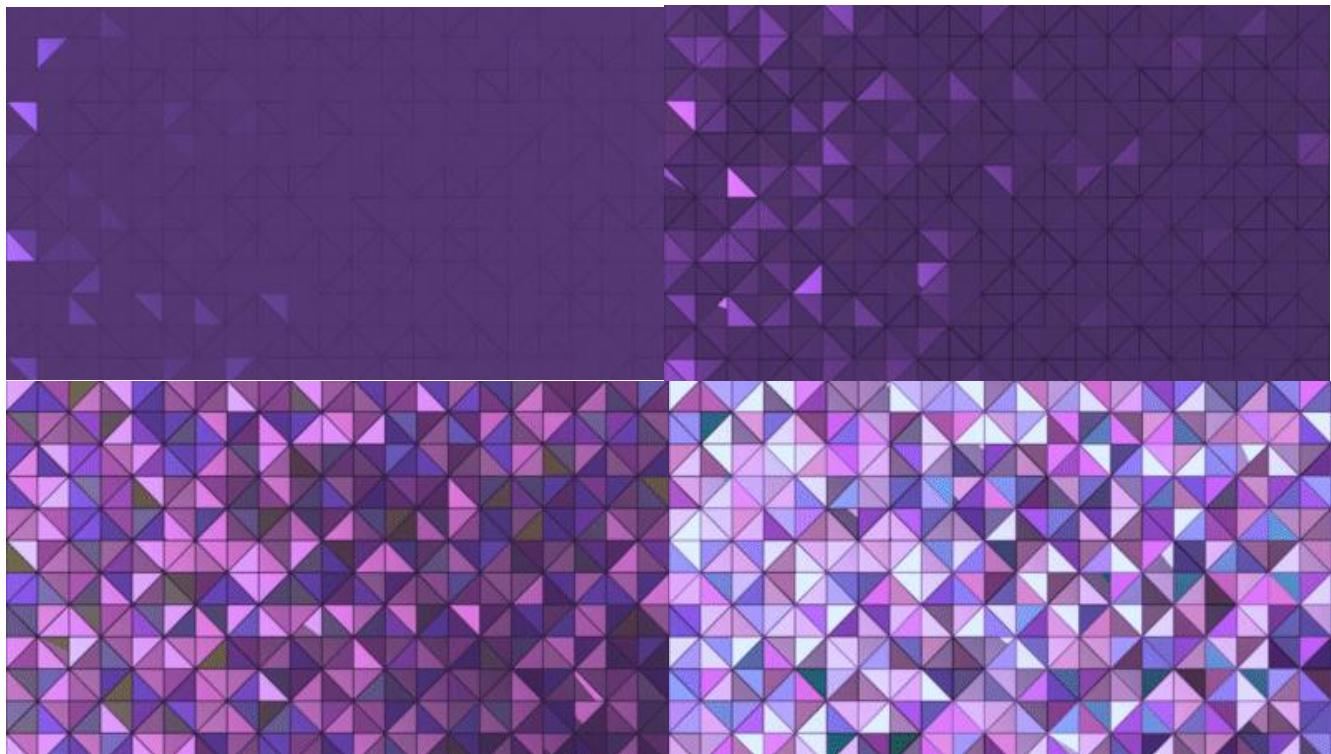


Specular Color에 연결된 fluidTexture2DShape

fluidTexture2D 효과를 보려면 애니메이션 시간 슬라이더를 재생해야 합니다.



Color Shaders를 사용한 모션 그래픽 효과



이 짧은 자습서에서는 Arnold의 일부 색상 쉐이더를 사용하여 모션 그래픽 애니메이션의 일부로 이용할 수 있는 흥미로운 효과를 만들어 보겠습니다. 이 애니메이션은 일부 키 프레임 애니메이션이 있는 Color Jitter 쉐이더에 연결된 **Utility** 쉐이더(Object ID 모드)에 의해 주로 구동됩니다.

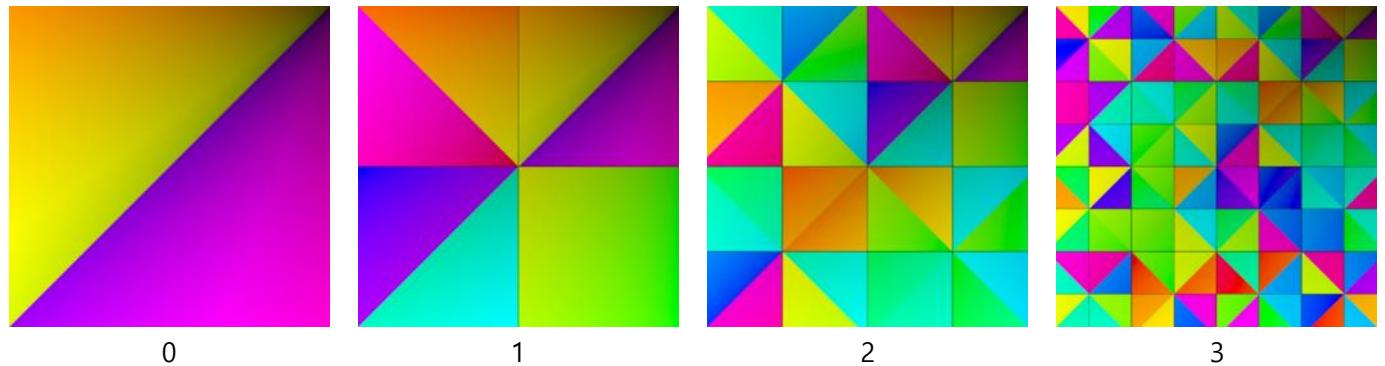
최종 장면은 하단 링크에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Motion+Graphic+Effect+Using+Color+Shaders>)

- 먼저 폴리곤 평면을 만들겠습니다.
- **Standard Surface** 쉐이더를 지정합니다.

Color Jitter 및 Utility Shaders

- **Color Jitter** 쉐이더를 Standard Surface 쉐이더의 **Emission Color**에 연결하고 **Emission Weight**를 1로 높입니다.
- **Utility** 쉐이더를 **Color Jitter** 쉐이더의 **Input**에 연결합니다. Utility 쉐이더에서 **Shade Mode**를 **Flat**으로 변경합니다. 그 이유는 Utility 쉐이더에서 순색만 사용하여 패턴을 만들고자 하기 때문입니다. **Overlay Mode**를 **Polywire**로 변경합니다. 그러면 색상 위에 와이어프레임이 오버레이됩니다.
- 이 경우 **Color Mode**가 **UV Coords**로 설정되었습니다. 애니메이트되면 패턴에서 대각선으로 닦아내는 효과가 나타납니다. 물론 *U* 또는 *V Coords*와 같은 다른 **Color Mode**나 완전히 다른 것을 선택할 수 있습니다. 이것은 여러 기법을 발견해내는 재미를 선사합니다!
- **Color Jitter** 쉐이더의 **Hue Max** (Face)를 높이면 Utility 쉐이더에 대한 효과를 볼 수 있습니다. 평면에 충분한 구획이 없으면 언제든지 해당 평면에 대한 **Subdivision Iterations** 수를 늘릴 수 있습니다.

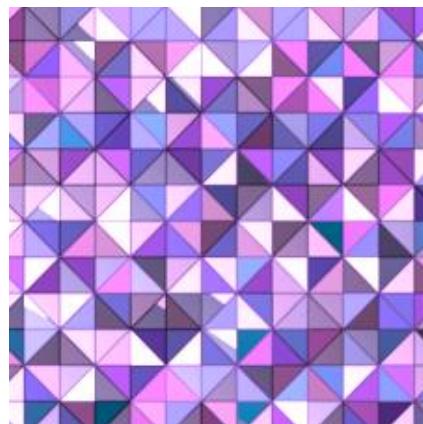


Subdivision Iterations의 수를 늘리면 더 많은 삼각형이 보입니다.

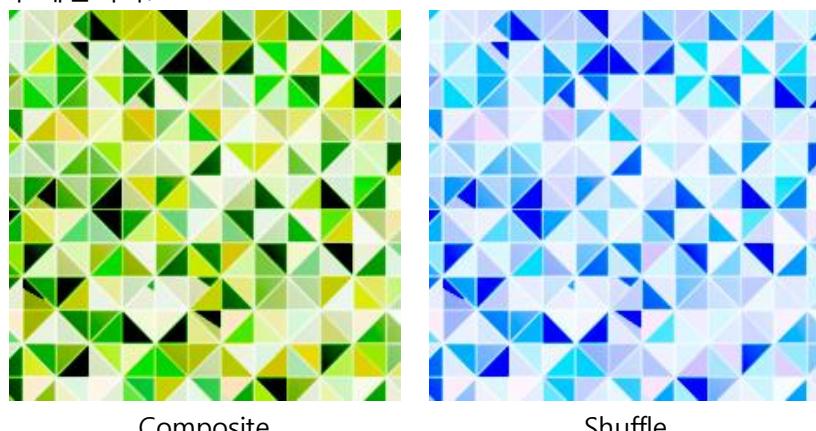
Color Correct

Color Correct 쉐이더를 사용하여 삼각형의 색상을 변경할 수 있습니다.

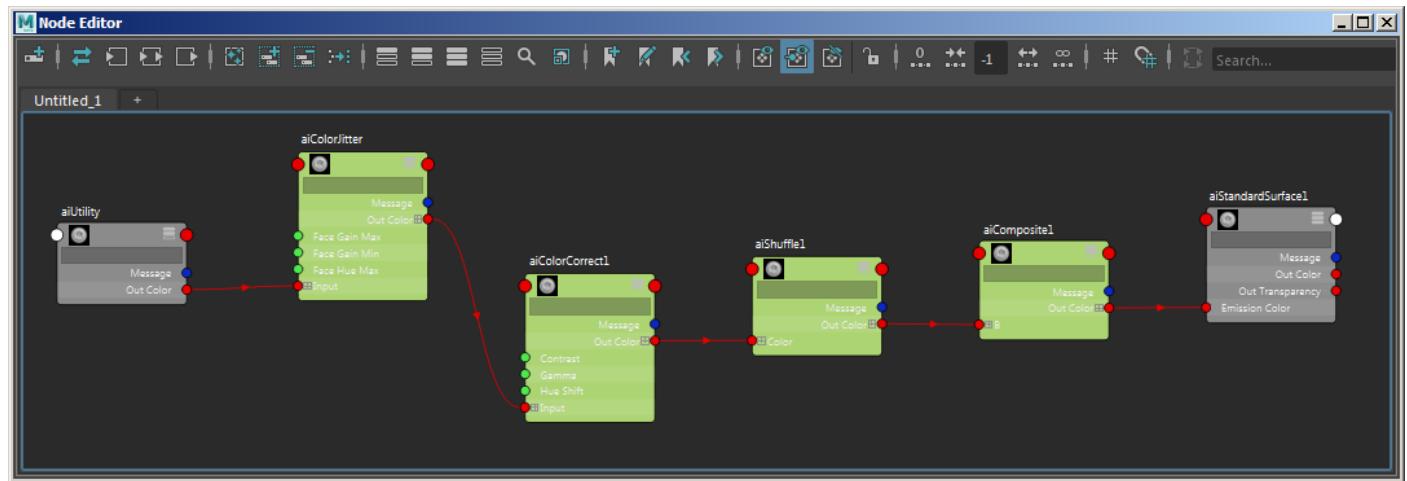
- Color Jitter와 Standard Surface 쉐이더 사이에서 **Color Correct** 쉐이더를 연결합니다. *Hue Shift*를 조절하고 *Multiplycolor*를 변경합니다. 마음에 들 때까지 Color Correct 쉐이더의 속성을 조절할 수 있습니다. 이 경우에는 *Gamma* 및 *Contrast*가 조절되었습니다.



Arnold의 다른 색상 쉐이더 중 일부를 사용하여 색상을 변경할 수도 있습니다. 다음은 **Composite** 및 **Shuffle** 쉐이더를 사용하는 몇 가지 예입니다.



마지막으로 Color Jitter 쉐이더의 *Gain* 및 *Hue (Face)* 쉐이더를 키프레임화합니다. Utility 쉐이더의 *Color Mode*가 *UV Coords*로 설정되었으므로 평면의 표면에 걸쳐 이 효과가 애니메йт화된 것을 볼 수 있습니다.



최종 쉐이더 네트워크

Emissive 쉐이더를 사용한 굴절 커스틱 효과

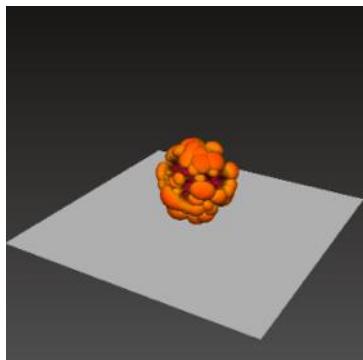


설정이 올바르다면 MtoA를 사용하여 커스틱 효과를 낼 수 있습니다. 이 짧은 자습서에서는 할당된 **Standard Surface** 쉐이더에서 **Caustics**가 활성화된 상태로 '액체' 메시가 포함되어 있는 장면을 설정하는 방법을 설명합니다. 이 장면은 굴절 커스틱 효과를 생성하는 높은 **Emission Scale** 값을 가진 평면을 사용합니다. 작지만 밝은 광원(예: 코냑 유리잔을 통과하는 스포트라이트)의 '단단한' 커스틱은 현재 불가능함을 유의하십시오.

Arnold는 간단한 단방향 경로 추적을 사용합니다. 광선은 조명이 아닌 카메라에서 시작됩니다. Arnold는 양방향 경로 추적을 사용하지 않습니다(또한 조명으로부터 광선을 발사하는 광자 매핑과 같은 다른 양방향 기술도 사용하지 않습니다). 포인트 라이트(**point light**) 및 스포트라이트(**spot lights**)와 같은 표준 조명을 사용할 때는 Arnold의 GI/반사/전달 광선이 조명에 부딪히는 것은 불가능합니다. 따라서 커스틱 효과가 없습니다. 그러나 'radius' 파라미터를 높여서 포인트 라이트와 스포트 라이트를 유한 크기의 조명으로 전환할 수 있습니다. 그러면 구체 조명이 만들어져서 부드러운 그림자와 부드러운 하이라이트가 제공됩니다. 그 다음에는 GI/specular reflection/transmission 광선이 이러한 조명을 '볼 수' 있을 것입니다. Arnold에서 조명은 광선-장면 교차 중에 광선이 횡단하는 지오메트리 데이터베이스에 저장된 지오메트리 오브젝트를 갖고 있지 않습니다. 따라서, 여전히 영역 조명이 GI/specular reflection/transmission 광선에 보이지 않습니다.

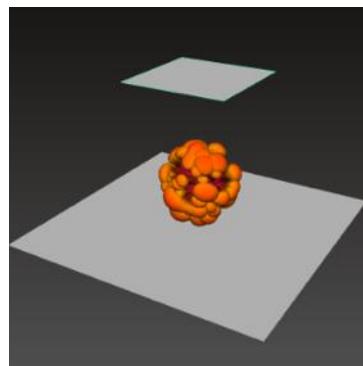
Arnold의 표준 조명을 사용하는 대신 폴리곤 메시를 만들고 여기에 평면 방사형 쉐이더를 적용하고 GI 엔진이 해당 조명을 '찾을 수' 있도록 할 수 있습니다. 그러면 커스틱 효과를 얻게 됩니다. 하지만 작은 방사체는 부딪치기가 힘들기 때문에 이는 매우 비효율적입니다. 이 노이즈가 받아들일 수 있는 수준이 되려면 많은 광선 또는 매우 큰 방사체가 필요할 것입니다. 사용자 설명서에서 큰 방사체로부터 나오는 '부드러운' 커스틱 효과를 낼 수 있다고 설명한 이유도 바로 이것 때문입니다.

- 먼저 Maya 장면 파일인 **Refractive-Caustics_Start.ma**를 열겠습니다. 이 장면에는 기본 물체가 포함되어 있어서 쉽게 시작할 수 있습니다. 폴리곤 바닥면을 만들어서 메시 아래에 놓습니다.

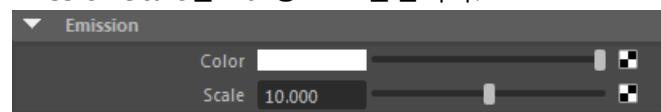


Emissive Plane

- 평면을 하나 더 만들고 아래와 같이 메시 위에 놓습니다.

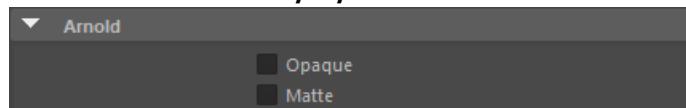


- 폴리곤 메시 위의 폴리곤 평면에 **Standard Surface** 쉐이더를 할당합니다. 그러면 광원이 됩니다. **Base Weight** 값을 낮추고 **Emission Scale**을 10 정도로 올립니다.



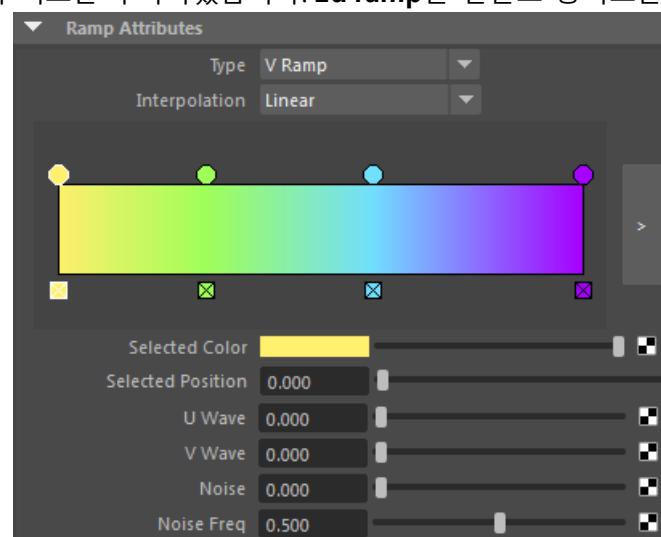
Mesh Shading

- 굴절 커스틱 효과를 만들고자 하는 폴리곤 메시에 Standard Surface 쉐이더를 할당합니다. 이것을 유리 쉐이더로 만들어 보겠습니다. 그 전에 **Opaque** 플래그를 비활성화해야 합니다. 해당 폴리곤 메시를 선택하고 **Attribute Editor**의 **Arnold** 속성에서 **Opaque**를 비활성화합니다.

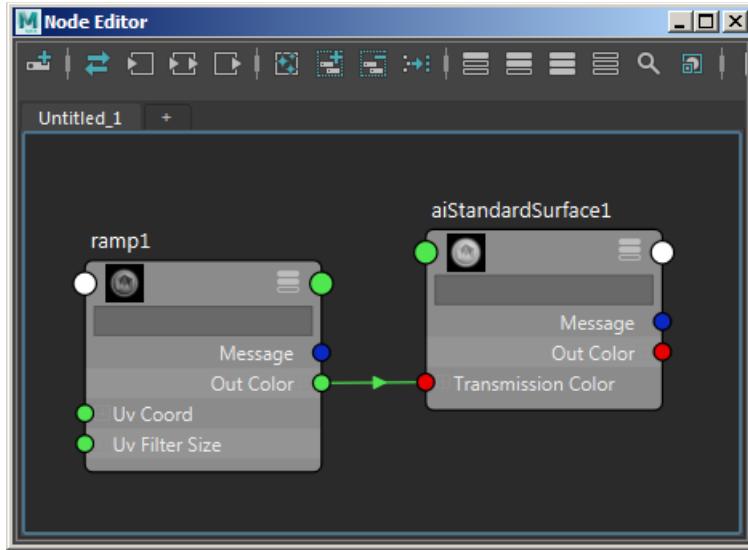


해당 유리 폴리곤 메시에 대해 Arnold 속성에서 Opaque를 비활성화합니다.

- Base Weight**를 0으로 낮추고 **Specular Roughness**도 0으로 낮춘 다음, **Transmission Weight**를 1로 높이고 **IOR**를 유리 값(1.5)으로 변경합니다.
- 그 다음에는 쉐이더에 색조를 추가하겠습니다. **2d ramp**를 만들고 흥미로운 색을 추가해 봅니다.

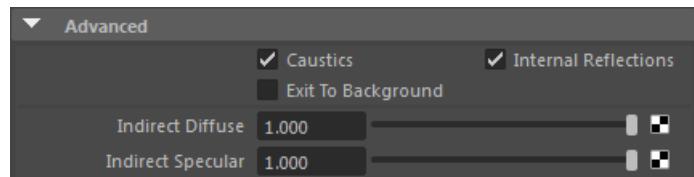


- 이 ramp를 Standard Surface 쉐이더의 **Transmission Color**에 연결합니다.

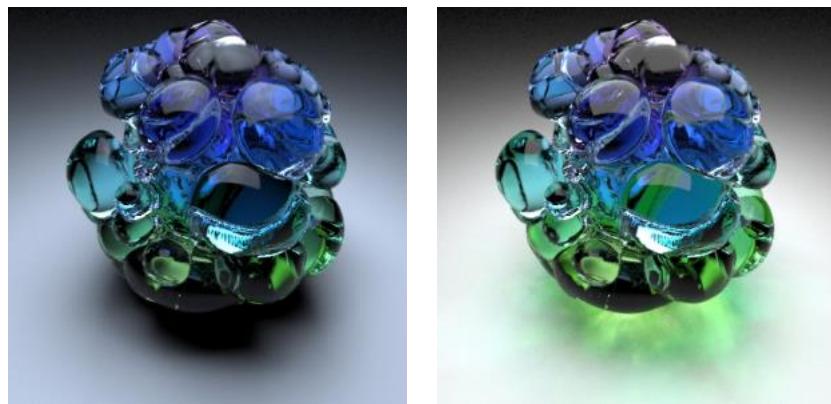


Refractive Caustics

유리를 통해 굴절 커스틱 효과를 보려면 유리 메시에 할당된 Standard Surface 쉐이더에서 이를 활성화해야 합니다. **Caustics**는 Standard Surface 쉐이더 내의 **Advanced** 아래에 있습니다. **Caustics**를 활성화합니다.

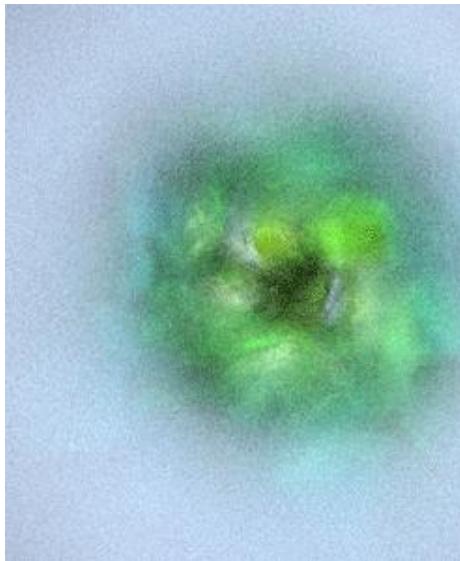


아래 이미지에서는 커스틱을 활성화할 때의 차이점을 볼 수 있습니다.

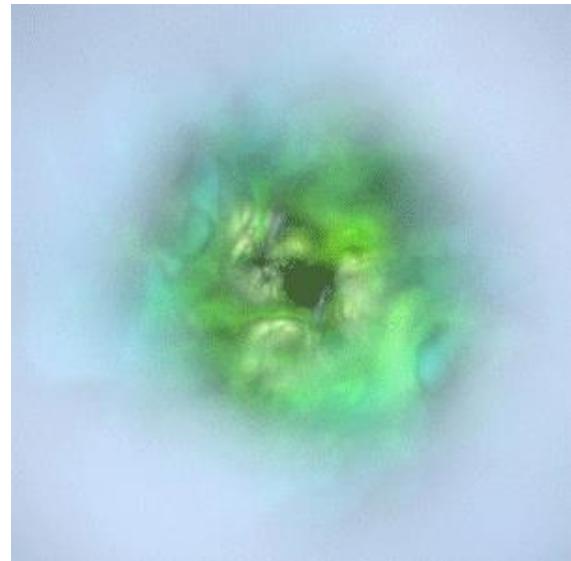


Noise

커스틱에서 노이즈를 줄이려면 **GI Diffuse Samples**의 수를 높여야 합니다. 아래 애니메이션들은 3개 및 6개의 GI Diffuse Samples로 굴절 커스틱을 렌더링할 때의 차이점을 보여줍니다. 이 값을 늘리면 렌더링 시간이 급격히 증가하므로 이 속성을 사용할 때는 주의해야 합니다.



Diffuse Samples 3 (3분 15초).



Diffuse Samples 6 (16분 55초).

이제 다 되었습니다. 렌더링을 시작하기만 하면 됩니다. 다른 물체들을 렌더링해보세요. 이 방법을 사용할 때는 다음을 고려해야 합니다.

1. Emissive 쉐이더가 할당된 지오메트리에 적절한 양의 **Emission**을 미세 조정하십시오. 그렇지 않으면 많은 화이트 노이즈가 발생할 수 있습니다.
2. **GI Diffuse Samples**를 높여서 노이즈를 추가적으로 줄이십시오.

Remap an Image Using UV Coords



이 튜토리얼에서는 Image 쉐이더의 'UV Coords' 속성을 사용하여 이미지를 재 매핑하여 추상적인 패턴화된 왜곡 효과를 만드는 방법을 보여줍니다.

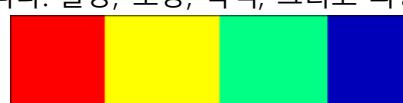
작업파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Remap+an+Image+Using+UV+Coords>)

- 폴리곤 면에 **Standard Surface** 쉐이더를 적용합니다.
- Standard Surface의 *Emission*을 1로 설정합니다. *Base*와 *Specular Weights*를 0으로 설정합니다.
- **Image** 쉐이더를 *Emission Color*에 연결하고 *Image Name*에 file texture를 추가합니다. 이 경우, 모나리자 이미지를 사용합니다.



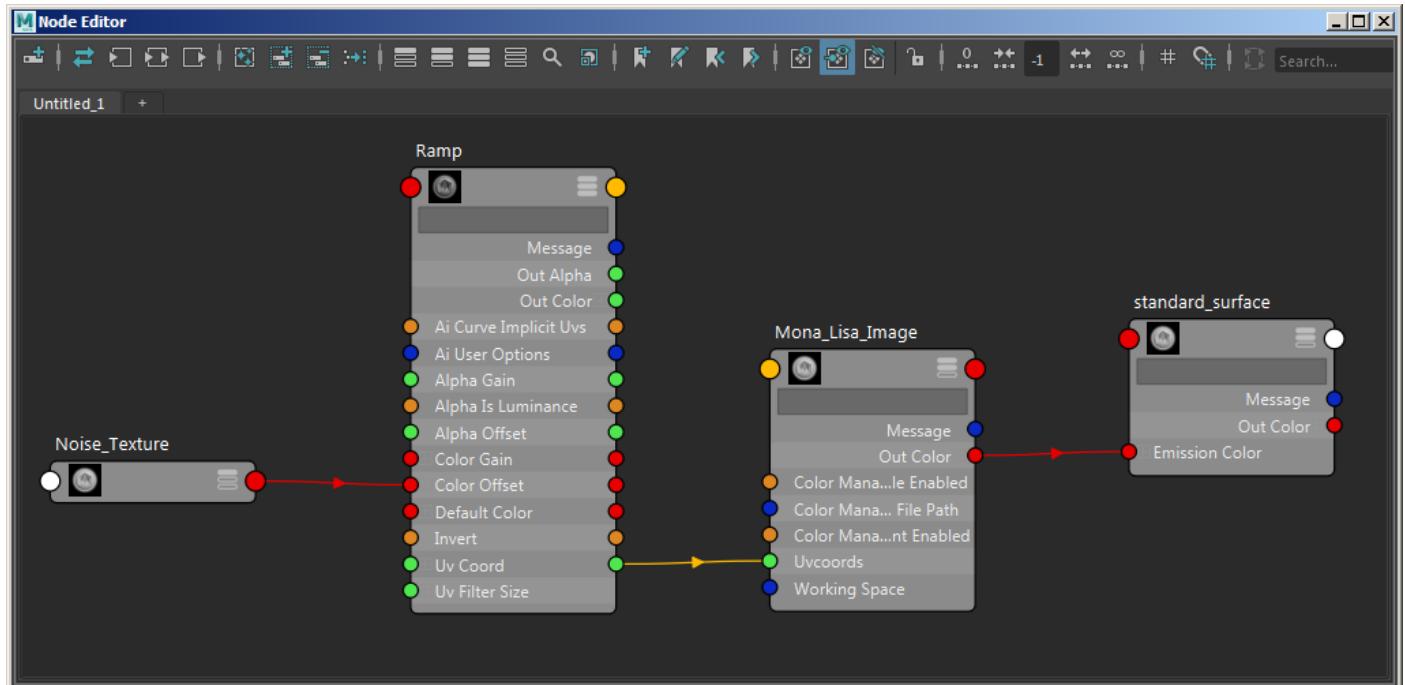
- Ramp (RGB) texture를 만들고 Image shader의 *UV Coords*에 연결합니다. Ramp의 *U Coords*를 Image shader의 *uvcoordsY*에 연결합니다. Ramp의 *V Coords*를 Image shader의 *uvcoordsX*에 연결합니다.
- Ramp type을 Four Corner Ramp로 변경합니다.
- Ramp에 4개의 다른 색을 만듭니다: 빨강, 노랑, 녹색, 그리고 파랑



Ramp의 HSV 값을 조절하여 이미지 재배치 방법을 변경할 수 있습니다.

파일 또는 Noise 텍스처를 Ramp의 Color Offset에 연결합니다. 이렇게 하면 왜곡 효과가 생깁니다.

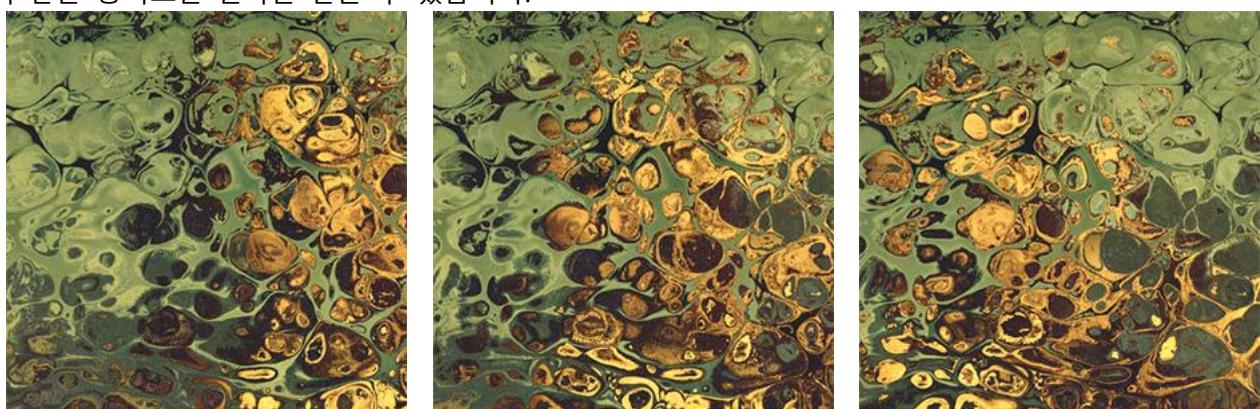
또한 파일 텍스처와 램프 사이에 Range 또는 Remap 쉐이더를 추가하여 효과를 추가로 제어할 수 있습니다.



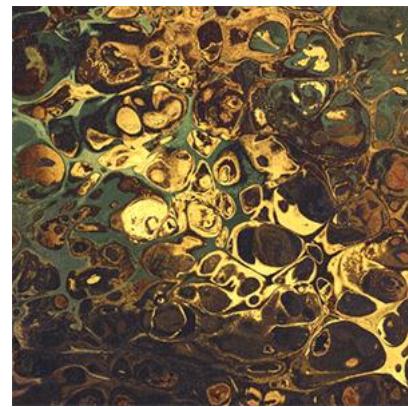
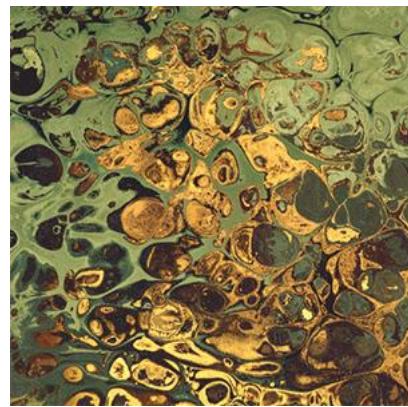
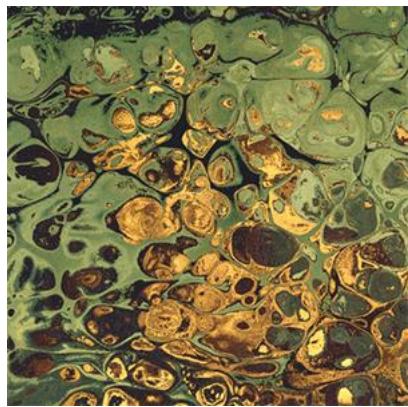
최종 쉐이딩 네트워크

Image: UV Coordinates

이미지 쉐이더의 UV 좌표에는 텍스처 맵의 위치를 더 변경하기 위한 컨트롤이 있습니다. 애니메이션을 적용하면 다음과 같은 흥미로운 결과를 얻을 수 있습니다.



Offset U (수평으로 애니메이션)



Offset V (수직으로 애니메이션)

이미지 쉐이더를 왜곡하는데 사용되는 텍스처의 노출(Exposure: **Color Correct** 사용)을 애니메이션하면 최종 결과가 생성됩니다.



Exposure (-10에서 0)

Rim 쉐이더

Facing Ratio 쉐이더는 쉐이딩 효과를 얻기 위해 사용할 수도 있습니다. Facing Ratio는 하단 링크의 사이트에서 볼 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Facing+Ratio>)



Rim 쉐이더는 물체의 윤곽선/가장자리와 관련된 조명 효과를 얻는 데 유용합니다. 이 자습서는 가짜 rim 조명 음영 효과를 시뮬레이션하는 간단한 쉐이더에 대해 설명합니다. 이것은 프레넬이나 만화 같은 스킨 효과를 만들 때 유용할 수 있습니다.

이 쉐이더는 하단 링크의 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Rim+Shader>)

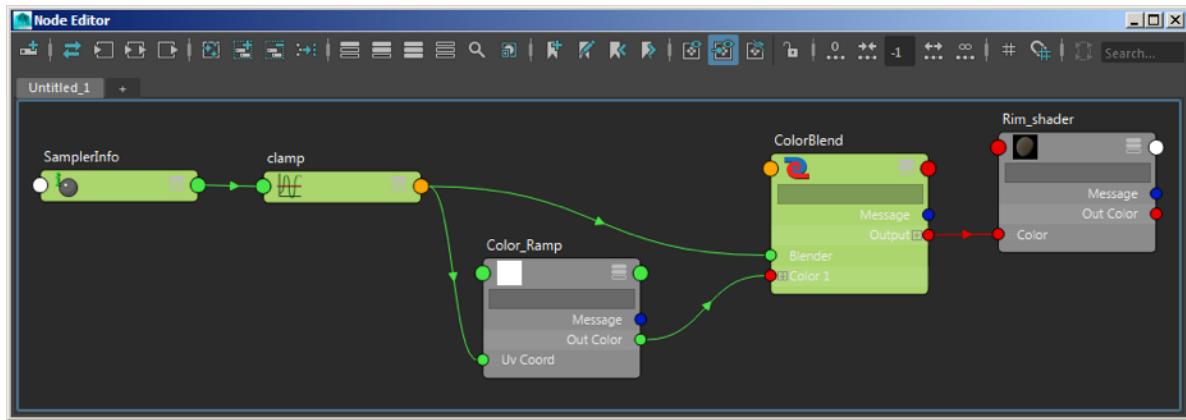


Rim 쉐이더 없음(Standard Surface)



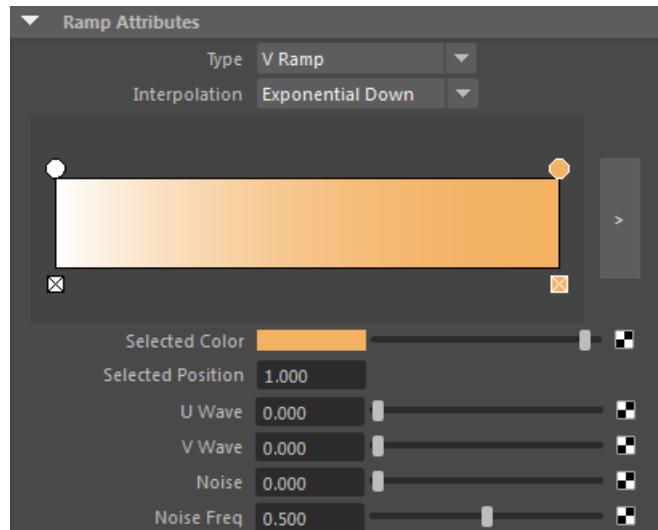
흰색으로 설정된 Rim 쉐이더 색상

- SamplerInfo 노드는 rim 조명 효과의 색상을 결정하는 ramp에 연결됩니다. 이것은 Standard Surface 쉐이더의 *Base Color* 속성에 연결됩니다.



Rim 쉐이더 네트워크

- Ramp의 색상을 변경하면 rim 조명 효과가 영향을 받습니다. ramp 텍스처를 사용하여 rim 효과의 폭과 강도를 조절할 수 있습니다.



노란색으로 설정된 ramp 색상

검은색으로 설정된 ramp 색상

지구 쉐이딩 하기



이 자습서에서는 쉐이더만 사용하여 사실적인 지구와 구름을 만드는 과정을 살펴 보겠습니다. Volume 및 Volume Collector 쉐이더로 지평선에 대기를 추가할 것입니다. 또한 Skydome 조명에 연결된 Physical Sky를 사용하여 빛을 비추겠습니다. 이 장면은 예를 들어, 불행한 우주 비행사에 관한 영화에 적절한 배경이 될 수 있습니다.

이 장면과 함께 사용할 고해상도 텍스처를 다운로드하여 사용해야 합니다. 렌더링 전에 반드시 텍스처를 .tx 형식으로 변환하십시오. 이미지는 하단 링크의 사이트에서 다운로드할 수 있습니다.

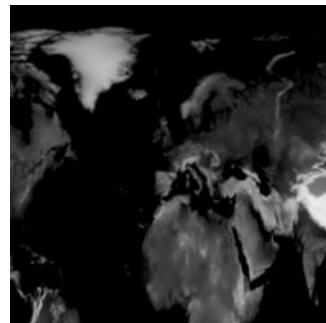
(<https://support.solidangle.com/display/A5AFMUG/Shading+a+Globe>)



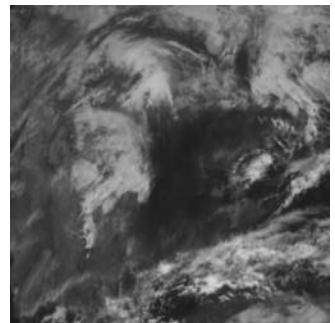
Earth Mask (specular)



Earth Color (diffuse)



Earth Elevation (displace)



Clouds (emission, opacity, displace)

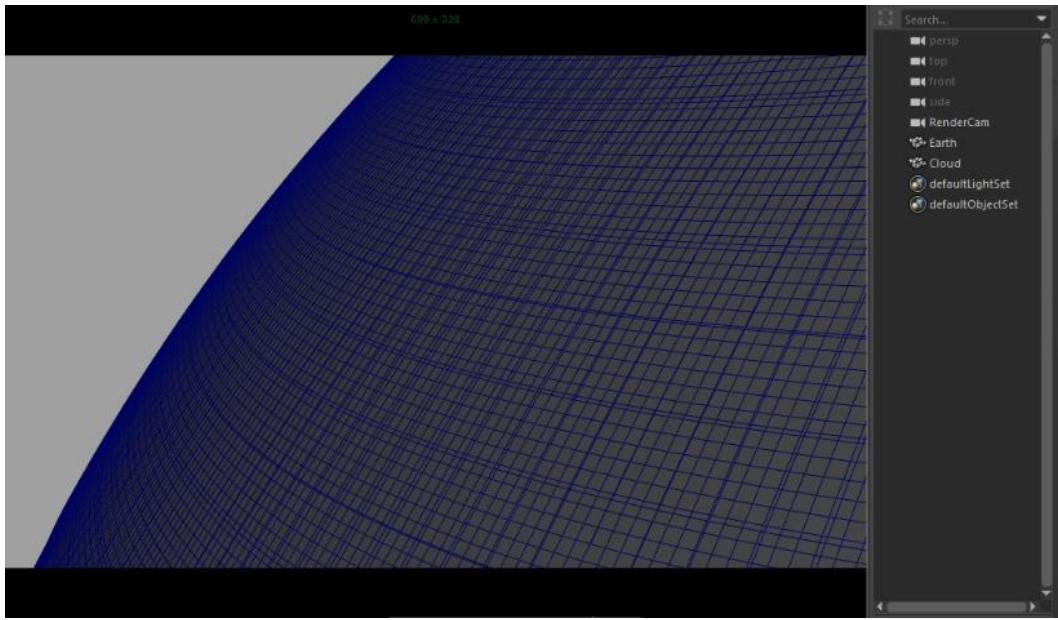
최종 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Shading+a+Globe>)

이 자습서는 다음 단계들로 구성되어 있습니다.

- [Earth Shader](#)
- [Cloud Shader](#)
- [대기에 Volume 사용하기](#)

- 먼저 시작 장면을 열겠습니다. 이 장면에는 Earth와 Cloud라는 이름의 오브젝트가 있습니다. 이들 두 오브젝트에 각 쉐이더를 적용하겠습니다.



Earth와 Cloud 오브젝트가 있는 시작 장면

- Skydome 조명을 만들어서 장면에 빛을 비춥니다. Physical Sky를 Skydome 조명의 **Color**에 연결합니다. Physical sky에서 태양의 방향을 변경하여 바다에 반사되도록 만들어야 합니다. *Skydome* 조명의 *Camera visibility*를 0으로 줄여서 카메라에서 *Physical Sky*가 보이지 않게 합니다.

Earth Shader

잠시 Cloud 오브젝트를 숨깁니다. 먼저 Earth 쉐이더에 집중하겠습니다.

Diffuse

- Standard Surface 쉐이더를 Earth 오브젝트에 할당하고 이름을 'Earth'로 변경합니다. 파일 텍스처를 만들고 **Earth Color** 텍스처를 엽니다. 파일 텍스처를 Standard Surface 쉐이더의 Base Color에 연결합니다. 그러면 Earth에 기본적인 확산 색이 적용됩니다.



Earth Standard Surface 쉐이더의 Base Color에 연결된 'Earth Color' 맵

Specular

이제 바다가 반사되고 육지에는 반사가 없는 곳을 찾기 위해 반사 맵을 추가하고자 합니다.

- Earth의 Standard Surface 쉐이더를 선택합니다. 파일 텍스처를 만들고 **Earth Mask** 파일을 엽니다. 이것은 광택(흰색)과 무광택(검정)이 무엇인지 결정하는 마스크입니다. 하지만 이미지를 보면 우리가 원하는 것과는 정반대인 것을 알 수 있습니다. 따라서 맵을 반전시켜야 합니다. 이것을 파일 텍스처의 Effects 부분에서 할 수 있습니다. 해당 파일 마스크 텍스처에 대해 *Invert*를 활성화합니다.
- **Earth Mask**를 Standard Surface 쉐이더의 **Specular Weight**에 연결합니다.



Earth 쉐이더의 specularity에 연결된 'Earth Mask' 텍스처로 결정되는 육지/바다의 빛남

Displacement

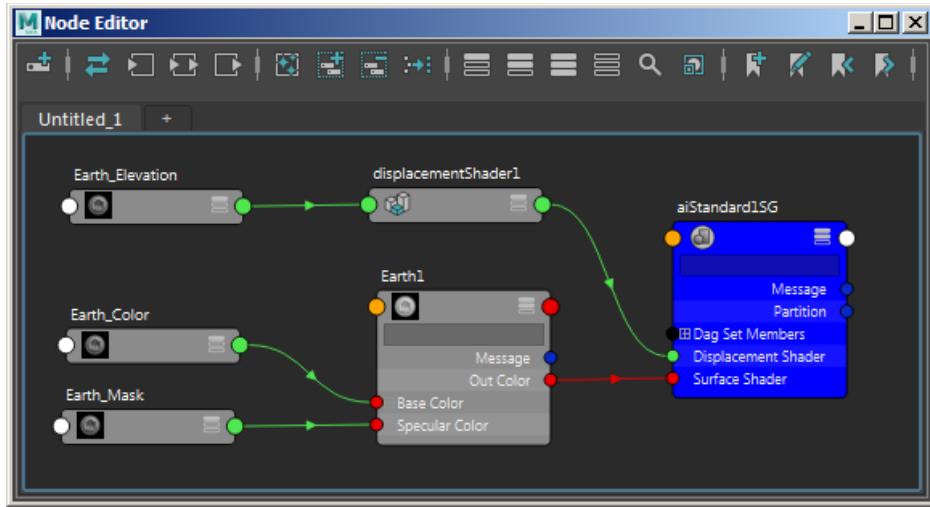
지구의 표면이 약간 깃깃해 보입니다. 변위 맵 - '**Earth Elevation**'을 사용하여 표면에 약간의 고도를 추가할 수 있습니다.

- **File Texture**를 만들고 Earth의 Standard 쉐이더의 Shading Group의 Displacement 속성에 연결합니다. **Earth Elevation** 텍스처를 엽니다. 변위의 세밀한 디테일을 보려면 Earth 지오메트리에 몇 개의 하위 구획을 추가해야 합니다.
- Earth 지오메트리를 선택합니다. Arnold **Subdivision** 탭 아래에 있는 **Type**을 **Catclark**로 변경하고 **Iterations** 수를 4로 올립니다.



Earth의 지형이 'Earth Elevation' 텍스처를 사용하여 바뀝니다(Rollover image).

Earth 표면의 최종 쉐이더 네트워크는 다음과 같습니다.

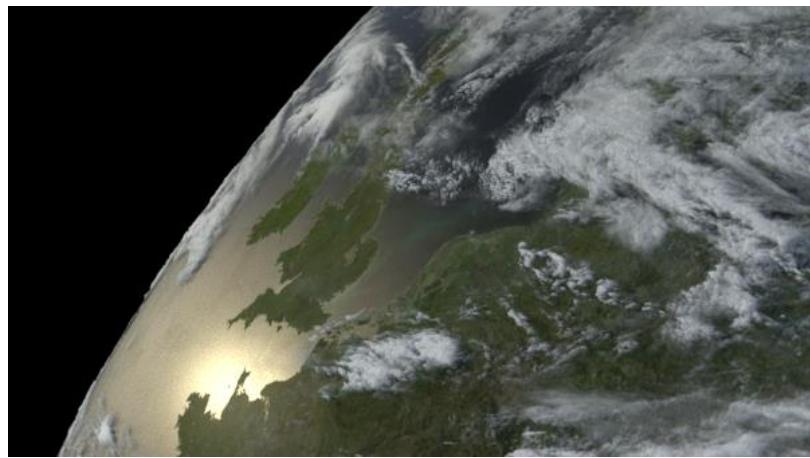


최종 Earth 쉐이더 네트워크

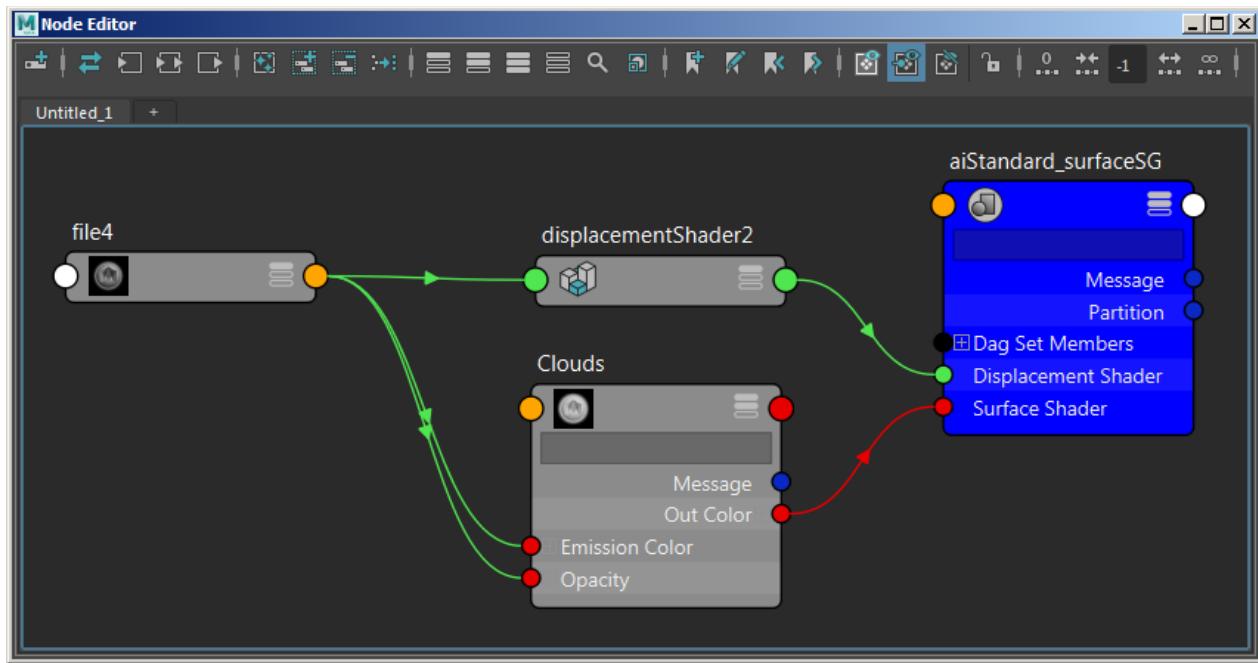
Cloud Shader

Clouds 쉐이더는 설정이 약간 더 쉽습니다. 구름 이미지를 사용하여 *Standard Surface* 쉐이더의 *Emission*, *Opacity*, *Displacement*에 연결하겠습니다.

- 구름에 대해 *Standard Surface* 쉐이더를 만듭니다. **Clouds**로 이름을 변경하고 **Cloud object**에 할당합니다.
- 이미지 텍스처를 만들고 Cloud 이미지를 엽니다. cloud 이미지 텍스처를 **Opacity**, **Emission**, **Displacement** 속성에 연결합니다.
- Emission Scale**을 1 정도로 높입니다. 그러면 구름의 밝기가 결정됩니다.
- 이제 구름의 변위를 위해 앞서 Earth 오브젝트에 실행한 단계를 Cloud 오브젝트에 대해서도 반복하여 구름이 지구의 표면에 떠있는 것처럼 보이게 해야 합니다. Cloud 지오메트리를 선택하고 **Subdivision** 탭 아래에 있는 **Type**을 **Catclark**으로 변경하고 **Iterations** 수를 4로 높입니다. **Displacement** 아래에서 **Height**를 1.5 정도로 높입니다.



Clouds 텍스처 맵에 의해 구동된 구름의 Displacement, Opacity, Emission

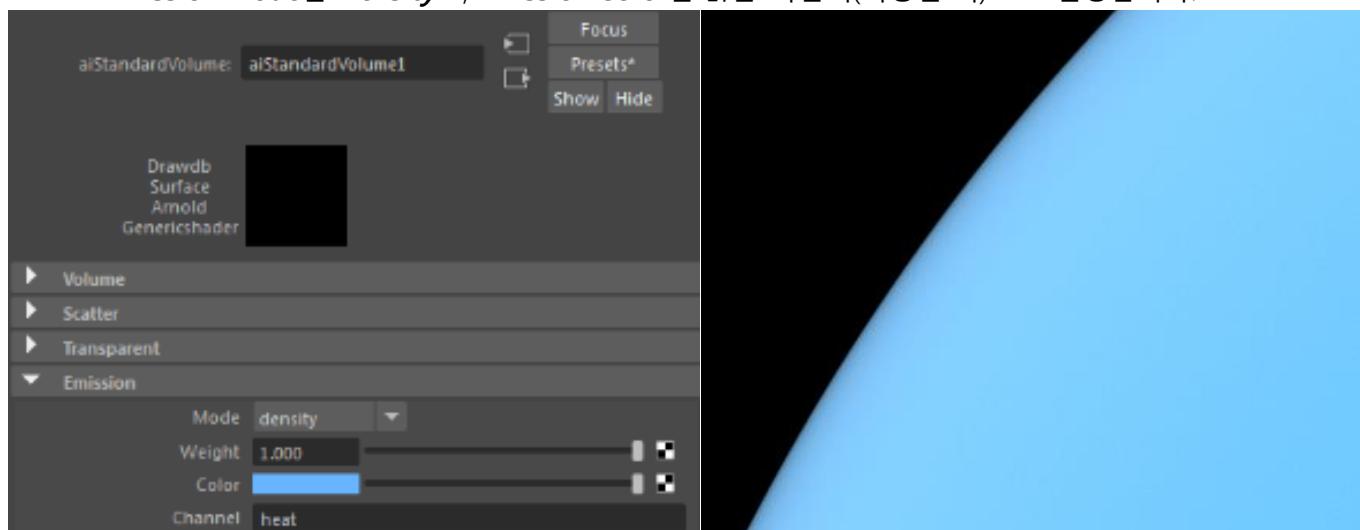


최종 Clouds 쉐이더 네트워크

대기애 Volume 사용하기

마지막으로 Volume을 사용하여 지평선에서 볼 수 있는 대기를 만들어 보겠습니다.

- 구체를 만듭니다. 크기를 키워서 Earth 오브젝트를 덮게 합니다.
- Arnold 속성에서 **Step Size**를 0.1로 높입니다.
- *Standard Volume* 쉐이더를 생성하고 구체에 할당합니다. 구체가 이제 volume으로 렌더링됩니다.
- **Emission Mode**를 *Density*로, **Emission Color**를 밝은 하늘색(지평선 색)으로 변경합니다.



Standard Volume 설정

지금까지 지구를 사실적으로 음영 처리하는 방법을 살펴봤습니다. 마지막으로, 구름과 지구의 회전을 키프레임화하여 가짜 타임 랙스 효과를 만들거나 우주 비행사가 죽음에 이르는 장면을 애니메이션으로 만들어보세요.

장미 쉐이딩 하기



이 자습서에서는 Standard Surface 쉐이더를 사용하여 장미를 사실적으로 음영 처리하는 방법을 설명합니다. 반투명과 하위 표면 산란을 포함하여 장미 쉐이더를 구성하는 미세한(그러나 중요한) 음영 처리 속성과 단면 지오메트리를 렌더링할 때 고려해야 할 몇 가지 사항을 살펴볼 것입니다.

Shading

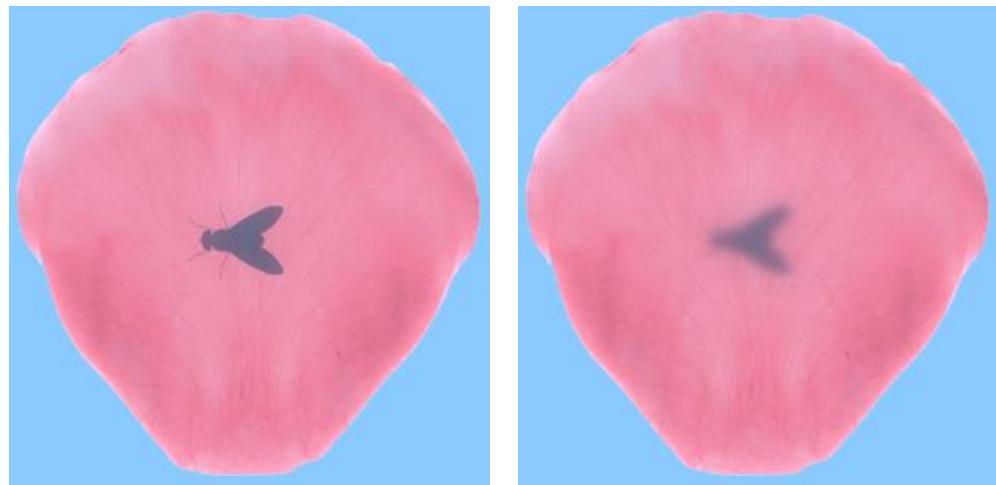
장미의 Standard Surface 쉐이더에는 *Base Color*, *Specular*, *SSS* 맵이 연결되어 있습니다. 약간 수정된 중복 색상 맵이 하위 표면 색상 속성에도 연결되어 있습니다. 백라이팅 효과는 *Thin Walled*를 활성화하여 얻을 수 있습니다. 아래의 움직이는 gif는 *Diffuse*(*Base Color*), *Specular*(높은 *Specular Roughness*) 및 *Subsurface* (*SSS*) 산란이 장미의 외관에 어떤 영향을 주는지를 보여줍니다. 이 장면은 카메라를 향하는 방향성 조명으로 빛이 처리되므로, 반투명 백라이팅 효과를 생성하고 Sky HDR 맵이 연결된 Skydome 조명이 색상 속성에 연결됩니다.



Translucency

*Thin Walled*와 결합된 *SSS*는 뒤에서 빛이 비춰지는 반투명 물체의 효과를 재현할 수 있습니다(이 효과는 단면 물체에 더 적합합니다). 아래 이미지는 두께 증가에 따라 꽃잎을 렌더링할 때의 차이를 보여줍니다. 파리 모델은 꽃잎 뒤에 있고 방향성 빛은 최대 효과를 내기 위해 카메라를 향하고 있습니다.

일반적으로 *Thin Walled*은 얇은 물체(단면 형상)에만 사용할 것을 권장합니다. 두께가 있는 물체는 잘못 렌더링 될 수 있습니다. 그러나 이 상황에서는 두께가 있어도 잘 작동합니다(*diffuse ray depth* 수준이 1 이상인지 확인).

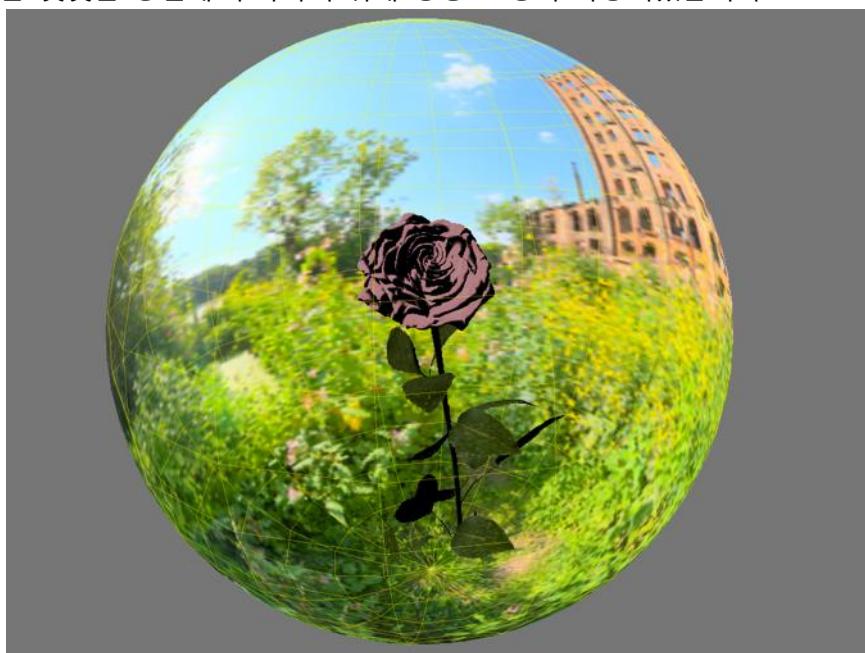


Mesh width scale: 1단위

Mesh width scale: 10단위

Lighting

HDRI는 Skydome 조명의 Color 속성에 연결됩니다. 이 예제에서 사용된 HDRI는 www.hdr-labs.com에서 다운로드했습니다. 또한 더 많은 햇빛을 장면에 추가하기 위해 방향 조명이 사용되었습니다.



방향 조명으로 빛이 비춰지는 장면과 HDR 맵이 있는 Sky 쉐이더

Shadow Terminator

강력한 방향 조명으로 뒤에서 빛을 비추면 그림자 경계선에 약간의 인공물이 생깁니다. `shadow_terminator_fix`로 활성화된 현재 알고리즘은 구체와 같이 볼록한 표면에서 잘 작동하지만 오목한 영역에서는 자체 그림자를 수정할 수 없습니다. 다음은 이를 해결할 수 있는 방법입니다.

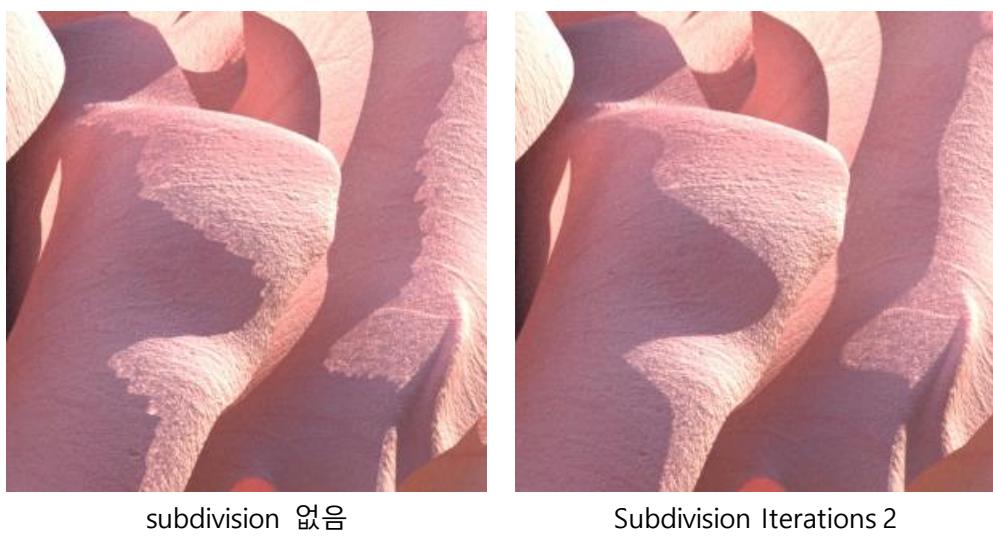
Shadow Angle

조명 반경을 늘리면(이 경우에는 태양에 대해 방향 조명이 사용되었기 때문에 **Angle**) 그림자 경계선의 현상이 줄어듭니다(그림자를 부드럽게 함으로써). 그러나 Angle을 높이더라도 그림자 경계선은 여전히 잘 보입니다.



Subdivision Iterations

또 다른 해결책은 꽃잎 지오메트리의 **Subdivision Iteration** 수치를 높이는 것입니다. 그러면 인공물이 감소합니다.



Extruded thickness

더 나은 '물리적으로 정확한' 방법은 꽃잎을 돌출시켜서 형상에 약간의 두께를 추가하는 것입니다. 왼쪽 이미지에 있는 꽃잎은 단면입니다. 꽃잎을 렌더링하면 얇은 종이로 보입니다(꽃잎의 뒤쪽으로 흘어져 있는 부분 주위에서 더 두드러짐). 오른쪽 이미지는 꽃잎이 돌출되었을 때의 결과를 보여줍니다. 렌더링하면 꽃잎이 약간 두꺼워 보입니다.



Single sided (메시에 두께가 없음).



메시에 돌출된 두께가 있습니다.

렌더링

Indirect Diffuse Ray Depth

Indirect Diffuse ray depth 값을 증가시키면 장면에 반사광이 추가됩니다. 이 값을 3으로 높이면 장미가 훨씬 더 밝아집니다. SSS는 Indirect Diffuse ray depth 값에 민감하므로 Indirect Diffuse ray depth 값을 높일 때는 주의해야 합니다. 서로 겹치거나 둘러싸인 표면(예: 장미)이 많은 경우 Indirect Diffuse ray를 2 이상으로 높이면 효과가 눈에 보입니다.

하위 표면 산란 값을 낮추면 이 증가된 밝기를 보완하는 데 도움이 됩니다.



Diffuse ray depth 1

Diffuse ray depth 3

지금까지 장미를 음영 처리하는 방법을 살펴봤습니다. 여기에 설명된 기술은 피부, 종이 및 왁스와 같은 다른 반투명 물질에도 적용할 수 있습니다.

Specular BRDF(Bidirectional Reflectance Distribution Function: 양방향 반사율 분포 함수)

이 자습서에서는 *Standard Surface* 쉐이더에서 발견되는 다양한 Specular 설정에 대해 설명합니다.

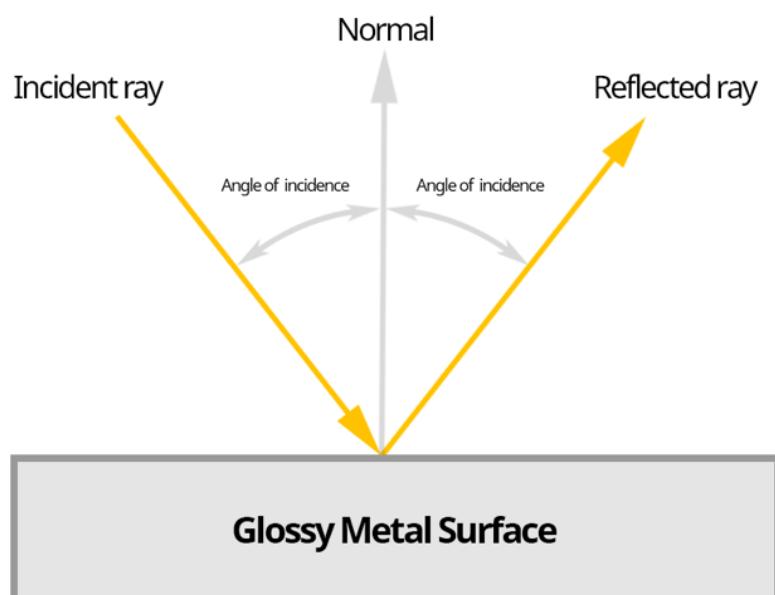
BRDF

양방향 반사율 분포 함수(BRDF)는 표면 법선과 관련하여 다른 방향으로 반사되는 한 방향으로부터 입사하는 양을 지정하여 표면의 반사 특성을 설명합니다.

물리적으로 타당한 BRDF의 주요 특징은 입사 방향과 반사 방향 사이의 대칭(Helmholtz 호환성)이며, 주어진 입사 방사선에 대한 전체 반사파는 입사광의 에너지와 같거나 그 이하입니다(에너지 보존).

Specular BRDF 모델은 뷰에 따라 다르며, 지표각에서 이방성 및 프레넬 효과를 고려할 수 있습니다. Arnold는 Cook-Torrance BRDF를 사용합니다.

표면에 부딪치는 광선을 입사광이라고 하며, 이 광선이 부딪치는 각도를 입사각이라고 합니다(아래 그림 참조). 뒤쪽으로 반사되는 빛 에너지는 표면에 들어오는 빛보다 적거나 같아야 합니다(빛을 "추가"할 수는 없음). 표면에서 광선은 반사되거나 굴절되어 결국 특정 매체에 의해 흡수될 수 있습니다.



입사각(Incidence Ray), 입사광(Incident Ray), 반사광(Reflected Ray)

Normal에서 Reflectance

프레넬 효과(Fresnel Effect)는 프랑스의 물리학자인 Augustin-Jean Fresnel에서 이름을 따왔습니다. 이 효과는 표면의 반사 강도가 시야각에 달려 있음을 나타냅니다. 반사의 양은 입사각에서 보았을 때 표면에서 증가합니다.

Standard Surface 쉐이더에서 IOR가 1보다 크면 물체의 반사율은 뷰에 따라 다르며, 프레넬 방정식을 따릅니다. *Standard Surface* 쉐이더는 Schlick의 프레넬 방정식 근사치를 사용하여 재료의 IOR를 사용하여 제어할 수 있습니다. 낮은 값에서 이 소재는 플라스틱과 같은 완벽한 유전체 역할을 합니다. 이 값이 높을수록 재료는 거의 완벽한 도체 또는 금속 표면이 됩니다.



법선에서 반사율: 0



법선에서 반사율: 0.1

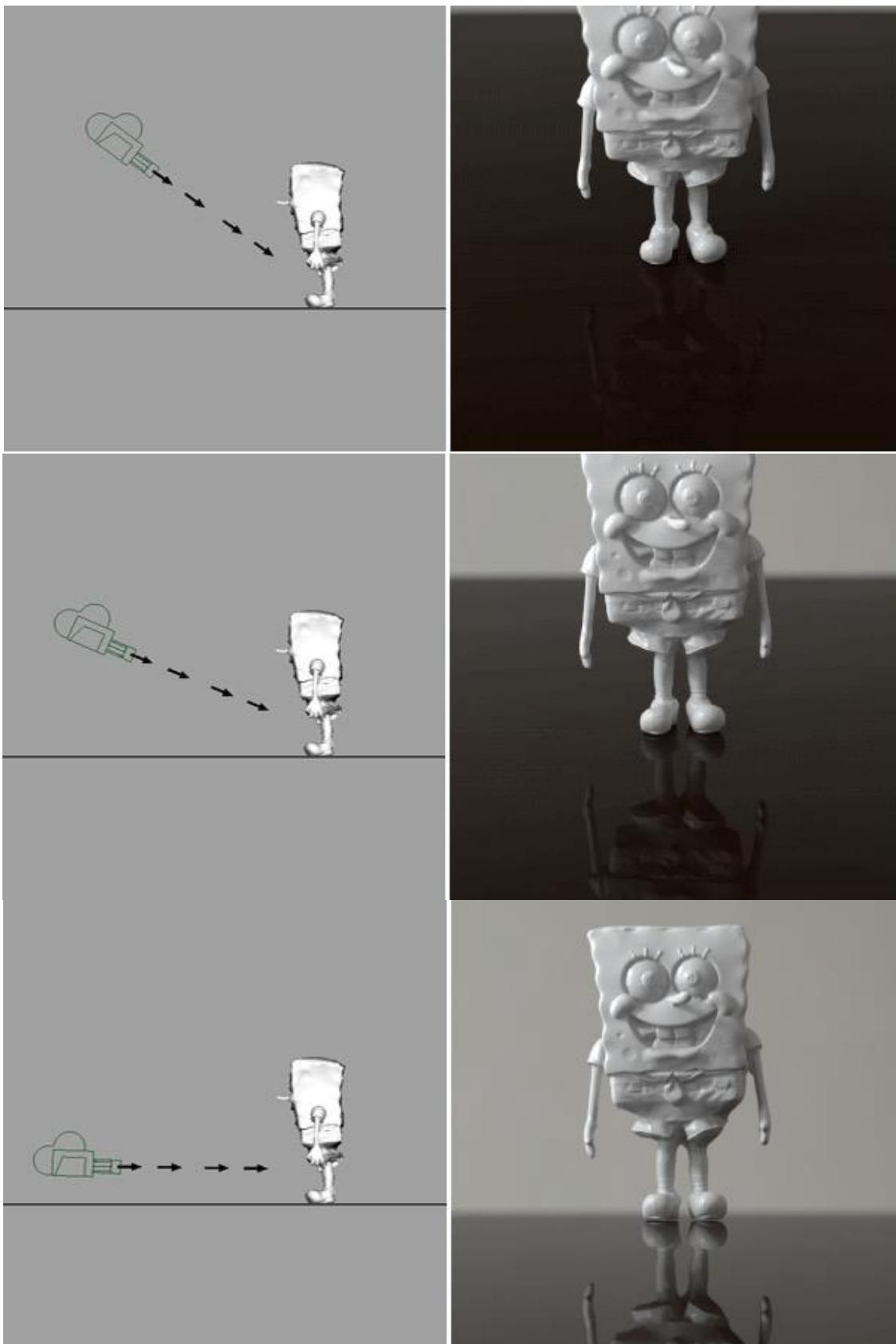


법선에서 반사율: 0.25



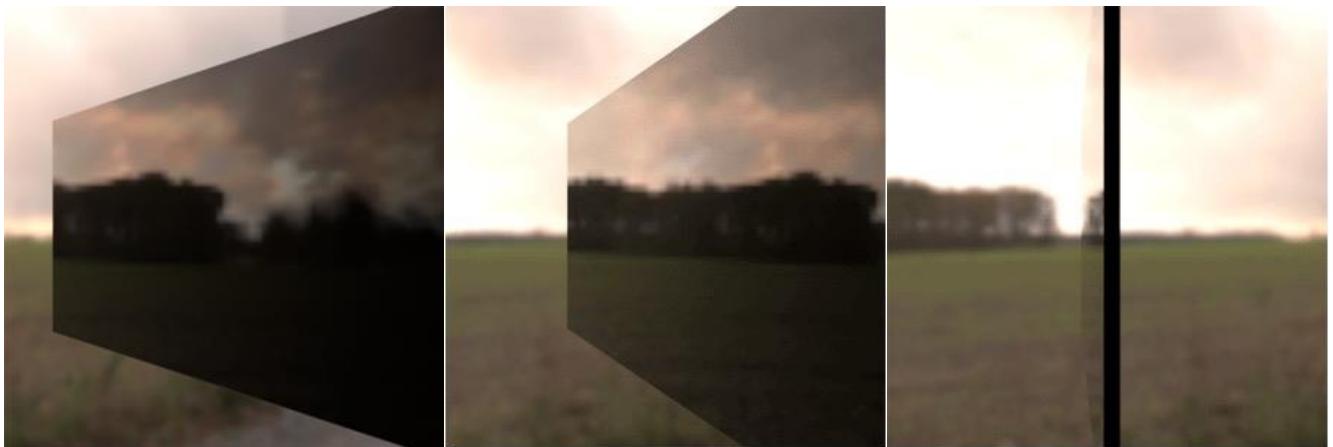
법선에서 반사율: 0.5

프레넬은 어떤 형태로든 모든 재료에 존재하며 유리와 물이 가장 일반적인 예입니다. 아래 예는 나무 테이블에서 프레넬의 효과를 보여줍니다. 카메라의 시야각이 알아짐에 따라 테이블 위의 반사가 어떻게 변하는지 확인하십시오.



보기 방향에 따른 Specular BRDF의 변화

아래 예에서는 IOR을 사용하여 큐브에 Standard Surface 세이더가 지정되어 있습니다. 프레넬 효과가 확실히 보입니다. 카메라의 시야각이 알아지면 광택 반사가 더 밝게 나타납니다.

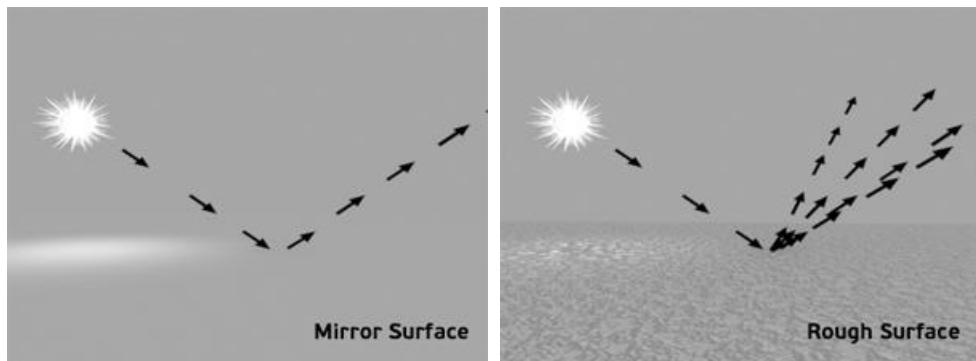


미세면의 표면

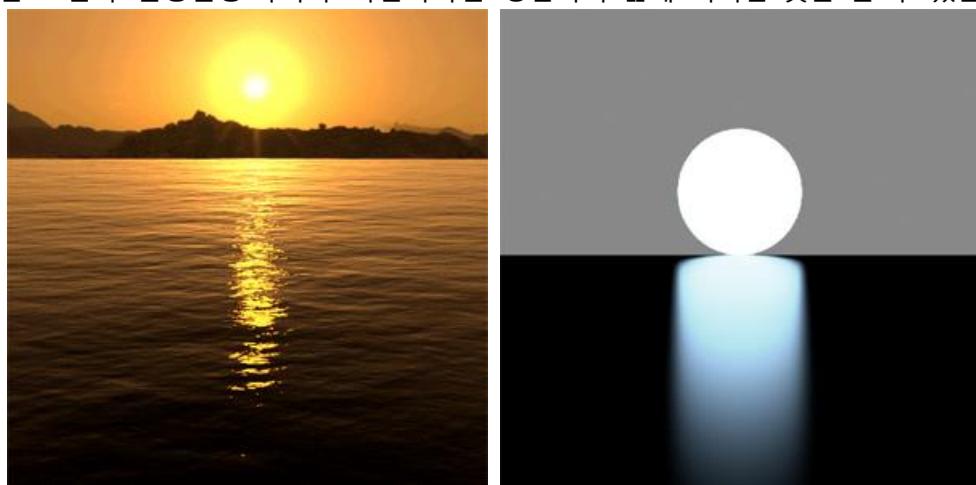
Cook-Torrance는 미세면 모델을 기반으로 하는 미세면 BRDF입니다(예를 들어 "**Microfacet Models for Refraction through Rough Surfaces**", Walter, Marschner, Li and Torrance, Eurographics 2007).

이들 모델은 완벽하게 부드럽지 않고 매우 작은 면들로 구성된 표면을 고려합니다. 각 면은 완벽한 반사경입니다. 이 미세면들은 근사화된 매끄러운 표면의 법선에 대해 분포됩니다. 미세면 법선이 매끄러운 표면 법선과 다른 정도는 표면의 거칠기에 의해 결정됩니다.

아래의 애니메이션은 표면의 거칠기가 정반사에 어떠한 영향을 주는지를 보여줍니다.



아래의 예제에서는 표면이 '울퉁불퉁'하거나 '거칠어지면' 정반사가 넓게 퍼지는 것을 볼 수 있습니다.



Anisotropy 및 Roughness

이방성 물질의 예제는 아스팔트 콘크리트일 것입니다. 이 효과는 젖은 도로의 표면에서 분명합니다. 젖은 표면은 확산적으로 더 어둡고 반사적으로는 더 밝게 보입니다. 표면으로부터 멀어지면 지면에서 늘어나는 이방성 반사가 더욱 명확해집니다. 그러나 가까워지면 늘어난 반사가 시야각의 변화로 인해 작아집니다. 이방성 하이라이트 크기는 시야각에 따라 다릅니다.



마른 확산 도로 표면 vs 젖은 이방성 도로 표면

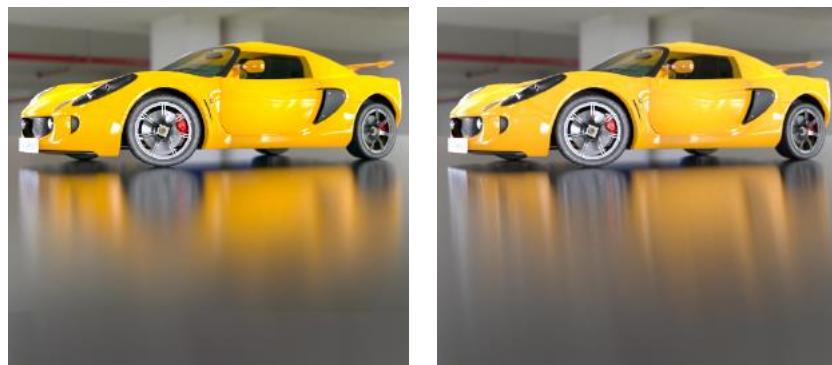
아래의 애니메이션은 이방성(0, 0.5, 1)을 사용할 때 반사 거칠기의 효과를 보여줍니다. 거칠기 값이 효과를 나타낼 수 있도록 0에서 1로 애니메이트되었습니다.



Anisotropy 값 왼쪽부터 오른쪽으로: 0, 0.5, 1

Anisotropy 예제

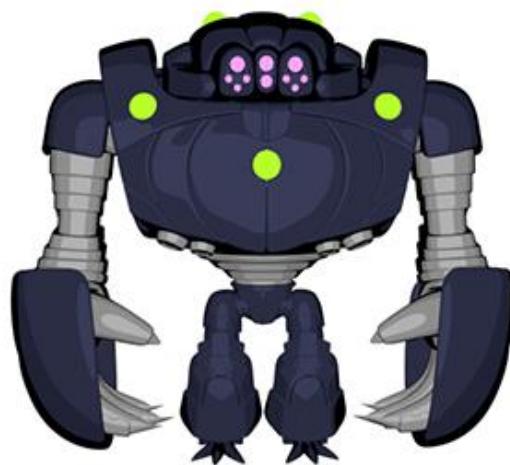
아래 이미지들은 바닥면에 이방성이 있거나 없을 때의 렌더링 차이점을 보여줍니다. 두 개 쉐이더 모두 Roughness가 0.2로 설정되어 있습니다.



Floor shader Anisotropy: 0

Floor shader Anisotropy: 0.8

Facing Ratio 쉐이더를 이용한 Toon Shading



이 튜토리얼에서는 Arnold를 사용하여 양식화된 만화 쉐이딩 효과를 만드는 방법을 보여줍니다. **Facing Ratio**를 다양한 다른 **Utility** 및 **Color** 쉐이더와 함께 사용하여 효과를 미세 조정합니다.

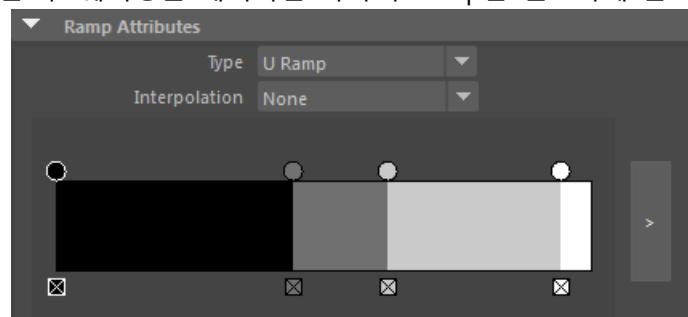
Scene 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Toon+Shading+with+the+Facing+Ratio+shader>)

Workflow

Standard Surface

- **Standard Surface** 쉐이더를 생성하고 이를 지오메트리에 할당합니다. 이 경우 두개의 Standard Surface 쉐이더가 Juggernaut 모델(회색 및 파랑)에 지정됩니다.
- Ramp 쉐이더를 만듭니다. 램프에 대해 네가지 색상을 만듭니다: 검정색(외곽선 용), 어두운 회색, 밝은 회색 및 흰색. 나중에 Color Correct 쉐이더를 사용하여 모델의 회색 및 파란색 부분에 개별 색상을 추가합니다. 그렇게하면 모든 투 쉐이딩을 제어하는 하나의 Ramp만 필요하게 됩니다.



Color Correct 쉐이더를 생성하고 이를 Standard Surface 쉐이더의 *Emission Color*에 연결합니다. *Emission*을 1로 증가시키고, *Base*와 *Specular Weight*를 0으로 낮추는데 이는 툰 쉐이딩에 이러한 속성이 필요하지 않기 때문입니다. Ramp를 Color Correct의 *Input0*에 연결합니다.

Facing Ratio

Ramp의 2d texture를 삭제하고 **Facing Ratio** 쉐이더를 Ramp의 UV Coords의 *U* 파라미터에 연결합니다. (Ramp Type이 U로 설정되어 있는지 확인하세요.)



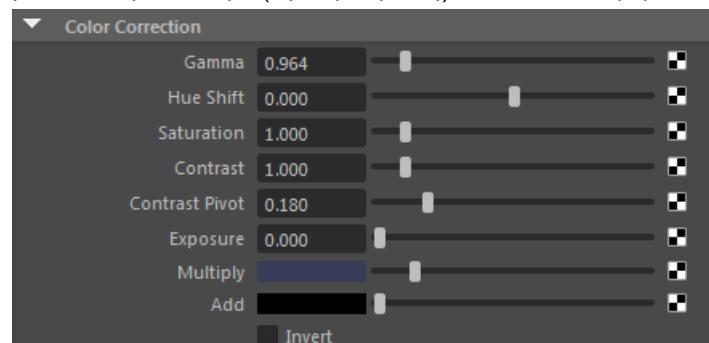
Ramp의 Out Value가 Ramp의 uCoord 연결

Utility Shader

Ramp 연결된 Utility 쉐이더(Shade Mode를 *nodoteye*로 설정)를 사용하여 유사한 대칭 비율 효과를 만들 수 있습니다. 또한 다른 툰 쉐이딩 효과를 내기 위해 *Color* 모드 중 일부를 시도해 볼 수 있습니다. *Ndoteye* 쉐이딩은 specular reflection에는 나타나지 않습니다.

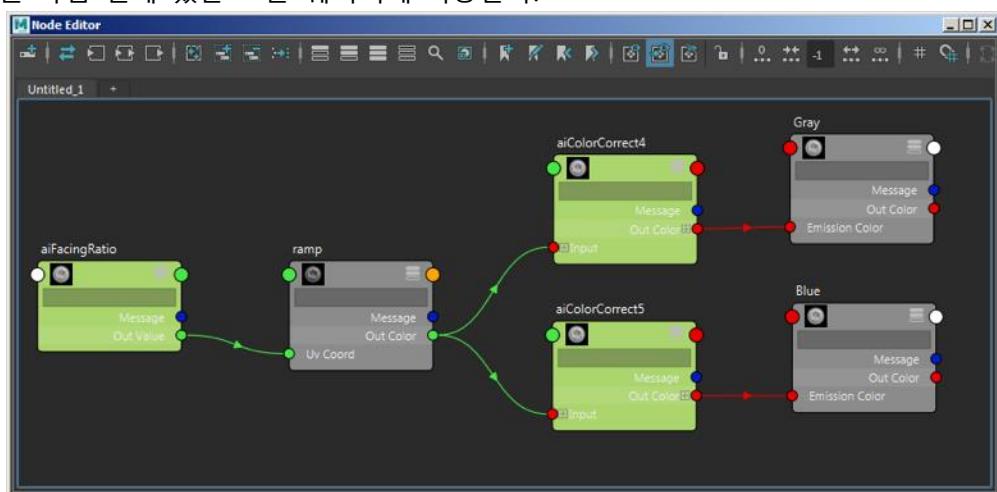
Color Correct

- **Color Correct** 쉐이더를 사용하여 램프에 색상을 추가하고 이를 사용하여 툰 쉐이딩 효과를 미세 조정할 수 있습니다.
- *Multiply* 색상을 쉐이딩으로 지정할 색상(이경우 파란색)으로 변경합니다.



파란색이 Color Correct 쉐이더의 Multiple에 연결

- 이 작업을 작업 씬에 있는 모든 쉐이더에 적용한다.



Multiple 쉐이더들이 동일한 Ramp와 Facing Ratio에 연결 됨

Facing Ratio 쉐이더를 사용하여 툰 쉐이딩 효과의 모양을 제어할 수 있습니다. 다음은 Bias와 Gain 속성을 조정할 때의 결과입니다.

Bias



0.25



0.5



0.75

Gain



0.1



0.5(default)



0.9

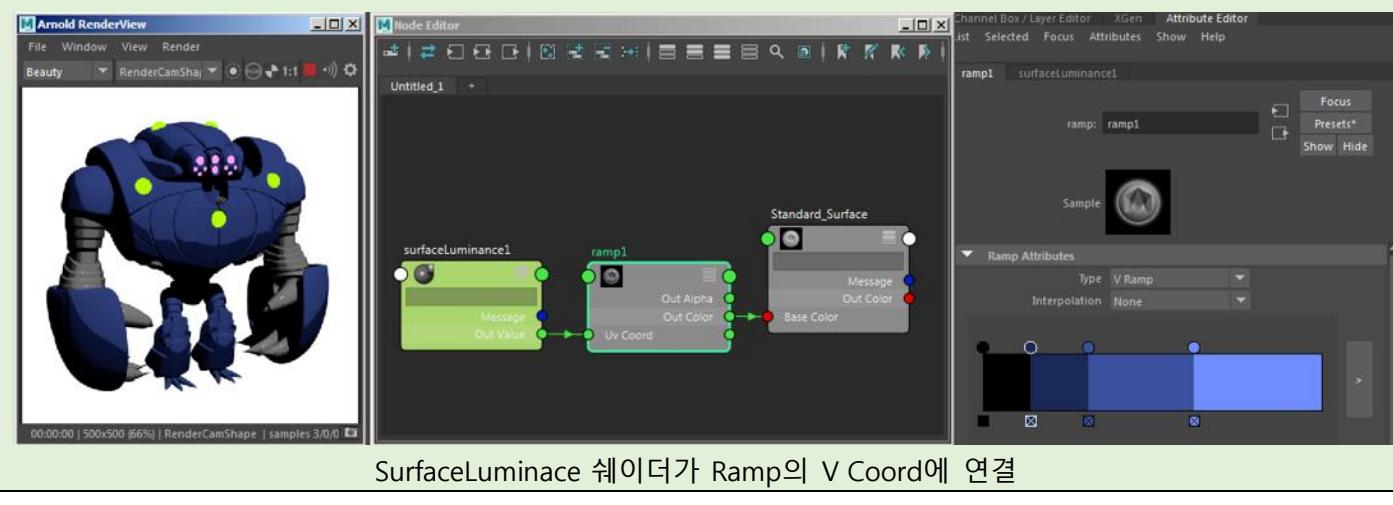
Range 쉐이더는 Ramp의 입력 값을 다시 매핑하여 툰 쉐이딩 효과를 미세 조정하는데도 사용할 수 있습니다.



Range 쉐이더가 Emission Color에 연결.

Surface Luminance

이 노드를 사용하여 환경의 광원을 기반으로 쉐이딩을 변경할 수 있습니다. 아래 예제에서 Surface Luminance 쉐이더는 Standard Surface 쉐이더의 **Base Color**에 연결된 Ramp의 V Coord를 구동하는데 사용됩니다.



User Data 튜토리얼

다음 튜토리얼은 MtoA에서 **User Data** 쉐이더의 사용에 대해 설명합니다.

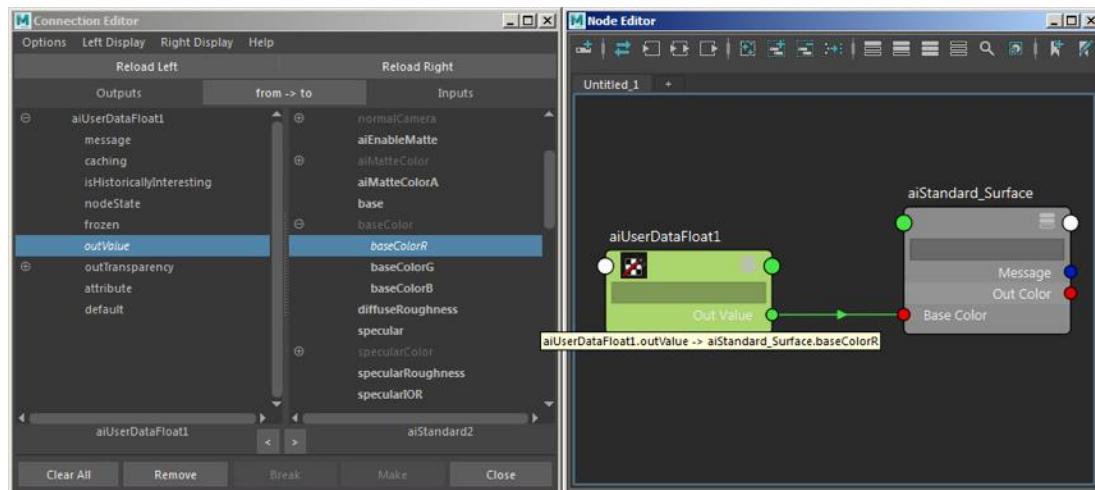
- [Ai User Data Float 튜토리얼](#)
- [Ai User Data String](#)
- [Ai Utility - Object ID Mode](#)
- [Ai Write Color](#)

Ai User Data Float 튜토리얼

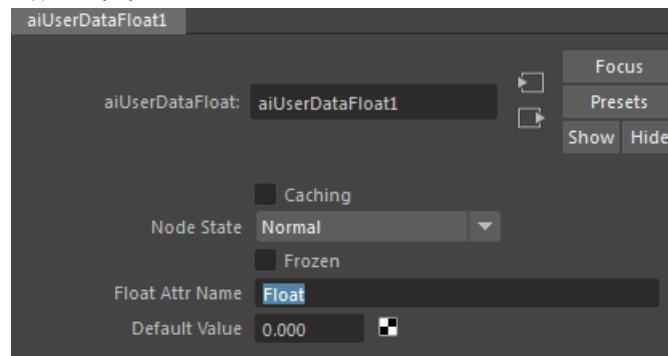
이 간단한 자습서에서는 *Ai User Data Float* 쉐이더의 사용 방법에 대해 설명합니다.

Maya Scene 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.
<https://support.solidangle.com/display/A5AFMUG/Ai+User+Data+Float+Tutorial>

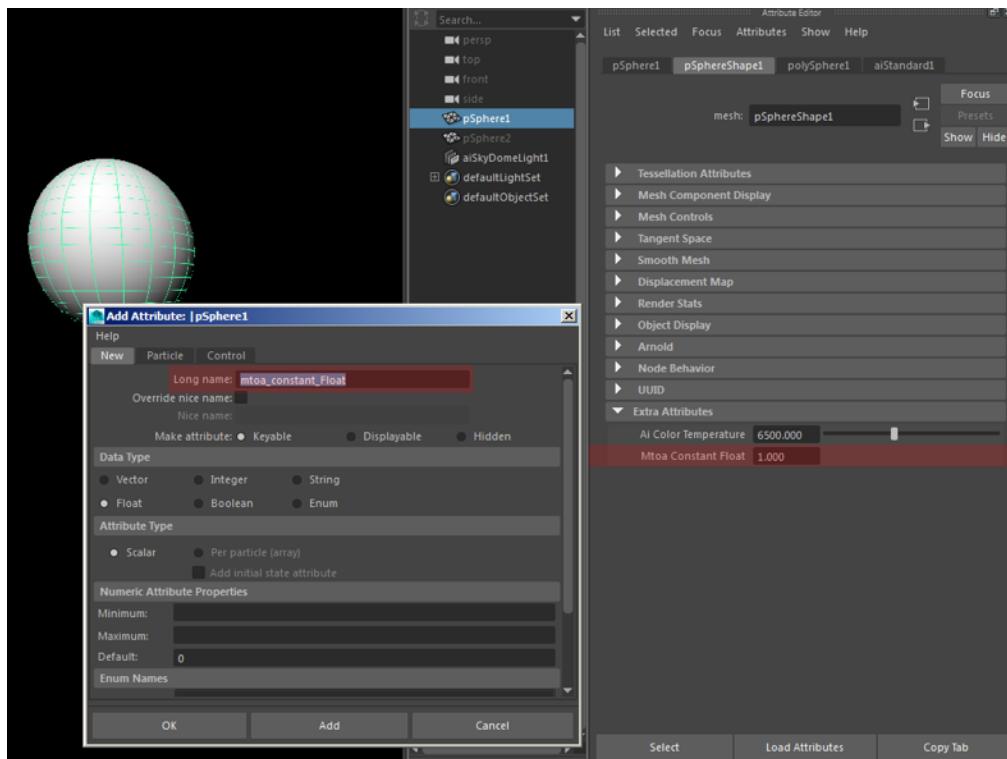
- 먼저 **Ai User Data Float** 쉐이더 및 **Standard Surface** 쉐이더를 만들어보겠습니다. 두 항목 모두에 대한 연결 편집기를 엽니다. Ai User Data Float 쉐이더의 **outValue**를 선택하고 이것을 Standard Surface 쉐이더의 **basecolorR** 속성에 연결합니다.



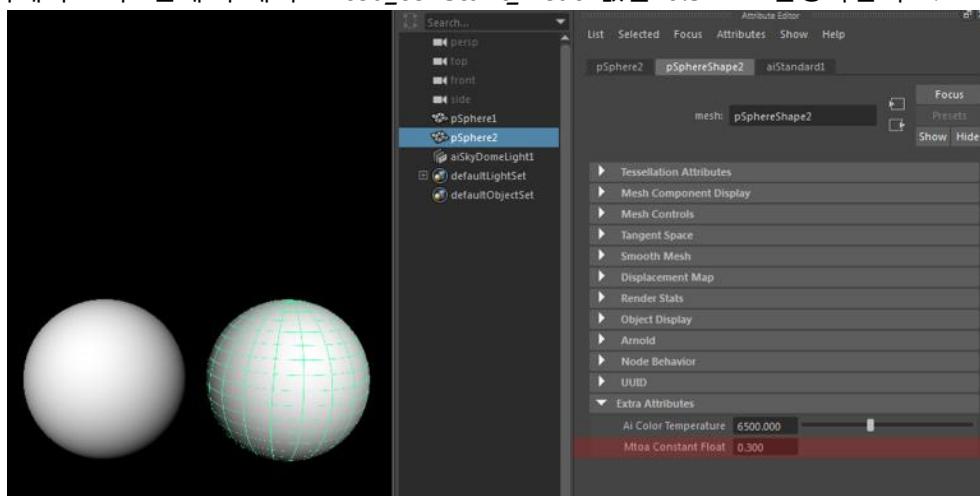
- Ai User Data Float 노드를 선택하고 속성 편집기의 Float Attr Name에 이름을 추가합니다. 이 예에서는 'Float'라는 이름을 사용했습니다.



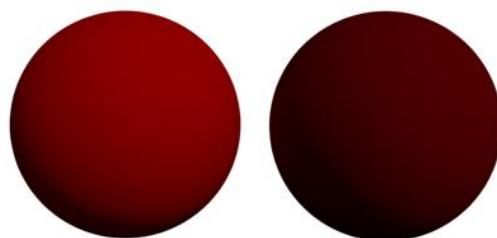
- 구체를 만들고 여기에 속성을 추가합니다(transform 노드가 아닌 shape 노드). 명명 규칙 'mtoa_constant_'를 사용하고 Ai User Data Float 노드에 대해 제공한 것과 동일한 이름인 'Float'를 추가합니다. 그러면 이름은 'mtoa_constant_Float'가 됩니다.
- 구체의 추가 속성에서 새 속성 'mtoa_constant_Float'이 1 값과 함께 표시되어야 합니다.



- 구체를 복제하고 두 번째 구체의 '*mtoa_constant_Float*' 값을 *0.3*으로 변경하십시오.



- 장면을 렌더링합니다. 왼쪽의 구체가 'Float' 속성 1.0에 의해 1.0의 순 적색 값을 가지게 됨을 볼 수 있습니다. 반면에 오른쪽의 구체는 'Float' 속성이 0.3으로 변경되었기 때문에 약간 어두운 적색입니다.

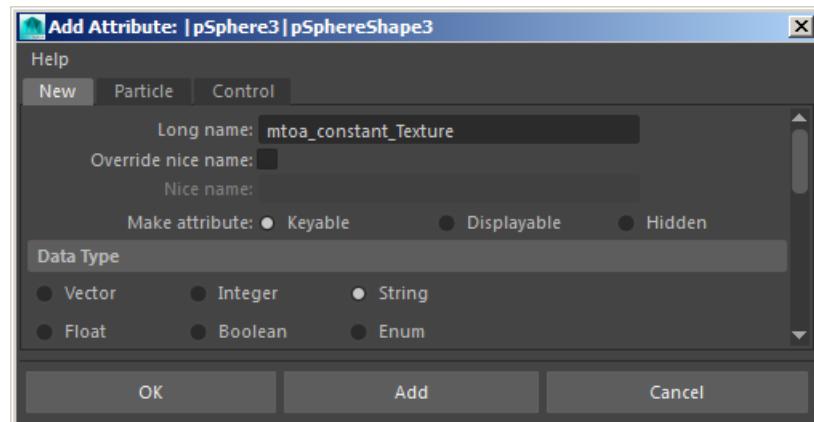


Ai User Data String

이 간단한 자습서에서는 **Ai UserData String** 노드의 사용 방법에 대해 설명합니다. 이 예제에서는 동일한 **Standard Surface** 쉐이더를 사용하여 여러 텍스처를 여러 오브젝트에 연결할 수 있습니다.

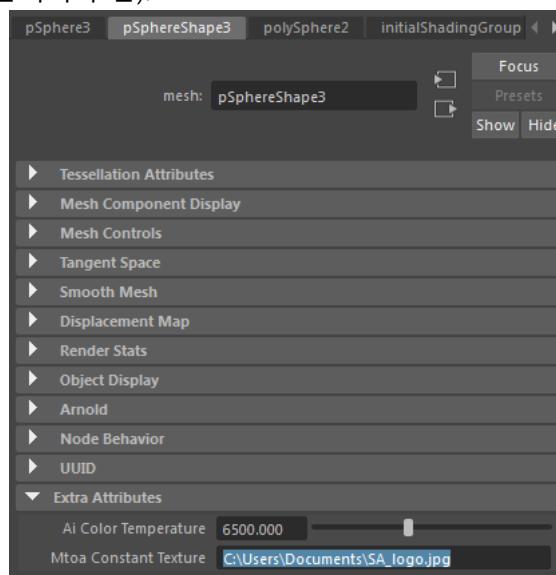
Maya scene파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.
(<https://support.solidangle.com/display/A5AFMUG/Ai+User+Data+String>)

- 구체를 만들고 이 형상 노드에 속성을 추가합니다(transform 노드가 아님). 아래 키를 (pickWalkDown) 사용하여 형상 노드를 선택합니다.
- 긴 이름 - '**mtoa_constant_Texture**'('Texture'는 Ai User Data String 노드에 대해 사용할 접두사이지만 원하는 대로 부를 수 있습니다('name'은 사용하지 마십시오)를 추가합니다. '**Data Type**'을 '**String**'으로 설정합니다.

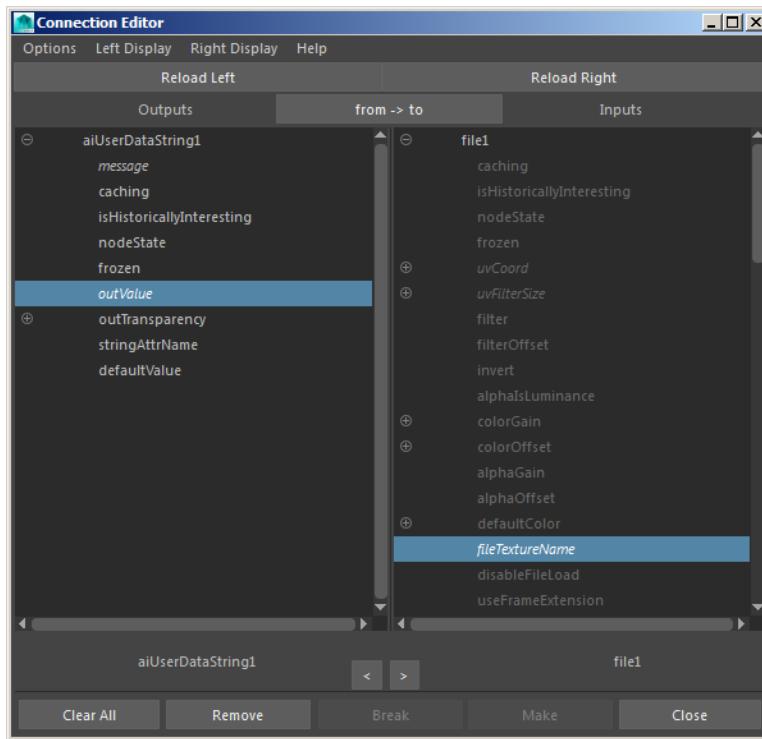


String Maya 속성

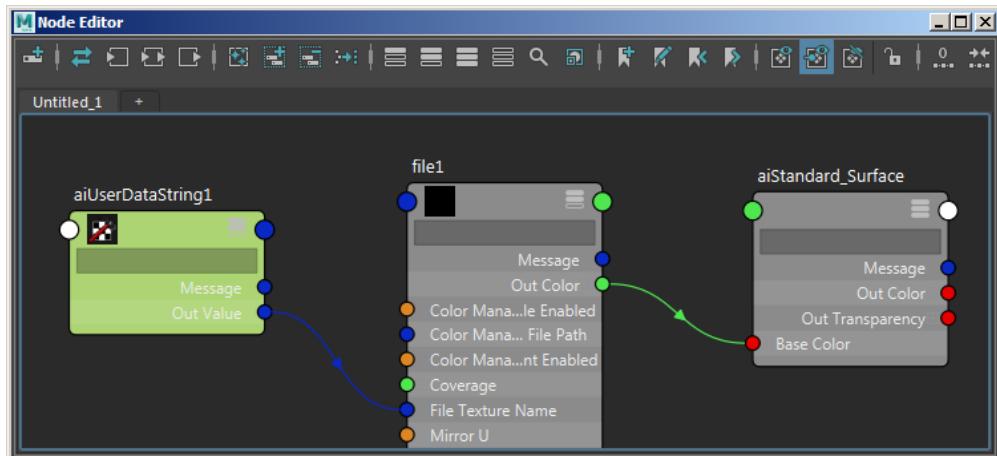
- 구체에 대한 형상 노드를 선택합니다. '**Extra Attributes**' 아래에서 새로 만든 Attribute - '**mtoa_constant_Texture**'를 볼 수 있습니다. 옆에 있는 텍스트 필드에 해당 텍스처의 전체 경로 이름을 입력합니다(절대 경로 이름이어야 함).



- Standard Surface 쉐이더를 구체에 할당합니다. **Base Color** 속성에 파일 텍스처를 연결합니다. **Ai User Data String** 노드를 만듭니다.
- 두 항목 모두에 대한 연결 편집기를 엽니다. **Ai User Data String** 노드의 **outValue** 속성을 "file texture" 노드의 **fileTextureName** 속성에 연결합니다.

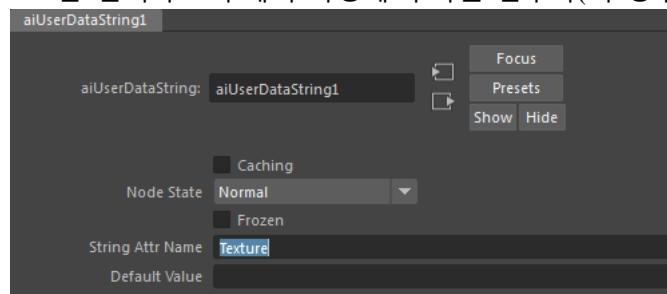


쉐이더 네트워크는 다음과 같은 모습입니다.



파일 텍스처 노드에 연결된 Ai User Data String

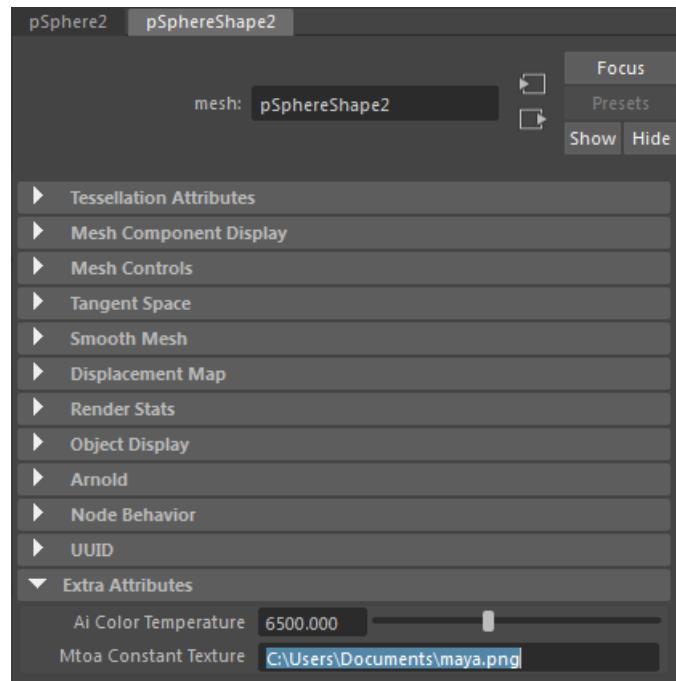
- Ai User Data String** 노드를 선택하고 구체의 속성에 추가된 접두사(이 경우 '**Texture**')를 입력합니다.



- 장면을 렌더링합니다. 구체는 구체의 속성에 추가된 텍스처 경로를 사용하여 렌더링해야 합니다.



- 구체를 복제합니다. 두 번째 구체에 대한 'Extra Attributes' 아래에 두 번째 텍스처 맵의 전체 경로 이름(절대)을 입력합니다.



- 장면을 렌더링합니다. 두 구체 모두 동일한 Standard Surface 쉐이더를 사용하지만 별도의 텍스처가 할당되었습니다.



Ai Utility - Object ID Mode

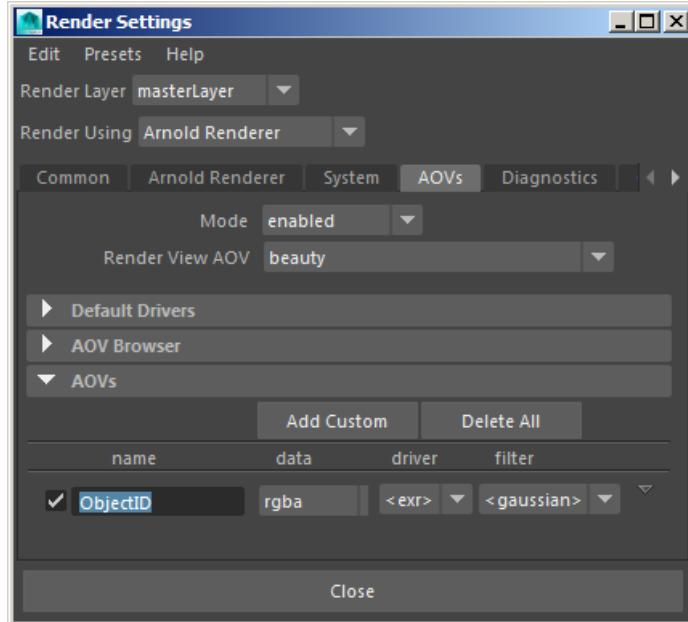
Ai Utility 쉐이더에 의해 트리거된 사용자 정의 색상 정보(beauty 렌더링으로 설정)를 특정 AOV에 쓰는 방법에 대해 알아봅니다. 예를 들어 AOV의 하나로 'Object ID'를 갖고자 합니다.

Maya scene은 하단 링크의 사이트에서 다운로드할 수 있습니다.

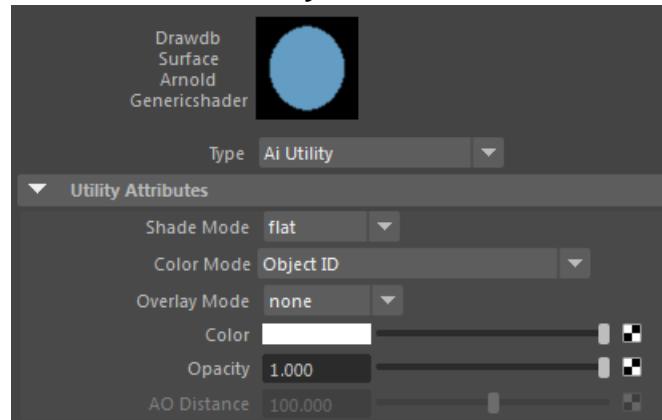
(<https://support.solidangle.com/display/A5AFMUG/Ai+Utility+-+Object+ID+Mode>)

또한 **Ai WriteColor** 쉐이더를 사용하여 유사한 효과를 얻을 수 있습니다.

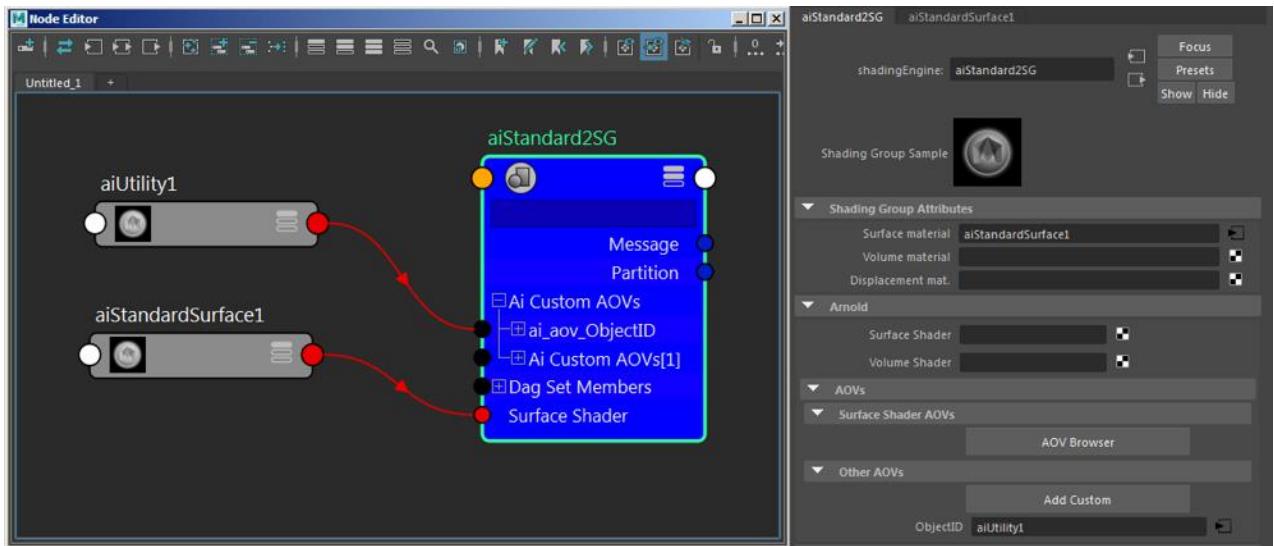
- Arnold 렌더링 설정 내의 AOV 탭에서 **ObjectID**라는 사용자 정의 AOV를 추가합니다.



- 오브젝트에 **Ai Standard** 쉐이더를 지정합니다.
- Ai Utility** 쉐이더를 만들고 **Color Mode**를 **ObjectID**로 설정합니다.

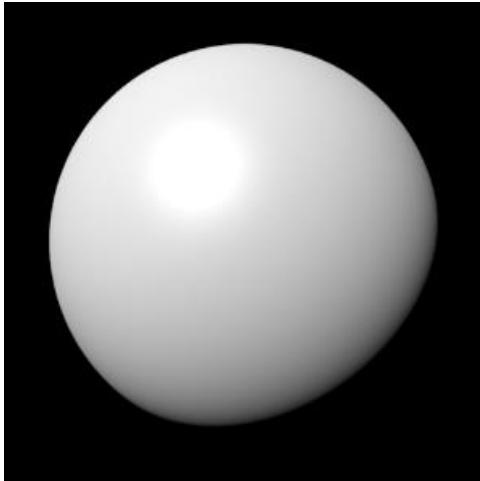


- Standard 쉐이더가 연결된 **Shading Group**으로 이동합니다. Arnold 섹션의 **Other AOVs** 아래에 **ObjectID** 속성이 있습니다.
- Hypershade로부터 **Utility** 쉐이더를 끌어서 Shading Group 내의 **ObjectID** 속성에 놓습니다.

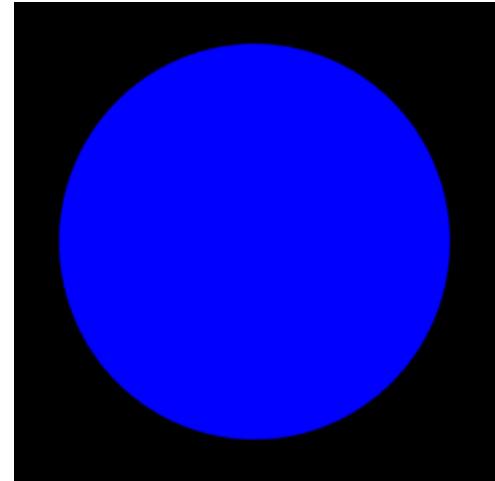


Utility 쉐이더를 Standard Surface SG의 ObjectID로 끌어오기

이제 사용자 정의 'ObjectID' AOV는 Ai Utility 쉐이더에서 값을 얻을 수 있지만 나머지 표준 AOV도 사용할 수 있습니다.



Beauty 렌더링



'ObjectID' AOV를 통해 렌더링된 Ai Utility 노드

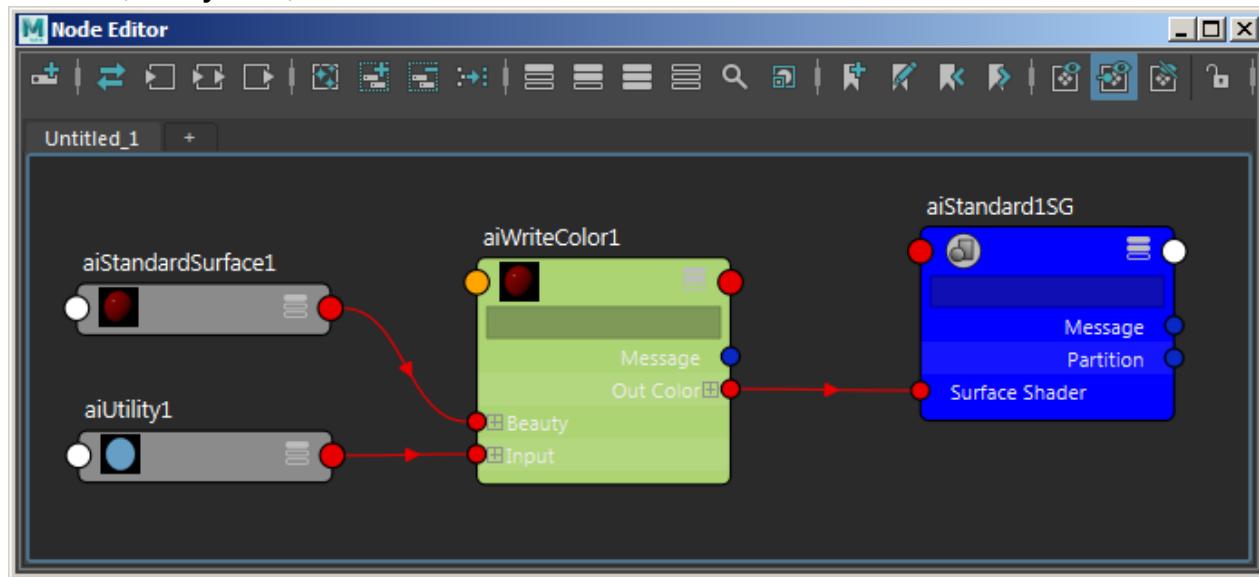
Ai Write Color

이 짧은 자습서에서는 Ai WriteColor 노드를 사용하여 다른 쉐이더(예: **Utility** 쉐이더)의 출력을 사용자 정의 AOV에 추가하는 방법에 대해 설명합니다.

Maya 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

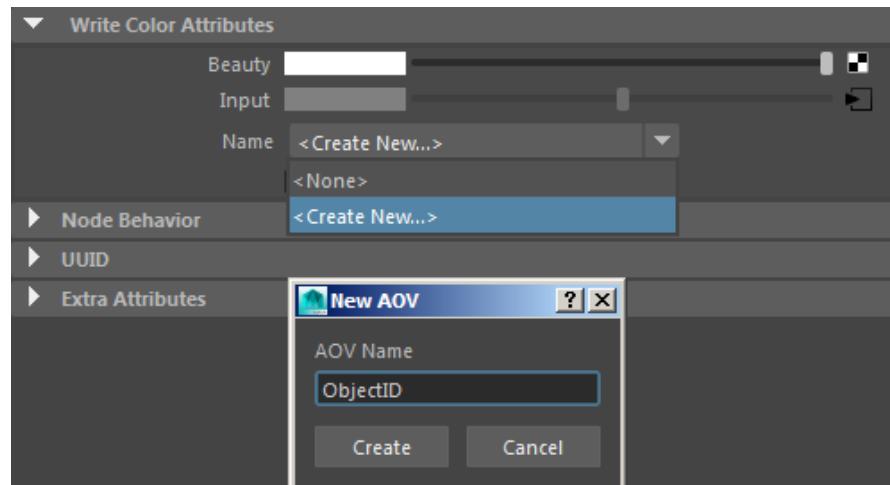
(<https://support.solidangle.com/display/A5AFMUG/Ai+Write+Color>)

- **Standard Surface** 쉐이더를 구체에 할당합니다.
- WriteColor의 beauty에 RGB를 출력하는 쉐이더(다른 Standard Surface와 같은)를 연결합니다.
- **WriteColor** 쉐이더를 Standard Surface 쉐이더의 **Color** 속성에 연결합니다.
- **Utility** 쉐이더를 **WriteColor** 쉐이더의 입력에 연결합니다. 원하는 값을 출력하도록 Utility 쉐이더를 구성합니다(예: **Object ID**):

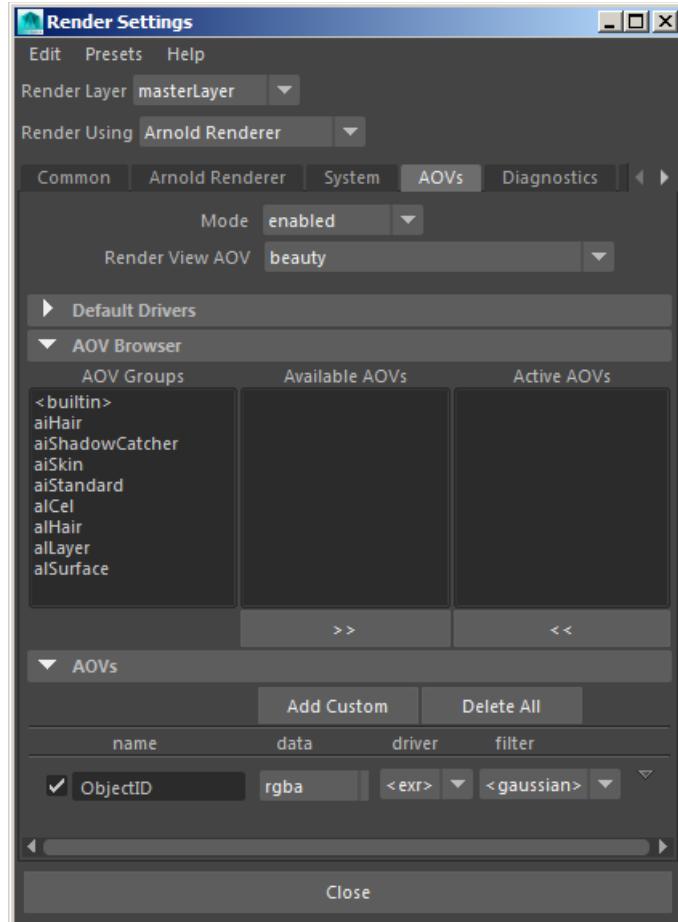


최종 쉐이더 네트워크

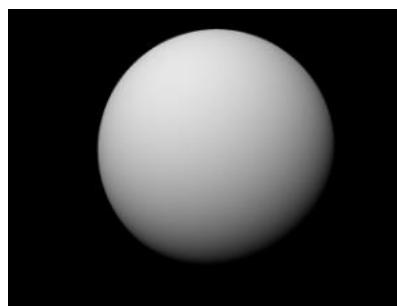
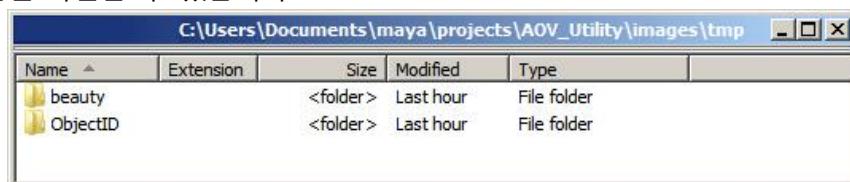
- **WriteColor**의 '**Name**' 속성에서 Utility 쉐이더의 출력과 연결할 사용자 정의 AOV(예: '**ObjectID**')를 선택(또는 작성)합니다.



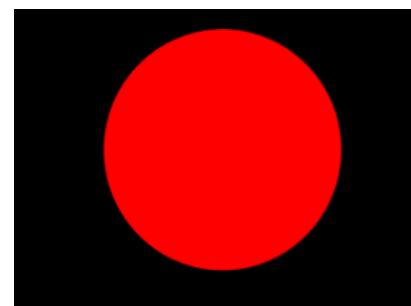
- Render Settings 창을 열고 AOVs 탭으로 이동합니다. 그러면 새롭게 만든 AOV가 보입니다.



- 장면을 렌더링합니다. 이미지 폴더에서 MtoA가 beauty 렌더링과 ObjectID AOV와 연관된 Utility 쉐이더를 모두 작성했음을 확인할 수 있습니다.



Beauty 렌더링

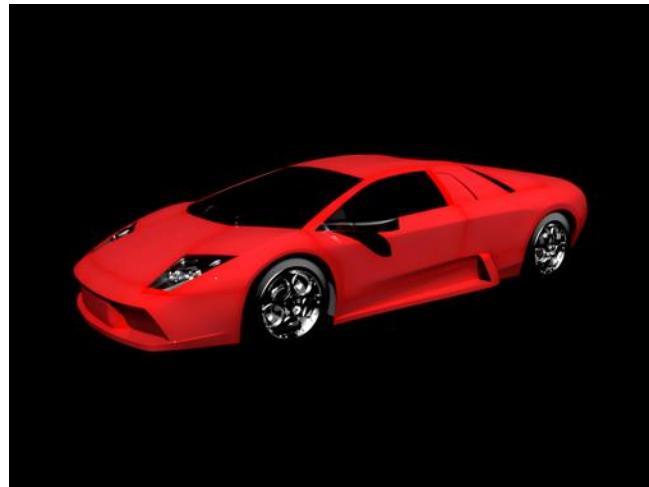


Ai Utility 렌더링

예 1

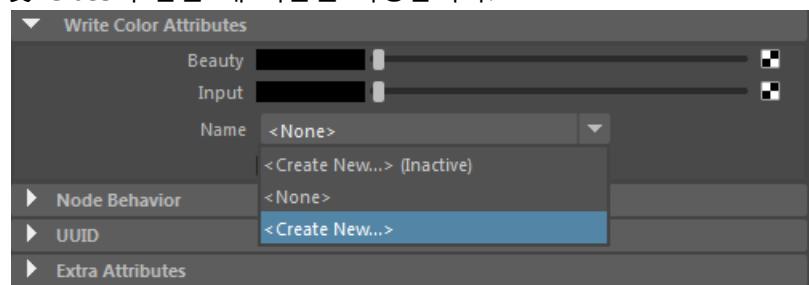
자동차 모델에 대해 다른 매트를 만들고 싶다고 가정해 보겠습니다. Ai WriteColor 노드는 모델을 여러 매트로 분할하는 유용한 방법이 될 수 있습니다. 이 예제에서는 자동차 페인트와 앞 유리 모두에 Ai Write Color 노드를 사

용할 것입니다.

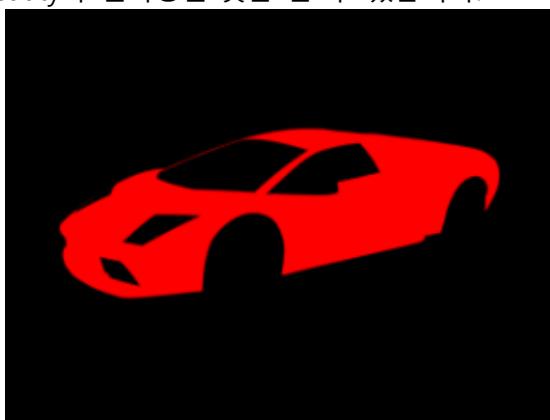


Car beauty 렌더링

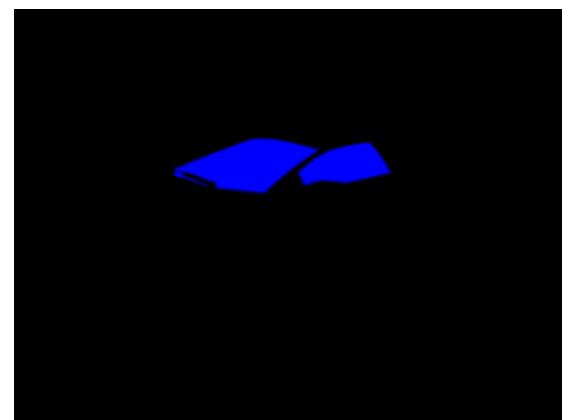
- 위의 구체에 사용된 방법을 사용하여 자동차 페인트와 앞 유리 모두에 대해 **Ai Write Color** 노드를 만듭니다. 자동차 페인트 재료와 앞 유리를 각각 지정하십시오.
- 각 **Ai Write Color** 노드를 선택하고 **Name** 옆에 있는 드롭다운 메뉴를 선택합니다. 각 **Ai WriteColor** 노드에 **Car Paint** 및 **Glass**와 같은 새 이름을 지정합니다.



- 장면을 렌더링합니다. 별도의 매트로 사용할 수 있는 자동차 페인트 및 앞 유리 이미지를 포함하여 car beauty가 렌더링된 것을 볼 수 있습니다.



별도의 매트 이미지로 렌더링된 자동차 페인트
유ти리티



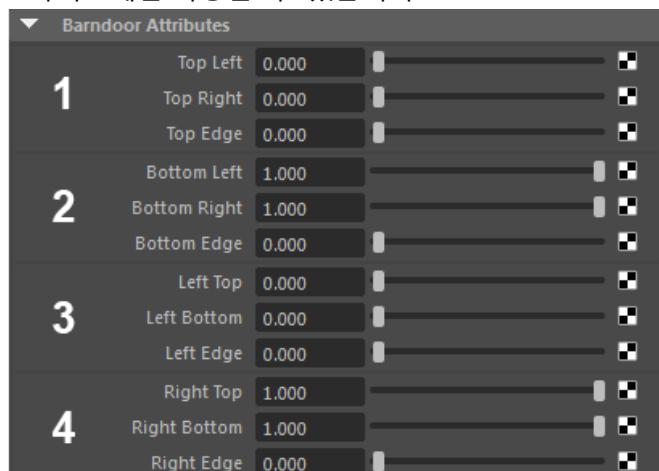
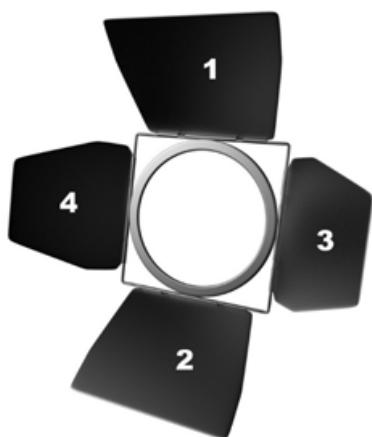
별도의 매트로 렌더링된 앞 유리 유ти리티

Barndoors Light Filter 사용하기(AiBarndoors)



Barndoors는 조명의 열린 부분 양쪽에 부착된 움직이는 불투명 패널입니다. 이들은 일반적으로 연극이나 영화 조명에서 조명 빔의 형상에 대한 추가적인 제어를 제공하기 위해 사용됩니다. 이 자습서에서는 Barndoors 조명 필터를 사용하는 방법과 다양한 설정을 보여줍니다.

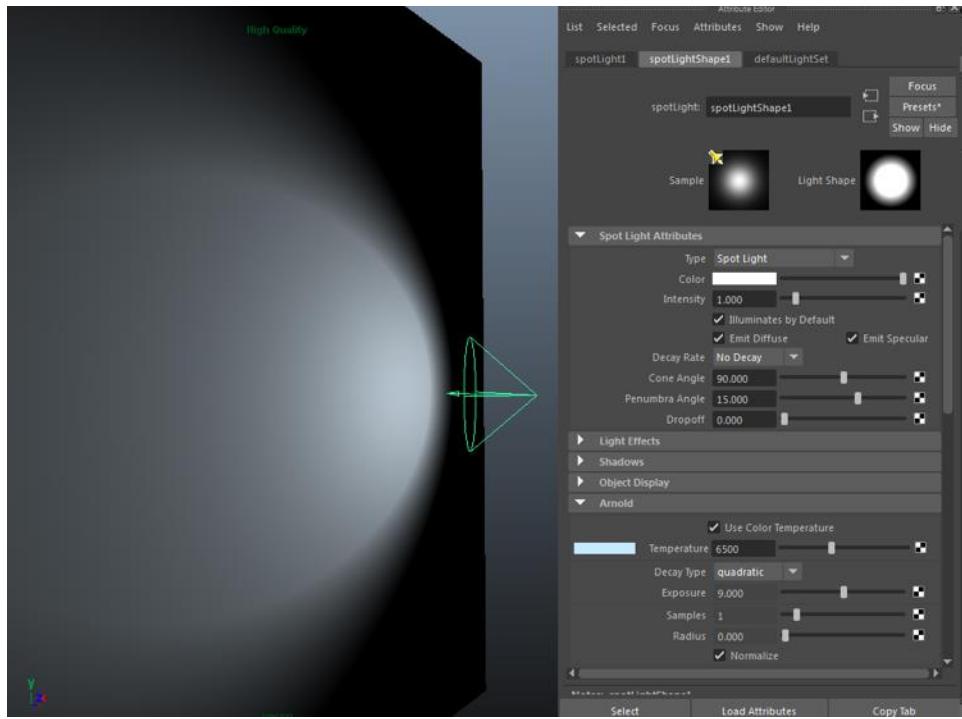
Barndoors 조명 필터는 4개의 반(barn) 도어 플랩이 있습니다. 각 barndoors 플랩에는 3개의 파라미터가 있습니다. 처음 2개의 파라미터는 플랩의 양쪽 끝을 조명의 면을 가로질러 배치합니다. 세 번째 파라미터(Edge)는 가장자리의 부드러움을 조절합니다. 이 조명 필터는 스포트라이트에만 사용할 수 있습니다.



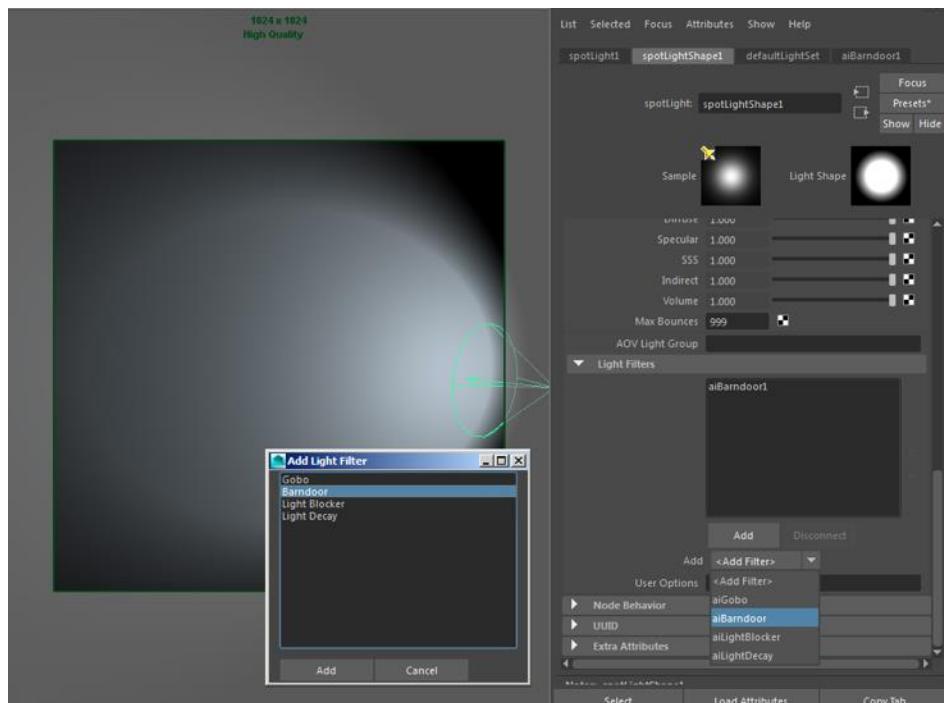
Barndoors 조명 필터 속성

다음 단계는 barndoors를 사용하는 방법을 보다 자세하게 설명합니다.

- 스포트라이트와 폴리곤 평면을 만듭니다. 평면에 스포트라이트를 비춥니다. **cone angle**을 높여서 조명의 효과를 볼 수 있게 합니다. 7(**Lighting>Use All Lights**)을 누르고 **High Quality Rendering**을 활성화하여 뷰포트에서 스포트라이트의 시각화를 개선시킵니다.

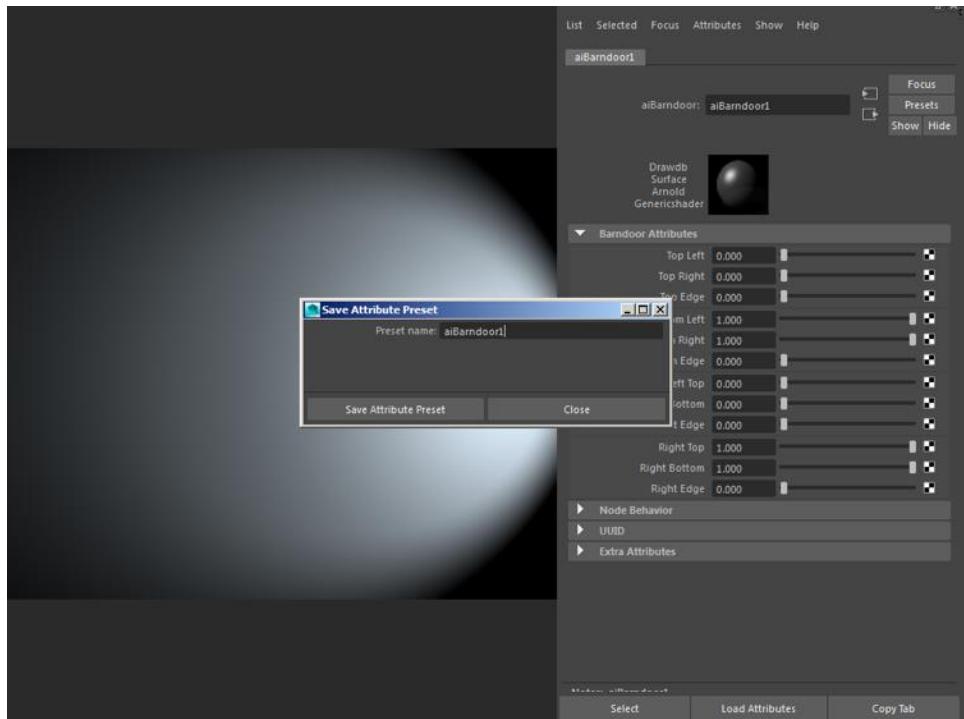


- 스포트라이트의 속성 편집기에서 Arnold로 스크롤 다운하고 **Light Filters** 아래에 있는 **Add**를 클릭합니다. 그러면 다양한 Arnold 조명 필터를 포함한 메뉴가 나타납니다. **Barndoar**를 클릭하고 **Add**를 추가합니다. 그러면 해당 스포트라이트와 해당 Barndoar 필터 사이에 연결이 구축됩니다. 이제 Light Filter 목록에 나타나야 합니다. 이를 두 번 클릭하면 attribute editor에서 해당 속성을 볼 수 있습니다.



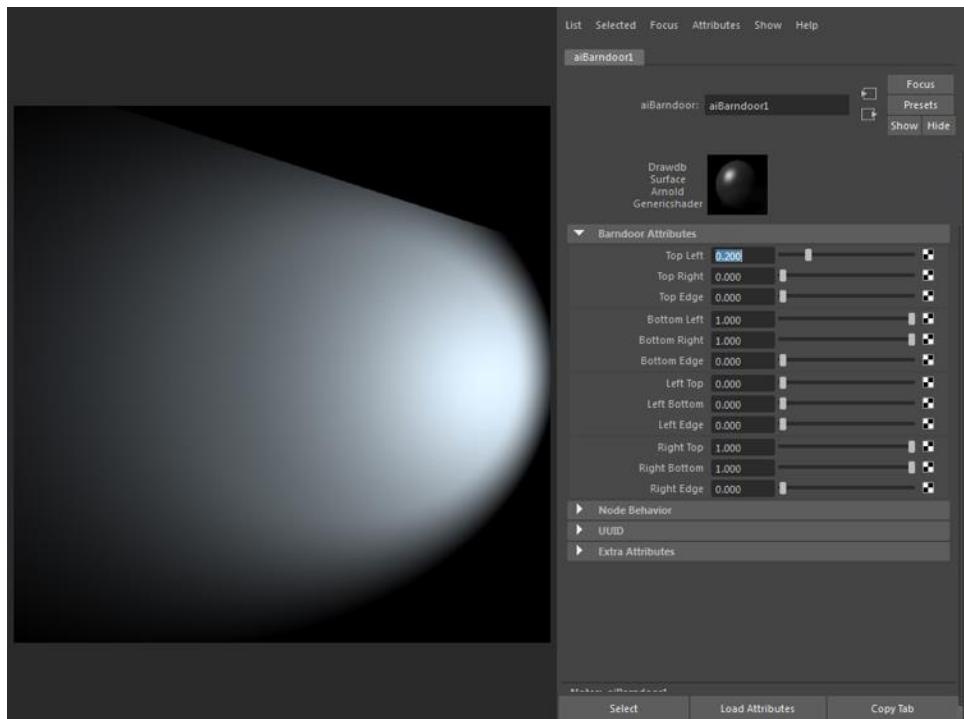
Spotlight의 Arnold 속성에서 Ai Barndoar를 생성합니다.

- 반 도어 플랩의 조정을 시작하기 전에 반 도어에 대한 속성 편집기에서 **Preset**을 만들고 이름을 **Ai Barndoar_default**라고 지정합니다. 그러면 Ai Barndoar의 기본 프리셋 값으로 돌아갈 수 있습니다. 이를 통해 다양한 값을 쉽게 테스트할 수 있습니다. 렌더링 카메라의 IPR 렌더링을 만듭니다.



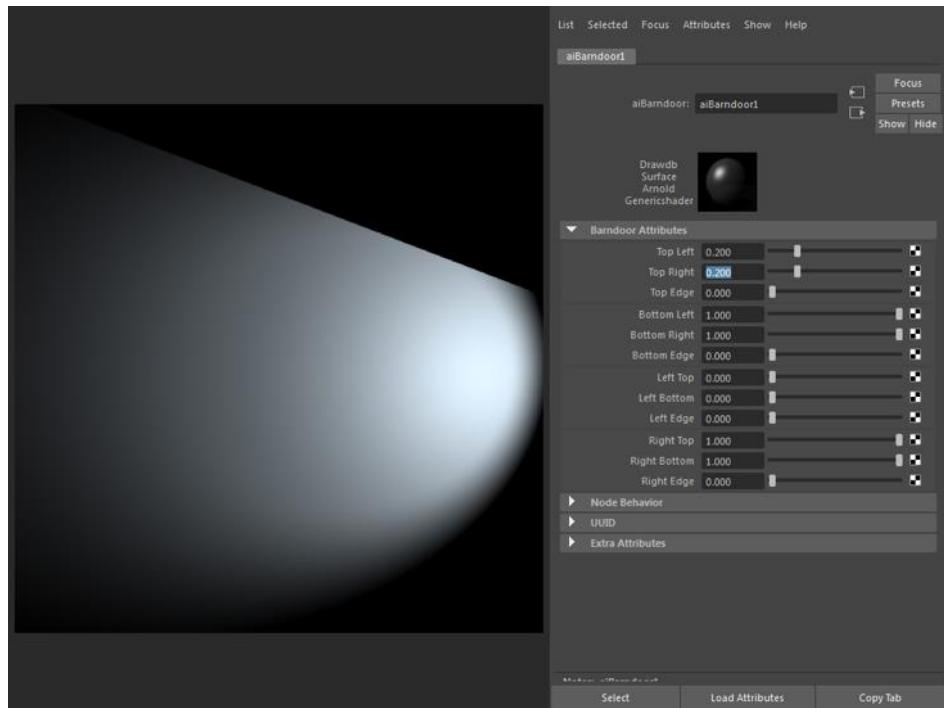
해당 Ai Barndoors에 대해 기본 프리셋을 생성합니다.

- **Top Left** 값을 0.2로 올립니다. 반 도어 맨 위 플랩이 이제 스포트라이트 효과를 클리핑하는 것을 볼 수 있습니다.



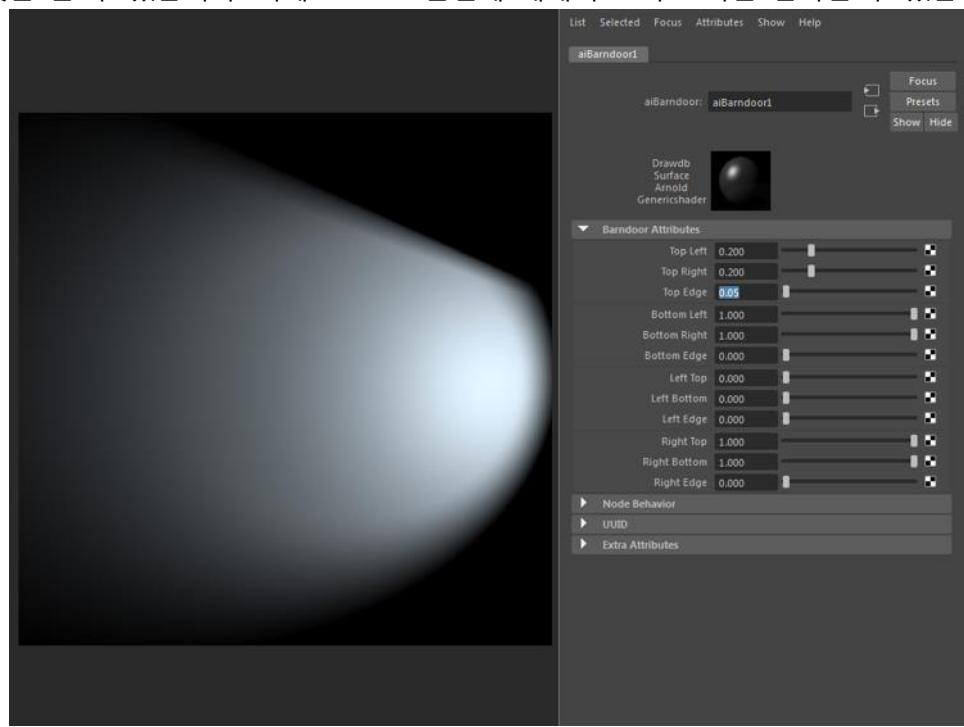
Top Left

- **Top Right**을 0.2로 높여 Top Left 값과 일치시킵니다. 반 도어 효과가 더 두드러진 것을 볼 수 있습니다. Top Left와 Right 모두 1.0의 값은 스포트라이트를 완전히 가리며 어떤 조명 효과도 볼 수 없게 됩니다.



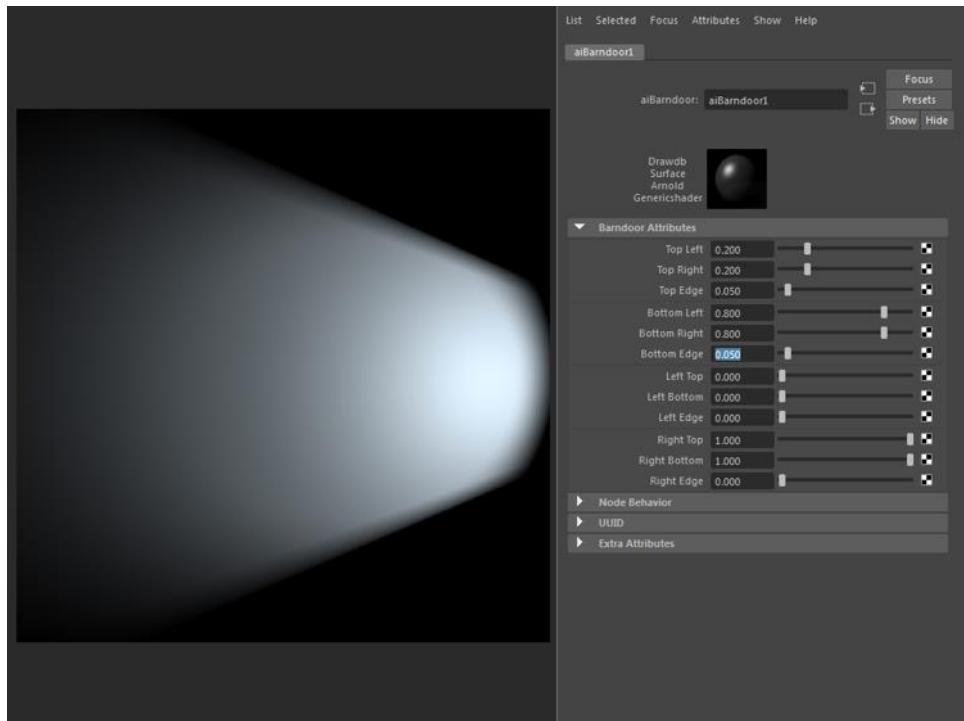
Top Right

- 이제 Top flap의 위치에 만족하므로 **Top Edge**를 **0.05**로 살짝 늘리십시오. 상단 반 도어의 효과가 부드러워지는 것을 볼 수 있습니다. 이제 Bottom 플랩에 대해서도 이 효과를 반복할 수 있습니다.



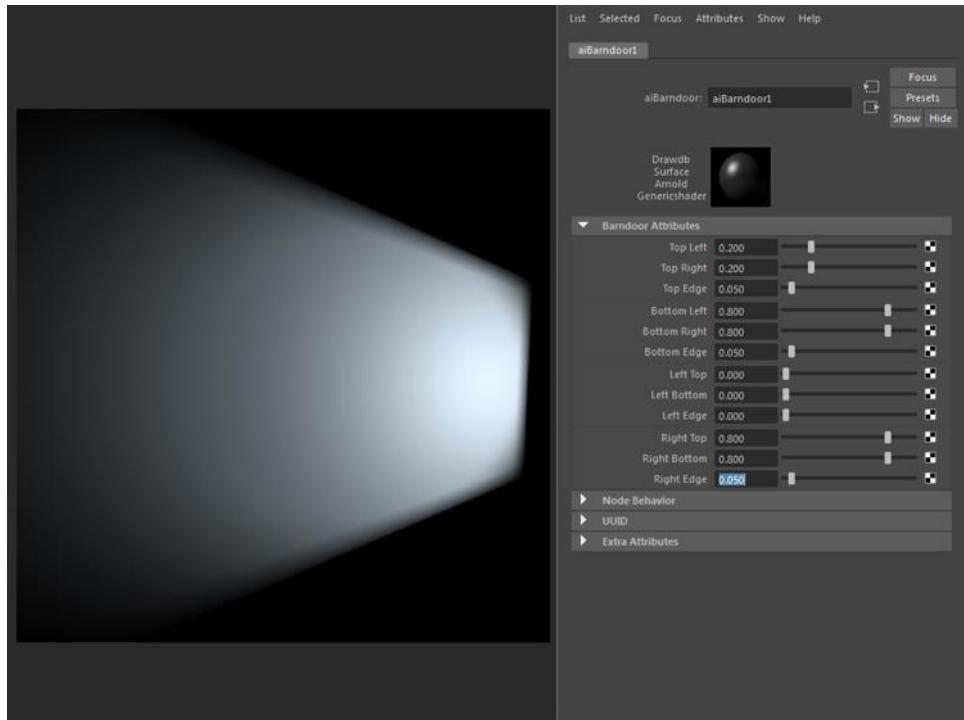
Top Edge

- 하지만 bottom 값은 1.0부터 시작합니다. 상단 플랩에 똑같이 위치시키려면 **Bottom Left** 및 **Bottom Right** 값을 **0.8**로 줄이십시오. Top Edge와 Bottom 가장자리에 사용했던 동일한 값을 사용하십시오.



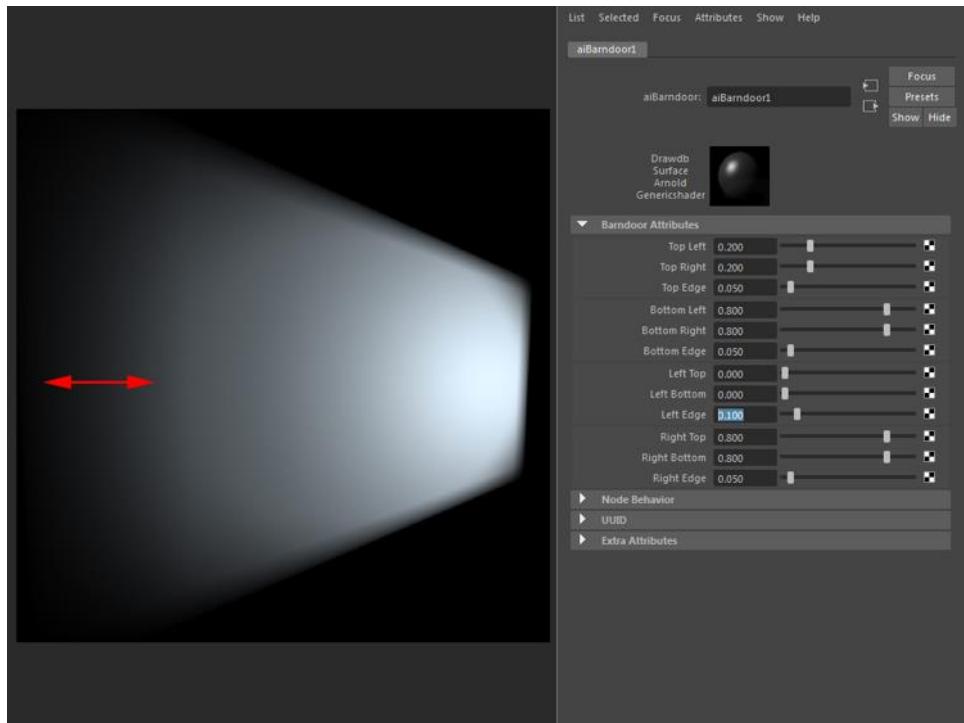
Bottom Left 및 Right

- 이제 스포트라이트의 Right 가장자리를 다듬어 보겠습니다. 아래에 있는 오른쪽 반 도어에 대해 위 단계를 반복합니다.



Right Barndoors

왼쪽 반 도어를 사용하여 화면의 왼쪽에 나타나는 빛의 양을 조절할 수 있습니다. **Left Top** 및 **Bottom** 값을 **0.2**로 높입니다. 빛을 더 많이 끌어 들이기 위해 이 값들을 더 늘릴 수도 있지만 딱딱한 모서리가 되면 안 됩니다. 그 대신 **Left Edge**를 **0.1**로 높일 수 있습니다. 그러면 빛이 떨어지면서 빛을 부드럽게 표현할 수 있습니다.



Left Edge를 높여서 빛의 왼쪽을 부드럽게 처리

- 이제 Ai Barndoors의 모습이 마음에 들기 때문에 Attribute Editor에서 Preset을 저장하여 추후 프로젝트에 사용할 수 있습니다.

물병 만들기



이 페이지는 플라스틱 병에 액체를 렌더링하는 데 유용한 고급 팁을 제공합니다. 사실적인 결과를 얻기 위해 필요한 다양한 음영 처리를 탐구합니다. 특정 광선의 계산을 최소화하기 위해 'Rayswitch' 쉐이더를 사용하는 방법도 살펴봅니다. 이 방식으로 렌더 품질을 저하시키지 않고도 렌더링 시간을 단축할 수 있습니다.

액체와 용기 표면을 다룰 때는 다양한 측면의 음영을 고려해야 합니다. 물리적 정확도가 우선 순위인 경우 기본 Standard Surface 쉐이더 설정이 충분해야 합니다. 하지만 Reflectance at Normal에서 Fresnel을 활성화하는 것은 좀더 선명하고 사실적인 표현을 하는 데 결정적이며 사진 스튜디오 이미지에 더 적합할 수 있습니다.

렌더링 속도가 기본 고려 사항이라면 일부 속성을 비활성화하는 것이 좋습니다(Internal Reflections, Caustics, Bounce Factor 0 활성화). 렌더링 시간을 추가적으로 줄이려면 보조 광선의 Transmission roughness와 같은 값비싼 렌더링 속성을 비활성화하는 rayswitch 쉐이더의 사용을 고려해야 합니다.

이 페이지를 읽기 전에 유리와 액체 표면을 렌더링하기 위한 모델링 요구사항을 더 잘 이해하기 위해 먼저 'Arnold로 유리 표면 렌더링하기' 자습서를 보는 것이 좋습니다.

Scene Setup

Transmission Ray Depth가 7로 증가되어 플라스틱 물병과 유체 표면에 충분한 빛이 통과되도록 합니다. 이 장면은 **Light Shape**이 **Cylinder**로 설정되어 있는 **Area Light**이 비추고 있습니다. 장면에 Sky도 있습니다.

물병 쉐이딩하기

아래 예제는 플라스틱 병 쉐이더에 사용된 Standard Surface 쉐이더의 다양한 속성을 변경할 때의 효과를 보여줍니다.

Internal Reflections 활성화

내부 반사를 선택 해제하면 간접 specular와 mirror perfect이 비활성화됩니다. 'Enable Internal Reflections'를 비활성화하면 렌더링 시간이 크게 단축될 뿐만 아니라 플라스틱 모양에도 영향을 미칩니다. 따라서 이 기능을 사용하기 전에 정확한 품질이나 속도가 더 중요한지 문의하십시오.



Internal Reflections 활성화: 켜짐(17분 45초)



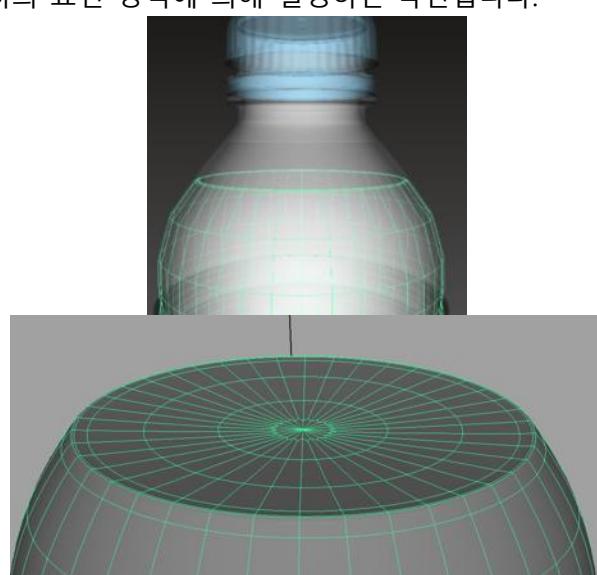
Internal Reflections 활성화: 꺼짐(7분)

플라스틱 병과 액체 표면 모두의 Arnold 속성에서 **Opaque**을 비활성화하는 것을 잊지 마십시오.

물 쉐이딩하기

Liquid Meniscus

액체의 윗부분은 메니스커스(meniscus)로 정확하게 모델링되었습니다. 메니스커스는 용기 또는 다른 물체의 표면에 가까운 액체의 상부 표면에서의 표면 장력에 의해 발생하는 곡선입니다.



액체 메니스커스 모델링 표면의 윗부분을 아래로 당기기에 충분한 가장자리 루프가 있는지 확인하십시오.

플라스틱 쉐이더는 플라스틱을 통과하는 광선에 영향을 미치며 내부의 액체에도 영향을 미칩니다. 따라서 Transmission **Ray Depth** 설정이 광선이 플라스틱 및 액체 표면을 통과할 만큼 충분히 높은지 확인해야 합니다. **Index of Refraction**이 물의 지수(1.33)로 변경되었습니다. **Base Weight**가 0.9까지 감소했습니다. 아래 예제는 물 쉐이더의 다양한 Standard 속성을 변경했을 때의 효과를 보여줍니다.

Base Weight

기본적으로 물은 약간 어둡게 보입니다. 소량의 diffuse를 추가하면 물의 외관을 '밝게' 하는 데 도움이 됩니다(물리적으로 정확한 방법은 아님).



Specular

Specular Weight는 0으로 두었습니다. 왜냐하면 물에서 병 안에 정반사를 표시하지 않기를 바라기 때문입니다.



Internal Reflections 활성화

아래 이미지들은 **Enable Internal Reflections**을 사용하거나 사용하지 않을 때 물 표면 렌더링 사이의 차이를 보여줍니다. 차이는 미묘하지만 주로 병의 바닥 주변에서 볼 수 있습니다.



Internal Reflections 활성화: 켜짐

Internal Reflections 활성화: 꺼짐

Caustics

Caustics 메뉴는 **Standard Surface** 쉐이더에 있습니다. Standard Surface 쉐이더의 이 스위치는 확산된 GI 광선이 '정' 반사 및 전송 광선을 '볼 수 있는지' 여부를 지정합니다(기본적으로 Standard Surface 쉐이더의 직접 및 간접 확산 조명만 확산 광선에서 계산됩니다).

아래 이미지는 커스틱이 있거나 없는 병과 액체의 렌더링 차이점을 보여줍니다.



커스틱 비활성화됨

커스틱 활성화됨

탁한 물

물을 탁하게 만들기 위해 'Transmission Roughness'를 0.4로 줄였습니다. 이 값을 높이면 렌더링 시간이 늘어납니다. 'Rayswitch' 쉐이더를 사용하여 렌더링 시간을 최소로 유지했습니다.

Rayswitch

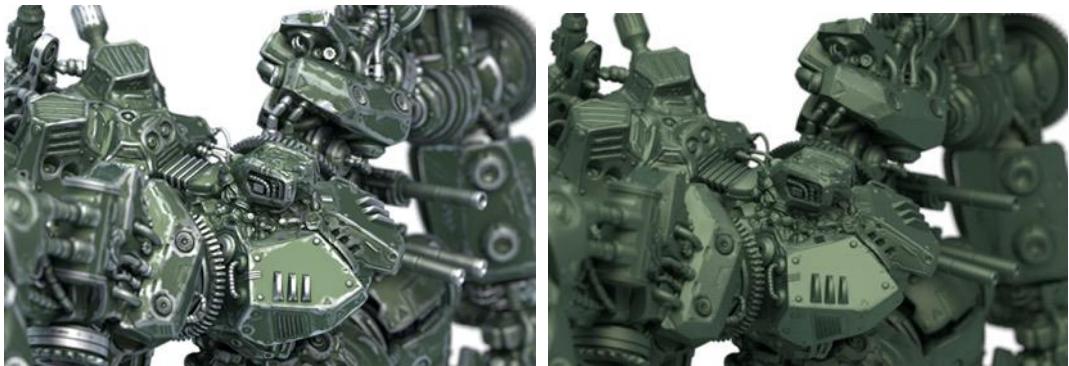
Rayswitch 쉐이더는 장면에서 계산된 광선 수를 줄이는 데 도움을 주어 품질을 저하시키지 않고 렌더링 시간을 향상시킵니다.

액체를 탁하게 만들려면 cloudy roughness 때문에 주요 광선이 굴절해야 합니다. 반면에 보조 광택, 그림자 및 확산 광선은 반사/투과 거칠기가 없는 두 번째 쉐이더 (Water_Secondary_Rays)를 사용합니다(렌더링 시간은 줄어듭니다).

아래 이미지들에서 볼 수 있듯이 렌더링 품질에는 약간의 차이가 있지만 렌더링 시간은 크게 단축되었습니다.



Curvature 쉐이더로 마모 효과 만들기



Curvature 쉐이더는 쉐이더에 디테일과 사실주의를 추가하는 쉬운 방법입니다. 이 짧은 자습서에서는 Curvature 쉐이더를 사용하여 마모 및 음영 처리 효과를 만들어 보겠습니다. 이것은 메시 모델에서 거친 표면의 가장자리들이 긁히는 부분을 표현하는 데 사용됩니다. 위의 그림에서는 페인트의 가장자리가 마모되어 밑에 있는 금속이 노출된 것을 볼 수 있습니다.

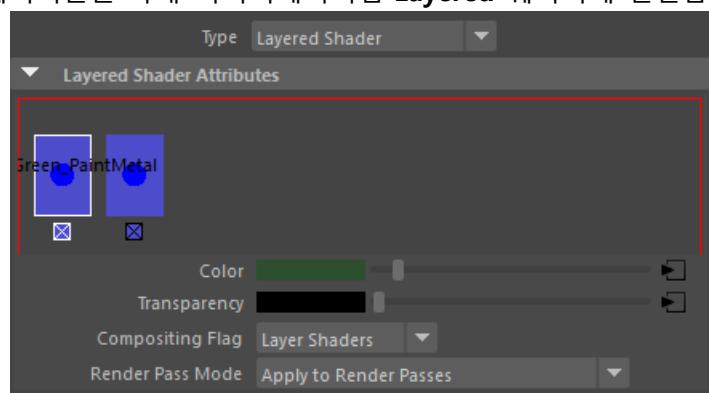
이 장면에서 사용되는 쉐이더는 하단 링크 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Wear+and+Tear+with+the+Curvature+Shader>)

Layer Standard Shaders

Layered 쉐이더는 두 개의 Standard 쉐이더를 레이어링하는 데 사용됩니다. 상위 녹색 페인트 레이어(녹색)에는 하나의 Standard 쉐이더를 사용하고 그 아래의 노출된 금속 표면을 표현할 때는 다른 Standard 쉐이더를 사용하겠습니다.

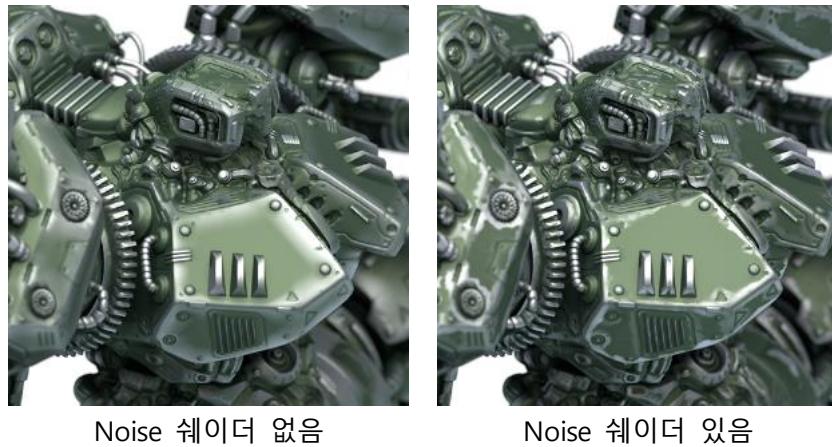
- Layered Shader를 생성하고 물체에 할당합니다.
- Standard Surface 쉐이더를 생성하고 이름을 **Green Paint**로 변경합니다.
- 다른 Standard Surface 쉐이더를 생성하고 이름을 **Metal**로 변경합니다.
- Standard Surface 쉐이더들을 아래 이미지에서처럼 Layered 쉐이더에 연결합니다.



Curvature 쉐이더

- Noise 쉐이더를 Curvature 쉐이더의 *radius*에 연결합니다(*bias* 및 *multiply*도 유효). 원하는 모습을 얻을 때까지 다양한 노이즈 설정으로 실험해 보십시오. Noise 쉐이더에 **Color Correct** 쉐이더를 연결하여 '마모된' 모양을 더 세밀하게 조정할 수 있다는 사실을 잊지 마십시오.

- **aiCurvature** 쉐이더를 Layered 쉐이더의 **Transparency** 속성에 연결합니다. 그러면 두 개의 Standard Surface 쉐이더 사이에서 혼합을 조절할 수 있습니다.



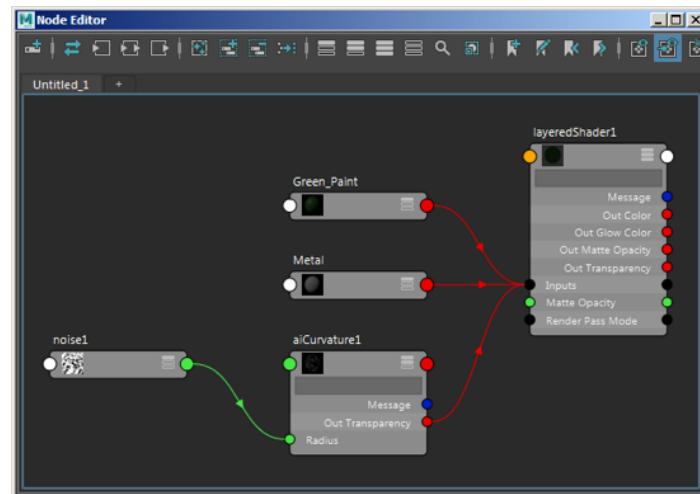
Bias

- 아래의 예제 이미지에서 **Curvature** 쉐이더의 **Bias** 파라미터가 그 효과를 더 잘 보여주기 위해 확대되고 과장되었습니다.



샘플

- **Curvature** 샘플의 수를 증가시키면 노이즈가 감소하고 더 나은 품질 결과를 얻을 수 있습니다.

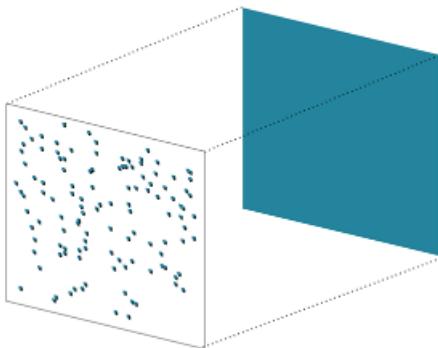


최종 쉐이더 네트워크

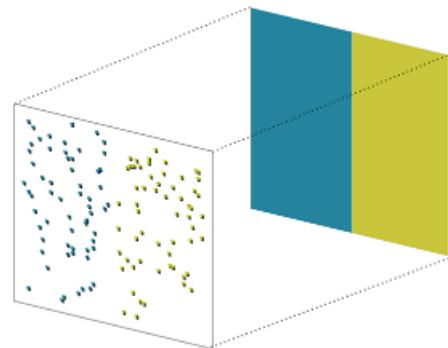
노이즈 제거

노이즈 및 샘플링

간단한 색상으로 구성된 장면에서 이 사진의 평균 색상이 궁금하다면 적은 수의 샘플을 사용하고 결과를 평균화 함으로써 어느 지점에 어떤 색상이 사용되었는지 쉽게 확인할 수 있습니다.

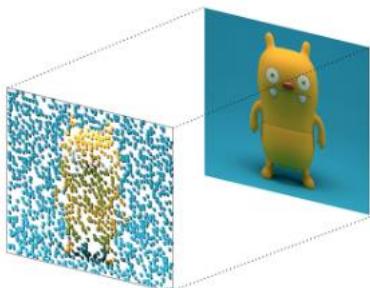


장면이 단색인 경우에는 몇 개의 샘플만으로도 올바른 평균을 낼 수 있습니다.

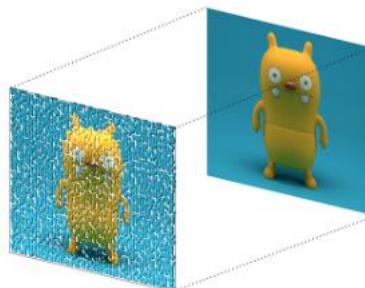


장면의 절반이 한 색상이고 나머지 절반이 또 다른 색상이라고 해도 샘플을 추가하면 우수한 평균을 낼 수 있습니다.

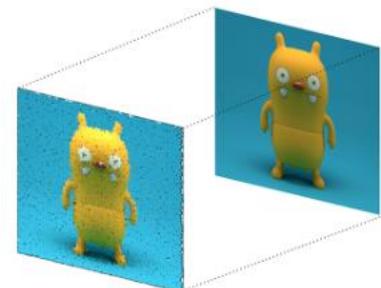
하지만 장면이 보다 복잡하고(예를 들어 캐릭터 포함) 몇 가지 샘플만 사용한다면 이들 중 절반은 노란색, 나머지 절반은 파란색이 되는 경우도 있습니다. 때로는 모든 샘플이 하나의 색이나 다른 색이 될 수 있습니다. 때로는 장면의 일부만이 노란색 캐릭터임에도 불구하고 우연히 모든 샘플이 노란색 캐릭터가 되는 경우도 있습니다. 평균값을 향상시키는 방법은 카메라 샘플의 수(AA)를 증가시키는 것입니다.



3 Camera (AA) samples. Low quality.



4 Camera (AA) samples. Medium quality



5 Camera (AA) samples. Higher quality

카메라 샘플 수(AA)를 증가시키면 노이즈가 해결되고 이미지의 품질이 향상됩니다.

샘플링에 대한 자세한 정보는 하단 링크의 사이트에서 다운로드 할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Removing+Noise>)

노이즈 유형

렌더링에서 노이즈를 제거하는 첫 번째 단계는 노이즈의 원인을 파악하는 것입니다. 노이즈의 원인은 다음과 같습니다.

Insufficient Sampling:

- Motion Blur

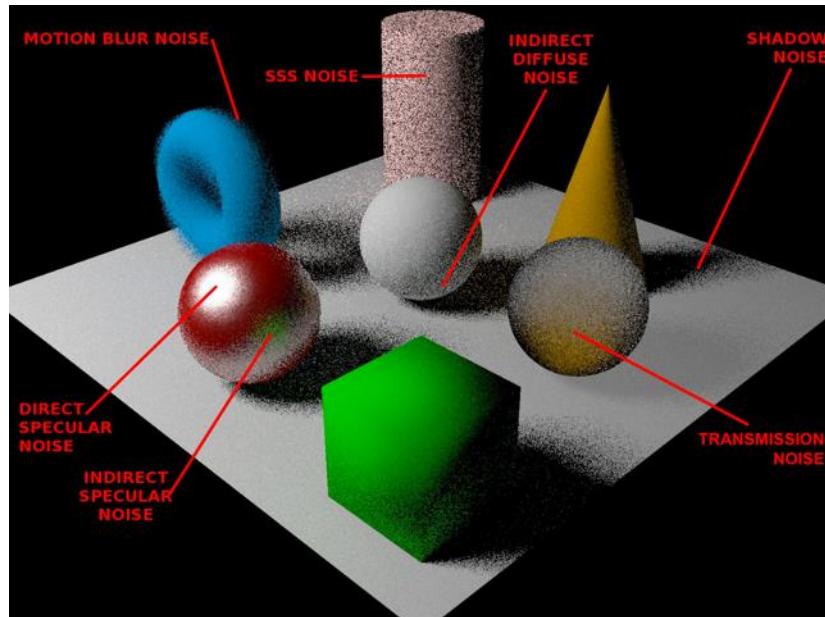
- Depth of Field
- Diffuse
- Specular
- Shadow
- Indirect Specular
- Transmission
- SSS
- Atmospheric Volume

기타:

- Fireflies이
- 비 에너지 보존 쉐이더, 네트워크 또는 설정.

노이즈는 거의 항상 불충분한 샘플링에서 유래하지만, 잘못된 광선에 대한 샘플링을 증가시키면 노이즈를 제거하지 않고 렌더링 시간을 늘릴 수 있습니다. 아티스트는 일반적으로 렌더링 시간이나 광선의 양이 제한되어 있기 때문에 이러한 광선을 최대한 효과적으로 할당하여 가장 효과적인 방법으로 노이즈를 최소화하는 것이 목표입니다. 따라서 DOF 노이즈를 제거하기 위해 카메라 샘플을 늘려야 하는 경우 렌더링 시간을 관리하기 위해 다른 설정을 낮춰야 합니다. 그러나 DOF 또는 모션 블러가 문제가 되지 않는 경우 Camera 샘플을 늘리면 다른 곳의 모든 노이즈가 해결될 수 있지만 불필요한 광선으로부터 렌더링 시간이 느려집니다.

노이즈 제거 절차는 워크플로우 다이어그램을 따르십시오.



Camera (AA) Samples: 1

Diffuse Samples: 1

Specular Samples : 1

Transmission Samples : 1

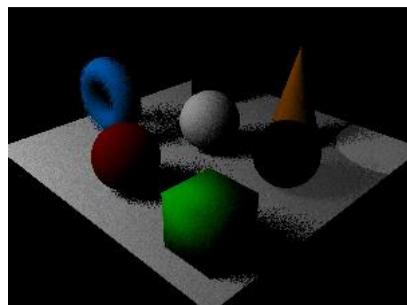
SSS Samples : 1

Light Samples : 1

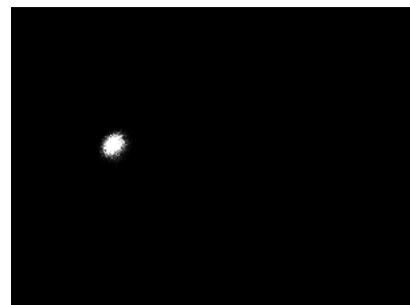
Atmospheric Volume Samples :

이것은 모든 샘플이 1로 설정되어 있는 매우 노이즈가 많은 장면입니다. 노이즈를 식별하기 위한 가장 효율적인 방법은 AOV를 렌더링하는 것입니다. 기능들을 켜거나 끄도록 설정을 수정하는 것은 시간이 추가로 소요되기 때문입니다.

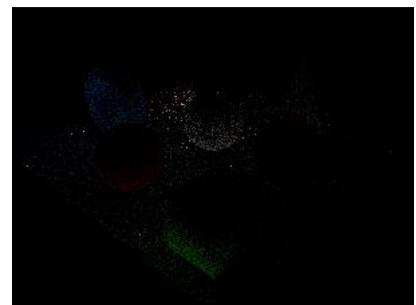
AOVs



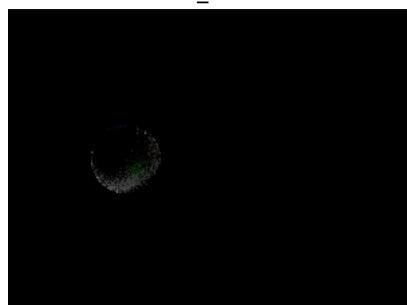
direct_diffuse



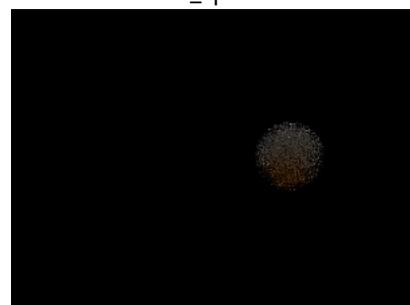
direct_specular



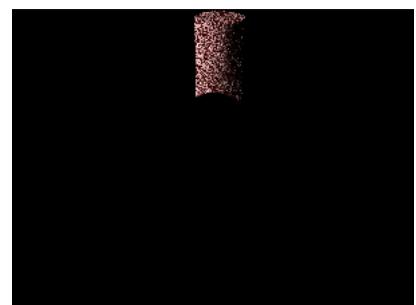
indirect_diffuse



indirect_specular



transmission



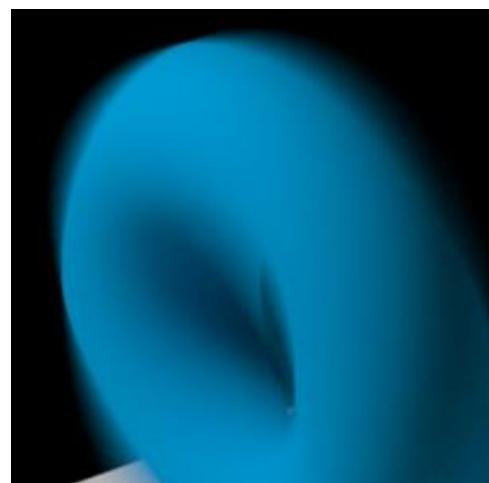
Sss

아래 예제에서 더 부드러운 옵션은 샘플 10으로 표시됩니다. 이것은 매우 높은 값이며 카메라(AA) 샘플과 다른 모든 샘플이 1로 설정되어 있기 때문에 노이즈가 많은 영역과 부드러운 영역을 대조하기 위한 예시 효과로 여기서만 사용됩니다. 일반적으로 카메라(AA) 샘플은 약 4-8이고 승수이기 때문에 다른 샘플에도 비슷한 값이 필요합니다.

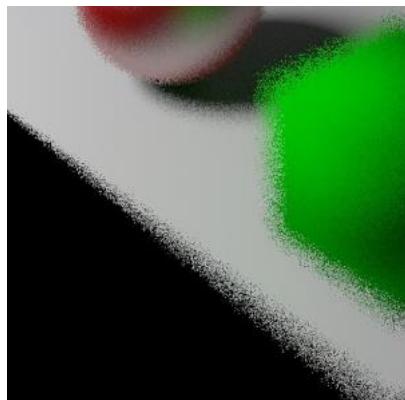
Motion Blur와 Depth of Field Noise



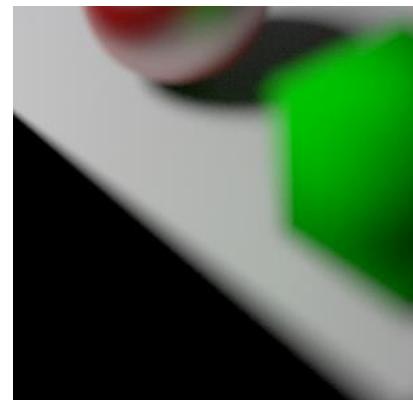
All Samples 1



Camera (AA) samples 10



Camera samples 1, All Other Samples 10

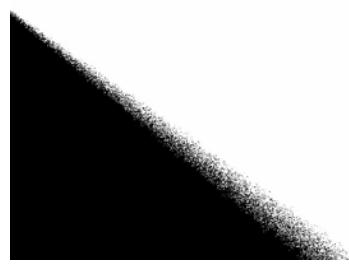


Camera samples 10, All Other Samples 1

모션 블러 노이즈는 움직이는 지오메트리의 흔적에 나타나고 DOF 노이즈는 초점이 맞지 않는 영역에 나타납니다. 둘 다 모두 알파 채널을 살펴봄으로써 노이즈 유무를 확인할 수 있습니다.



Motion Blur Alpha

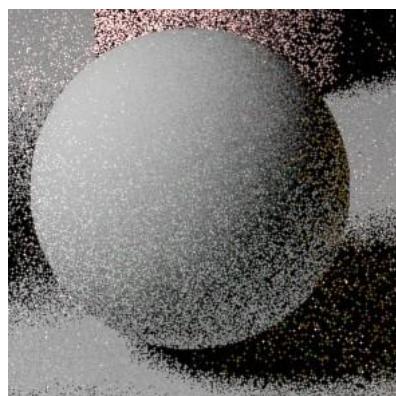


Depth of Field Alpha

모션 블러 및 피사계 심도 노이즈는 카메라 광선의 부족으로 인해 발생하므로 카메라(AA) 샘플을 늘려야 해결할 수 있습니다. 실제 카메라 샘플 수(AA)는 이 숫자의 제곱입니다. 4의 카메라(AA) 샘플 수는 16개의 광선을 비춥니다. 카메라(AA) 샘플을 늘리면 다른 샘플도 증가하는데, 이 숫자를 줄여야 보정됩니다.

카메라(AA) 샘플을 늘리면 렌더링 시간이 크게 변합니다. 모션 블러나 DOF의 문제가 아니라면 카메라(AA) 샘플은 다른 노이즈 유형을 수정하는 데 있어 마지막 고려 사항이 되어야 합니다.

Diffuse Noise



All Samples 1

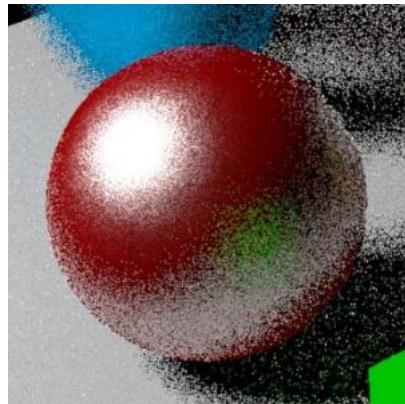


Diffuse Samples 10

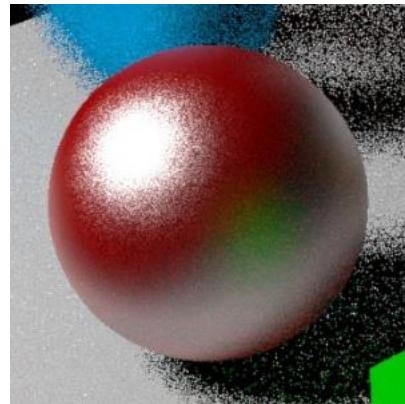
간접 Diffuse 노이즈는 아마도 가장 자주 처리하게 될 노이즈 유형일 것입니다. 이것이 원인인지 확인하는 가장 쉬운 방법은 간접 Diffuse AOV를 확인하는 것입니다. 다른 방법은 Diffuse Samples를 0으로 설정하여 간접 Diffuse를 끄는 것입니다. 이를 통해 노이즈가 사라진다면 간접 Diffuse이 원인이라고 할 수 있습니다.

Diffuse 샘플이 0보다 크다면 Diffuse 표면과 교차하는 카메라 광선이 간접 Diffuse 광선을 발사합니다. 광선이 반구형 스프레드 내에서 임의의 방향으로 방사됩니다. 환경으로부터 값 범위를 해결할 수 있는 충분한 광선이 없는 경우 노이즈가 발생합니다. **Diffuse** 샘플을 높이면 노이즈를 제거할 수 있습니다.

Specular Noise



All Samples 1

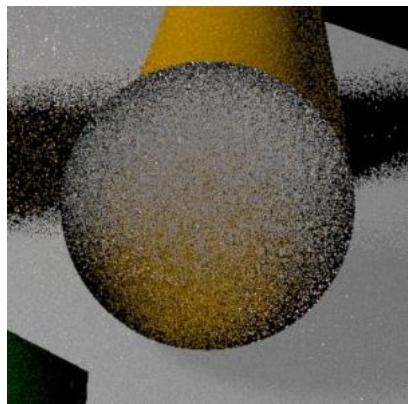


Specular Samples 10

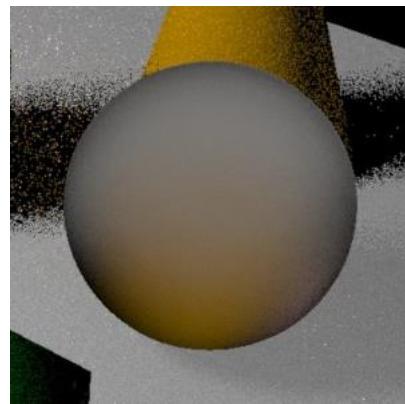
Specular Roughness 파라미터가 0보다 클 경우 간접 반사 노이즈가 발생합니다. 구체의 하단 오른쪽(상단 왼쪽이 아닌)에서 특히 뚜렷하게 나타납니다. 이것이 원인인지 확인하는 가장 쉬운 방법은 간접 반사 AOV를 확인하는 것입니다. 다른 방법은 **Specular** 샘플을 0으로 설정하여 블러 처리된 반사를 제거하는 것입니다. 이를 통해 노이즈가 사라진다면 간접 반사가 원인인 것입니다.

간접 반사 노이즈는 **Specular** 샘플의 부족으로 인해 발생합니다. 이를 샘플은 반사 BRDF에 의해 가중치가 적용된 반구에 대해 통합되어 있는 반사된 간접 광도를 계산할 때 방사되는 광선 수를 제어합니다. 광선의 정확한 수는 이 값의 제곱입니다. 이 숫자를 늘리면 간접 반사 노이즈가 감소합니다. 샘플링은 각 카메라(AA) 샘플에 대해 수행되므로 카메라(AA) 샘플과 Specular 샘플의 값이 모두 높으면 렌더링이 느려지는 경향이 있음을 기억하십시오.

Transmission Noise



All Samples 1

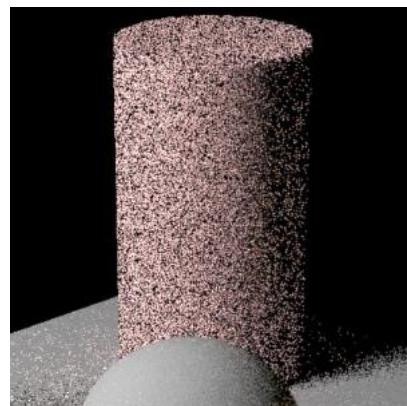


Transmission Samples 10

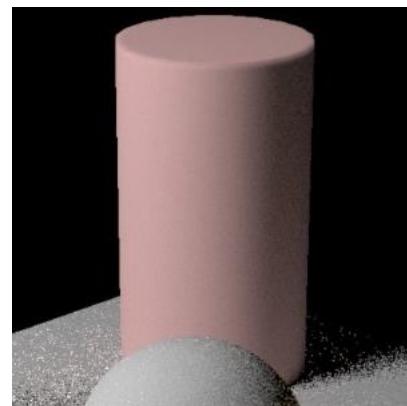
전송 노이즈는 specular Roughness 0보다 큰 투명 물체의 흐려진 굴절에서 눈에 띕니다. 이것이 원인인지 확인하는 가장 쉬운 방법은 전송 AOV를 확인하는 것입니다. 다른 방법은 **Transmission Samples**를 0으로 설정하여 블러 처리된 굴절을 제거하는 것입니다. 이를 통해 노이즈가 사라진다면 Transmission 원인인 것입니다.

전송 노이즈는 Transmission Samples의 부족으로 인해 발생합니다. Transmission Samples는 미세먼 기반의 광택이 있는 굴절 평가를 시뮬레이션하는 데 사용되는 샘플 수를 제어합니다. 광선의 정확한 수는 이 값의 제곱입니다. 이 숫자를 늘리면 Transmission을 사용하여 쉐이더 안의 노이즈가 감소합니다.

SSS 노이즈



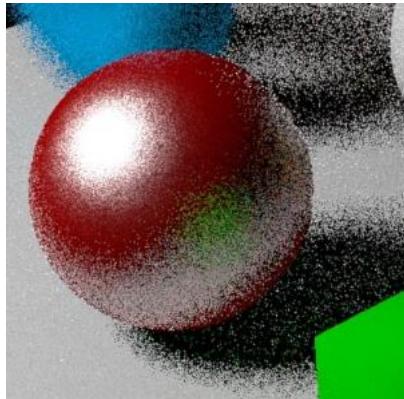
All Samples 1



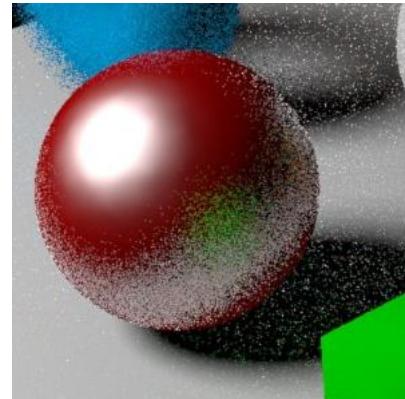
SSS Samples 10

SSS 노이즈는 SSS가 활성화된 Standard Surface 쉐이더를 사용하는 표면에서 발생합니다. 확인하려면 SSS AOV를 살펴보십시오. SSS 샘플을 높이면 SSS 노이즈를 제거할 수 있습니다.

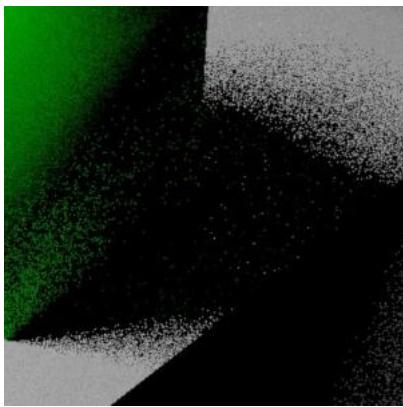
Direct Specular와 Shadow Noise



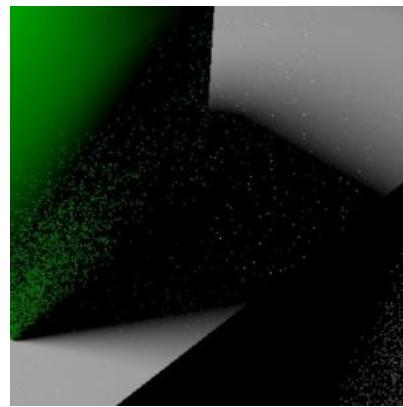
All Samples 1



Light Samples 10



All Samples 1



Light Samples 10

마찬가지로 그림자 노이즈는 간접 Diffuse 노이즈와 혼합되는 경우가 많습니다. 특히 반경이 큰 조명의 경우 그림자가 더 부드러워지기 때문에 그렇습니다. 직접 Diffuse AOV를 보고 그림자에 노이즈가 많은지 확인하십시오. 또는 Diffuse 샘플과 Specular 샘플을 0으로 설정하면 GI가 제거되어 직접 조명 효과를 없앨 수 있습니다. 간접 반사 노이즈로 오인될 수 있어서 직접 반사 노이즈를 식별하는 것은 까다로울 수 있습니다. 직접 반사는 빛 자체를 표면에 반사시키는 것으로, 색상과 강도에 기반하여 주변 물체의 반사로부터 구별하기 위해 무엇을 확인해야 하는지를 보여줍니다.

직접 반사 및 그림자 내의 노이즈는 조명 샘플의 부족으로 인해 발생합니다. 일반적으로 직접 반사 노이즈를 제거하려면 약간의 샘플이 필요하지만 그림자의 노이즈를 제거하려면 더 많은 샘플이 필요할 수 있습니다. 광원의 반지름이 클수록 그림자가 부드러워지므로 그림자 노이즈를 제거하려면 더 많은 샘플이 필요합니다.

노이즈는 높은 Specular Weight 값을 가진 쉐이더를 사용하는 얇은 지오메트리의 반사 하이라이트에도 나타날 수 있습니다. 카메라(AA) 샘플 수를 높이면 이 효과를 한 점으로 줄일 수 있습니다. 그러나 높은 카메라(AA) 샘플을 사용하더라도 앤티앨리어스 가공물이 나타날 수 있습니다. Specular Roughness 값을 높이면 이러한 유형의 노이즈를 줄이는 데 도움이 됩니다. 조명의 크기를 늘리고 강도를 줄여도 이러한 상황에서 도움이 될 수 있습니다.



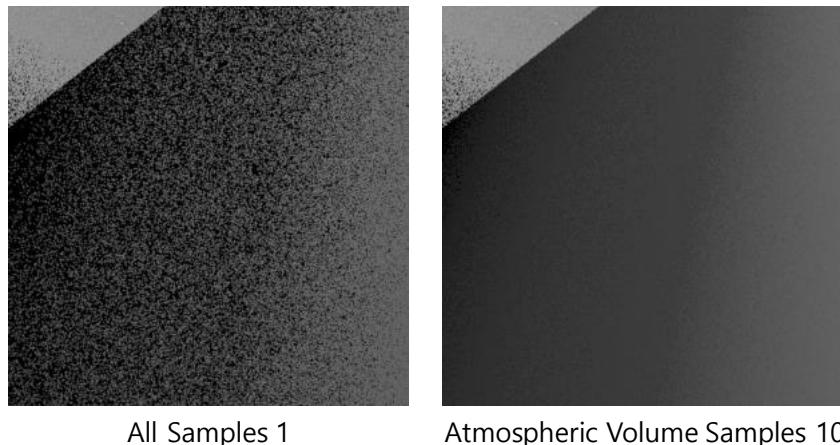
Specular Roughness 0.2



Specular Roughness 0.5

Specular Microfacet Distribution을 'GGX'로 변경해도 특정 상황에서 이러한 인공물이 감소합니다.

Atmospheric Scattering Noise



Atmospheric Volume 노이즈는 광선의 그림자 영역에서 발생하며 이 노이즈는 *Atmospheric Volume* 쉐이더의 샘플 부족이 원인입니다. 샘플들은 체적 밀도에 따라 분산됩니다. 샘플이 더 많으면 솔루션의 품질이 향상됩니다.

Fireflies와 Sparks

Fireflies이(밝은 '스파이크' 유형의 노이즈)는 일반적으로 반짝이는 낮은 조도의 광택 표면에 강한 빛이 반사되어 생성됩니다. 보통 수백 개의 샘플이 최종 픽셀 색상에 참여합니다. 그 중 하나가 높은 값의 반사 광선(강렬한 빛의 반사)이면 낮은 값을 가진 많은 샘플과 수천 개의 값을 가진 단일 샘플이 존재하게 됩니다. 이 단일 샘플은 전체 픽셀을 흰색으로 만듭니다("Fireflies").

Clamping

샘플의 고정은 이러한 종류의 노이즈를 제거해 줍니다. 높은 샘플 값을 고정하면 흩어진 단일 샘플이 희석되어 최종 색상에 큰 영향을 미치지 않습니다. 하지만 렌더러의 최종 동적 범위에 영향을 줍니다. 주의해서 사용하십시오.

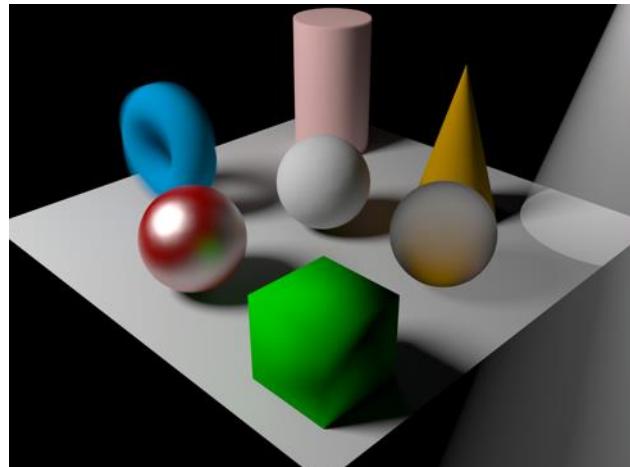
Light Decay Filter

Fireflies은 매우 밝은 곳에서 간접 Diffuse 조명으로 인해 발생할 수 있습니다. 특히 광원이 캐비닛이나 인클로저 등 근처 지오메트리에 매우 근접한 경우(또는 터치하는 경우)에 그 가능성이 높습니다. 빛 안의 역 square decay는 근처의 이러한 표면을 매우 밝게 만듭니다. 해결 방법은 시작 값이 매우 낮은 조명 decay 필터를 추가하여 빛에 매우 가까운 샘플을 피하는 것입니다. 이 역시 주의를 기울여 실행해야 합니다.

기타 고려사항

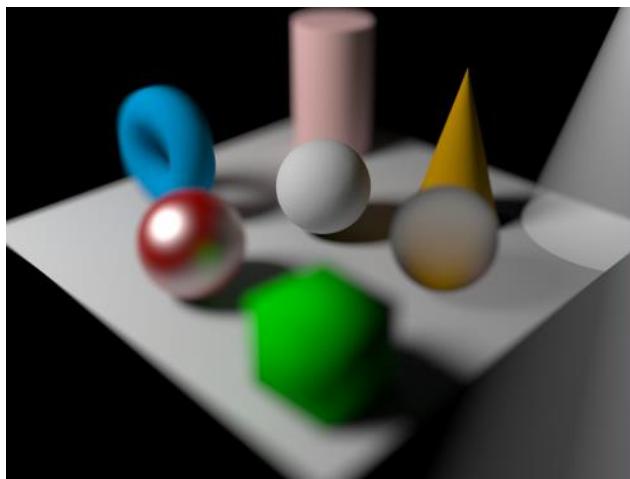
- 노이즈는 렌더링에서 볼 수 없는 것들이 원인입니다(카메라 뒤).
- 노이즈는 에너지가 보존되지 않는 쉐이더, 네트워크 또는 설정에 의해 유발되거나 악화될 수 있습니다.
- 노이즈를 줄이는 또 다른 방법은 원인을 제거하고 특수 조명으로 속이는 것입니다. 예를 들어 반사광으로만 얼굴을 비추는 캐릭터: 특정 반사광을 추가하는 것이 훨씬 더 노이즈가 적습니다.
- Arnold는 조명을 직접 샘플링하여 조명이 어디에 있는지 알 수 있는 경우에는 노이즈를 쉽게 제거할 수 있지만, 현장의 조명에 크게 영향을 주는 조명으로 태그가 지정되지 않은 밝은 "방향성" 패치가 있는 경우에는 문제가 발생합니다.

Clean Renders



DOF 없음

- Camera (AA) Samples: 7
- (Indirect) Diffuse Samples: 3
- (Indirect) Specular samples : 5
- Transmission Samples : 3
- SSS Samples : 3
- Light Samples : 2
- Atmosphere Volum Samples : 2



DOF 있음

- Camera (AA) Samples: 16
- (Indirect) Diffuse Samples: 1
- (Indirect) Specular samples : 2
- Transmission Samples : 1
- SSS Samples : 1
- Light Samples : 1
- Atmosphere Volum Samples : 1

렌더러에서 노이즈를 식별하려고 할 때 AOV를 렌더링하고 확인하는 것이 유용합니다. 이를 통해 노이즈 유형을 분리하고 관련 샘플을 조정할 수 있습니다. 일반적인 노이즈 유형의 순서를 식별하고 우선 순위를 정하는 일반적인 지침은 아래 표를 참조하십시오. 이것은 일반적인 가이드일 뿐이며 장면마다 다를 수 있습니다.

Noise Visible in:	Samples to Adjust
Alpha Channel	Camera (AA) samples
Indirect Diffuse	Diffuse samples
Direct Specular (specular noise)	Light samples
Direct Diffuse (shadow noise)	Light samples
Indirect Specular	Specular samples
Refraction	Refraction samples
SSS (direct and indirect)	SSS (direct and indirect) samples
Volume	Volume samples (note that there are also volume samples in lights too)

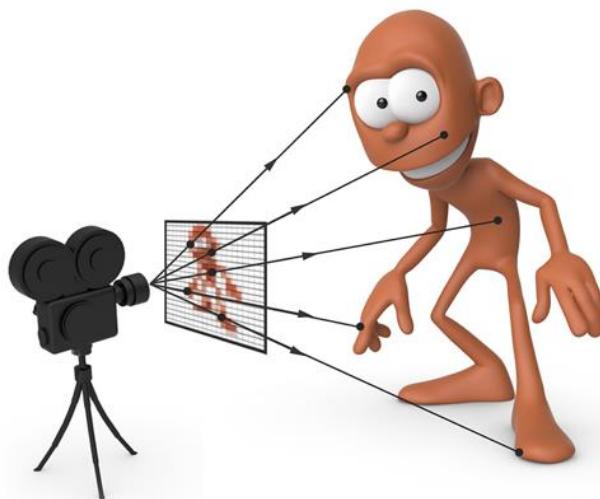
다음은 노이즈 관련하여 자세히 알아보겠습니다.

- [샘플링이란 무엇인가요?](#)
- [노이즈 제거 워크플로우](#)
- [Fireflies \(반딧불 노이즈\) – 보트 장면](#)
- [Volume – UFO 장면](#)

샘플링이란 무엇인가요?

Arnold와 같은 렌더러로 작업할 때는 광선 추적 렌더링의 핵심 원리에 대한 기본적인 이해가 필요합니다. 가상 장면의 사실적인 이미지를 생성하려면 장면 내에서 광원부터 카메라까지 빛의 전파를 시뮬레이션해야 합니다. 각 이미지 픽셀의 색상을 파악하기 위해 Arnold는 장면 지오메트리에서 쉐이더, 라이트 등의 정보를 수집하고 픽셀을 통해 보이는 물체를 광원에 연결하는 무작위적인 빛 전달 경로 여러 개를 추적합니다. 이 과정을 '샘플링'이라고 합니다. 생성되는 이미지의 품질은 주로 각 픽셀에 대해 생성된 경로 또는 샘플의 수에 따라 달라집니다.

이미지를 렌더링 할 때 Arnold는 장면을 조사하여 각 픽셀에 대한 색상 값을 결정해야 합니다. Arnold는 카메라의 위치에서 여러 개의 광선을 발사하여 장면 내의 물체에 닿을 때까지 이 작업을 수행합니다. 광선이 물체에 닿을 때마다 궁극적으로 해당 객체에 대한 정보(예: 색상)를 반환하는 계산을 수행합니다. 이 프로세스는 기본적으로 가상 카메라 이미지 평면의 픽셀을 '샘플링'하는 것으로 설명할 수 있습니다.



각 픽셀에서 너무 적은 샘플로 계산된 렌더링은 노이즈가 발생할 수 있습니다. 픽셀당 샘플 수를 늘리면 실제 장면을 더 잘 표현할 수 있도록 노이즈가 적은 이미지를 얻을 수 있습니다. 아래 이미지는 장면에서 카메라(AA) 샘플의 수를 늘릴 경우를 비교해서 보여줍니다. 카메라(AA) 샘플을 늘리면 앤리어싱 노이즈가 감소할 뿐만 아니라 조명에서 노이즈를 줄이는 보조 광선 샘플이 추가됩니다.



1개의 카메라(AA): 픽셀당 $1 \times 1 = 1$ 개의 샘플(이미지에 노이즈가 생김)



6개의 카메라(AA): $6 \times 6 = 36$ (노이즈가 감소됨)

Camera (AA) 샘플링

그렇다면 카메라 샘플링이란 무엇일까요? 기본적으로, 여러 광선은 카메라로부터 나와서 렌더링된 화면 창의 각 필수 픽셀을 통해 장면으로 들어갑니다. 이를 '기본 광선'이라고 하지만, 렌더링 뷰로부터 비춰지기 때문에 '눈' 또는 '카메라 광선'(카메라 (AA) (앤티앨리어싱))이라고 부르는 경우도 있습니다. 때때로 '픽셀 샘플'이라고도 합니다.

카메라(AA) 값은 픽셀 수퍼 샘플링 속도 또는 카메라로부터 추적되는 픽셀당 광선 수를 제어합니다. 샘플 수가 높을 수록 앤티앨리어싱 품질이 우수하지만 렌더링 시간은 더 오래 걸립니다.

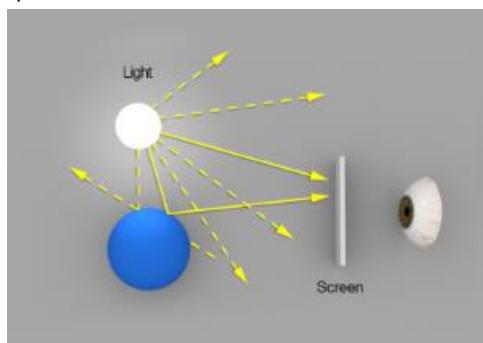
이 샘플링 속도 각각에 대해 실제 샘플 수는 입력 값의 제곱이므로 이 프로세스는 선형이 아닙니다. 예를 들어, 카메라(AA) 샘플이 3이면 $3 \times 3 = 9$ 샘플이 앤티앨리어싱에 사용된다는 의미입니다. Diffuse 샘플이 2인 경우에는 Diffuse GI에 $2 \times 2 = 4$ 샘플이 사용됩니다. 이것은 다른 값들에도 적용됩니다.

이러한 AA 샘플은 1차 광선(또는 픽셀 샘플)이라고 묘사되어 렌더링되는 이미지의 전체 품질을 결정합니다. AA 샘플을 늘리면 이미지의 전체 품질이 향상되지만 모든 샘플링 속도가 높아지므로 낭비일 수 있습니다. 노이즈의 원인이 장면에서 식별되면 샘플링 값들을 설정하여 특정 광선만 증가시키는 것이 더 효율적입니다. 예를 들어, 모션 블러가 많은 장면이 있으면 더 큰 카메라(AA) 샘플이 필요하므로 다른 샘플 값들을 줄일 수 있습니다. 마찬가지로, 증가하는 카메라(AA)는 간접 조명에도 도움이 될 수 있으므로 많은 확산 샘플이 필요하지는 않습니다. 그러나 캐릭터의 피부 노이즈를 줄이려면 SSS 샘플을 더 높이는 것이 효율적이며 카메라(AA) 샘플에는 그렇게 높은 값은 사용하지 않는 것이 좋습니다.

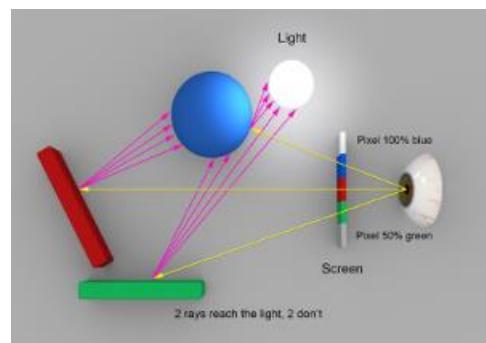
Light Sampling

직접 조명은 조명을 다루는 광선으로 설명할 수 있습니다. 이들 광선은 장면의 한 위치에서 다양한 광원을 향해 이동합니다. 이 광선은 표면이 그림자에 있는지 여부를 파악하고 그렇지 않은 경우 조명 정보를 계산할 수 있습니다.

조명으로 인해 발생하는 노이즈는 진단하기 어려울 수 있습니다. 특히 넓은 지역 또는 크기의 조명인 경우 그렇습니다. 이 경우에는 간접 확산 노이즈로 오인될 수도 있습니다. 그렇기 때문에 다양한 노이즈 광선 유형을 테스트해야 할 필요성이 더 높아집니다. 그림자 노이즈의 경우, Arnold 렌더링 설정에서 그림자 무시 기능을 간단하게 전환함으로써 노이즈 문제를 완전히 해결할 수 있습니다. 아래 이미지들은 Arnold에서 어떻게 조명이 추적되는지를 보여줍니다.



일부 광선은 물체에 닿지만 그렇지 않은 광선
도 있습니다.



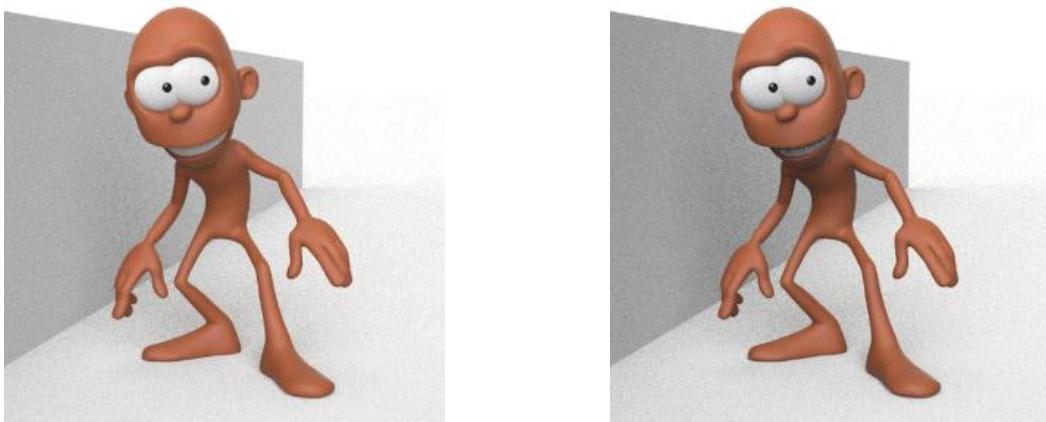
Arnold에서 조명 샘플링

앞에서 설명한 동일한 노이즈 문제가 직접 광선과 간접 광선에서도 발생할 수 있습니다. 직접 광선의 노이즈는 대개 반사 하이라이트 또는 영역 조명의 크고 부드러운 그림자에 나타납니다. 이 경우에는 조명 샘플의 수를 증가시켜야 합니다. 아래 이미지는 빛 샘플의 수를 1에서 3으로 늘릴 때 그림자 노이즈의 차이를 보여줍니다.



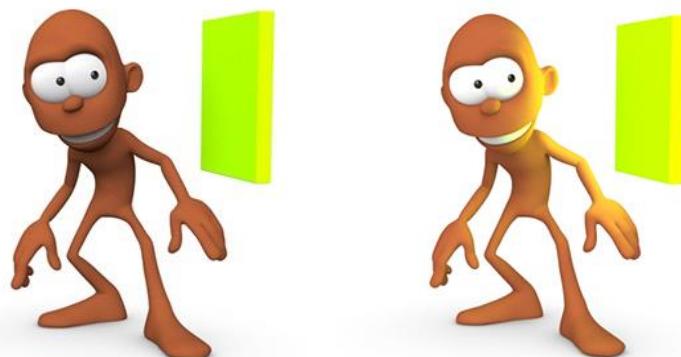
1개의 조명 샘플. 매우 노이즈가 많습니다. 3개의 조명 샘플. 노이즈가 개선되었습니다.

장면에서 노이즈를 평가하고자 할 때는 한 가지 노이즈 원인을 다른 노이즈 원인과 분리하는 것이 유용할 수 있습니다. 'Diffuse' 광선 (0)을 비활성화하면 어떤 유형의 노이즈가 있는지 식별하는 데 도움이 됩니다. 장면에서 간접 확산 광선의 수를 최적화하려고 시도할 때 Diffuse 광선의 수를 늘리는 것이 아니라 (렌더링 시간이 각 Diffuse 샘플마다 두 배가 됨) 노이즈의 원인을 먼저 격리해야 합니다.



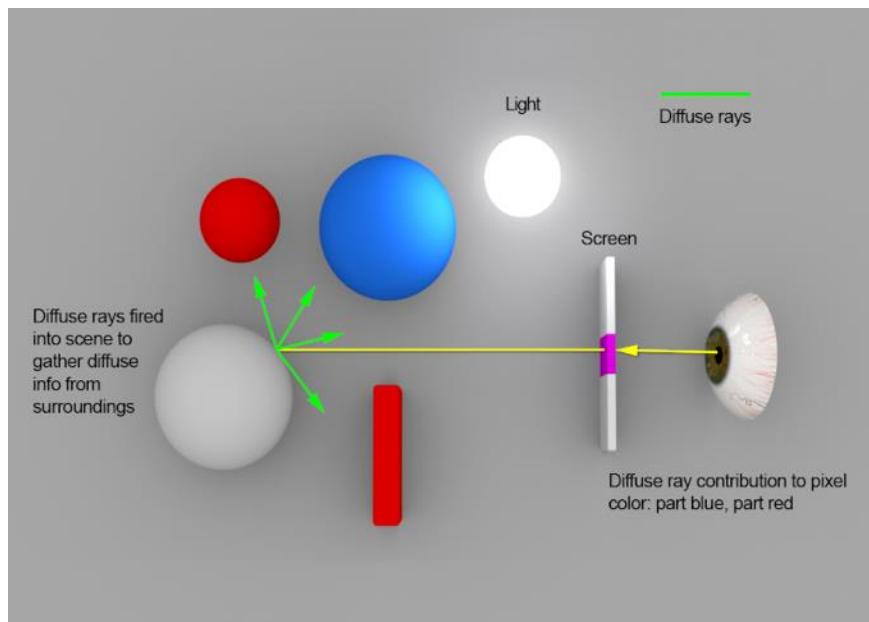
조명 노이즈와 Diffuse 노이즈(구분이 어려움). 0으로 설정된 Diffuse(비활성). 빛의 노이즈를 더 쉽게 식별할 수 있습니다.

Diffuse Ray Sampling



녹색 방출 큐브에서 나오는 간접(확산) 조명.

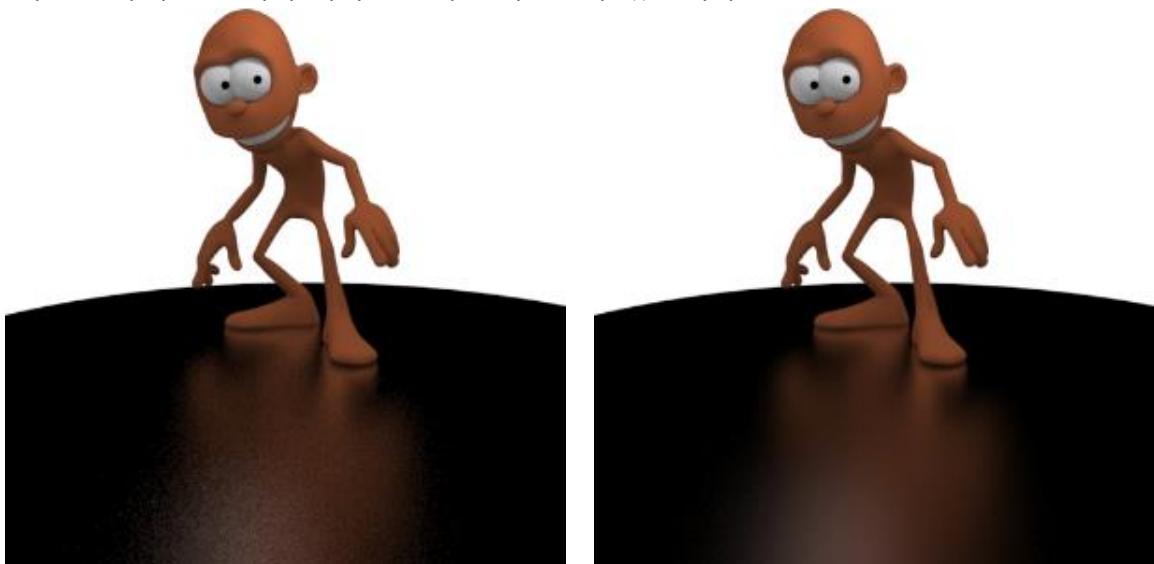
Indirect Diffuse 광선은 물체 및 표면 쉐이더와 상호 작용하는 광선입니다. 따라서 광선은 장면에서 물체에 지정된 쉐이더에 의해 결정된 방향으로 이동합니다. 전송 광선은 개체를 통해 이동합니다. 광선이 거울과 같은 반사체에 닿으면, 들어오는 광선에 의해 결정된 방향으로 해당 지점으로부터 하나의 반사 광선만 추적됩니다. 반대로, 확산 광선은 접촉 지점 주변의 반구에서 무작위로 샘플링됩니다.



Arnold 렌더링에서 확산 광선이 전파되는 방식을 보여주는 다이어그램

Specular Ray 샘플링

광택 있는 반사면을 렌더링할 때는 확산 광선에서처럼 노이즈도 문제가 될 수 있습니다. 밝은 핫스팟은 간접적인 반사 샘플에서 노이즈를 유발할 수 있습니다. 일례로, 작고 밝은 빛의 넓은 정반사가 있습니다. 반사 하이라이트에서 노이즈가 문제라면 원인이 직접 광선이 아닌 보조 광선 유형(예: 반사)인지 확인해야 합니다. 이것은 GI_diffuse_depth, GI_specular_depth를 0으로 설정하면 쉽게 달성할 수 있습니다(이것은 기본적으로 모든 전역 조명을 끔). 여전히 노이즈가 존재한다면 그것이 조명 모델의 직접적인 반사 요소라는 것을 알 수 있습니다. 대부분의 경우 반사 샘플의 수를 높여서 이러한 문제를 해결할 수 있습니다.

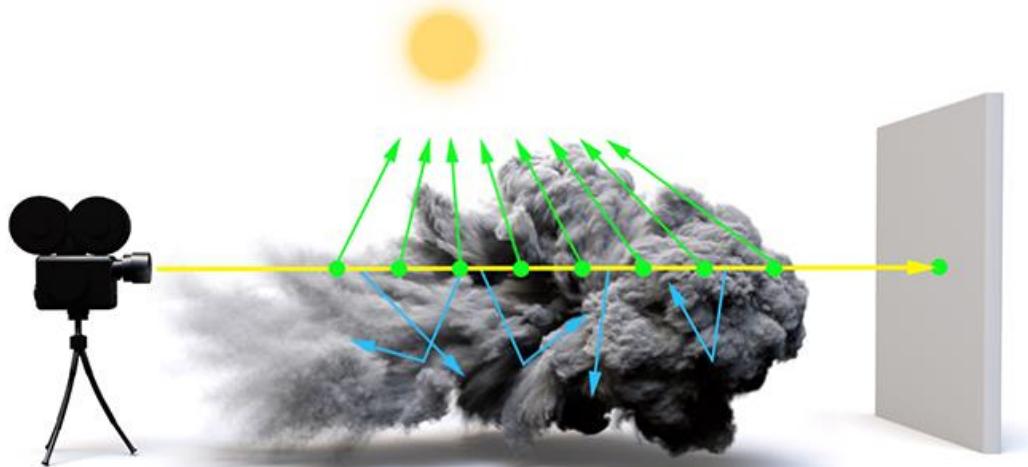


1 (부족한 샘플)

4 (충분한 샘플, 노이즈 감소)

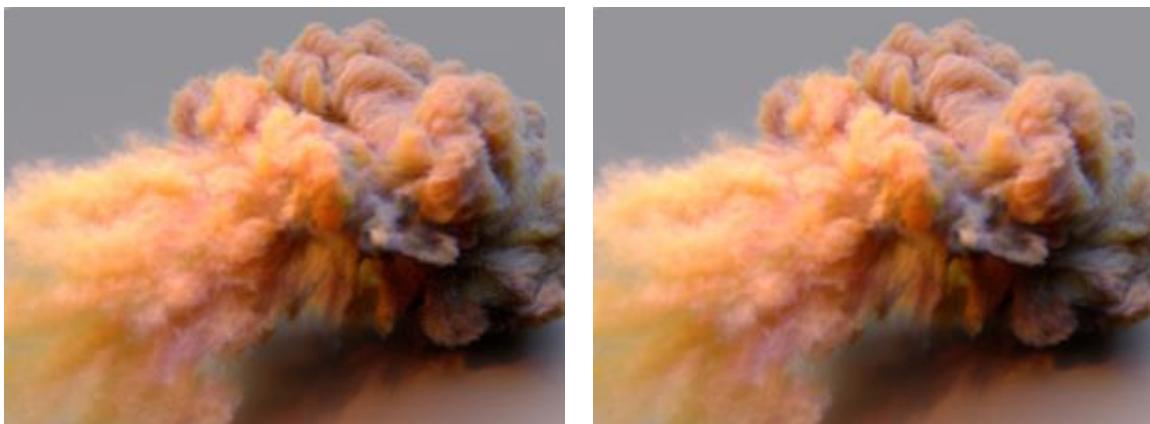
Volumes

볼륨을 샘플링 경우, 직접(AA) 광선은 볼륨을 통과할 때 여러 번 볼륨을 샘플링합니다. 간접 광선(Volume Indirect)도 비슷한 방식으로 작동하는데, 광선이 체적 오브젝트를 통해 '단계별'(볼륨 스텝 크기)로 통과하면서 여러 번 보내집니다. 따라서 볼륨 샘플링이 일반적으로 표면을 샘플링하는 것보다 계산이 오래 걸립니다.



볼륨 샘플링. 직접 조명(녹색). 간접 조명(파란색)

광선이 이동하는 각 단계에서 쉐이더를 평가하고 볼륨의 밀도를 누적합니다. 이러한 밀도 값이 볼륨 전체에서 보다 불규칙하게 되면 인근 광선들이 서로 매우 다른 값을 계산할 수 있게 되어 렌더링에 노이즈가 생길 수 있습니다. 이 경우 더 많은 조명 볼륨 샘플을 사용하여 렌더링하거나 볼륨 내의 스텝 크기를 줄여야 할 수도 있습니다. 샘플링 값이 낮더라도 볼륨이 있는 깨끗한 이미지를 렌더링할 때는 비용이 많이 듭니다.



Light volume samples: 1

Light volume samples: 6

볼륨을 렌더링할 때는 광선 marching, 직접 조명, 간접 조명 샘플링, 이 세 가지를 고려해야 합니다. 이들은 모두 각자 다른 샘플링 설정을 가지고 있습니다.

Step Size

광선 marching 프로세스의 `step_size`입니다. 샘플링 속도가 너무 낮으면 볼륨 렌더링이 볼륨의 작은 세부 사항을 놓치게 되어 너무 얇게 되거나 화산 구름과 같은 '고체' 부피의 정확한 '표면' 위치를 잘못 추정하게 될 수 있습니다. 불량한 스텝 크기는 눈덩이 효과를 가지고 있기 때문에 나머지 샘플링 과정에서 노이즈를 발생시킬 수 있

습니다. 알파 채널을 보거나 볼륨에 방출을 추가하고 모든 조명을 꺼서 테스트하는 것이 매우 쉽습니다. 원하는 AA 설정에서 방출/알파 채널에 노이즈가 있으면 `step_size`가 너무 큼 것입니다.



Step 크기가 올바름



Step 크기가 너무 크면 인공물이 알파에 나타남.

스텝 크기는 실제로 월드 공간이 아닌 오브젝트 공간에 있습니다. 이는 볼륨의 변환을 스케일링 할 때 샘플링 품질이 동일하게 유지되도록 하기 위한 것입니다. 그러나 불합리하게 큰 스텝 크기는 피해야 합니다. 그렇지 않으면 볼륨이 얇게 나타나고 색이 바래집니다. 아래의 예제 이미지에서는 이 효과를 과장하기 위해 볼륨을 최대 100단위까지 확장했습니다(1/25/50은 이 볼륨의 크기에 비례함).



1



25



50

시각적으로 눈에 띄는 인공물이 발생하려면 스텝 크기가 가능한 한 커야 합니다. Step Size가 작으면 렌더링 시간이 늘어납니다. 예를 들어, 스텝 크기가 0.1이고 월드 공간에서 볼륨이 10 단위로 큰 경우, 약 100개의 기본 샘플이 있으므로 볼륨 쉐이더가 100번 호출됩니다.

Direct Lighting

스텝 크기가 올바르면 직접 또는 간접 조명으로 인해 노이즈가 발생할 수 있습니다. 직접 조명은 감지하기가 상당히 쉽습니다. 간접 조명을 꺼야 하고, 렌더링에 노이즈가 있다면 각 조명을 한 번에 하나씩 렌더링하여 노이즈의 원인이 되는 조명을 발견한 다음 `volume_samples`를 표시해야 합니다. `volume_samples`가 증가해도 직접 조명 노이즈가 현저하게 감소하지 않는다면 `step_size`가 너무 작거나 조명의 상호 작용을 샘플링하기가 힘든 것일 수 있습니다(높은 이방성 볼륨, 심하게 텍스처링된 `mesh_lights`, 2차 감쇠 등). 이 경우에는 클램핑 또는 필터링을 통

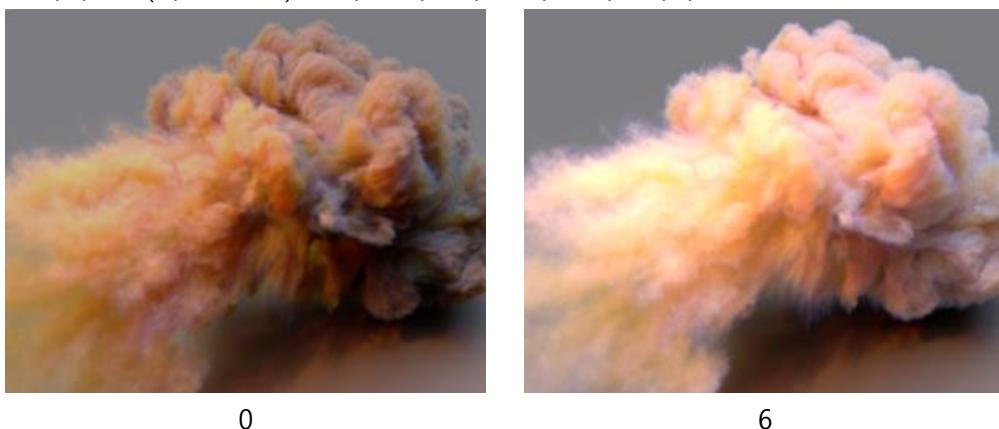
해 샘플을 제거하는 것 이외에는 해결 방법이 없습니다.

Indirect Lighting

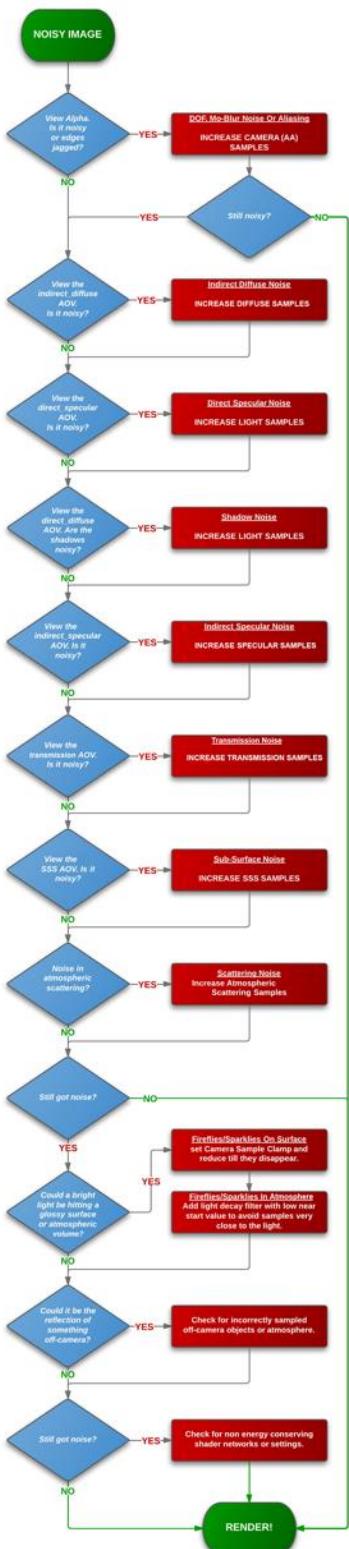
간접 노이즈 역시 식별이 쉽습니다. 노이즈가 많은 볼륨의 GI_volume_depth를 비활성화해야 하며 렌더링에 더 이상 노이즈가 없다면 원인은 간접 샘플링입니다. 품질을 개선해야 하는 경우 volume_indirect_samples 수를 늘려야 합니다. 이 값을 높이면 노이즈가 줄어들지 않으므로 조명 상호 작용을 샘플링할 때 노이즈가 문제가 될 수 있습니다(다중 산란, 이방성 볼륨, 볼륨에 포함된 빛 등). 그리고 앞서 언급했듯이 클램핑 또는 필터링을 통해 샘플을 제거하는 것 이외에는 해결 방법이 없습니다.

Volume ray depth

볼륨 광선 깊이를 늘리면 볼륨 모양이 크게 달라질 수 있습니다. 그러나 볼륨 광선 깊이를 늘리면 볼륨 내의 다중 산란 횟수가 증가하므로(기본값은 0) 렌더링 시간이 크게 늘어납니다.



노이즈 제거 워크플로우



옆의 다이어그램은 장면의 노이즈 유형과 이를 개선하기 위해 사용할 샘플링 파라미터를 식별하는 데 유용한 방법을 보여줍니다.

Fireflies (반딧불 노이즈) - 보트 장면



Physical Sky의 햇빛에 의한 반딧불 노이즈

이 장면은 바다를 표현하는 평면상의 보트 모델로 구성됩니다. 두 모델 모두 Standard Surface 쉐이더가 지정되어 있습니다. 위 이미지에서 볼 수 있듯이 이 장면에는 많은 '반딧불'이 있습니다. 이 상황에서 반딧불이들은 SkyDome 조명에 연결되어 있는 Physical Sky 쉐이더의 밝은 태양 원판에 의해 생깁니다. 이러한 종류의 노이즈는 Render Settings에서 샘플의 수를 늘리는 것만으로는 제거할 수 없습니다. 하지만 이러한 노이즈를 해결할 수 있는 몇 가지 방법이 있는데, 아래에서 살펴보겠습니다.

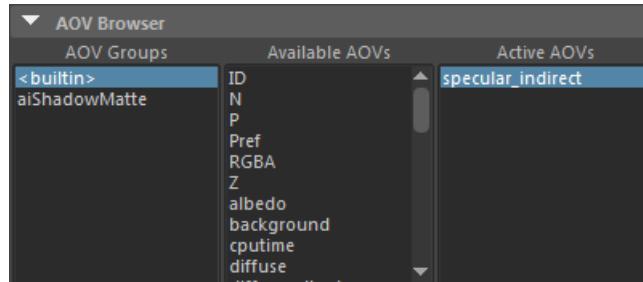
장면은 하단 링크의 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Fireflies+-+Boat+Scene>)

AOV

반딧불이 어디에서 오는지 확인할 수 있는 한 가지 방법은 AOV를 사용하여 장면을 렌더링하는 것입니다.

- Render Settings 창을 열고 AOV 탭을 선택합니다. **AOV Browser**에서 **Builtin**을 선택합니다. **Available AOVs**에서 *indirect_specular*를 두 번 클릭합니다. 그러면 **Active AOV** 아래에 새로운 AOV가 나타납니다.
- 장면을 렌더링하고 AOV를 열거나 **Arnold RenderView** 창에서 확인합니다.



반딧불이 나타나는 원인은 다음 두 가지입니다.

1. 약간의 반사가 있는 보트의 쉐이더
2. 약간의 반사가 있는 바다의 쉐이더

이러한 AOV를 렌더링하면 반딧불이들이 확실하게 눈에 보입니다.



Clamp Max Value 10 (기본 값)



Indirect Specular AOV (보트에서 보이는 반딧불)

Specular Roughness

일부의 경우 **Specular Roughness**를 증가시키면 반딧불이 감소할 수도 있습니다. 더 크고 부드러운 반사 하이라이트가 효과적으로 생기기 때문입니다. 보트에 지정된 Standard Surface 쉐이더의 Specular Roughness를 높이면 반딧불이의 수가 감소합니다. 하지만 이러한 경우에는 이 방법을 사용해도 반딧불이 여전히 남아 있습니다.



Low Specular Roughness 값



Specular Roughness: 1. 반딧불이 감소했지만 여전히 확실하게 보입니다.

Specular Samples

바다 쉐이더의 표면이 거울과 흡사하기 때문에 태양으로부터 매우 높은 Specular 값이 Specular 바다 표면에 반사되고 있습니다. 이렇게 높은 Specular 값은 샘플링이 상당히 어렵기 때문에 이러한 상황에서는 **Specular Sample**의 수를 늘린다고 해서 반딧불이 감소하지 않습니다.



2



4

Rayswitch

더 나은 방법은 rayswitch 쉐이더를 사용하여 Specular 및 확산 광선에 대해 더 간단한 쉐이더를 지정하는 것입니다.

- 보트에 **RaySwitch** 쉐이더를 지정합니다.
- **Standard Surface** 쉐이더를 RaySwitch 쉐이더의 **Diffuse** 및 **Specular** 광선에 연결합니다. **Specular Weight**가 0으로 설정되어 있어야 합니다. 그래야 반사 및 확산 광선이 반딧불의 근원인 쉐이더의 반사 요소들을 평가하지 않게 됩니다.
- 보트의 Standard Surface 쉐이더(**Specular Weight**가 있는)를 나머지 속성(카메라, 정반사, 전송과 그림자)에 연결합니다.

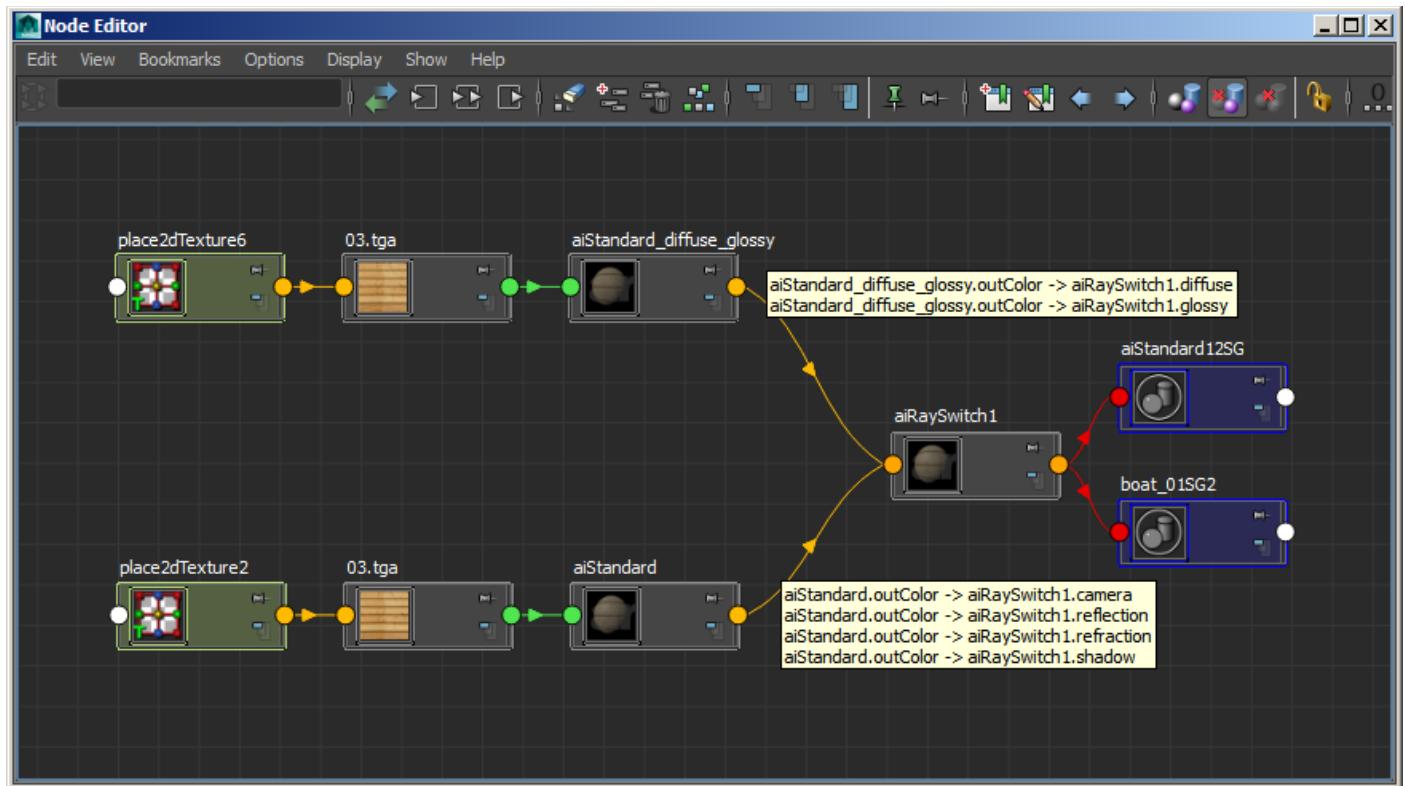
이제 AOV를 살펴보면 반딧불이 보이지 않습니다.



Indirect Specular AOV



Rayswitch Beauty (반딧불 사라짐)



Rayswitch 쉐이더에 연결된 2개의 Standard Surface 쉐이더

Clamp Samples Values

또 다른 옵션은 장면에서 픽셀의 출력 값을 고정하는 것입니다. 이 컨트롤은 픽셀 샘플을 장면의 모든 쉐이더에 대해 지정된 최대 값으로 고정시킵니다.

아래 이미지들은 반딧불에 대한 고정 효과를 보여줍니다. 1 값은 반딧불을 완전히 제거하지만 렌더링에서 동적 범위도 제거합니다.



Clamp Max Value 5



Clamp Max Value 1 (반딧불이들이 사라짐)

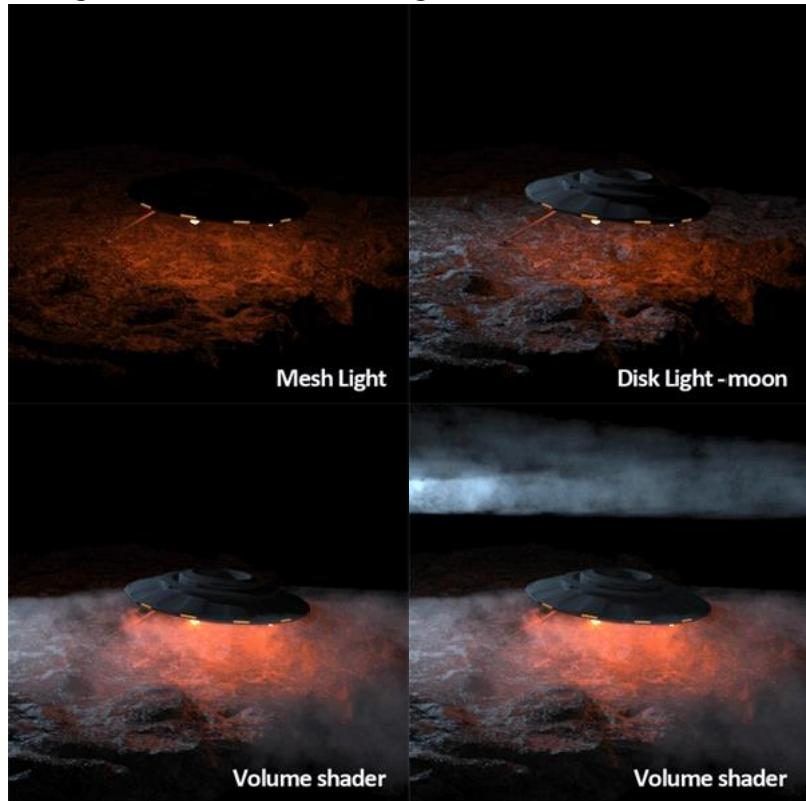
Volume - UFO 장면

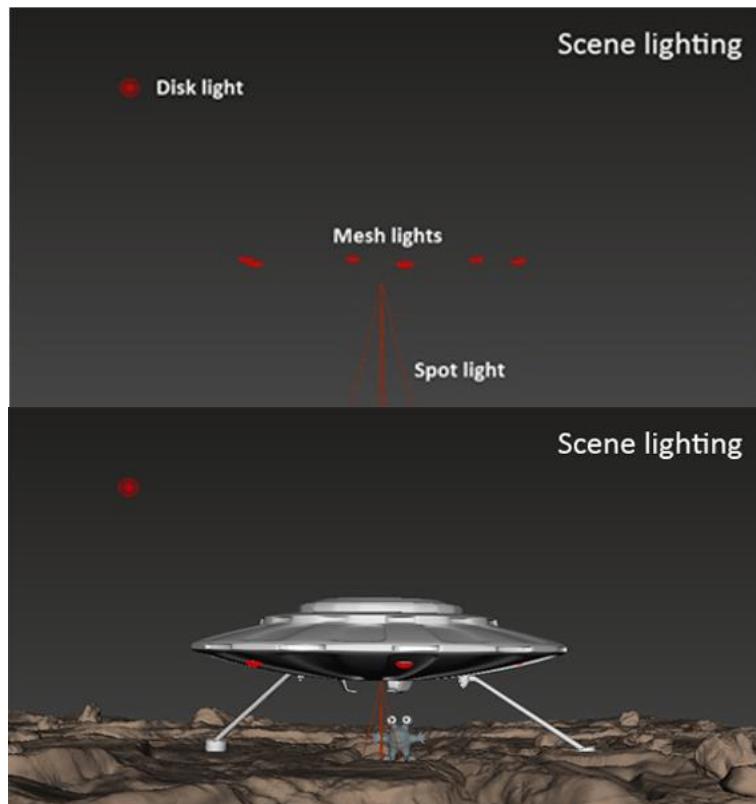


이 '만들기' 자습서에서는 렌더링 시 특정 종류의 노이즈가 발생할 때 조명 및 렌더링 볼륨뿐만 아니라 몇 가지 기타 문제를 다룹니다. 이 장면은 작은 가시 광원, 높은 반사면 및 체적 산란이 포함되어 있어서 '반딧불'이 나타날 가능성이 있습니다.

Scene Lighting

이 장면은 Light Shape이 'Disk'(달빛)로 설정되어 있는 '**Arnold Area Light**'이 비추고 있습니다. UFO 조명은 Mesh Light으로 변환된 지오메트리입니다. 또한 UFO 아래쪽을 향하는 스포트라이트와 오른쪽에 영역 라이트가 있습니다. MtoA Render Settings 창에서 Volume Scattering이 활성화되었습니다.

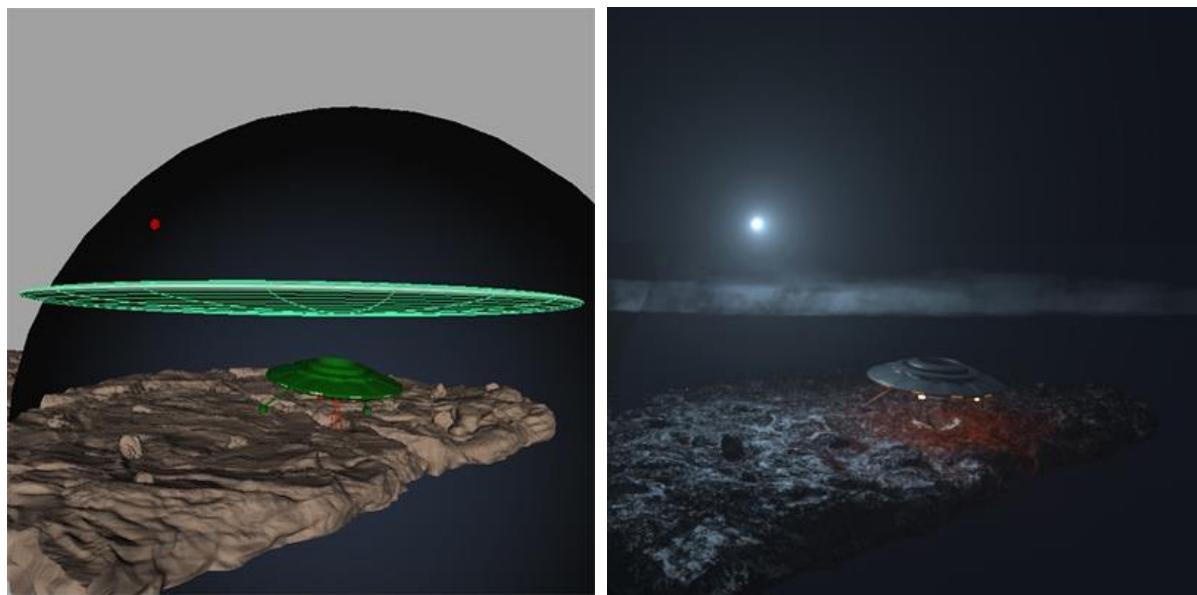




Moonlit Clouds – Volume 쉐이더

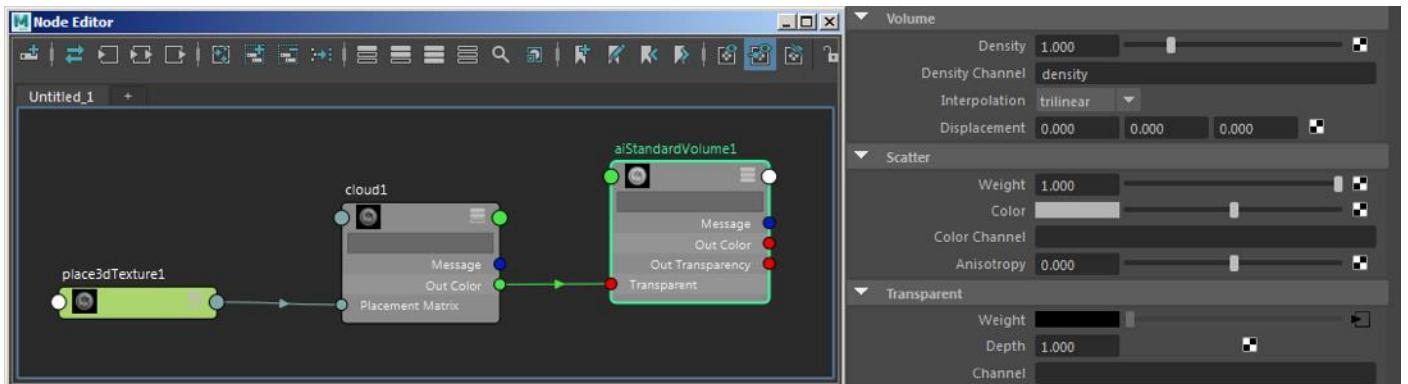
- 폴리곤 구체가 이 장면 위와 Disk 조명 아래에 위치합니다. 이것이 '구름층'으로 사용됩니다. 이것은 Disk 조명과 장면 사이에 위치하고 있습니다. 지면 안개를 표현하는 중복된 구체도 UFO 아래의 지면 근처에 배치되었습니다.

구름으로 볼륨을 렌더링하는 것에 대한 추가 정보는 [여기에서\(Volume Shader를 사용하여 구름 렌더링하기\)](#) 있습니다.

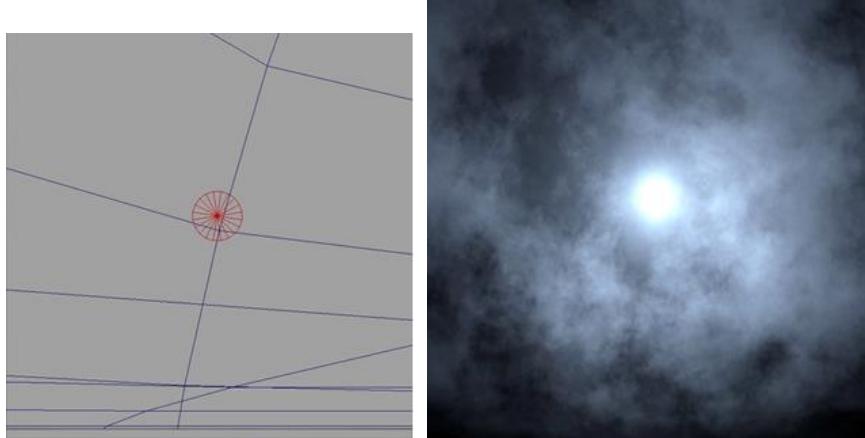


구름층을 표현하는 구체에 지정된 구름 쉐이더

- **Standard Surface** 쉐이더를 구체에 지정합니다. Maya '구름' 텍스처는 Standard Surface Shading Engine의 Volume Material 속성에 연결된 aiStaVolumeCollector 노드의 Scattering Color에 연결됩니다. 쉐이딩 네트워크는 다음과 같은 모습입니다.



렌더링되면 구름이 아래 이미지처럼 보입니다.



Volume Scattering이 활성화되어 달빛을 표현하는 Disk 조명

Noise

이 장면에는 일반적으로 '스파이크 노이즈' 또는 '반딧불이'라고 부르는 샘플링 노이즈가 있습니다. 이들은 격리된 매우 밝은 픽셀로 애니메이션에서 여러 프레임 사이를 이동합니다. 이러한 종류의 노이즈는 렌더러에서 샘플의 수를 증가시키는 것만으로는 제거할 수 없습니다.

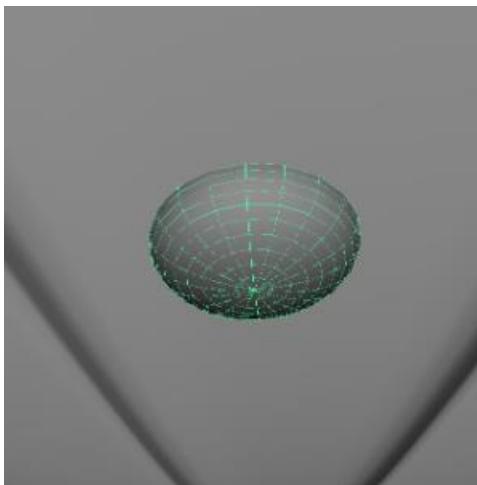


렌더러에 나타나는 반딧불 노이즈(빛과 광택이 있는 반사 금속 UFO 물질 주위에 집중).

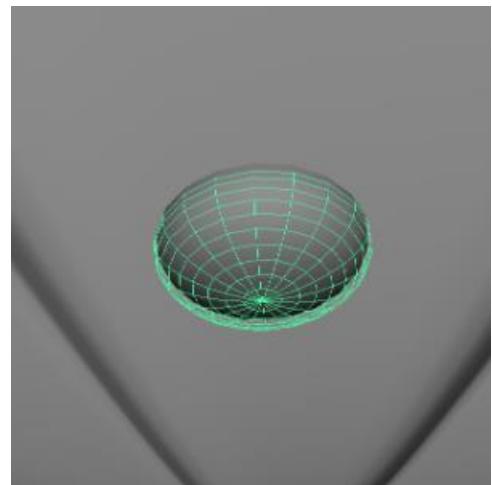
이 장면은 높은 반사 표면, 체적 산란 및 작은 광원과 같은 여러 가지 노이즈 요인을 가지고 있습니다. '노이즈' 문제를 수정하는 방법을 살펴보면 다음과 같습니다.

Mesh Light

UFO의 조명은 Mesh Light로 변환된 지오메트리로 구성됩니다. Mesh Light가 생성되면, 생성이 시작된 지오메트리와 동일한 위치에 배치됩니다. 이것은 샘플링 광선과 그림자 노이즈가 나타나는 경우 문제가 될 수 있습니다. 한 가지 해결 방법은 해당 지오메트리에 대해 'Casts Shadows' 및 'Receive Shadows'를 꺼는 것입니다. 또 다른 해결책은 빛을 해당 물체로부터 멀리 이동시키는 것입니다(아래 이미지 참조).



Mesh Light는 메시와 같은 위치에 생성됩니다.

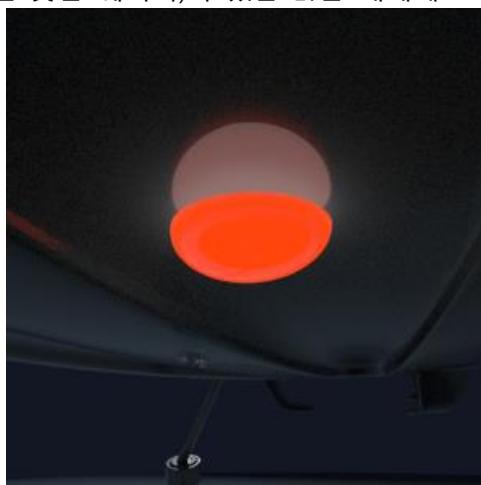


Mesh Light를 메시에서 멀리 이동하면 노이즈가 감소합니다.

Rayswitch

반딧불이를 줄일 수 있는 또 다른 방법은 메시를 반사 광선에 보이지 않게 만드는 것입니다. 이것은 **Ray Switch** 쉐이더를 통해 가능합니다. 이 방법은 '반딧불이'를 유발하는 원인이 될 수 있는 모든 쉐이더에 대한 컨트롤을 제공합니다. 예:

- *Specular Reflection* 속성에서 수정된 쉐이더(예: 검정색을 반환하는 쉐이더 또는 훨씬 낮은 Specularity Weight를 갖는 쉐이더)가 있는 밝은 개체에 Ray Switch 쉐이더를 지정합니다.



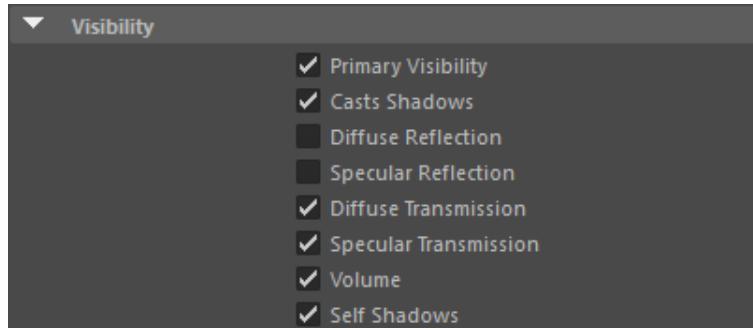
Rayswitch



Rayswitch 없음

Visible in Diffuse / Specular

노이즈를 줄일 수 있는 또 다른 방법은 노이즈를 유발하는 메시에 대해 *Diffuse Reflection* 및/또는 *Specular Reflection* 속성을 비활성화하는 것입니다. 이러한 속성은 메시의 Arnold 속성에 있습니다.



Visible in Diffuse/Glossy: Enabled(specular/diffuse
노이즈)

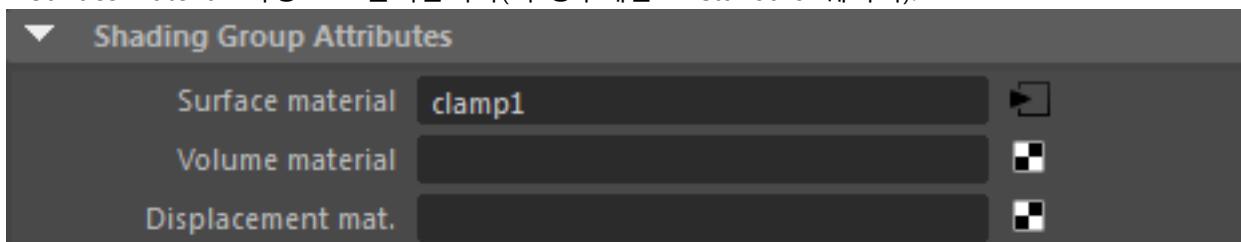


Visible in Diffuse/Glossy: Disabled (노이즈 없음)

Clamp Sample Values

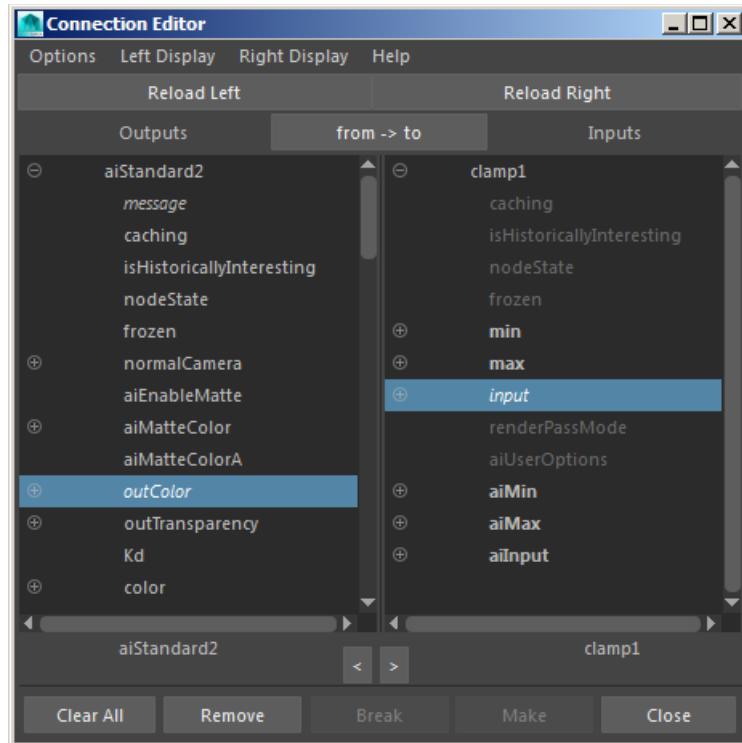
노이즈를 줄일 수 있는 한 가지 옵션은 렌더링된 픽셀의 샘플 값을 고정(Clamp)하는 것입니다. 하지만, 이것은 전체 이미지에 대한 픽셀을 고정합니다. 한 가지 해결 방법은 특정 쉐이더에 대한 픽셀을 고정하는 것입니다. 이를 위해서는:

- Maya 'Clamp' 노드를 마우스 가운데 버튼으로 누른 채 고정하고자 하는 재료에 대한 Shading Group의 'Surface material' 속성으로 끌어옵니다(이 경우에는 Ai Standard 쉐이더).

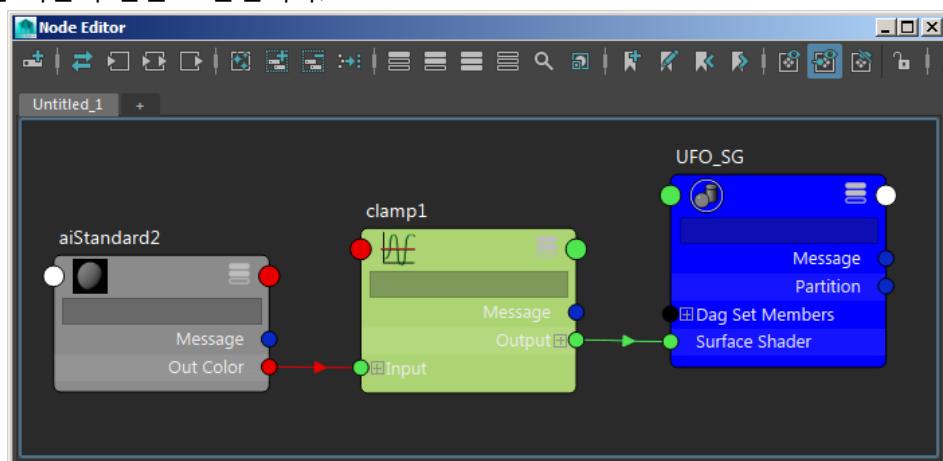


Ai Standard 쉐이더의 Shading Group의 Surface 재료 속성에 연결된 Clamp 노드

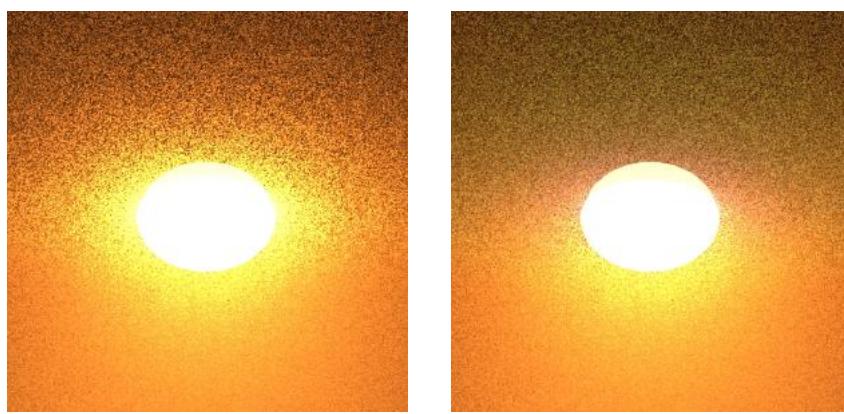
- Standard Surface의 'outColor'를 Clamp 노드의 'input' 속성에 연결합니다.



쉐이딩 네트워크는 다음과 같은 모습입니다.



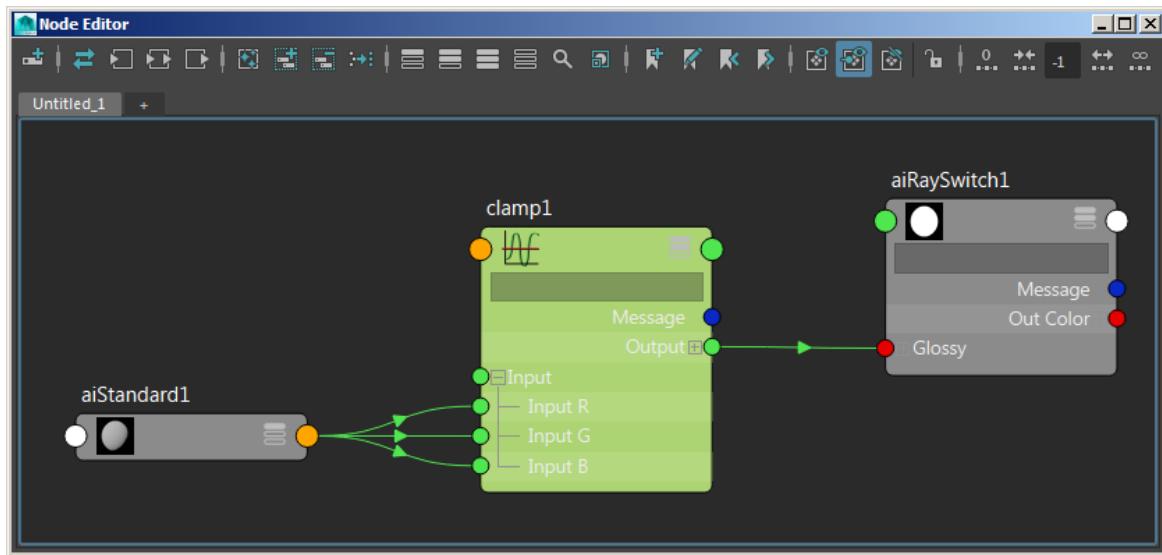
'Clamp Color Attributes'에서 'Max' 값을 낮춰보십시오.



No clamp

Max clamp 1

아래 예제에서는 Ai Standard의 반사 속성이 Maya 클램프 노드에 연결되었습니다. 그 다음 클램프 노드가 Ai Ray Switch 쉐이더의 광택 속성에 연결됩니다. 이것은 Render Settings 창에서 일반 클램프 대신 보조 광선 클램프로 작동합니다. 이 방법을 사용하면 직접 하이라이트에서 범위를 유지하는 이점을 얻을 수 있을 뿐만 아니라 보조 광선 노이즈도 제어할 수 있습니다.



Atmosphere Volume의 반딧불 노이즈 효과

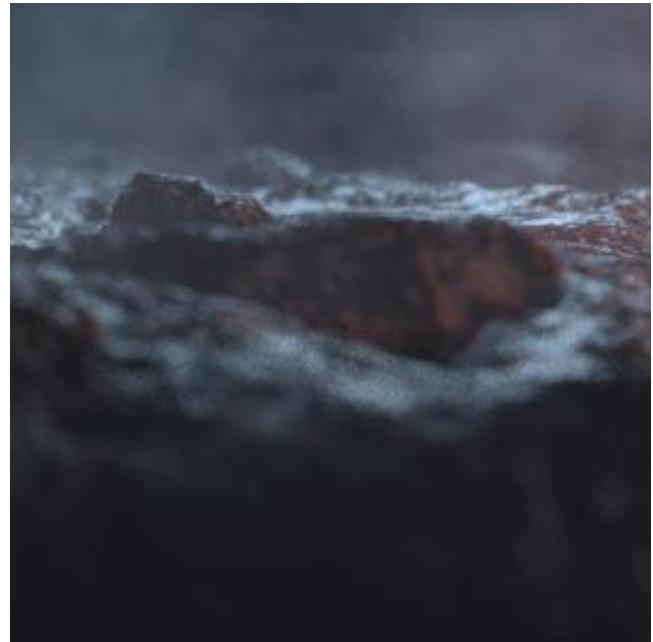
아래 이미지에서는 장면에 **Atmosphere Volume**을 활성화할 때의 효과를 볼 수 있습니다.



Atmosphere Volume을 활성화하면 지면 재료에 '반딧불'이 생깁니다. 이 머티리얼은 반딧불의 원인이 되는 Specularity를 가지고 있습니다. 이것은 이미지의 전경에서 더 두드러집니다. '반딧불'을 줄이는 간단한 방법은 바닥 재료에서 Specularity 양을 줄이는 것입니다. 이것이 가능하지 않은 경우에는 **Specular Samples**를 4로 높이면 반딧불이 제거됩니다.

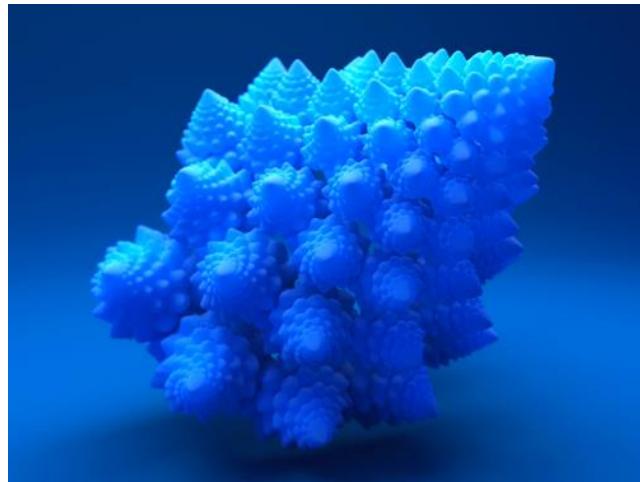


Gi Glossy Samples 2: 반딧불이 보임



Gi Glossy Samples 4: 반딧불이 안 보임

렌더링 튜토리얼



이들 자습서는 MtoA를 통한 렌더링에 대해 설명합니다.

- [Stand-in 소개](#)
- [Physical Sky를 사용한 구름 유체](#)
- [Depth of Field](#)
- [머리카락 렌더링](#)
- [만델벌브\(Mandelbulb\) 렌더링 방법](#)
- [MakeTX: 나무 바닥](#)
- [HDRI 및 Backplate를 사용하여 자동차 렌더링하기](#)
- [Curve 렌더링하기](#)
- [Arnold로 유리 표면 렌더링하기](#)
- [페인트 효과와 함께 불투명도 맵 렌더링하기](#)
- [Subdivision Displacement 예](#)
- [안개 낀 수중 장면](#)

Stand-in 소개



Stand-in을 사용하면 렌더링 시간까지 지오메트리 데이터 로딩을 지연하여 작업 장면을 가볍고 효율적으로 유지할 수 있습니다. 아주 적은 노력으로 막대한 수준의 복잡성을 장면에 추가하는 것이 가능합니다. 이 소개서에서는 stand-in을 사용하여 복잡한 장면을 매우 간단하게 설정하는 데 필요한 단계를 설명합니다.

장면은 하단 링크의 사이트에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/An+Introduction+to+Stand-ins>

다음 단계에서는 stand-in 사용 시 몇 가지 유용한 점을 소개합니다(사용자 각자에게 적합한 모델을 사용하여 따라해 볼 수 있음). 이 자습서는 매우 강력한 도구인 stand-in에 대한 단순한 소개입니다.

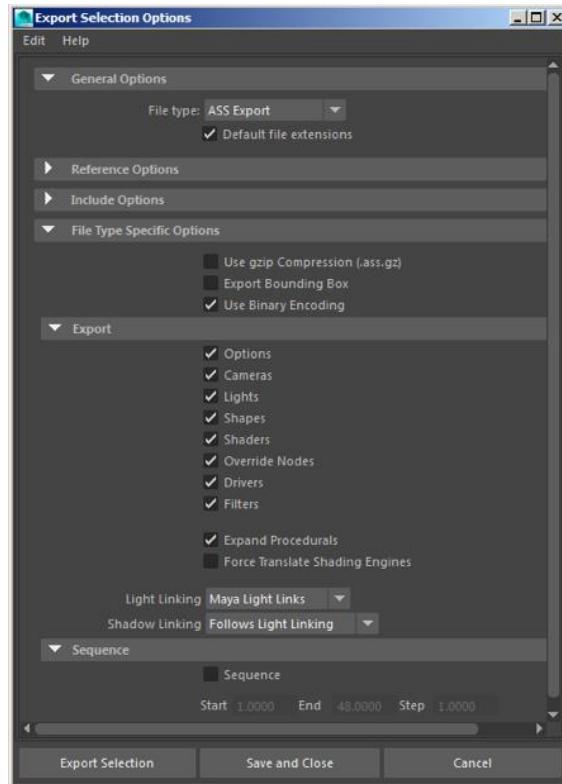
- Arnold의 Stand-in 노드를 사용하여 복제하고자 하는 모델을 엽니다. 모델을 내보내면 수정할 수 없으므로, 모델을 내보내기 전에 모델과 음영 처리가 완료되었는지 확인하십시오. 모델을 내보내려면 해당 모델을 선택하고 **Arnold > StandIn > Export** 메뉴 옵션으로 이동하십시오. 또는 **File>Export All** 옵션을 사용하여 모델을 Stand-in으로 내보낼 수 있습니다.



Arnold 메뉴에서 stand-in으로 모델을 Export

Export Stand-in

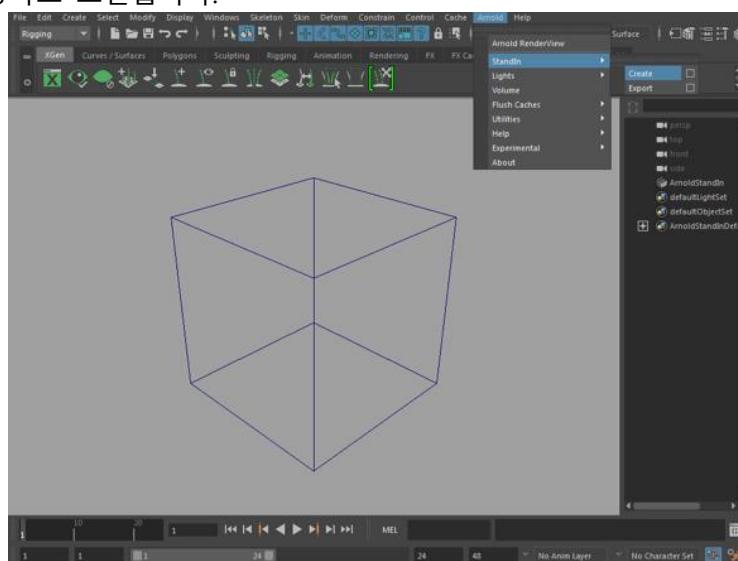
- Stand-in으로 사용하고자 하는 모델을 내보내려는 곳을 선택하고 OK를 클릭합니다. Arnold는 메시를 렌더링하는 데 필요한 모든 것이 담겨 있는 .ass 파일로 메시를 내보냅니다. 모델의 파일 크기가 매우 큰 경우 **Compressed**을 선택하여 압축하도록 할 수 있습니다. 모델에 애니메이션이 포함되어 있으면 **Sequence**를 클릭하고 내보내려는 프레임 범위를 선택합니다(만든 후에는 stand-in의 프레임 속성에 '=frame'을 쓰는 것을 잊지 마십시오).



Ass 내보내기 옵션

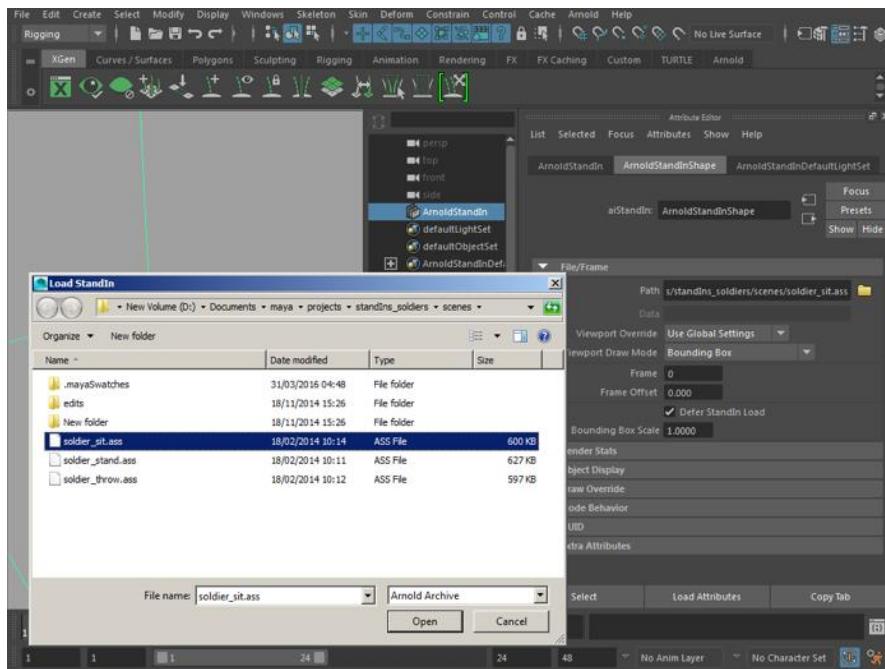
Stand-in 생성

- 새로운 장면을 생성합니다. **Arnold>Stand-in>Create**을 선택하여 새로운 stand-in을 생성합니다. 그러면 stand-in이 경계 상자로 표현됩니다.



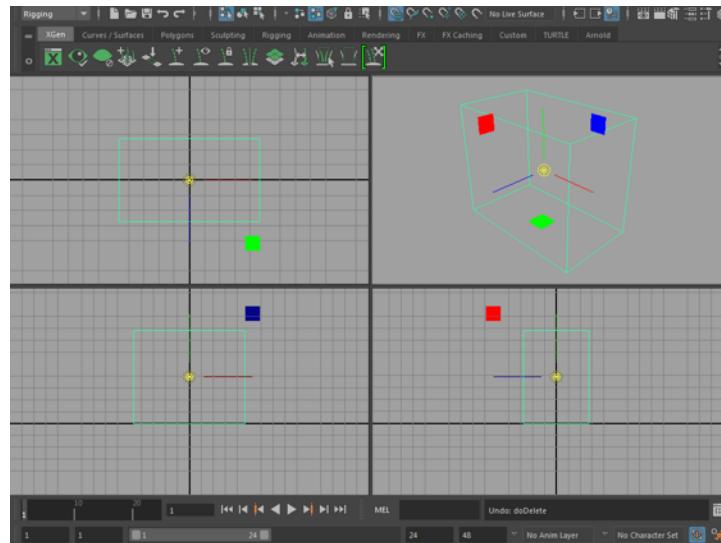
Maya 뷰포트에서 경계 상자로 표시되는 Stand-in

- 해당 stand-in을 선택하고 속성 편집기를 엽니다. 경로 아래에서 stand-in 모델로 내보낸 .ass 파일을 선택합니다.



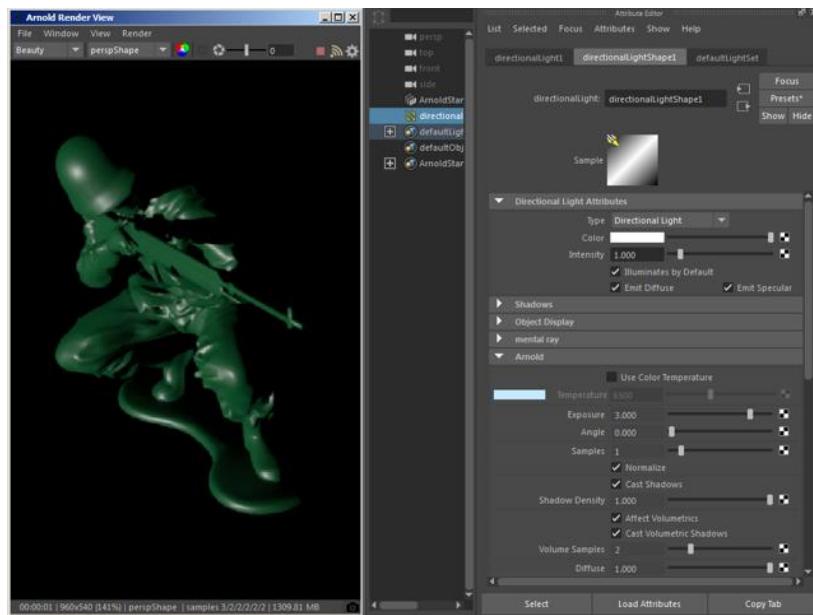
stand-in ass 파일 로드

- stand-in 경계 상자의 모양이 원래 모델의 모양과 일치하도록 변경됩니다. 격자의 중앙에 오도록 배치합니다. 중심점을 stand-in의 아래에 배치합니다. 이것은 인스턴스화된 stand-in 모델을 배치할 때 도움이 될 것입니다.

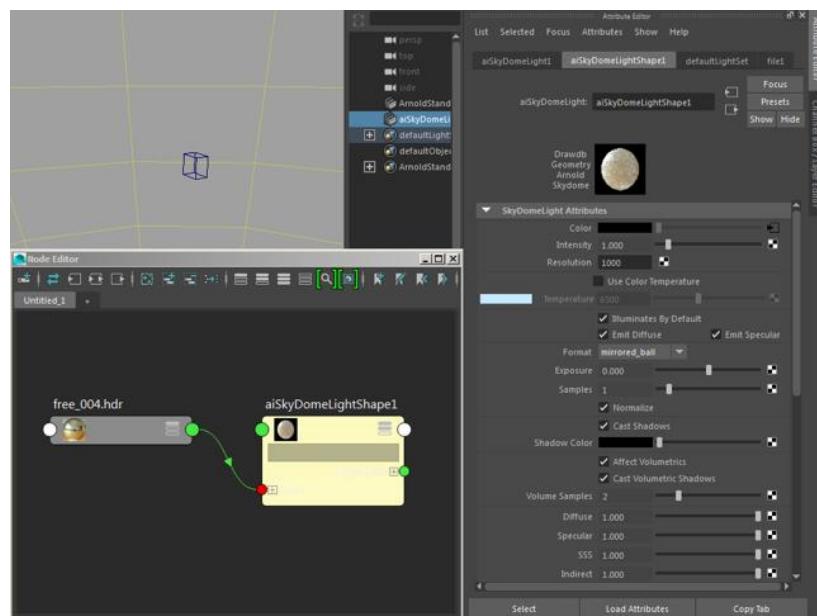


조명

- 방향 조명을 생성합니다. 조명이 stand-in과 렌더링을 향하게 합니다. 원래 쉐이더 지정을 가진 원래 모델이 표시됩니다. 쉐이더를 변경해야 한다면 stand-in을 다시 내보내서 위 절차를 반복하거나 새 쉐이더를 할당해야 합니다.



- 방향 조명을 HDR 조명으로 교체하겠습니다. Hypershade 창을 열고 **Ai SkyDomeLight**을 만듭니다. 파일 텍스처를 **Color** 슬롯에 할당하고 적절한 HDR 맵을 선택합니다.

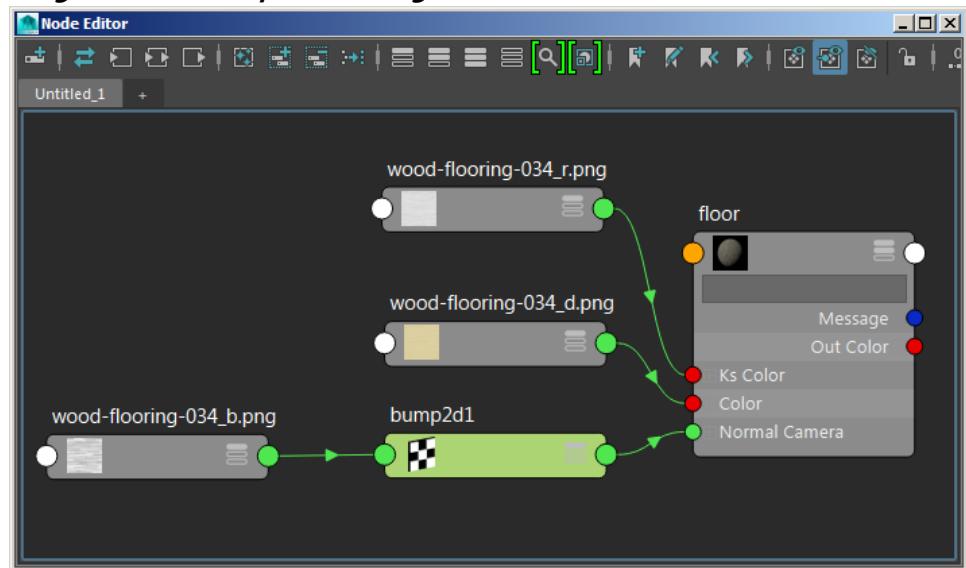


렌더링되면 다음 이미지와 유사한 것들이 보입니다.



Ai SkyDome Light가 비춰진 Stand-in

- 폴리곤 면을 만들고 stand-in에 대한 바닥면으로 사용할 수 있을 정도로 높게 조절합니다. **Standard Surface** 쉐이더를 지정하고 **Base Color, Specular Color, Bump** 맵 슬롯에 나무 바닥 텍스처를 지정합니다. **Specular Weight**을 0.3으로, **Specular Roughness**를 0.2로 낮춥니다.



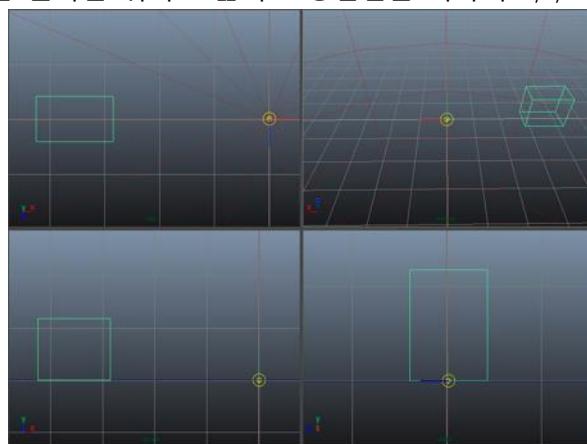
Standard Surface 쉐이더를 사용하여 나무 바닥 만들기

stand-in 지오메트리의 한 인스턴스를 사용하면 다음 이미지처럼 보입니다.



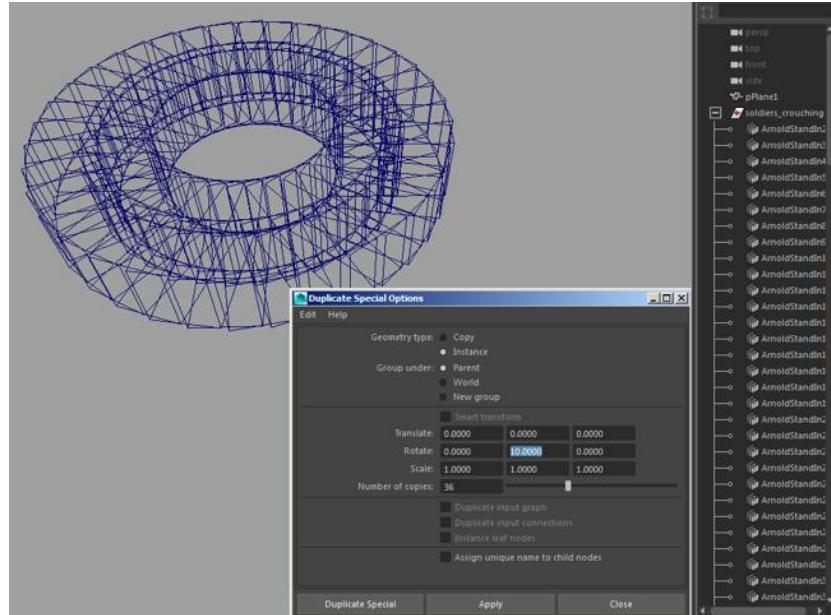
Stand-in 인스턴스화

- 이제 재미있는 것을 보여드리겠습니다. stand-in 인스턴스화를 시작하겠습니다. stand-in을 원으로 복제하기 위해 Maya의 **Duplicate Special** 옵션을 사용할 것입니다. 먼저 중심점을 올바르게 설정해야 합니다. stand-in을 중심에서 약간 떨어진 위치로 옮기고 중심점을 격자의 0,0,0 중앙에 놓습니다.



중심으로부터 stand-in을 멀리 이동하고 중심점을 0 0 0으로 이동합니다.

- stand-in을 선택하고 **Duplicate Special** 옵으로 이동합니다. **Instance**를 선택하십시오. 이것은 매우 중요합니다. 메모리가 절약되고 장면이 더 빠르게 렌더링되기 때문입니다. **Y 회전**을 **10**으로 변경하고 복사본의 수를 **36**으로 높입니다. 모델의 크기에 따라 회전 값을 변경해야 할 수 있습니다. 그렇지 않으면 인스턴스화된 stand-in이 서로를 침투할 수 있습니다. 원래 stand-in을 삭제하지 않으면 지오메트리가 중복되므로 주의해야 합니다.

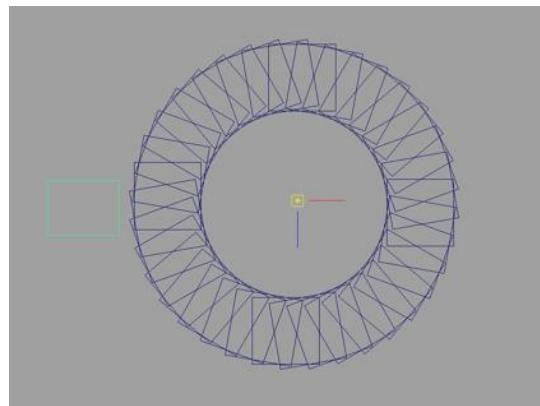


원으로 인스턴스화된 Stand-in

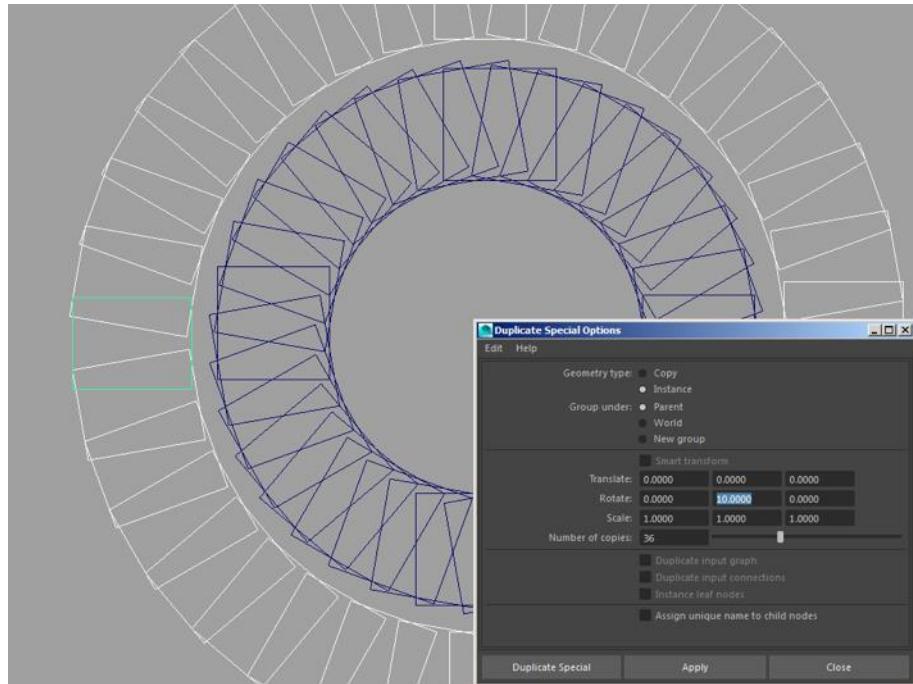
이것은 다음과 같은 이미지가 됩니다.



- stand-in 중 하나를 복제하여 바깥쪽 옆으로 이동하고 중심점을 격자의 0,0,0 중앙에 놓습니다.

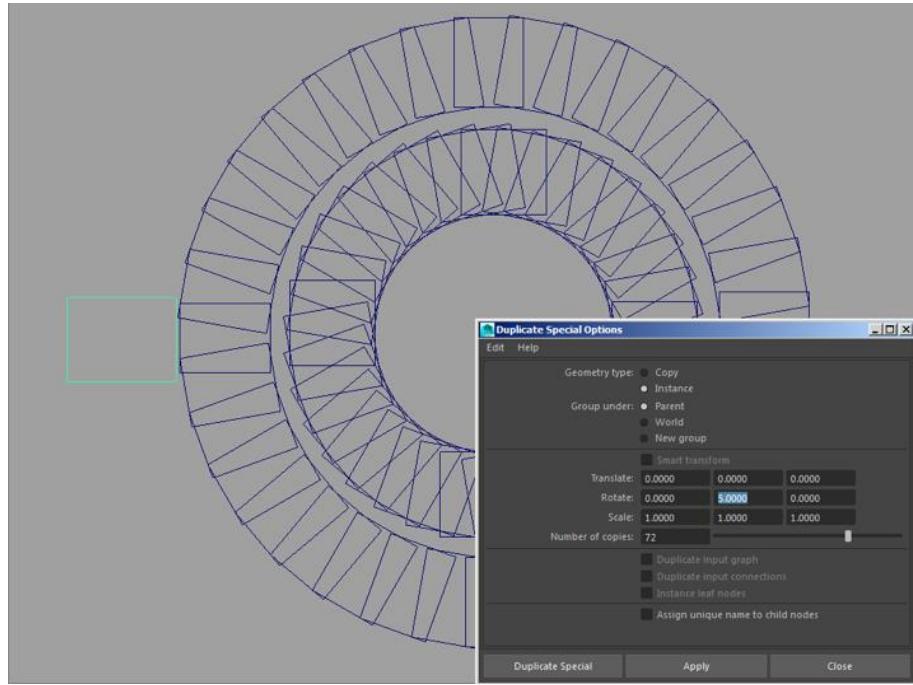


- 위 절차를 사용하여 stand-in을 복제합니다. 인스턴스화하는 것을 잊지 마십시오.



군인의 두 번째 행에 대해 이 절차를 반복합니다.

이제 stand-in 모델이 두 행이 됩니다. 동일한 방법을 사용하여 서 있는 군인에 대해 다른 stand-in을 만들었습니다. 하지만 더 큰 둘레를 보완하기 위해 복사본 수를 72개로 늘렸습니다.

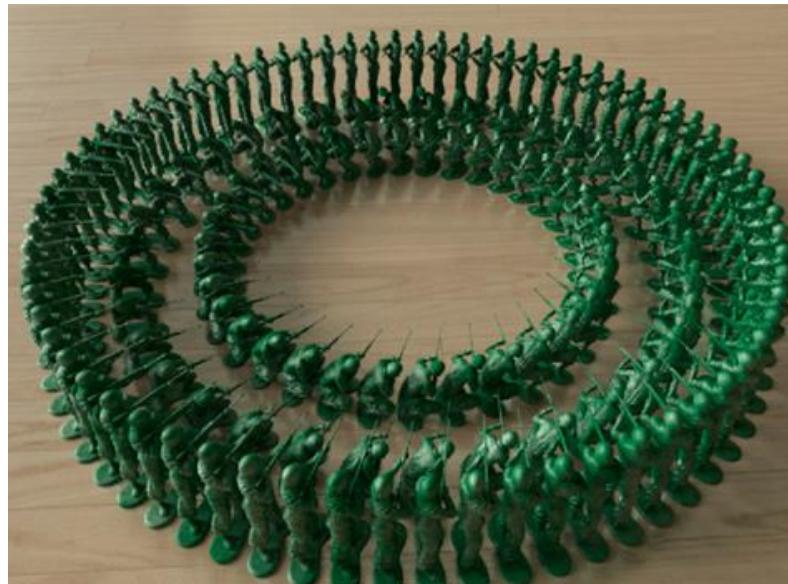


서 있는 군인의 stand-in을 생성 및 복제

또한 서있는 군인의 stand-in이 앉아 있는 군인의 stand-in과 교차하고 있습니다. 이 경우에는 앉아있는 군인 위로 돌출된 것은 총이므로 상관이 없습니다.



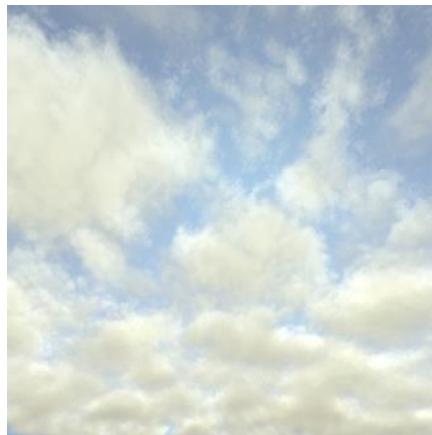
stand-in은 렌더링에 약간의 오버헤드로 여러 번 인스턴스화 될 수 있습니다(원래 모델은 약 20,000개의 폴리곤을 가지고 있으므로 stand-in을 사용하여 인스턴스화하지 않으면 수백 명의 군인이 있는 다음 장면이 상당히 커집니다). Arnold는 물론 장면 렌더링에 문제가 없습니다.



결론

이 소개서에서는 stand-in 사용의 기본만 설명하고 있음을 다시 한 번 알려드립니다. 예를 들어, .ass 아카이브 파일을 사용할 뿐만 아니라 stand-in은 동적 라이브러리의 형태로 제공할 수 있는 사전 컴파일된 사용자 정의 절차 프로그램과도 작동하므로 파라미터화된 절차 지오메트리를 사용할 수 있습니다. Stand-in은 또한 재귀적일 수도 있고, 렌더링하는 동안 실제적으로 필요할 때까지 절차 지오메트리의 로딩을 연기할 수 있습니다(예: 광선이 물체의 경계 상자 안에 들어갈 때까지 지오메트리가 로드되지 않으며 메모리를 사용하지 않습니다). 이러한 기술을 통해 모듈 방식으로 장면들을 모을 수 있습니다.

Physical Sky를 사용한 구름 유체



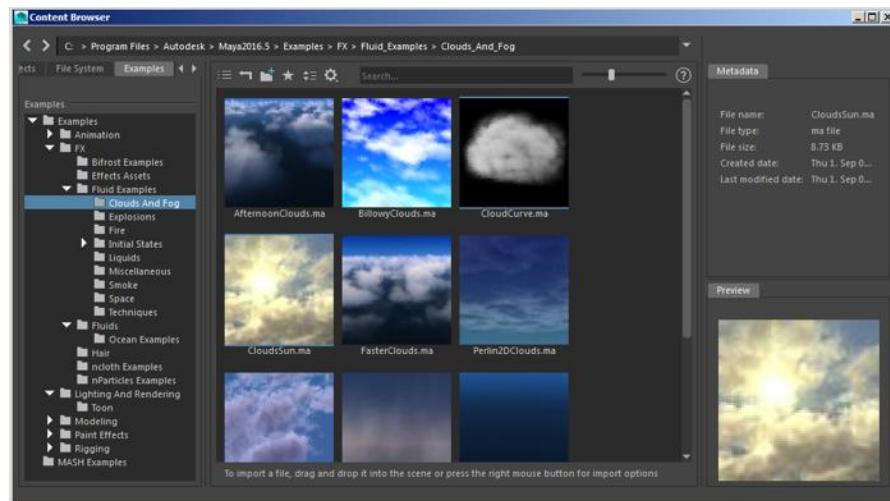
이 자습서에서는 Maya Fluid 컨테이너와 Physical Sky를 사용하여 흐린 하늘을 렌더링하는 방법을 설명합니다. 구름 프리셋으로 시작하여 Maya Fluid의 음영 속성 중 일부를 미세 조정하여 최상의 결과를 얻을 수 있습니다. 마지막으로 타임랩스 또는 카메라 fly-through에 적합한 효과를 얻기 위해 유체를 애니메이트하는 방법을 알아봅니다.

파일은 하단 링크의 사이트에서 다운로드 할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Cloud+Fluids+with+the+Physical+Sky>

Visor로부터 Fluid 가져오기

- 이 예제에서는 먼저 **Content Browser** 창 안의 **Fluid Examples** 아래에 있는 **CloudsandFog** 폴더 안의 구름 유체 프리셋 중 하나를 사용해 보겠습니다. **CloudsSun.ma**를 마우스 오른쪽 버튼으로 클릭하고 **Import**를 선택합니다.



Fluid 프리셋 'CloudsSun.ma' 가져오기

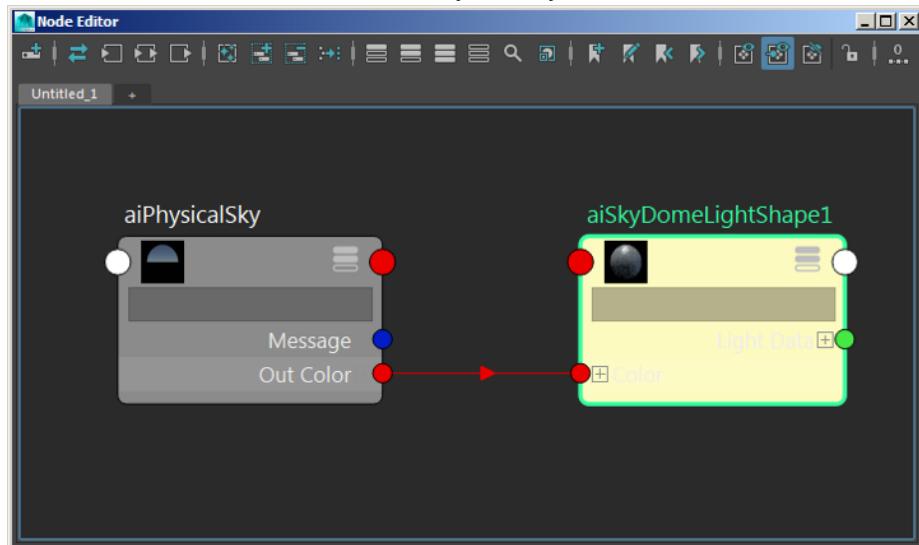
- 점 조명 - 'Sun'을 삭제합니다. '구름' 유체만 남아야 합니다. 카메라 **CloudsSunCamera**를 사용하여 장면을 렌더링합니다. 장면 조명이 없을 때는 렌더링이 다음 이미지처럼 보입니다.



장면 조명이 없는 'CloudLayer' 유체 렌더링

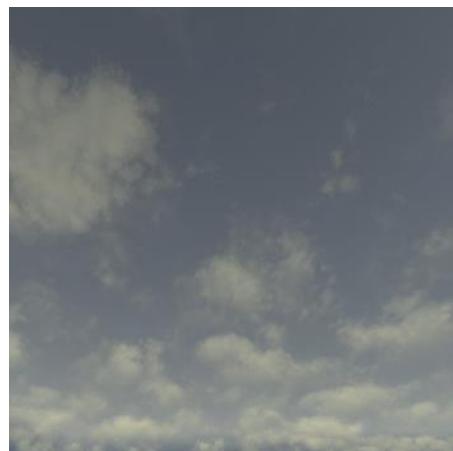
Physical Sky 생성

- **SkyDomeAi Skydome Lightlight**를 만들고 SkyDome 조명의 Color 속성에 **Physical Sky**를 연결합니다 (Hypershade 창에 있는 텍스처 탭으로부터 aiPhysicalSky를 마우스 가운데 버튼으로 드래그할 수 있음).

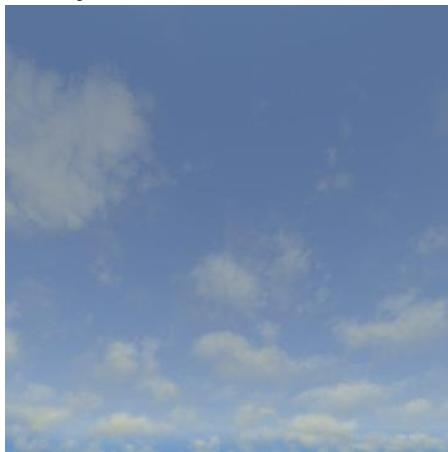


Skydome 조명에 연결된 Physical sky

- 장면을 렌더링합니다. 기본 설정을 사용하면 이미지가 다음과 같습니다. 지금도 괜찮지만 Physical Sky를 추가하고 Maya Fluid의 음영 속성을 조정하면 더 멋지게 만들 수 있습니다.



- 하늘이 약간 침침해 보입니다. 하늘을 좀더 밝게 만들고 파란색을 추가해야 합니다. **Physical Sky**를 선택합니다. **Intensity**를 3으로 높이고 **Sky Tint**를 연청색으로 변경합니다.



- 마지막으로 **Physical Sky**의 **elevation** 및 **azimuth** 속성을 조정하여 태양이 카메라에 표시되도록 배치합니다.

Fluid 쉐이딩하기

이 특별한 유체에 대한 몇 가지 사항:

- 일정한 밀도만 사용합니다.
- 음영 Y 그라디언트에 짙은 모서리 드롭오프가 있습니다.
- Y 그라디언트 색상, 백열 및 불투명도 입력은 구름의 색상과 모양을 크게 만들어내는 반면 Perlin 노이즈가 있는 텍스처 불투명도는 구름에 대한 세부정보를 제공합니다.

Opacity

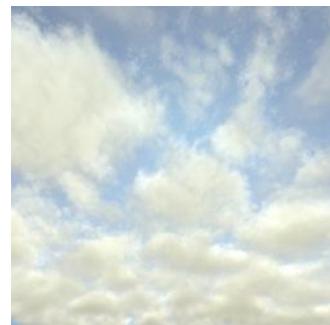
구름이 약간 부드럽게 보입니다. 투명도와 불투명도를 줄임으로써 구름을 더 짙게 만들 수 있습니다. 불투명도와 결합된 투명도는 유체의 밀도에 얼마나 많은 빛이 침투할 수 있는지를 결정합니다. 먼저 불투명도부터 시작하겠습니다. 아래에서 Opacity 'Input Bias'를 조절할 때의 결과를 볼 수 있습니다. 이 값으로 실험해 보십시오. 밀도 높은 많은 구름을 만들어주는 0.3 값을 시도해 보십시오.



Opacity 'Input Bias' 0.1



Opacity 'Input Bias' 0.2



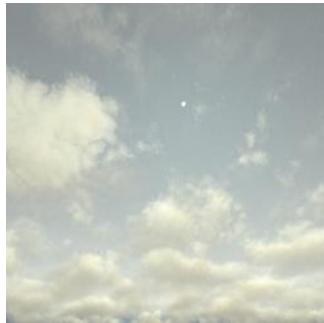
Opacity 'Input Bias' 0.3



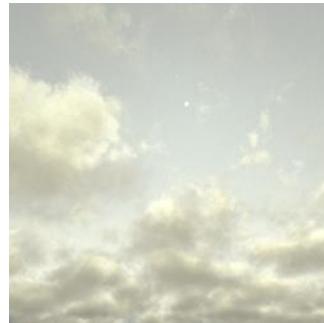
Opacity 'Input Bias' 0.4

Transparency

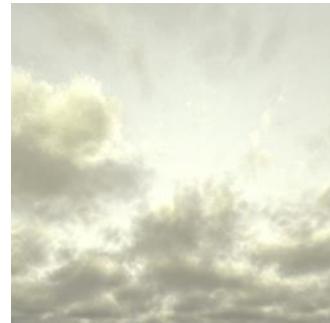
투명도 값을 사용하여 구름의 불투명도를 추가적으로 세밀하게 조절할 수 있습니다. Transparency 값을 높이면 다음 효과가 나타납니다.



Transparency 0.1



Transparency 0.05



Transparency 0.03



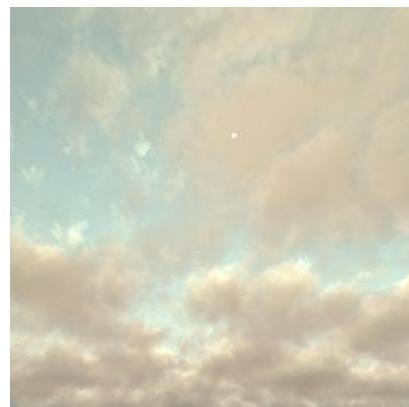
Transparency 0.01

투명도 값의 색상을 조절하여 컬러 불투명도를 만들 수도 있습니다.

- **Transparency** 옆에 있는 색상 선택기를 클릭하고 **saturation value**를 높입니다. 아래 예제에서는 다음 값들이 사용됩니다.

Hue: 1.042. Saturation: 0.243. Value: 0.067

HSV를 조정하여 재미있는 색깔의 구름을 표현할 수 있어야 합니다. **Physical Sky**의 **Azimuth**과 결합하여 이 값을 실험함으로써 흥미로운 하늘을 연출할 수도 있습니다.

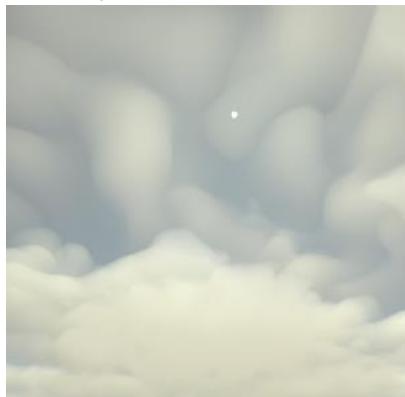


Transparency의 HSV를 변경하면 흥미로운 색상의 구름 하늘을 표현할 수 있습니다.

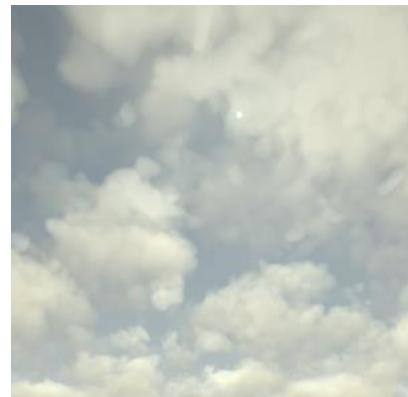
다음 Maya Fluid Shading 속성 값을 증가시켜 유체 내의 디테일 및 복잡성을 개선할 수 있습니다.

Depth Max와 같은 일부 값을 증가시키면 렌더링 시간이 늘어납니다.

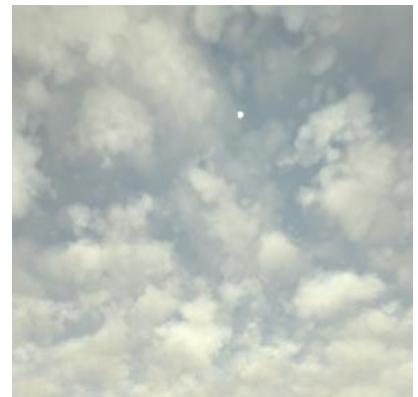
Frequency



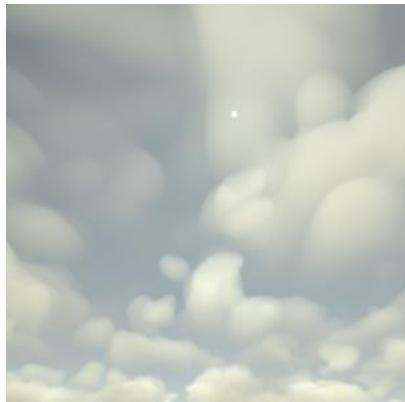
Frequency 0.5



Frequency 1.5



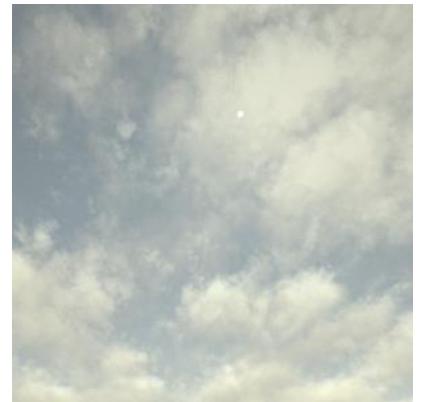
Frequency 3

Frequency Ratio

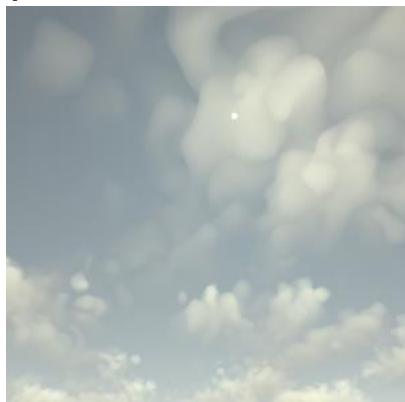
Frequency Ratio 1.5



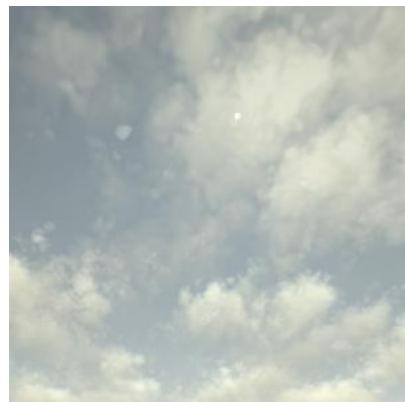
Frequency Ratio 2



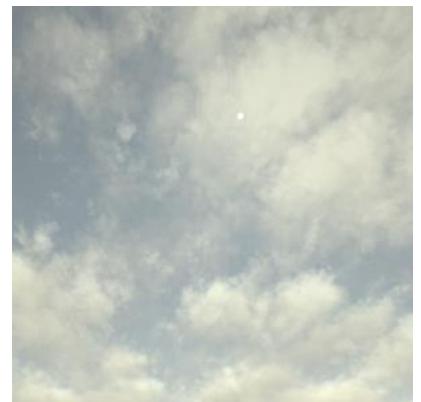
Frequency Ratio 4

Depth Max

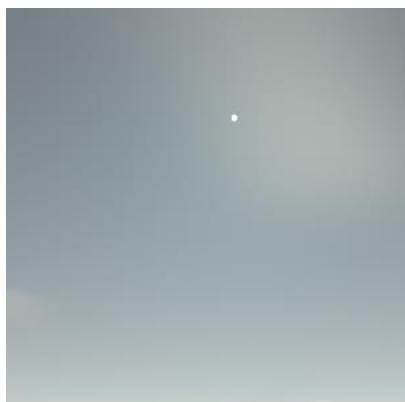
Depth Max 2



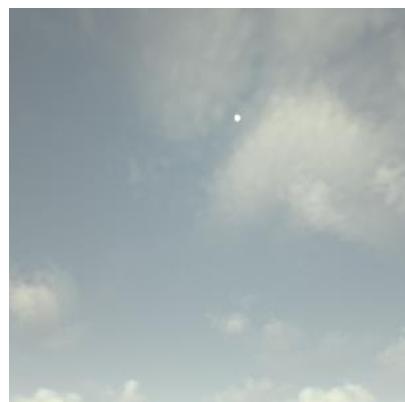
Depth Max 3



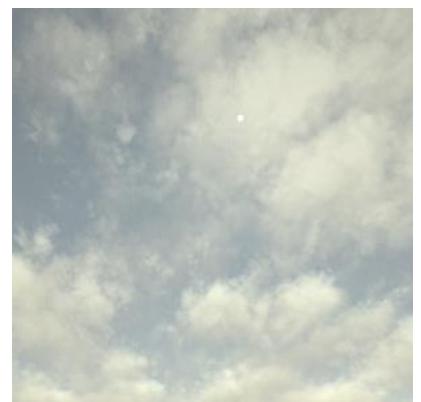
Depth Max 4

Ratio

Ratio 0.25



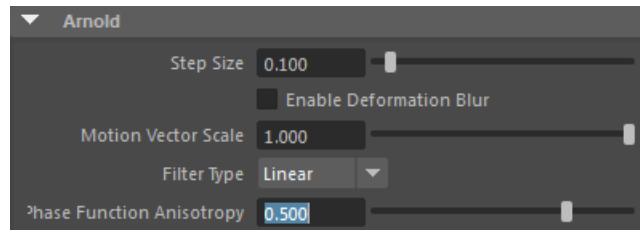
Ratio 0.5



Ratio 0.75

Phase Function Anisotropy

Phase Function Anisotropy를 조절하여 볼륨 내에서 햇빛이 '산란'되는 방법을 변경할 수 있습니다. 이것은 Maya Fluid의 Arnold 속성에 있습니다.



- **Phase Function Anisotropy**를 0.5 정도로 높입니다. 산란 효과가 태양 원판 주위에서 더욱 분명해짐을 알 수 있습니다.



Phase function anisotropy 0



Phase function anisotropy 0.5



Phase function anisotropy 0.75

아래는 'Phase Function Anisotropy'의 보다 '극단적인' 사용을 보여주는 또 다른 예제입니다.



Phase function anisotropy -1



Phase function anisotropy -0.5



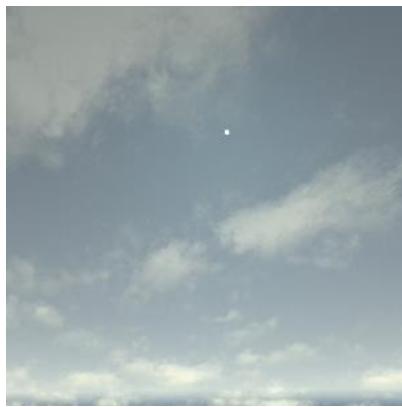
Phase function anisotropy 0.5

구름 움직임 효과 만들기

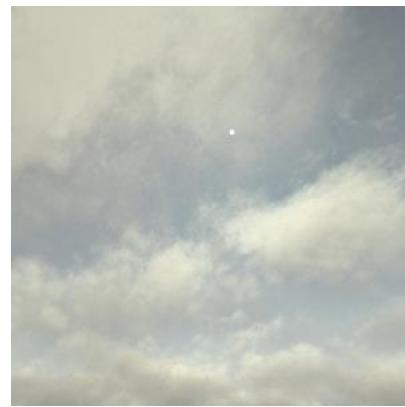
구름을 움직이게 만들어서 타임랩스 효과를 연출할 수 있습니다. 먼저 파란 하늘부터 시작하고 **Shading>Edge Dropoff**을 사용하여 구름을 만들어보겠습니다. 그 다음에는 **Texture Origin Z**를 움직여서 Z 축을 통해 구름을 이동시키겠습니다.

Edge Drop Off

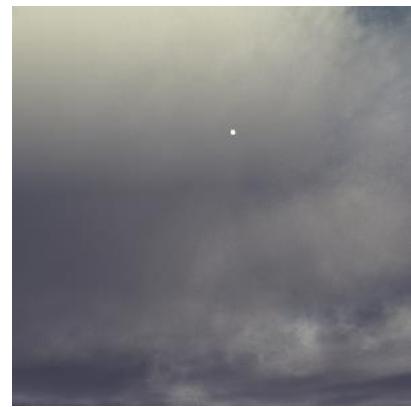
'Edge Drop Off'를 높여서 구름을 보다 빽빽하고 '짙게' 만들 수 있습니다. 이 값을 높이면 유체 내의 구름 가장자리 쪽으로 밀도가 떨어집니다. 이 값을 애니메이트하여 구름의 형성을 증가시킬 수 있습니다.



Edge Dropoff 0.3

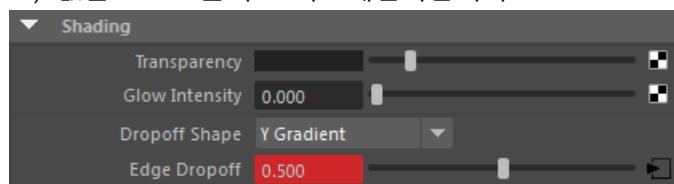


Edge Dropoff 0.45



Edge Dropoff 0.5

- 프레임 1에서 0.2와 같은 낮은 값을 사용하여 **Edge Dropoff**를 키프레임화합니다. 애니메이션의 마지막 프레임에서(이 경우 100) 값을 0.5로 높이고 키프레임화합니다.



Edge Dropoff를 높여서 유체 구름이 형성되는 것을 보십시오(0.2-0.5).

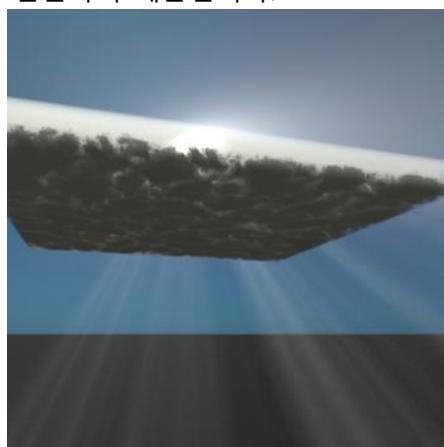
- 유체를 선택하고 **Textures**로 이동합니다. 프레임 1에서 **Texture Origin Z**를 0으로 키프레임화합니다. 프레임 100으로 이동하고 **Texture Origin Z**를 -2로 키프레임화합니다. 그러면 음의 Z 방향을 통해 유체 텍스처의 움직임이 애니메이트됩니다.



Texture Origin Z를 애니메이트하여 Z 축을 따라 구름이 움직이는 것을 확인하십시오.

Volume Scattering 적용

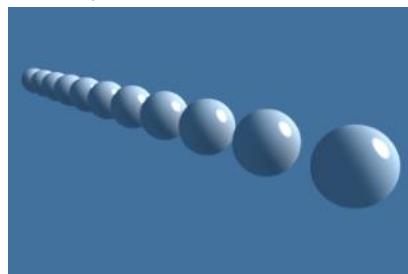
현재 'Volume Scattering'은 유체 볼륨에 잘 맞지 않습니다. 이것은 대기가 픽셀의 배경에 있는 모든 항목 위에 매핑된 불투명도인 단일 평면 결과를 반환하기 때문입니다.



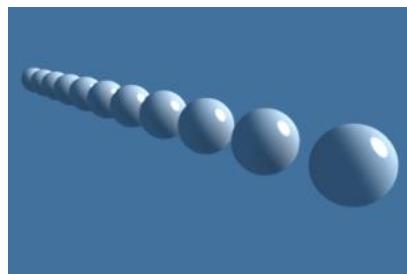
지금까지 Maya Fluid 및 Physical Sky를 사용하여 구름을 렌더링하는 방법을 알아봤습니다. 이들 설정을 다른 Maya Fluid 프리셋에 적용할 수도 있습니다. 다른 구름 프리셋을 사용하거나 이를 기술을 여러분 자신의 Maya Fluid에 적용해 보세요.

Depth of Field

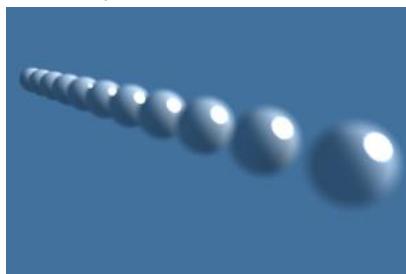
다음 이미지들은 카메라의 다양한 피사계 심도 값의 효과를 설명합니다. 아래 예제들은 반지름이 1인 구체를 사용합니다(장면 스케일 크기가 사용하는 DOF 값에 영향을 줄 수 있음을 기억하십시오.).



Depth of field 없음



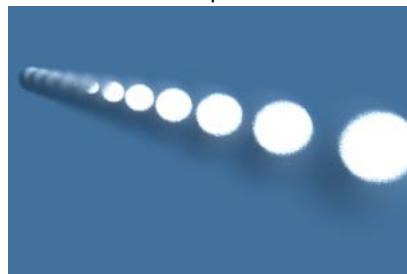
Size 0.01. Aspect Ratio 1



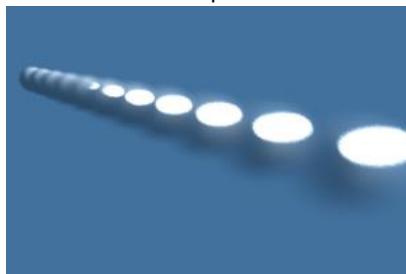
Size 0.25. Aspect Ratio 1



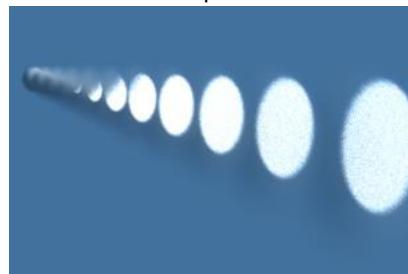
Size 0.5. Aspect Ratio 1



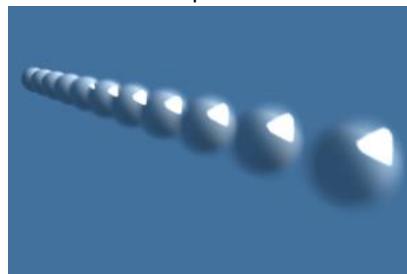
Size 1. Aspect Ratio 1



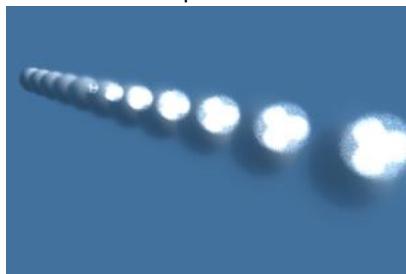
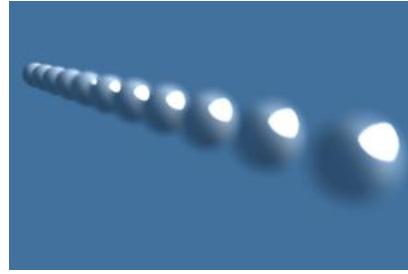
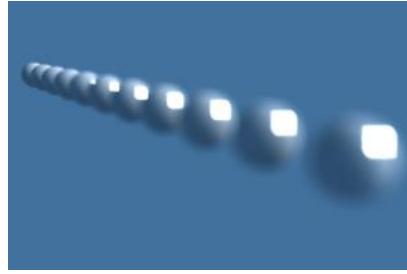
Size 1. Aspect Ratio 0.5



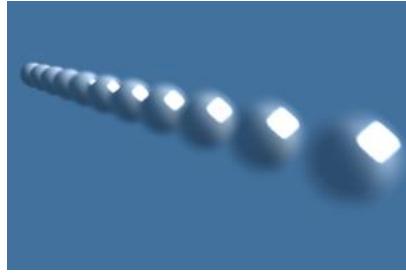
Size 1. Aspect Ratio 2



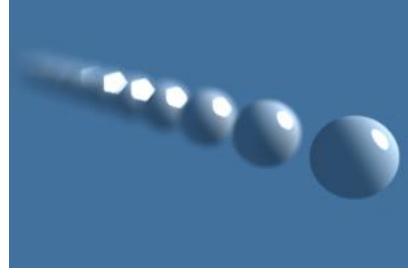
Size 0.5. Aspect Ratio 1. Blades 3

Size 1. Aspect Ratio 0.5 Blades 3.
Blade Curvature -5Size 1. Aspect Ratio 0.5 Blades 3.
Blade Curvature 0.5

Size 1. Aspect Ratio 0.5 Blades 4.



Rotation 45

Distance 8.5. Size 1. Aspect Ratio
0.5 Blades 3. Blade Curvature -5

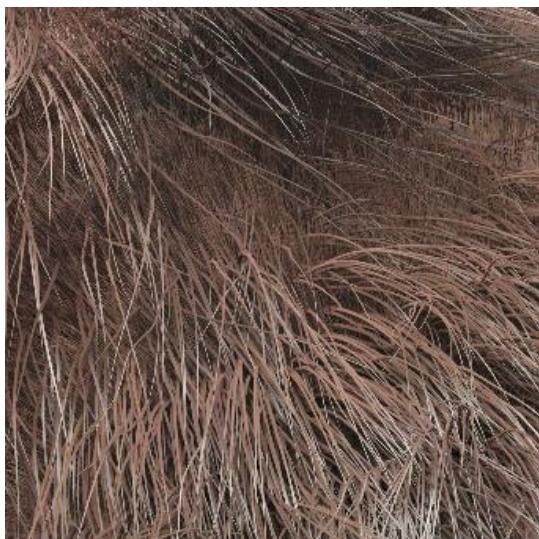
머리카락 렌더링



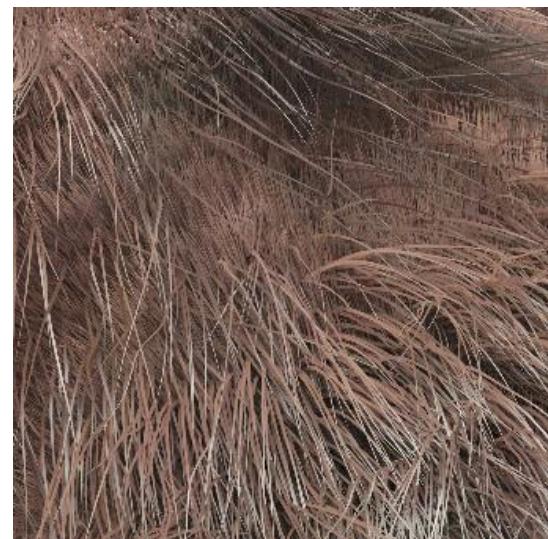
이 섹션의 목적은 사용자에게 Arnold에서 머리카락을 효율적으로 렌더링하는 방법을 보다 잘 설명하는 것입니다. 머리카락 렌더링 속도에 영향을 주는 주요 파라미터를 렌더링 예제와 함께 설명합니다. 일반적인 최적화 팁과 비결에 대해서도 살펴봅니다. 다음 자습서에서는 Arnold 곡선 형상 노드, **Standard Hair** 쉐이더 모두와 관련된 측면을 알아보겠습니다.

머리카락 렌더링 모드

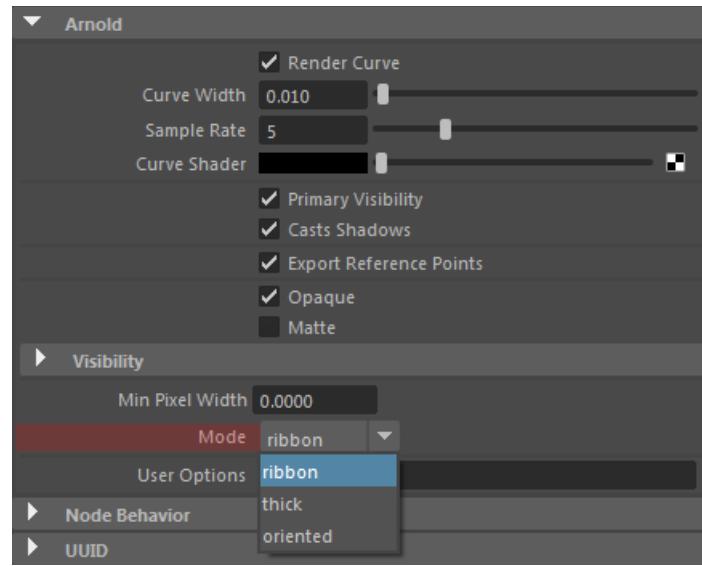
MtoA는 Arnold 곡선 프리미티브를 사용하여 머리카락을 렌더링합니다. 곡선은 **Ribbon**, **Thick**, **Oriented**, 세 가지 모드로 렌더링할 수 있습니다. 다음 이미지는 이들 사이의 차이점을 보여줍니다. 이 문서에서는 얇은 머리카락/털(이 자습서의 기반)의 렌더링에 가장 적합한 모드인 Ribbon 모드만 사용합니다.



Thick 모드. Thick 모드는 머리카락을 마치 투브처럼 만듭니다.



Ribbon 모드

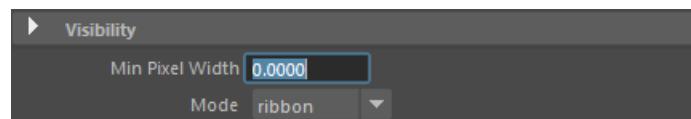


Arnold 속성에서 머리카락 렌더링 모드

최소 픽셀 너비(MPW)

머리카락 또는 털의 본질은 매우 미세하고 얇은 가닥으로 이루어져 있다는 것입니다. 일반적으로 화면 공간에 있는 것이 작을수록 더 많은 앤티앨리어싱(AA) 샘플이 필요합니다. 그 이유는 모발의 렌더링을 어렵게 만드는 깜박임 없이 부드럽게 렌더링할 수 있기 때문입니다. 샘플링에서 털 렌더링을 보다 쉽게 지원할 수 있도록 Arnold에는 "최소 픽셀 너비"(MPW)라는 기능이 있습니다.

MPW의 개념은 매우 얇은 머리카락을 화면 공간에서 자동으로 두껍게 만드는 동시에 투명하게 만드는 것입니다. 이렇게 하면 더 낮은 AA 설정에서도 머리카락이 부드럽게 렌더링되고 애니메이션이 깜박거리지 않게 됩니다. 주어진 머리카락이 지정된 MPW 값보다 화면 공간에서 더 얇을 경우에만 MPW 투명도가 적용됩니다. 예를 들어 프레임 내의 모든 머리카락이 1 픽셀 너비 이상이고 MPW 값이 0.5로 설정된 경우에는 투명도가 전혀 사용되지 않습니다. 예를 들어, 카메라가 해당 지점에서부터 멀어져 축소되기 시작하여 화면 공간에서 모든 털이 더 얇아지는 경우, 0.5픽셀 이하가 되자마자 MPW 시스템이 적용되어 머리카락이 0.5 픽셀 너비로 자동으로 유지되고 더 얇아지고 멀어질 수록 더 투명하게 됩니다.



MPW 정확성은 대부분 자동 투명도에 달려 있습니다.

아래 이미지에서는 0과 0.5 사이의 MPW 값을 사용합니다. Transparency Depth가 증가하면 머리카락이 동일한 AA 설정으로도 훨씬 더 부드러워집니다.

Transparency Depth가 0으로 설정된 경우에도 머리카락이 투명하게 보이지는 않지만 항상 화면 공간에서 최소 0.5 픽셀 두께가 됩니다.



MPW 0



MPW 0.25

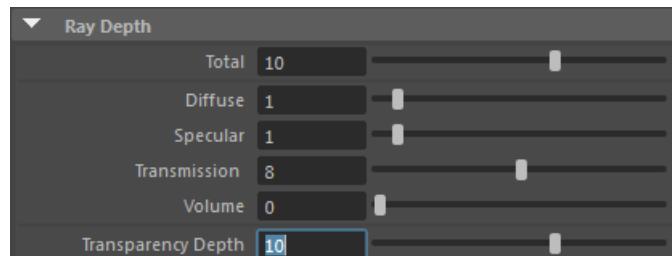


MPW 0.5

Min Pixel Width도 포인트 프리미티브에 적용됩니다. 하지만 모션 블러가 적용된 반지름이 이 효과를 비활성화 합니다.

Transparency Depth

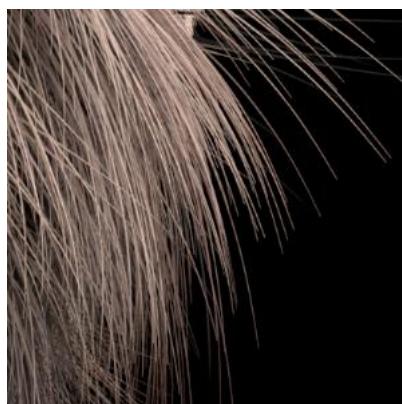
투명하게 만들 수 있는 머리카락의 층 개수는 Render Settings의 Transparency Depth에 따라 다릅니다. 아래에서 볼 수 있듯이 이 값을 조절하면 품질과 렌더링 시간 모두에 큰 영향을 미칩니다. Transparency Depth를 0으로 설정하면 장면 내 모든 머리카락에서 MPW가 비활성화됩니다.



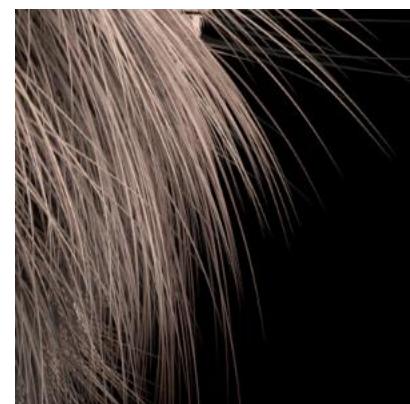
Ray Depth 설정에서 Transparency Depth(Render Settings)

참고: MPW는 Ribbon 모드의 머리카락에만 작용합니다.

아래 이미지들은 0.1의 MPW 값을 사용합니다. Transparency Depth가 4로 설정되어 있어도 머리카락이 여전히 들쭉날쭉합니다. 더 부드러운 결과를 얻으려면 더 높은 AA 설정 값이 필요합니다.



Transparency Depth 0



Transparency Depth 4

최적화

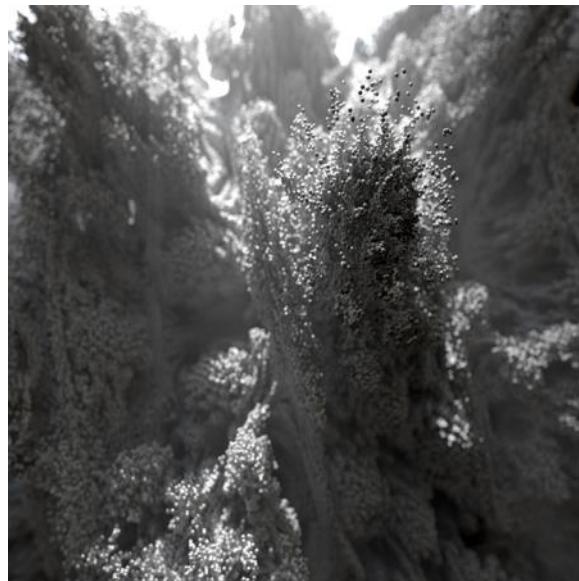
머리카락을 투명하게 만들면 렌더링 시간에 큰 영향을 미칩니다. 최적의 렌더링 속도를 위해 합리적인 값을 사용하는 것이 매우 중요합니다. 부드러운 머리카락이 어떻게 렌더링되는지는 다음 3가지에 따라 다릅니다.

- Min Pixel Width(최소 픽셀 너비)
- Transparency Depth(투명도)
- Specular(반사) 샘플의 수
- 카메라(AA) 샘플의 수

카메라(AA) 샘플이 높을수록 부드러운 모발을 얻기 위해 MPW에 의존할 필요성이 줄어듭니다. 장면이 이미 10개의 카메라(AA) 샘플을 사용하여 부드러운 모션 블러 및 DOF를 얻는 경우, 예를 들어, 4개의 카메라(AA) 샘플만 사용하는 경우보다 훨씬 작은 MPW 값을 사용할 수 있습니다.

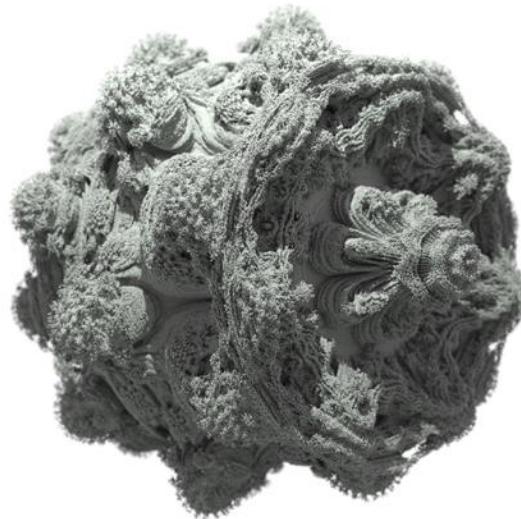
위에서 언급했듯이 MPW는 Transparency Depth가 0으로 설정된 경우에도 영향을 미칩니다. 따라서 머리카락을 투명하게 만들지 않아도 머리카락 한 가닥을 얼마나 작게 만들 것인지 대한 '한도'로 사용할 수 있습니다. MPW를 사용하여 머리카락을 굵게 만드는 위의 렌더링 시간을 보면 Transparency Depth가 0으로 설정된 경우 렌더링 시간에 긍정적 영향을 미칠 수 있습니다.

만델벌브(Mandelbulb) 렌더링 방법



이 짧은 자습서에서는 .ass 파일을 통해 만델벌브를 렌더링하는 방법을 설명합니다. 여기서는 "만델벌브" 3D 버전의 만델브로(Mandelbrot) 및 Julia set을 만들어 보겠습니다. 더 자세한 만델벌브 생성 방법은 Procedurals 자습서의 **Large Datasets from Procedurals**를 참조하십시오.

(<https://support.solidangle.com/display/A5ARP/Large+Datasets+from+Procedurals>)



Julia set은 복소 평면상의 점 집합으로, 상수(C)에 제곱되고 반복적으로 추가되면 '포로(prisoner)' 점이 반지를 2의 원을 벗어나지 못합니다. C 의 모든 가능한 값에 대해 전체 Julia set이 설정되어 있습니다. C 는 복소 평면상의 2차 원 값입니다.

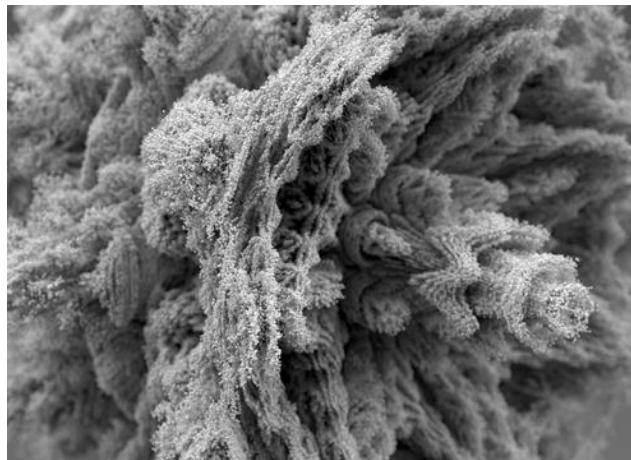
Julia set은 다음 두 가지 유형이 있습니다:

1. 완전히 연결됨
2. 연결이 끊긴 점들의 Cantor dust(Cantor dust에 대한 자세한 정보 - http://en.wikipedia.org/wiki/Cantor_set)

Julia set의 중심점(0,0)이 포로 점인 경우 Julia set이 완전히 연결된 것이며, 포로 점이 아니라면 Julia set이 Cantor Dust입니다.

만델브로 집합은 모두 완전히 연결된 Julia set인 경우의 집합입니다. 이것은 가능한 모든 Julia set의 중심점을 샘플링하여 생성됩니다. 즉, 만델브로 집합에서 모든 점에 대해 Julia set을 생성할 수 있습니다. 두 집합이 복소 평면에서 연속 함수이지만 일반적으로 짹수 격자(픽셀)를 샘플링하여 렌더링되며 해당 점이 포로 점이면 검정색으로 렌더링됩니다.

만델버브는 이 수학을 3차원으로 확장합니다 이 생성기는 일반 격자 내의 모든 점을 샘플링하여 작동하며, 해당 점이 포로 점인 경우 렌더링을 위해 AiPoints 지오메트리 노드에 추가합니다.



만델벌브의 근접 촬영. 수천 개의 작은 구체들로 구성되어 있음을 명확히 볼 수 있습니다.

다음 파일들이 있어야 만델벌브를 생성 및 렌더링할 수 있습니다.

컴파일된 만델벌브 절차 Windows | Mac | Linux 는 하단 링크 사이트에서 다운로드할 수 있습니다.
<https://support.solidangle.com/display/A5AFMUG/How+to+Render+a+Mandelbulb>

이 자습서를 따라하기 전에 Arnold의 최신 버전을 사용하고 있는지 확인하십시오. 그렇지 않으면 만델벌브를 렌더링할 때 문제가 발생할 수 있습니다.

Ass 파일

- 먼저 standin을 생성합니다(Arnold 절차로 알려짐). **Arnold>Standin>Create**로 이동합니다. 해당 standin 을 선택하고 속성 편집기에서 Mandelbulb.ass 파일을 찾습니다.

만델밸브 절차를 설치하려면 플러그인 검색 경로에 추가해야 합니다(ARNOLD_PLUGIN_PATH 환경 변수를 사용하거나 kick -i 실행). 플러그인이 로드되면 장면에서 **mandelbulb**라는 이름의 Arnold 절차 노드를 사용할 수 있습니다. Mandelbulb Arnold Procedural(standin)에 쉐이더를 지정할 수 있습니다. 조명을 만들고 장면을 렌더링합니다. 렌더링하여 만델밸브를 생성하기 전까지 잠시 시간이 걸립니다.

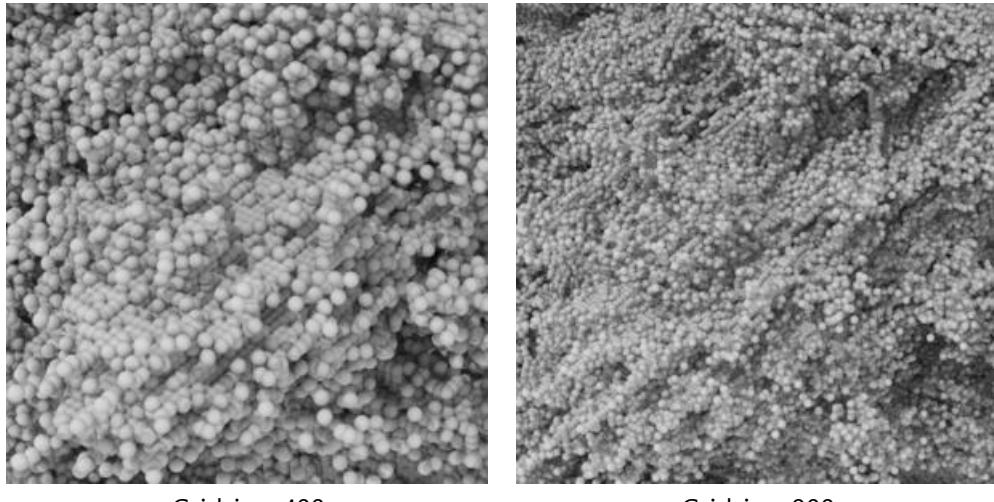
또한 Arnold Procedural에서 직접 **mandelbulb.dll** 파일을 설정하고 **User Options** 필드를 사용하여 사용자 정

의 파라미터를 정의할 수 있습니다.

Gridsize

Gridsize는 격자 안의 셀 개수입니다(800 x 800 x 800).

만델브르의 해상도를 높이려면 격자 크기를 키워야 합니다. 숫자가 클수록 구체가 더 많아집니다. 메모리 사용량과 렌더링 시간이 늘어나므로 이 숫자를 높일 때는 주의하십시오.

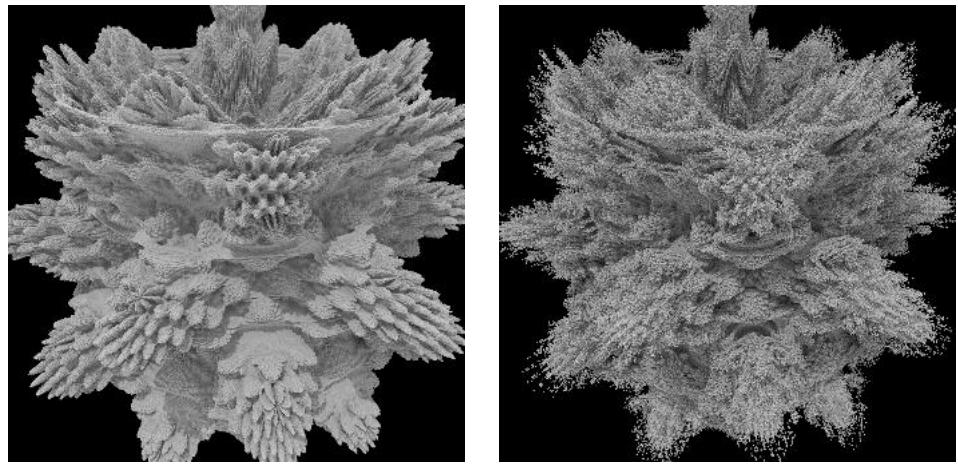


Gridsize: 400

Gridsize: 800

Maximum iterations

최대 반복 시도 횟수입니다. 이 반복 횟수 후에 점이 아직도 반지름이 2인 구체 내에 있는 경우 포로 점으로 간주합니다.

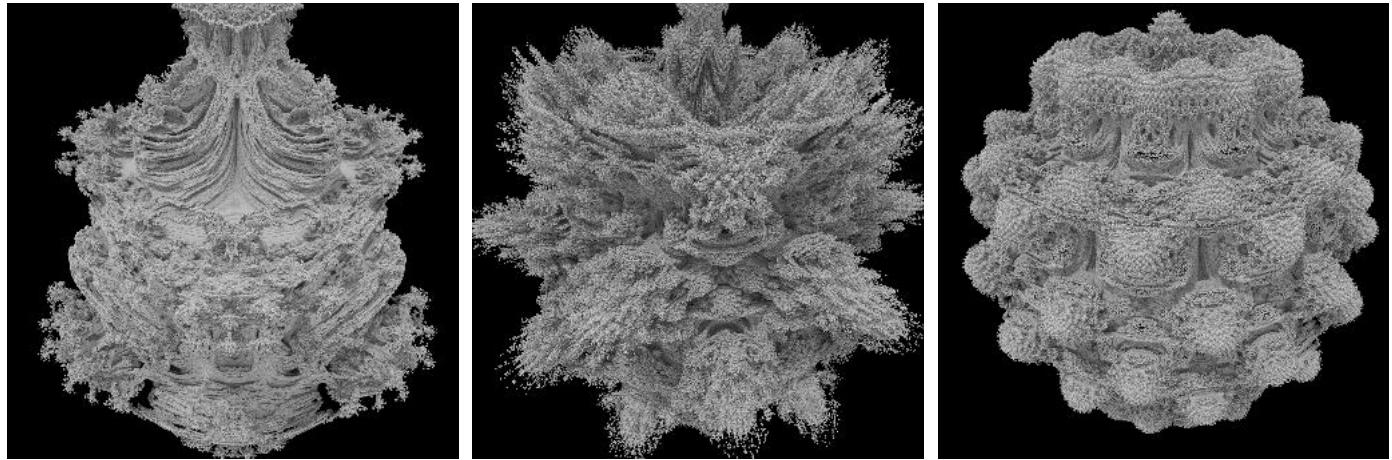


Maximum iterations: 3

Maximum iterations: 9

Power

2차원에서는 Julia set과 만델브르 집합이 C에 더해진 Z의 거듭제곱을 반복하여 계산됩니다. 3차원에서는 8의 제곱이 더 흥미로운 결과를 낳지만 원한다면 이를 변경하려고 시도할 수 있습니다.



Power: 4

Power: 8

Power: 12

Spheremult

각 셀에는 구체의 반지름에 대한 승수가 있습니다. 구체가 클수록 더 잘 섞이지만 '폭신하게' 보일 것이며 구체가 작을수록 격자가 더 많이 나타납니다.



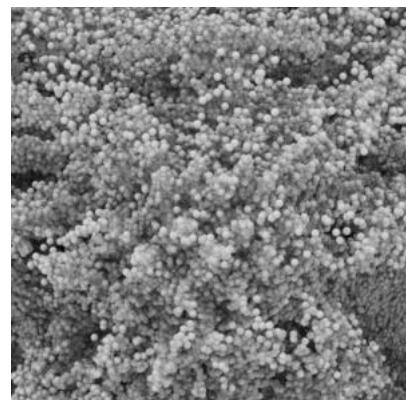
Spheremult 1



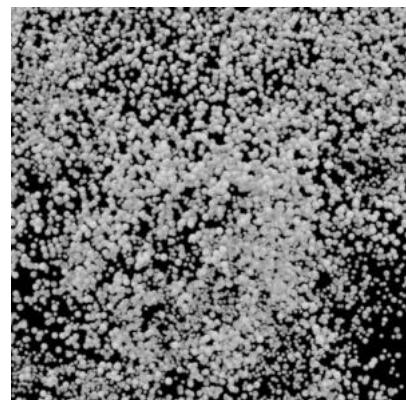
Spheremult: 2

Orbitthresh

점이 이 거리보다 더 멀리 반복되지 않으면, 만델밸브 안에 있으며 렌더링을 위해 추가되지 않습니다. 이것은 만델밸브의 중심에 큰 빈 공간을 없애서 지오메트리를 약간 더 가볍게 만듭니다.



Orbitthresh: 0.05



Orbitthresh: 2

Chunks

이것은 만델밸브를 분할할 덩어리의 개수이므로, 램에 전체 집합을 한 번에 만드는 것은 아닙니다. 이 숫자가 높

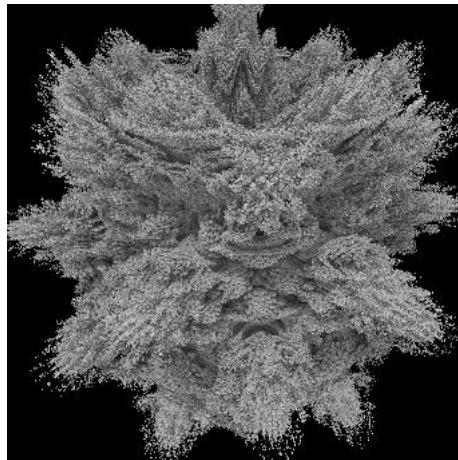
을수록 ram을 덜 사용하며 포화점은 약 30입니다.

Threads

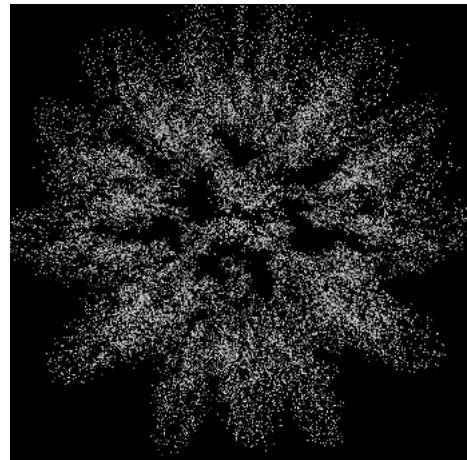
격자에서 값들을 샘플링할 때 생성될 가닥의 수입니다.

Julia

만델벌브 또는 Julia set으로 렌더링하기 위한 스위치



Julia: Off



Julia: On

Cval

Julia set을 생성하는 데 사용되는 C에 대한 값; julia == off인 경우 무시됨

이것으로 본 자습서를 마무리하겠습니다. .ass 파일에서 눈금 크기를 키우고 아래 예제 이미지에서처럼 좀 더 재미있는 조명 효과를 추가해 보세요.

MakeTX: 나무 바닥



이 자습서에서는 간단한 나무 바닥 장면을 사용하여 maketx를 통해 텍스처를 사전 처리할 때의 장점에 대해 설명합니다. 이 자습서는 다음 페이지들로 구성되어 있습니다.

- [파트 1 – Scene 셋업](#)
- [파트 2 – Render & Conversion](#)

Arnold에서 최상의 결과를 얻으려면 이미지를 바둑판식으로 배열된 MIP 매핑 텍스처로 변환하는 것이 좋습니다. 고해상도의 압축되지 않은 텍스처 맵은 해상도가 낮지 않고 거리에 관계없이 메모리에 로드해야 하기 때문에 렌더링에는 매우 비효율적입니다. 마찬가지로, 텍스처 맵 역시 렌더러가 하나의 텍셀에 액세스해야 할 때 완전히 메모리에 로드되어야 하기 때문에 비효율적입니다. 이 자습서의 끝에서는 렌더링 시간을 절반 이하로 줄일 수 있습니다.

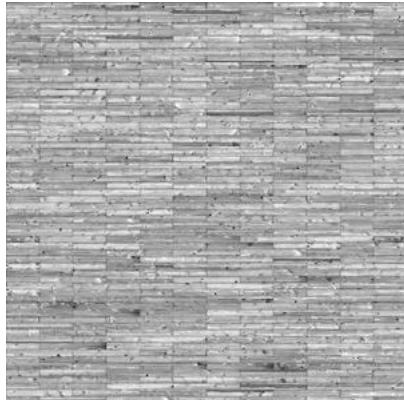
FORMAT	AUTO_MIPMAP	RENDER TIME	FILE I/O	PEAK CACHE MEMORY
JPG	On	43s	4m 10.4s	394.9 MB
JPG	Off	36s	1m 47.1s	173.5 MB
TX	-	20s	0m 0.6s	34.5 MB

기본적으로 Maya 2017에서는 각 이미지 쉐이더에 대해 타일링되고 mipmap된 TX 텍스처가 자동으로 생성됩니다.

파트 1 – Scene 셋업

이 자습서에서는 Standard Surface 쉐이더에 연결된 고해상도 diffuse, bump 및 specular 맵을 사용하여 위의 렌더링을 다시 만드는 방법을 알아봅니다. Skydome 조명을 사용하여 나무 재료로 사실적인 정반사를 만들어보겠습니다.

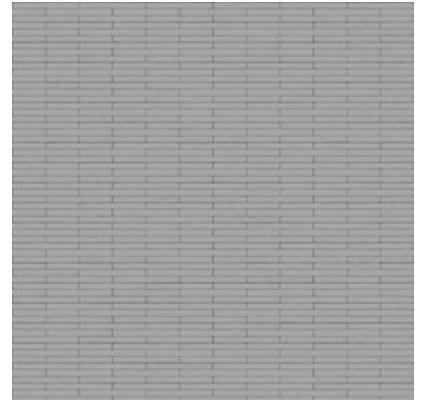
- 먼저 Standard Surface 쉐이더에 연결할 고해상도(6k) 나무 텍스처를 다운로드합니다.



Specular 맵



Diffuse 맵



Bump 맵

- 먼저 바닥에 대한 폴리곤 평면을 만들고 Standard Surface 쉐이더를 지정합니다.
- Skydome 조명을 만들고 HDR 맵을 Color 속성에 연결합니다.

Standard Surface

- Diffuse map을 **Base Color** 속성에 연결합니다.
- Specular map을 **Specular Color** 및 **Specular Weight** 속성에 연결합니다.
- Bump map을 **Bump Mapping** 속성에 연결합니다.

Bump 매핑

매우 미세한 **Bump depth** 값만 필요합니다.



Bump Depth: 1 (너무 높음)



Bump Depth: 0.1

파트 2 – Render & Conversion

이 자습서를 시작하려면 여기서 텍스처와 이전 섹션의 결과를 다운로드해야 합니다.

미처리 텍스처를 사용하여 렌더링

장면이 완료되었으므로 이제 렌더링할 수 있습니다.

텍스처가 사용될 때 로그에 OpenImageIO ImageCache의 통계가 표시됩니다.

```
00:00:43 536MB | OpenImageIO ImageCache statistics (0x7f3b5c00c2c0) ver 0.10.19
00:00:43 536MB | Images : 3 unique
00:00:43 536MB |   ImageInputs : 3 created, 3 current, 3 peak
00:00:43 536MB |   Total size of all images referenced : 208.9 MB
00:00:43 536MB |   Read from disk : 203.2 MB
00:00:43 536MB |   File I/O time : 4m 10.4s (4.5s average per thread)
00:00:43 536MB |   File open time only : 0.0s
00:00:43 536MB |   Tiles: 545 created, 518 current, 521 peak
00:00:43 536MB |   total tile requests : 341522036
00:00:43 536MB |   micro-cache misses : 26264003 (7.69028%)
00:00:43 536MB |   main cache misses : 545 (0.00015958%)
00:00:43 536MB |   Peak cache memory : 394.9 MB
00:00:43 536MB |   3 not tiled, 3 not MIP-mapped
```

시스템의 성능에 따라 통계가 달라집니다. 나중에 비교로 사용할 통계를 메모하십시오.

- 렌더링 시간 - 00:00:43

- 파일 I/O 시간: 4분 10.4초

- 최고 캐시 메모리: 394.9 MB

Auto-mipmap

렌더링이 시작되기 전에 상당한 지연이 있음을 알 수 있습니다. 이것은 **Auto-mipmap**이 기본적으로 활성화되어 있기 때문입니다. 텍스처 맵 파일이 고유하게 mipmap되지 않은 경우(현재 사용중인 것과 같이), 이 옵션을 켜면 렌더링 기간 동안 메모리에 저장될 mipmap이 자동으로 생성됩니다. 가장 최근에 사용된 타일만 기본 크기 1024MB의 텍스처 캐시로 메모리에 보관됩니다. 이 mipmap 생성은 특히 고해상도 텍스처가 많은 장면의 렌더링 시간을 늘립니다. 여기서 비활성화할 수 있습니다.

- 다시 렌더링하고 로그를 살펴봅니다.

```
00:00:36 241MB | OpenImageIO ImageCache statistics (0x7f522800c2c0) ver 0.10.19
00:00:36 241MB | Images : 3 unique
00:00:36 241MB |   ImageInputs : 3 created, 3 current, 3 peak
00:00:36 241MB |   Total size of all images referenced : 208.9 MB
00:00:36 241MB |   Read from disk : 118.8 MB
00:00:36 241MB |   File I/O time : 1m 47.1s (1.9s average per thread)
00:00:36 241MB |   File open time only : 0.0s
00:00:36 241MB |   Tiles: 176 created, 151 current, 151 peak
00:00:36 241MB |   total tile requests : 165597670
00:00:36 241MB |   micro-cache misses : 17463992 (10.546%)
00:00:36 241MB |   main cache misses : 176 (0.000106282%)
00:00:36 241MB |   Peak cache memory : 173.5 MB
00:00:36 241MB |   3 not tiled, 3 not MIP-mapped
```

- 렌더링 시간 - 00:00:36

- 파일 I/O 시간: 1분 47.1초

- 최고 캐시 메모리: 173.5 MB

따라서 속도는 더 빠르지만 텍스처를 변환할 때 어떤 효과가 나타날지 살펴보겠습니다.

MakeTX로 텍스처 변환

텍스처가 OpenImageIO의 일부인 명령줄 유ти리티인 maketx로 변환됩니다.

- Linux에서는 터미널을 열고 텍스처 위치로 이동한 후 다음 명령을 실행합니다.

maketx -v --oiiو wood-flooring-041_d.jpg -o wood-flooring-041_d.tx

-v 플래그는 상태 메시지를 출력하기 위한 자세한 정보입니다. --oiiو 플래그는 타일 크기를 OpenImageIO(64x64)에 맞게 강제 설정하여 더 빨리 렌더링할 수 있는 Arnold에 최적화된 .tx 파일을 생성합니다. 또한 인터리브된 채널(RGBARGBA ...)을 저장하여 읽기가 더 빨라지고 지속적인 색상 감지가 가능합니다. 이미지가 일정한 색상이면 작은 단일 색상 .tx 파일로 변환합니다. -o은 출력 파일 앞에 와야 합니다.

```
[michael@ws1 Flooring]$ maketx -v --oiiو wood-flooring-041_d.jpg -o wood-flooring-041_d.tx
Reading file: wood-flooring-041_d.jpg
  SHA-1: 8B295EB7453A1DEF5F70DE6A172BAF2944AD8CD3
  Writing file: wood-flooring-041_d.tx
  Filter "box" width = 1
  Mipmapping...
    3000x3000
    1500x1500
    750x750
    375x375
    187x187
    93x93
    46x46
    23x23
    11x11
    5x5
    2x2
    1x1
  Wrote file: wood-flooring-041_d.tx
maketx Runtime statistics (seconds):
  total runtime: 15.82
  file read: 1.78
  file write: 11.34
  initial resize: 0.90
  mip computation: 0.45
  unaccounted: 1.35
maketx memory used: 1021.9 MB
```

- 다른 텍스처들도 동일한 방식으로 변환합니다.

텍스처 교체

모든 이미지 노드를 살펴보고 텍스처를 업데이트할 수도 있지만 더 빠른 방법이 있습니다. **Render Settings** 창을 열고 **Textures** 아래에서 **Use Existing .tx Textures**를 활성화합니다.

미처리 텍스처가 여전히 쉐이더에 지정되어 있지만 이 파라미터를 사용하면 Arnold는 동일한 파일 이름이지만 .tx 확장자를 가진 파일을 검색합니다.

다시 렌더링해 보십시오. 렌더링이 훨씬 더 빨라진 것을 알 수 있습니다. 로그에서 통계를 확인할 수 있습니다.

```
00:00:20 102MB | OpenImageIO ImageCache statistics (0x7fdb4800c2c0) ver 0.10.19
00:00:20 102MB | Images : 3 unique
00:00:20 102MB | ImageInputs : 3 created, 3 current, 3 peak
00:00:20 102MB | Total size of all images referenced : 278.5 MB
00:00:20 102MB | Read from disk : 34.5 MB
00:00:20 102MB | File I/O time : 0.6s (0.0s average per thread)
00:00:20 102MB | File open time only : 0.0s
00:00:20 102MB | Tiles: 3889 created, 3838 current, 3838 peak
00:00:20 102MB | total tile requests : 265906254
00:00:20 102MB | micro-cache misses : 30221065 (11.3653%)
00:00:20 102MB | main cache misses : 3889 (0.00146255%)
00:00:20 102MB | Peak cache memory : 34.5 MB
```

- 렌더링 시간 - 00:00:20
- 파일 I/O 시간: 0.6초
- 최고 캐시 메모리: 34.5 MB

원래 렌더링과 비교해서 이제 렌더링 시간이 절반으로 줄었습니다. 파일 I/O는 사실상 아무 것도 아니며 사용된 캐시 메모리는 원래의 9%입니다. 렌더링 시간을 크게 단축할 수 있기 때문에 텍스처를 변환하기 위해 처음에 조금 더 시간을 들이는 것이 그럴 만한 가치가 있음을 알 수 있습니다.

HDRI 및 Backplate를 사용하여 자동차 렌더링하기



바닥에 aiShadow Catcher 노드를 사용하여 이미지 평면에 렌더링된 자동차

이 자습서에서는 배경 이미지 위에 자동차나 다른 객체를 렌더링하는 방법을 설명합니다. 또한 Ai Shadow Catcher 노드를 사용하여 그림자 레이어를 렌더링하는 방법에 대해서도 알아봅니다. HDRI로 장면에 조명을 적용해 보겠습니다. HDRI와 백플레이트는 Jeff Patton (<http://jeffpatton.net/>)이 제공해주셨으며 무료로 다운로드할 수 있습니다.

프로세스를 몇 가지 단계로 나눠보겠습니다.

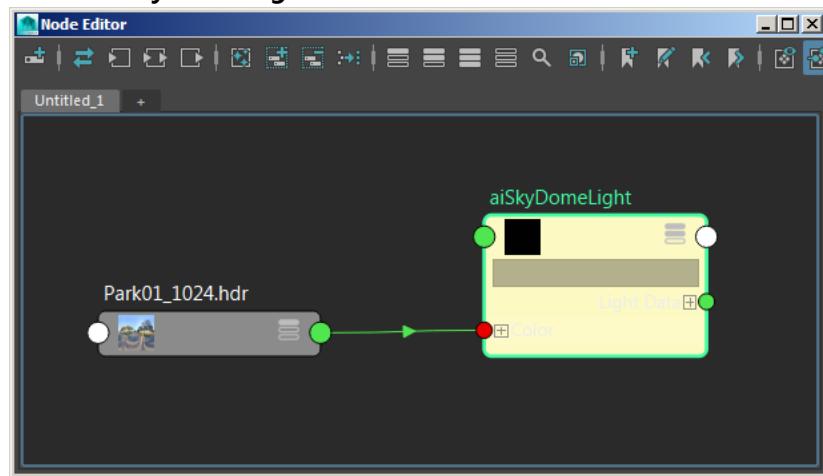
- [HDRI 준비하기](#)
- [Shadow Matte 생성하기](#)
- [최종 렌더링](#)

HDRI와 백플레이트를 다운로드하고 직접 시도해 보세요!

HDRI 준비하기

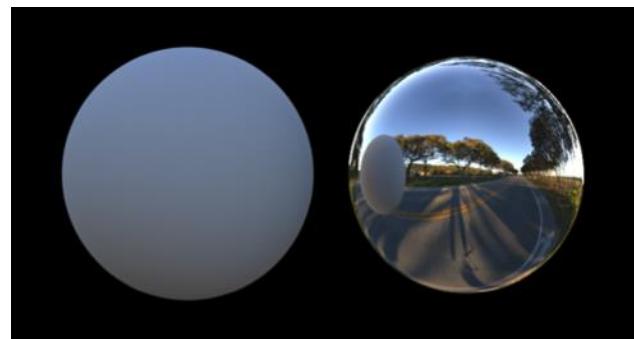
먼저 HDRI 맵을 테스트하겠습니다. 자동차를 바로 렌더링하기 보다는(렌더링이 더 오래 걸림) 두 영역을 렌더링 할 수 있습니다. 하나의 구체는 조명을, 다른 하나는 광택 반사를 테스트하기 위한 것입니다.

- **Standard Surface** 쉐이더를 아래 이미지에서처럼 구체 하나에 적용합니다. Specular Weight를 0으로 줄입니다. 이것은 조명과 그림자를 테스트하는 데 사용됩니다. HDRI 맵의 소스에 따라 장면 내의 조명 노출을 늘리거나 줄여야 할 수 있습니다. 이 회색 구체는 설정된 조명 조건에 맞추려고 할 때도 유용합니다(회색과 크롬 구체는 나중에 3D에서 조명 조건을 일치시키기 위한 세트에서 사용할 수 있음).
- 또 다른 **Standard Surface** 쉐이더를 다른 구형에 할당합니다. 이 재료를 크롬처럼 보이게 하려면 Metalness를 1로 늘리고 Base Weight를 1로 늘리고 Specular Roughness를 0으로 낮춥니다.
- **Skydome** 조명을 만들고 HDRI 맵을 **Skydome** 조명의 **Color** 속성에 연결합니다. 현재 사용 중인 HDRI 맵은 **lat-long** 형식입니다. **Resolution**을 높여서 HDRI 맵의 더 큰 너비 해상도에 맞춥니다. 이 경우에는 4096입니다(기본적으로 이 값은 1000으로 설정됩니다. 이것은 1000 픽셀보다 큰 환경 맵을 사용하는 경우 **Skydome** 조명에서 **Resolution**을 그에 맞게 조정해야 함을 의미합니다). 예를 들어, 환경 맵이 3000*1500 픽셀인 경우 **Skydome Light**의 **Resolution**을 3000으로 설정해야 합니다.



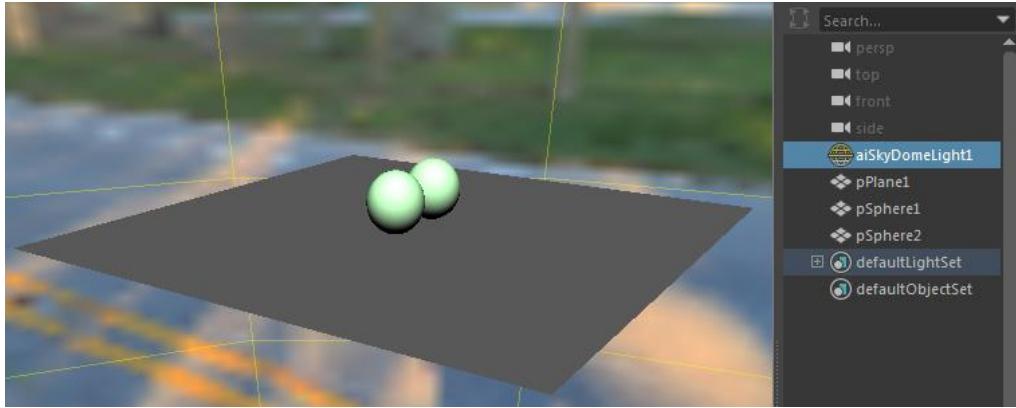
Ai SkyDomeLight의 색상 속성에 연결된 HDRI 맵

- 장면을 렌더링하면 아래와 같은 이미지가 생성됩니다. 하얀 구체가 너무 어둡게 보인다면 노출을 증가시켜야 할 수도 있습니다. 이 경우에는 노출을 2로 높였습니다.

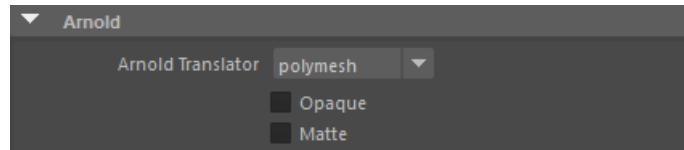


Shadow Matte 생성하기

- 이제 장면에 바닥 그림자가 필요합니다. 장면에서 그림자를 커버할 만큼 큰 바닥 평면을 만듭니다.



- 바닥 평면을 선택하고 Arnold 속성에서 **Opaque**을 비활성화하십시오. 이것은 **Shadow Matte**를 작동시키려면 필요한 단계입니다.



바닥 평면에 대해 Opaque를 비활성화해야 합니다.

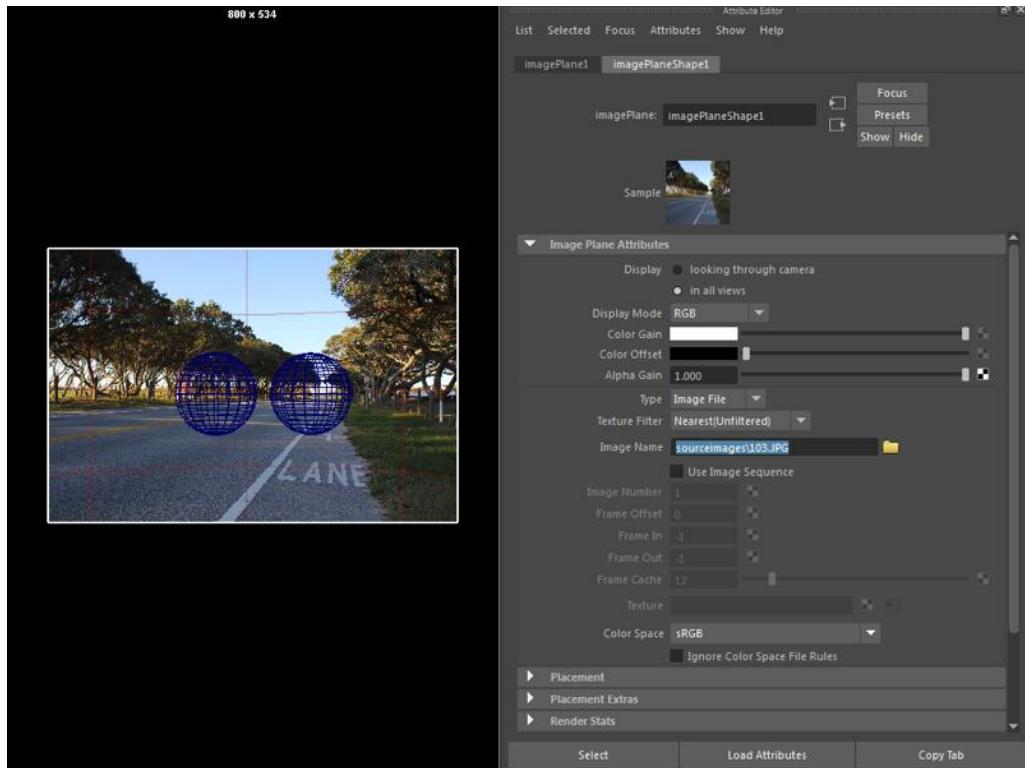
Shadow Matte

- **Shadow Matte** 쉐이더를 만들고 바닥 평면에 지정합니다.



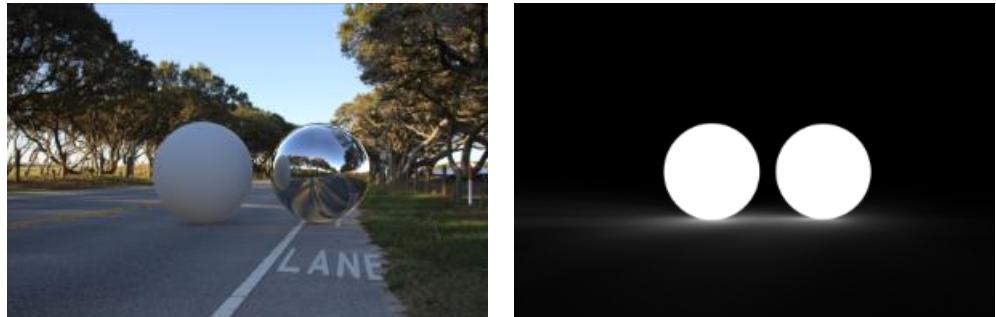
Image Plane

- 렌더링 카메라를 선택하고 'Environment' 아래에 새 이미지 평면을 만듭니다. 아래와 같이 이미지 평면 내에서 사진 백플레이트를 엽니다. 이 예에서는 '103.jpg'를 사용합니다.



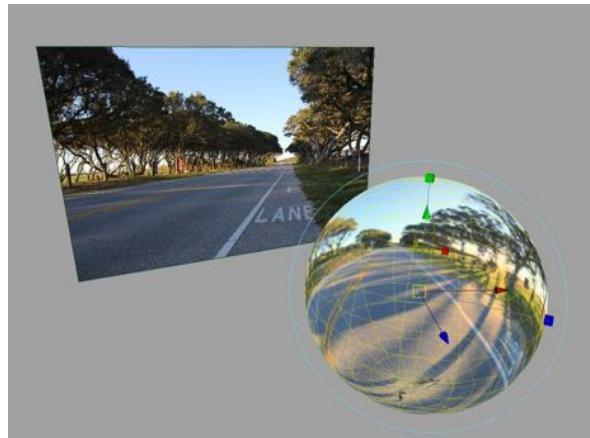
Maya에서 이미지 평면 열기

- 장면을 렌더링합니다. 이미지 평면에 Shadow Matte 쉐이더에 의해 생성된 그림자로 렌더링된 구체들이 보입니다. 다음과 같습니다.



Shadow Matte 쉐이더를 사용하여 백플레이트에서 렌더링된 구체바닥 그림자가 표시되는 알파 채널(이미지 평면 숨겨짐).

- 반사가 부적절해 보이는 경우 Skydome 조명을 백플레이트와 일치할 때까지 회전합니다.

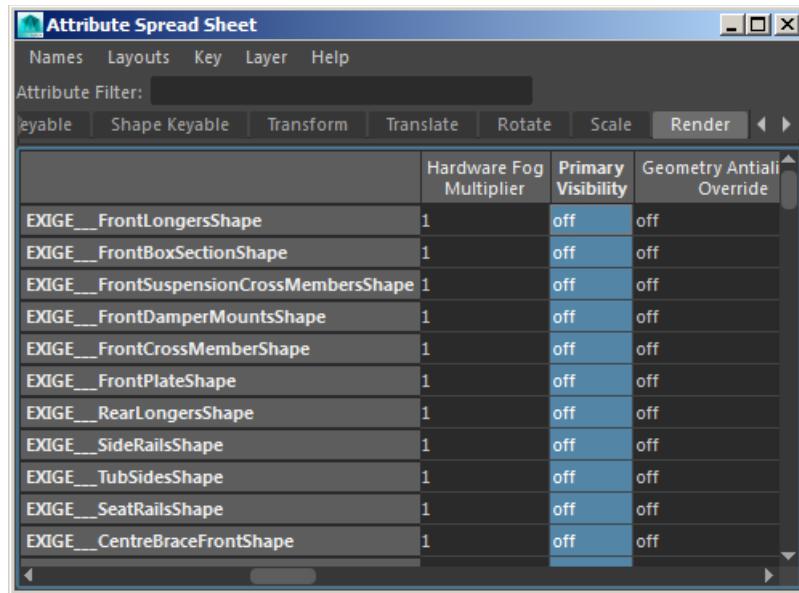


최종 렌더링

마지막으로 가능한 최상의 결과를 얻기 위해 미세 조정할 수 있는 여러 가지 항목이 있습니다.

쉐도우 렌더링

- 그림자를 별도로 렌더링하려면 먼저 모든 자동차 오브젝트에 대한 Primary Visibility를 비활성화해야 합니다. 자동차를 선택하고 Maya의 Attribute Spreadsheet 창을 엽니다. Primary Visibility 열을 선택하고 선택한 열에서 **off** 또는 **0**을 입력합니다.



The screenshot shows the Maya Attribute Spreadsheet window with the following data:

	Hardware Fog Multiplier	Primary Visibility	Geometry Antialiasing Override
EXIGE__FrontLongersShape	1	off	off
EXIGE__FrontBoxSectionShape	1	off	off
EXIGE__FrontSuspensionCrossMembersShape	1	off	off
EXIGE__FrontDamperMountsShape	1	off	off
EXIGE__FrontCrossMemberShape	1	off	off
EXIGE__FrontPlateShape	1	off	off
EXIGE__RearLongersShape	1	off	off
EXIGE__SideRailsShape	1	off	off
EXIGE__TubSidesShape	1	off	off
EXIGE__SeatRailsShape	1	off	off
EXIGE__CentreBraceFrontShape	1	off	off

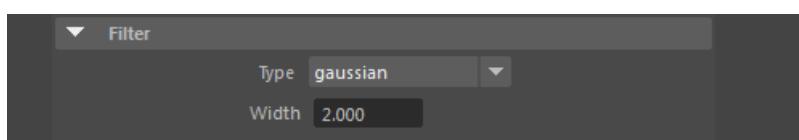
Attribute Spreadsheet에서 Primary Visibility 비활성화

- 자동차를 렌더링하면 보이지 않게 렌더링되지만 그림자는 그대로 유지됩니다. 알파 채널은 아래와 같이 렌더링되어야 합니다.



Ai Shadow Matte 렌더러

Filter Width



기본적으로 Arnold는 **Filter Width** 값을 2로 렌더링합니다. 그러나 렌더링할 때 자동차가 원래 사진용 백플레이트와 비교하여 약간 선명하게 보입니다. 또한 자동차 가장자리의 앤티앨리어스가 사진 배경에 대해 너무 날카롭게 보입니다. 포스트에서 이 문제를 해결하기 위해 노력할 수 있지만 포스트 내의 이미지를 흐리게 처리하면 픽셀들이 평균화되고 불만족스러운 결과가 나올 수 있습니다. 또한 포스트 프로덕션 작업에 불필요한 단계가 추가됩니다. 더 나은 방법은 더 넓은 Filter Width 설정으로 렌더링하는 것입니다.

다음은 Filter Width 2와 3으로 렌더링된 동일한 장면을 비교한 것입니다.



Filter Width 2. 렌더링이 백플레이트에 대해 너무 날카로움.



Filter Width 3. Filter Width를 높이면 백플레이트에 대해 렌더링이 혼합됩니다.

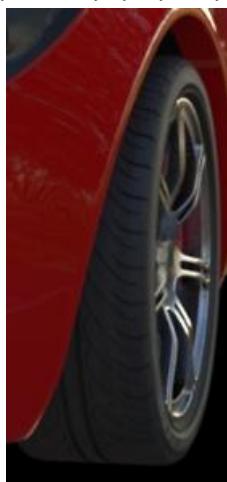
3의 값이 약간 희미하게 보일 때는 2.5의 Filter Width 값이 바람직합니다. 이런 방식으로 실험해볼 수 있으며 물론 장면에 따라 달라집니다. 이 경우에는 사용되는 HDR 맵의 해상도가 원하는 것보다 작습니다. 이것은 자동차의 재료에서 저해상도의 반사를 일으킬 수 있습니다. 이 경우에는 Filter Width를 높이는 것도 도움이 됩니다.

Shadow Matte: Indirect Diffuse

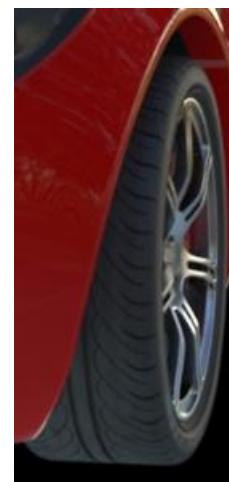
사용하는 HDR 맵과 백플레이트에 따라, 타이어에서 접지면과 맞닿는 부분이 사진 배경 내 빛과 비교해서 약간 어둡게 보일 수 있습니다. 더 많은 빛이 바닥에서부터 타이어의 바닥으로 반사될 것입니다.

Shadow Matte shader의 Indirect Diffuse 속성을 높임으로써 이러한 문제를 해결할 수 있습니다. 색상을 수동으로 선택하거나 Diffuse Color를 선택하고, 색상 선택기로 이미지 평면 배경의 지면에 있는 픽셀을 클릭하십시오. 두 번째 방법을 통해 동일한 색상이 타이어에 다시 반사되도록 할 수 있습니다.

다음은 이러한 효과의 극단적인 예입니다.



Indirect Diffuse: 0

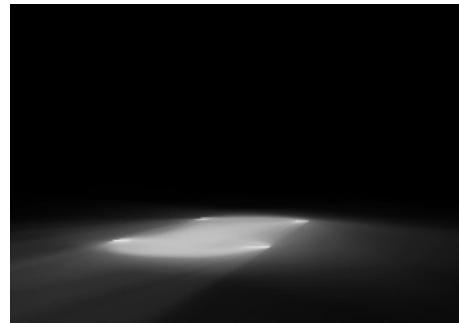


Indirect Diffuse: 1

이제 자동차를 백플레이트에 합성하는 데 필요한 렌더링 요소가 모두 있습니다.



Car beauty 렌더링



Shadow Matte 렌더링



사진 백플레이트

아래 이미지는 6개의 AA 샘플을 사용하여 렌더링한 최종 결과입니다. 최종 렌더링을 위해 Ai SkyDomeLight 샘플이 3으로 증가되었습니다.



최종 합성된 렌더링



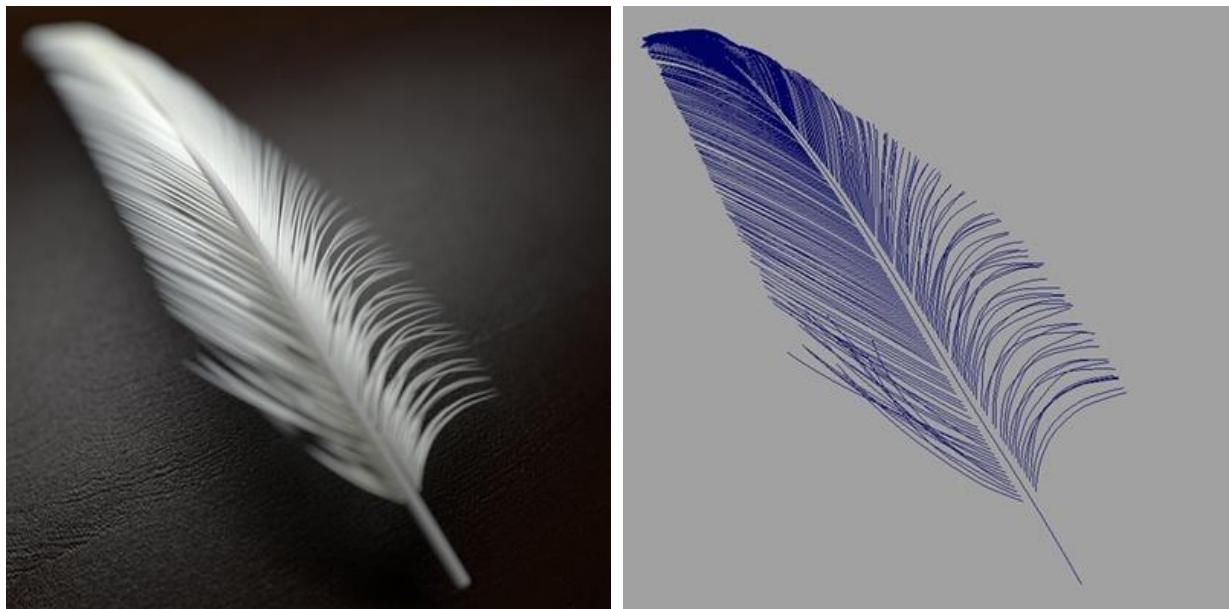
위와 동일한 워크플로우를 사용하여 다른 백플레이트의 렌더링을 시도할 수 있습니다.

Curve 렌더링하기



- [깃털처럼 Curve 렌더링하기](#)
- [Override Sets을 사용하여 Paint Effect Curves 렌더링하기](#)
- [Render Curve를 사용하여 추상적인 머리 렌더링](#)

깃털처럼 Curve 렌더링하기



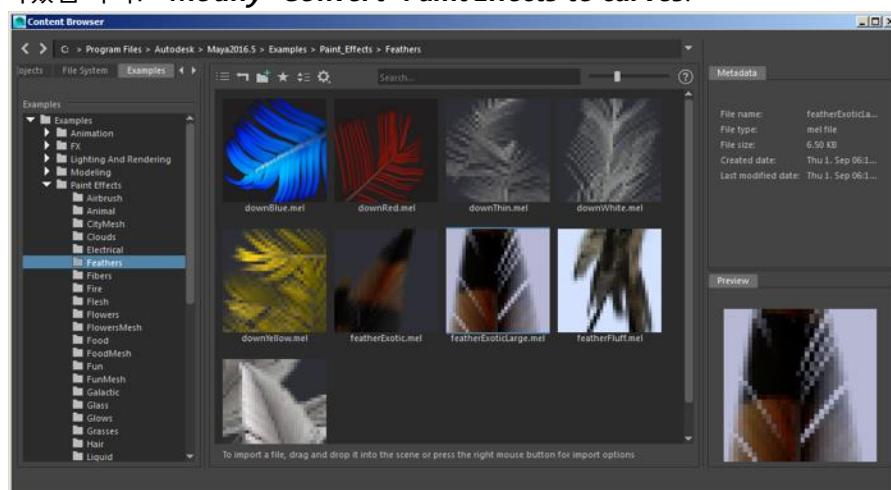
Curve으로 렌더링된 깃털

이 자습서에서는 곡선으로 변환된 페인트 효과 브러시를 사용하여 깃털을 렌더링하는 방법에 대해 설명합니다.

Maya 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Rendering+Curves+as+Feathers>

- 곡선들은 처음에는 **Paint Effects** 브러시 - *strokeFeatherExoticLarge.mel*로 생성되었으며, 그 다음에는 곡선으로 변환되었습니다. - **Modify>Convert>Paint Effects to curves**.

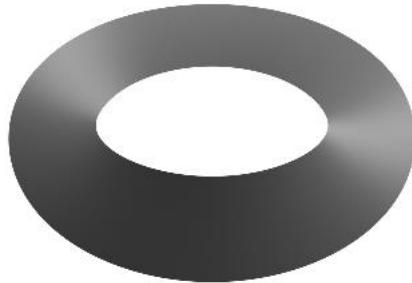


- 이 많은 커브를 렌더링하려면 먼저 해당 커브에 대한 **Override set**을 만들어야 합니다. 이렇게 하면 각각의 개별 곡선에 추가하지 않고도 다음과 같은 '재정의 속성'을 곡선에 추가할 수 있습니다.

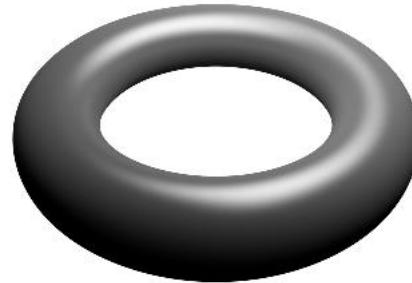
여러 개의 곡선을 렌더링하는 더 빠르고 효율적인 방법은 **Curve Collector**를 사용하는 것입니다.

Mode

Arnold에서 곡선을 렌더링하는 데는 **Ribbon**, **Thick** 두 가지 알고리즘이 있습니다. 기본 줄기는 **Thick** 모드를 사용하여 곡선을 렌더링해야 합니다.



Mode: Ribbon

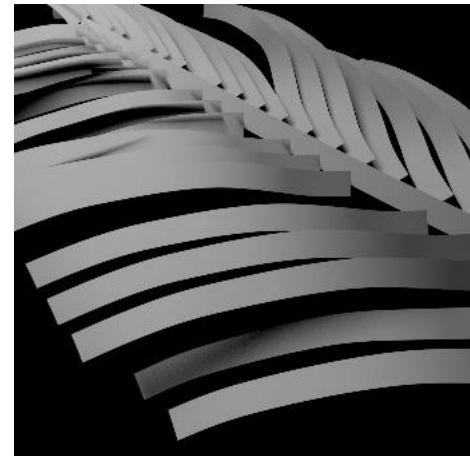


Mode: Thick

개별 깃가지는 **Ribbon** 모드로 렌더링할 때 더 잘 표현됩니다.



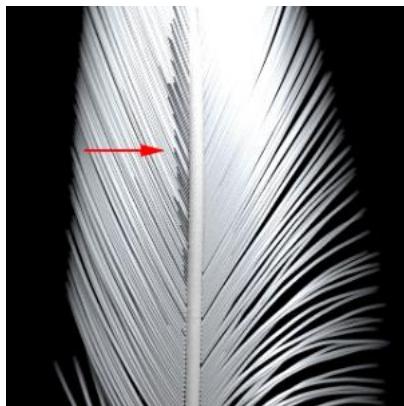
Mode: Thick



Mode: Ribbon. 깃가지 렌더링에 보다 적합.

Ai Curve Width

Ai Curve Width가 너무 높게 설정되어 있으면 곡선들이 겹치기 시작하고 샘플링 오류가 발생합니다.



Ai Curve Width: 0.01. 곡선들이 겹칩니다.



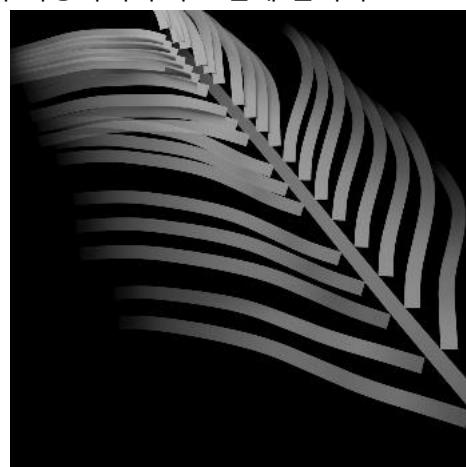
Ai Curve Width를 0.005로 줄이면 문제가 해결 됩니다.

Opacity

Standard Hair의 *Opacity*에 램프 텍스처 맵을 추가하면 곡선의 가장자리가 부드럽게 됩니다.



불투명도 없음



Standard Hair 쉐이더의 Opacity 속성에 연결된
램프 텍스처

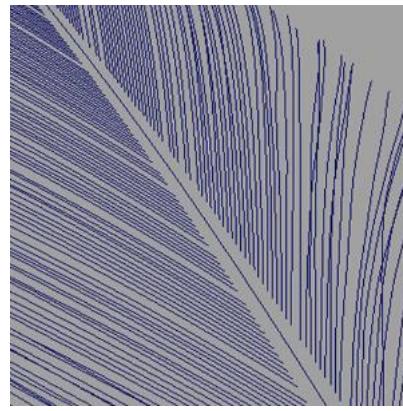
Ai Opaque 속성이 세트에 추가되어야 하며 불투명도가 제대로 작동하려면 이것이 비활성화되어야 합니다.

Stem

*curve width*를 깃털 곡선 너비의 최소 두 배가 되도록 늘립니다(이 경우에는 **0.04**의 값이 사용됨). 모드를 **Thick**으로 변경하여 곡선이 원형으로 보이게 합니다. 깃털의 기본 줄기는 일반적인 광택 재료가 필요합니다. Ai Standard Surface 쉐이더를 줄기 곡선의 *Curve Shader*에 연결합니다.



'Thick' 모드로 렌더링된 줄기 곡선



와이어프레임

쉐이딩

Standard Hair 쉐이더를 생성하고 곡선 세트의 *Ai Curve Shader* 속성에 연결합니다.

Root Color

아래 이미지는 뿌리 색을 변경할 때의 효과를 보여줍니다.



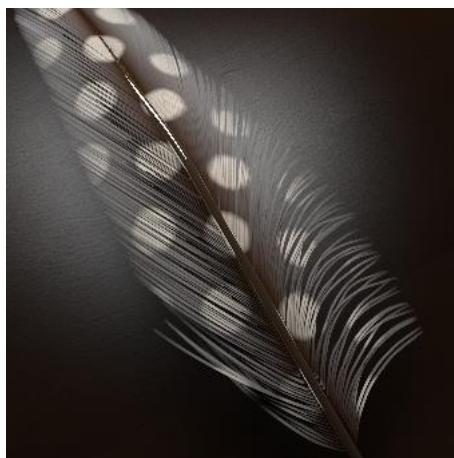
Root Color: 흰색



Root Color: 검정

텍스처

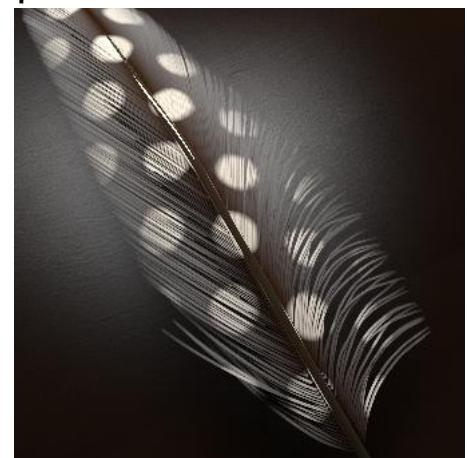
다음 이미지들은 Standard Hair 쉐이더의 *Specular Tint*에 유색 *circular ramp*를 추가할 때의 효과를 보여줍니다.



Specular color에 연결된 램프 텍스처



Root Color에 연결된 램프 텍스처



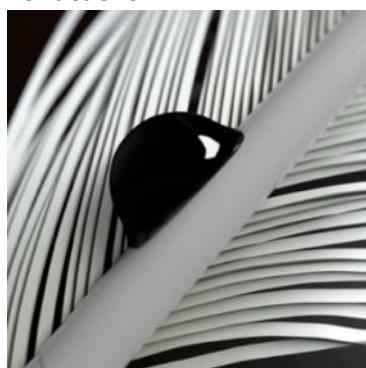
Root 및 Specular color에 연결된 램프 텍스처

SSS는 폴리 메시에서만 작동하므로 곡선 프리미티브에 의해서는 지원되지 않습니다.

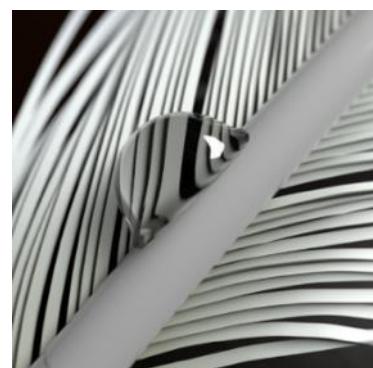
Visible in Specular Reflections / Transmission

광택이 있는 반사 및 굴절에서 곡선을 보려면 다음과 같은 Override Attributes을 추가해야 합니다.

- Ai Visible in Glossy
- Visible in Refractions



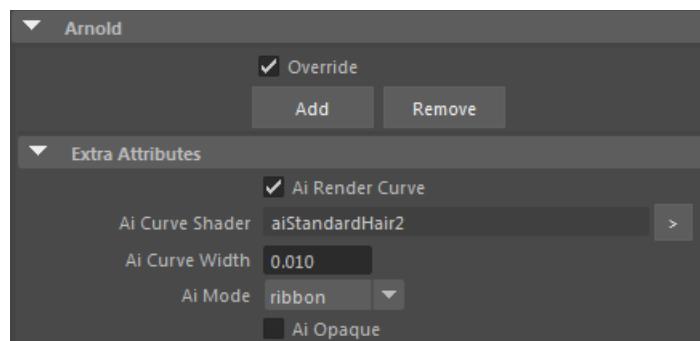
Ai Visible in Glossy/Visible in Refractions: 꺼짐



곡선은 기본적으로 반사에서 보이지 않습니다. 곡선이 반사되기를 원한다면 곡선들이 포함된 Override Set에 **Visible in Reflections** 속성을 추가해야 합니다.

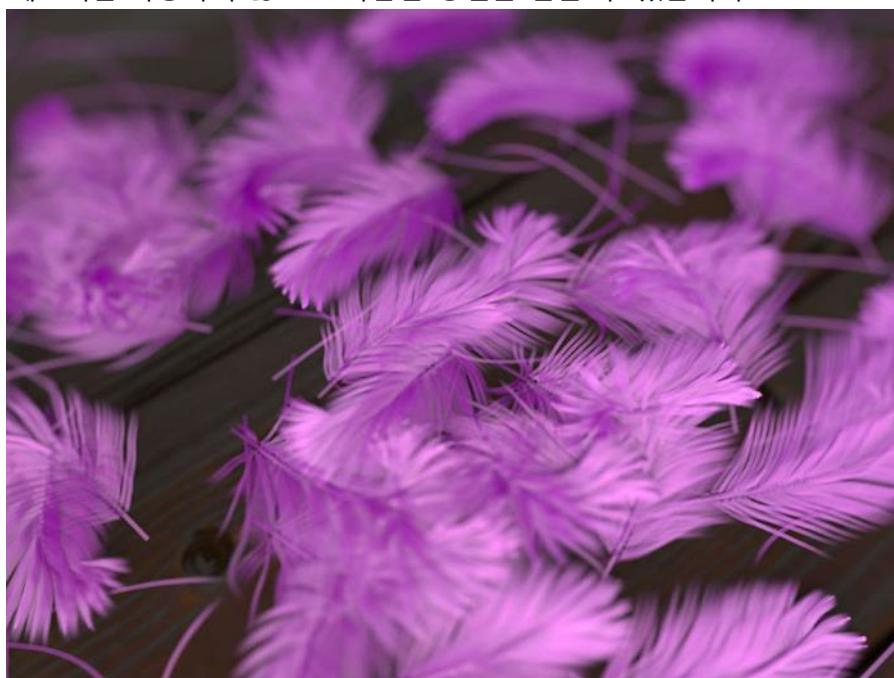


반사에서 곡선이 보이지 않습니다. 곡선의 Override set에 추가된 'Visible in Reflections' 재정의



Arnold Set 속성

지금까지 깃털로 곡선을 렌더링하는 방법을 알아봤습니다. Override Set을 사용하여 곡선을 머리카락으로 렌더링하면 많은 양의 지오메트리를 사용하지 않고도 복잡한 장면을 만들 수 있습니다.



Override Sets을 사용하여 Paint Effect Curves 렌더링하기



곡선으로 변환되고 Override Set을 사용하여 렌더링된 Paint Effects

이 자습서에서는 Maya의 Paint Effects를 MtoA의 Override Set과 함께 사용하여 복잡한 '추상' 곡선 장면을 렌더링하는 방법을 알아봅니다. Maya의 Paint Effects 도구를 사용하여 흥미로운 모양을 만들어보겠습니다. 그런 다음 결과를 곡선으로 변환하고 Override Set을 사용하여 매우 빠르고 쉽게 렌더링합니다. 그럼 시작해 보겠습니다.

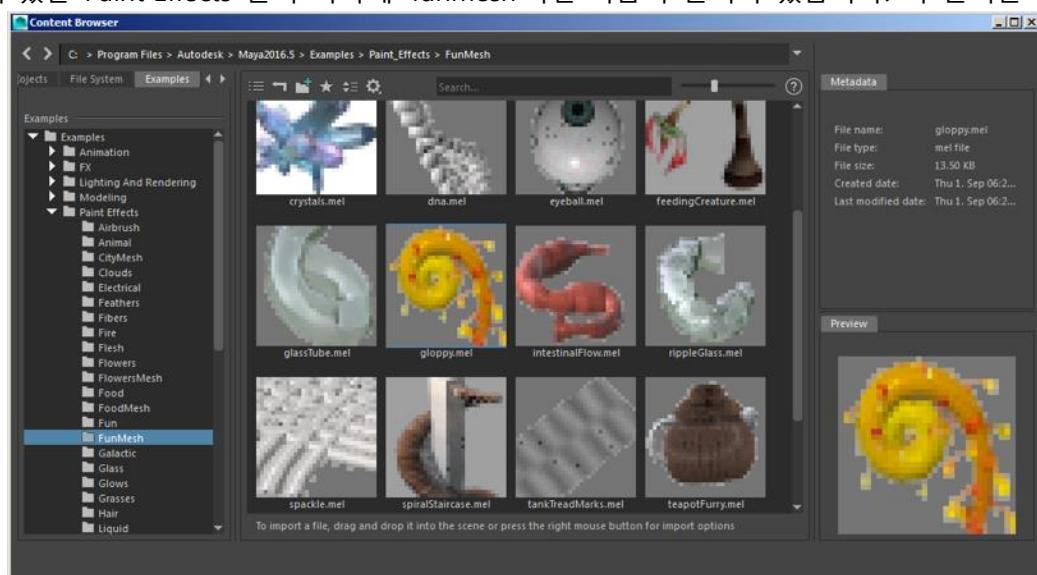
Curve Collector는 많은 수의 곡선을 렌더링할 수 있는 더 빠르고 쉬운 방법입니다.

장면 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

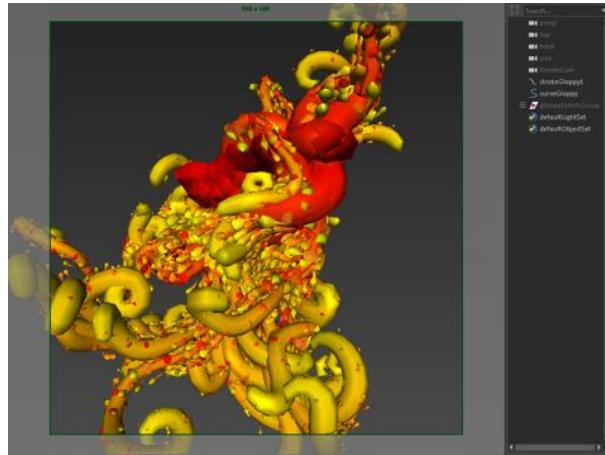
(<https://support.solidangle.com/display/A5AFMUG/Rendering+Paint+Effect+Curves+using+Override+Sets>)

Paint Effects Gloppy 브러시

- 새로운 장면을 생성합니다. **Content Browser** 창을 엽니다.
- 왼쪽에 있는 Paint Effects 폴더 목록에 'funmesh'라는 이름의 폴더가 있습니다. 이 폴더를 선택합니다.



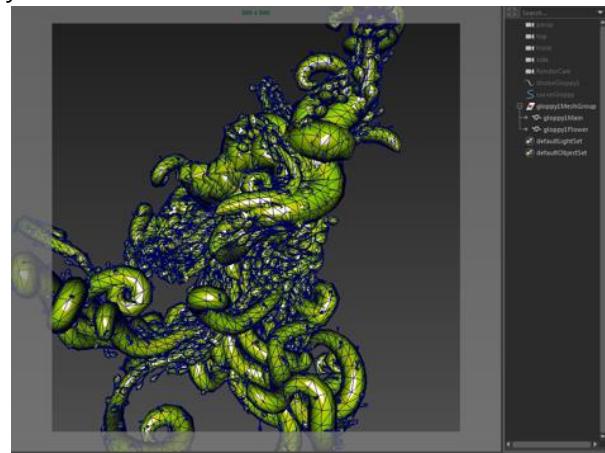
- 오른쪽 창에는 'floppy.mel'이라는 브러시가 있습니다. 이 브러시를 선택하고 Maya 뷔포트에서 아래와 같은 이미지를 얻을 때까지 그리십시오(이 Floppy strokeshape에 사용되는 설정: - **Sample Density**: 80 및 Global Scale: 4.



Paint Effect를 폴리곤으로 변환

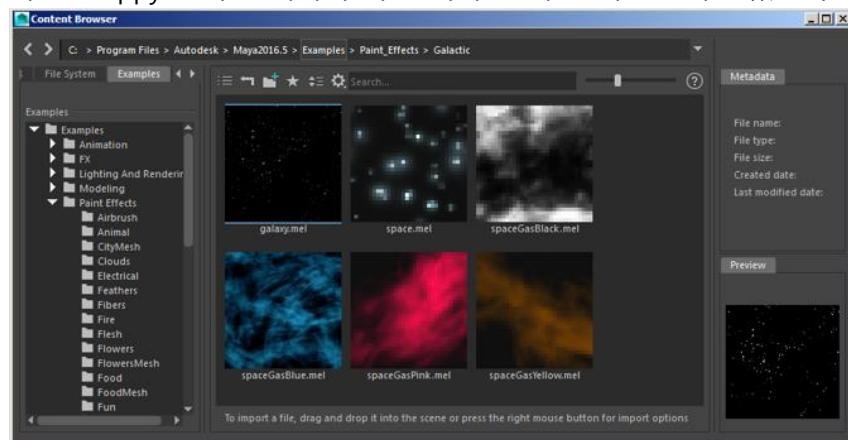
- 이제 Floppy paint effect 브러시를 폴리곤으로 변환하여 다른 브러시를 페인트할 수 있습니다. Floppy 브러시를 선택하고 **Modify>Convert Paint Effects to Polygons**로 이동합니다.

그러면 아래 그림과 같이 Floppy 페인트 효과 브러시의 폴리곤 버전이 나타납니다.

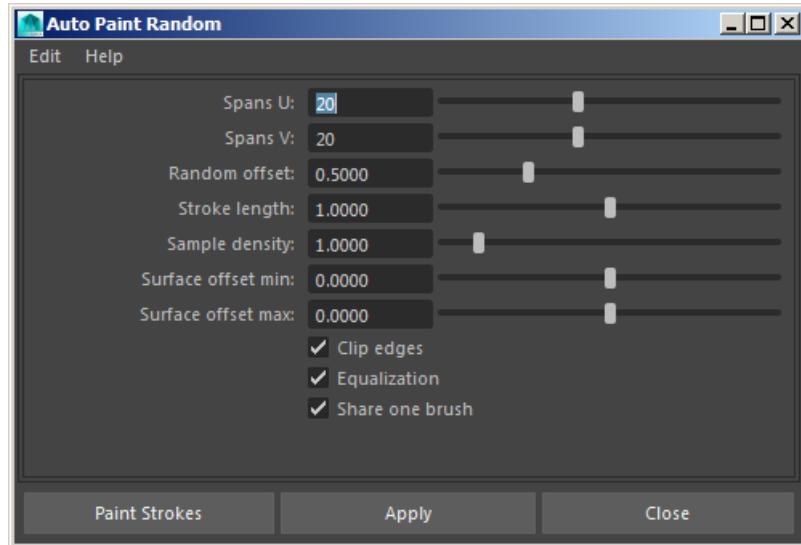


Paint Effects Galaxy 브러시

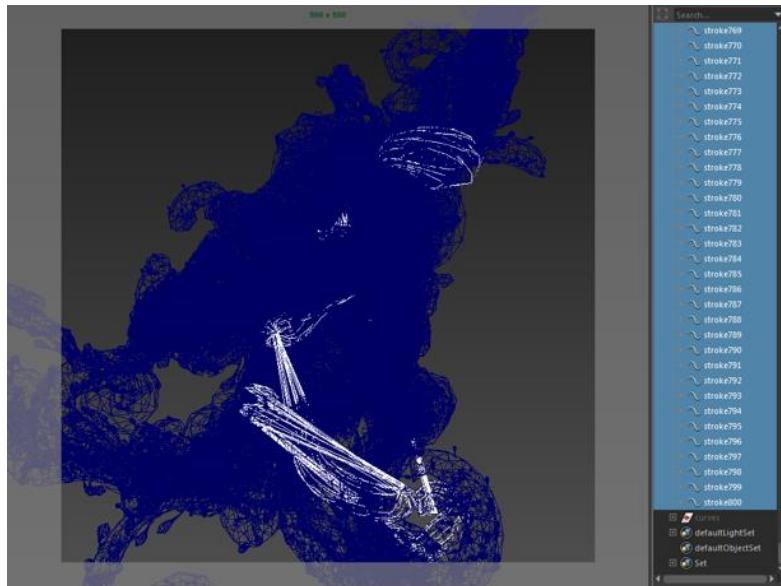
- Content Browser** 창을 다시 엽니다. 이번에는 'Galactic' 폴더를 선택하고 오른쪽 창에서 'galaxy.mel' 브러시를 선택하십시오. Floppy 폴리곤 메시에 이 브러시를 '자동 페인트'해 보겠습니다.



- Floppy 지오메트리가 선택되고 'galaxy.mel' 페인트 효과 브러시가 강조 표시된 상태로 'Paint Effects>Auto Paint> Paint Random' 옵션 창을 엽니다. 'Auto Paint Random'은 선택한 페인트 효과 브러시를 선택한 지오메트리에 무작위로 적용하여 수동 작업을 없애줍니다. Spans U 및 V는 지오메트리에 몇 번의 스트로크를 적용할지를 결정합니다. 이 수치가 너무 높으면 안 됩니다. 너무 높으면 Maya에서 모든 곡선을 계산하는 작업이 오래 걸리거나 최악의 경우 메모리가 부족하게 됩니다. **Spans U and V**를 20으로 변경합니다.

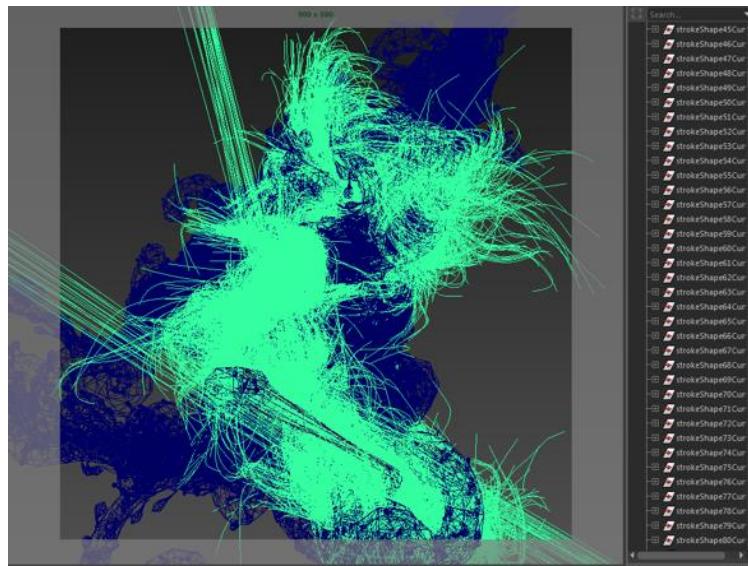


- Apply를 누릅니다. 새로운 'galaxy' 스트로크가 다음과 같이 모델링 및 아웃라이너 창에 표시되어야 합니다.

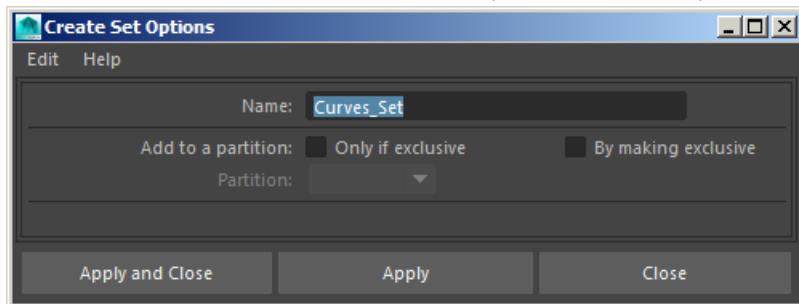


Paint Effect를 곡선으로 변환

- 이제 이러한 페인트 효과 스트로크를 곡선으로 변환할 수 있습니다. 이를 위해서는 'galaxy' 스트로크를 선택하고 **Modify>Convert Paint Effects to Curves**로 이동합니다. 어느 정도 계산한 후 Maya는 모든 'galaxy' 스트로크를 곡선으로 변환합니다(곡선은 galaxy 페인트 효과 스트로크와 매우 다름을 볼 수 있습니다). 하지만 이것은 우리가 정확히 원하는 효과이므로 걱정하지 마십시오!

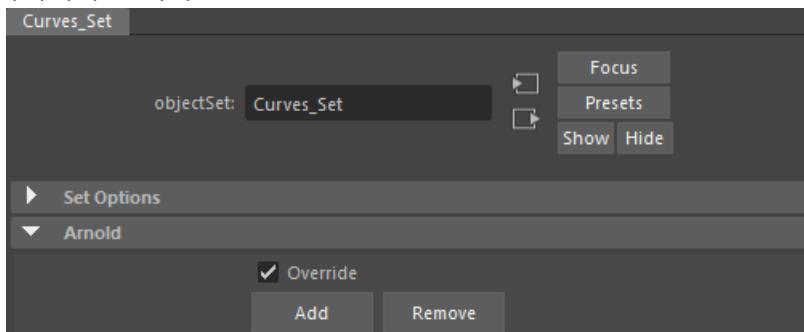


- 곡선이 선택된 상태에서 이들에 대한 Set을 생성합니다(Create>Sets>Set).



Set에 Add Curves

- 해당 Set을 선택하고 Attribute Editor를 엽니다. Arnold를 클릭합니다. Override의 속성을 추가하거나 제거하는 기능이 표시되어야 합니다.



여러 개의 곡선을 렌더링하는 더 빠르고 효율적인 방법은 Curve Collector를 사용하는 것입니다.

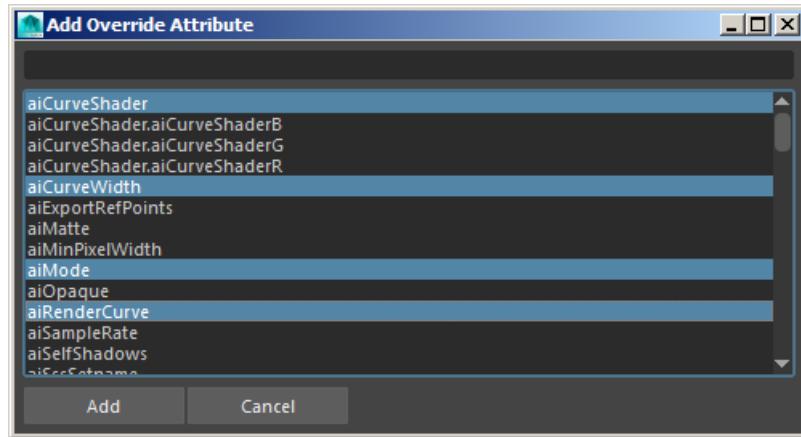
곡선 Set을 선택할 때 지연이 발생할 수도 있습니다. 이것은 많은 수의 곡선이 사용되기 때문입니다.

Add Extra Attributes

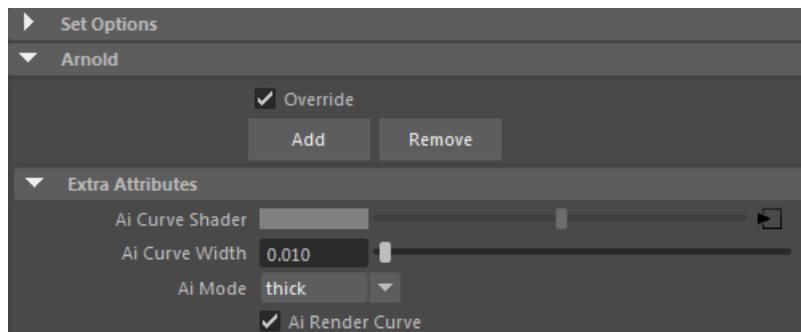
- 'Add'를 클릭합니다. 'Add Override Attribute' 창이 나타납니다. 다음 속성들을 선택합니다.
 - 'aiRenderCurve' - 곡선을 렌더링할 것인지 여부를 결정합니다.
 - 'aiCurveShader' - 곡선에 쉐이더를 지정할 수 있습니다. Standard Hair 또는 Standard Surface 쉐이더

를 이 속성에 연결합니다.

3. 'aiCurveWidth' - 곡선에 대한 너비 값을 설정할 수 있습니다. 두께를 얻으려면 최소 0.010의 값을 선택하십시오. 하지만 너무 높게 선택하면 안 됩니다.
4. 'aiMode' - 'Ribbon' 또는 'Thick' 중에서 사용할 곡선 렌더링 모드를 선택할 수 있습니다. 'Thick'을 선택합니다.



추가한 'Override Attributes'가 해당 세트의 Attributed Editor에서 'Extra Attributes' 아래에 나타나야 합니다. 쉐이더가 Ai Curve Shader(이 경우 Standard Surface)에 연결된 것을 볼 수 있습니다. Ai Curve Width가 0.01로 설정되었습니다. 사용자의 선호에 따라 이 값을 늘리거나 줄일 수 있습니다(모든 예제 렌더링에서 0.01 사용). Ai Mode가 thick으로 설정되었습니다. 이것은 원형 단면을 가지기 때문에 보기에 더 좋습니다.



결론

이제 곡선의 렌더링을 시작할 수 있습니다. 다음 단계는 흥미로운 조명(아래 이미지와 같은 Physical Sky)과 카메라의 피사계 심도를 추가하는 것입니다. 곡선을 좀 더 재미있는 모양으로 '스타일링'하기 위해 'floppy' 또는 'galaxy' 스트로크에 Paint Effects '선 수정자'를 추가할 수도 있습니다.



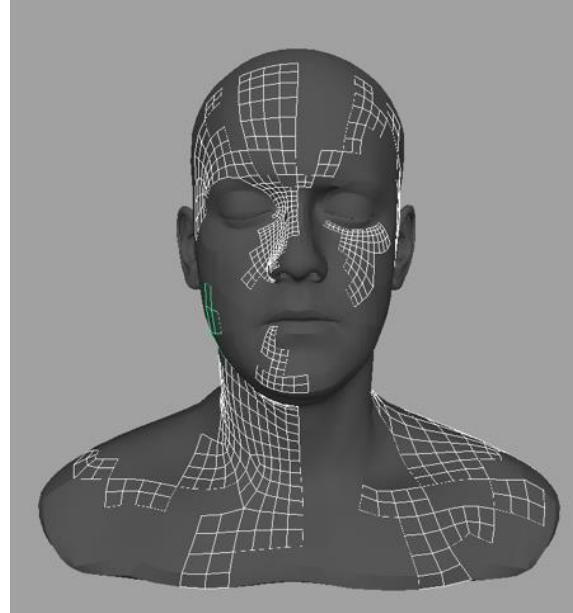
Render Curve를 사용하여 추상적인 머리 렌더링



Maya의 Paint Effects는 복잡한 곡선 모양을 쉽고 재미있게 만들 수 있는 방법입니다. 이 자습서는 Maya의 Paint Effects 브러시, 렌더링 곡선 및 **Override set**을 조합하여 추상적인 초상화 머리를 렌더링하는 방법을 설명합니다.

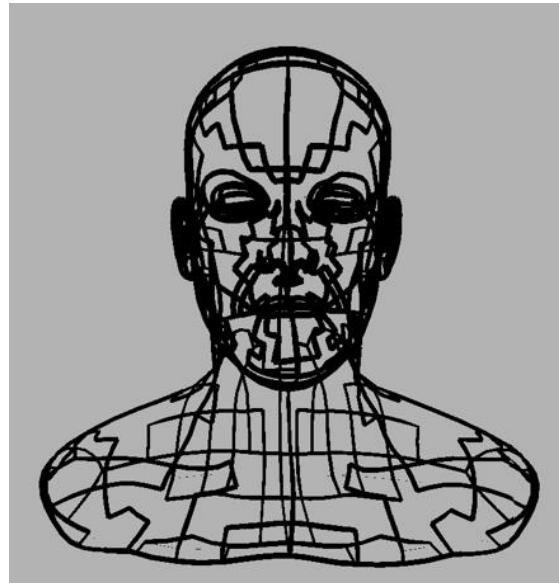
- 먼저 Lee Perry Smith 모델을 다운로드하고 Maya에서 엽니다. 모델이 몇 가지 조각으로 구성되어 있어야 합니다. 그렇지 않다면 모델을 분리해야 합니다. - **Mesh>Separate**.
(모델은 링크의 사이트에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Render+an+Abstract+Head+Using+Render+Curves>)



Toon Outline

- 전체 모델을 선택하고 **Toon>Assign Outline>Add New Toon Outline**으로 이동합니다.
- 머리 모델을 숨깁니다. Toon Outline(**pfxToonShape1**)을 선택하고 아래 이미지에서처럼 분명하게 보이도록 **Line Width**를 줄입니다. 이 경우에는 **0.005**이 사용되었습니다.

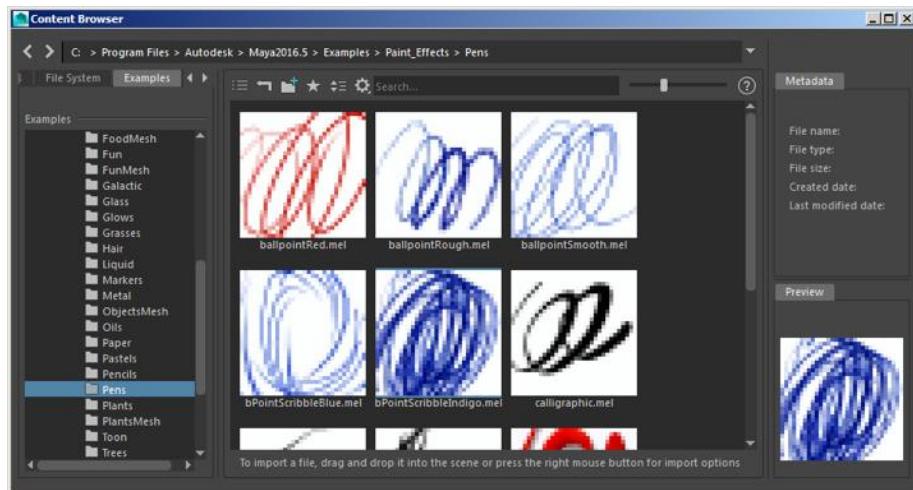


Toon 윤곽

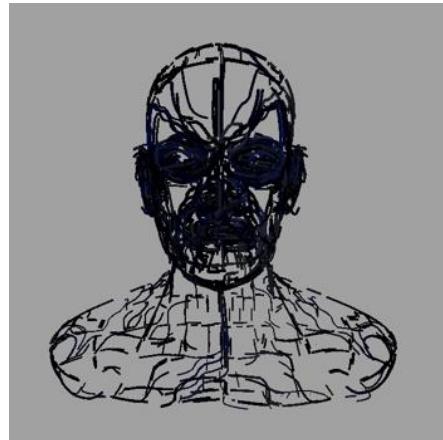
Paint Effects Brush를 Toon Line에 지정

Toon Outline을 그대로 곡선으로 전환할 수도 있지만 먼저 Toon Outline에 Paint Effects 브러시를 추가하여 보다 흥미롭게 표현할 수 있습니다.

- Content Browser 창을 엽니다. Paint Effects 아래에서 **pens** 폴더를 선택합니다. 사용하고자 하는 브러시는 **bpointScribbleIndigo.mel**입니다. 이 브러시는 곡선으로 변환할 때 더 많은 볼륨을 추가하는 여러 스트로크를 가지고 있습니다.

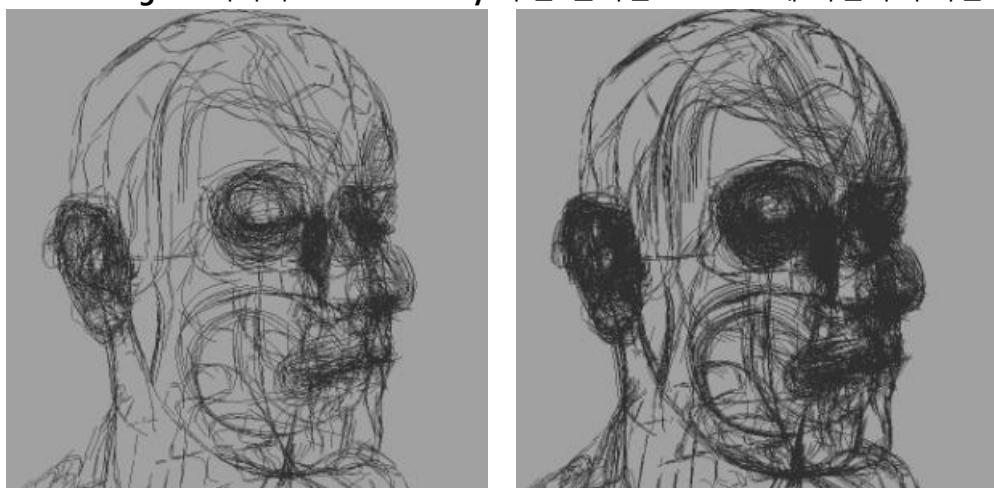


- Toon Outline을 선택하고 Visor에서 Shift를 누른 상태로 '**bpointScribbleIndigo.mel**' 브러시를 선택합니다. Toon 메뉴 아래에서 **Assign Paint Effects Brush to Toon Lines**에 할당합니다. 글로벌 스케일에 따라 속성 편집기에서 **bpointScribbleIndigo** 브러시의 전체 축척을 줄여야 할 수도 있습니다(Toon Outline을 선택한 상태). 이 경우에는 1이 사용되었습니다.



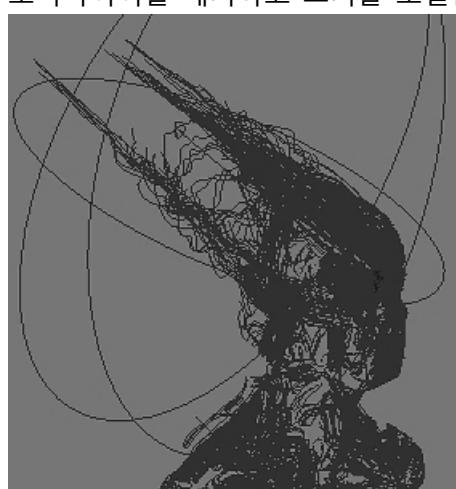
Toon Outline에 지정된 'bpointScribbleIndigo' (Global Scale 1).

- **bpointScribbleIndigo** 브러시의 **Tubes Per Step** 수를 높이면 스트로크에 곡선이 추가됩니다.



Paint Effects Modifier

- 수정자를 사용하여 곡선의 모양을 왜곡시켜 더욱 흥미로운 머리 모양을 만들 수 있습니다. **pfxToon** 브러시가 선택된 상태로 **Paint Effects>Create Modifier**로 이동합니다. 아래 그림과 같이 Paint Effects 브러시가 머리 뒷쪽으로 당겨지도록 모디파이어를 배치하고 크기를 조절합니다.



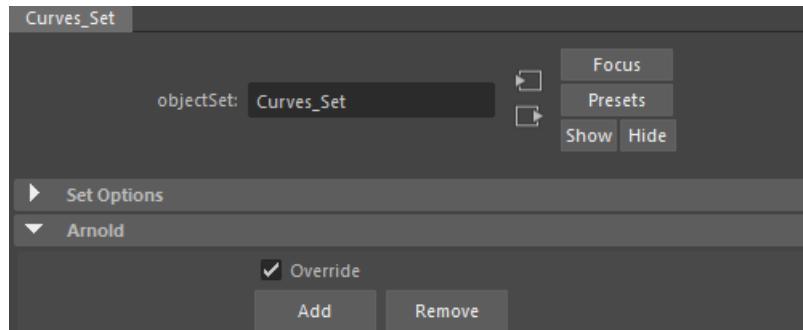
Paint Effects Modifier를 사용하여 변형된 Paint Effect

Paint Effect를 곡선으로

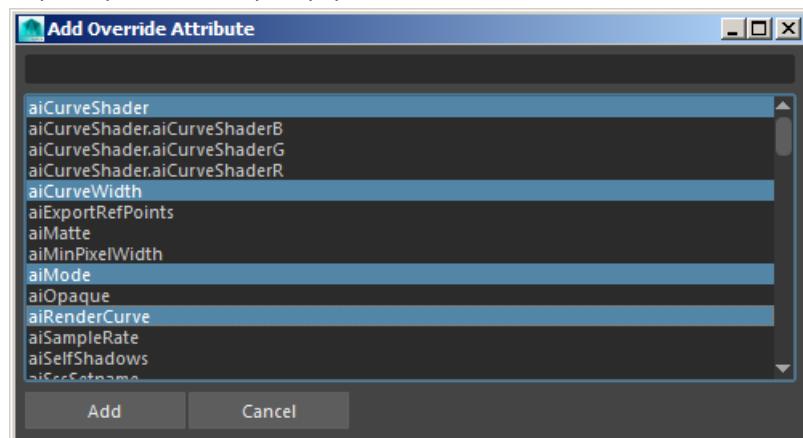
- Paint Effect의 모양이 마음에 들면 이것을 곡선으로 변환할 수 있습니다. **bpointScribbleIndigo** 브러시가 선택된 상태로 **Modify>Convert>Paint Effects to Curves**로 이동합니다.
- 이제 MtoA로 곡선을 렌더링할 수 있게 만들어야 합니다. 이를 위해서는 해당 곡선에 대해 **Override Set**을 만들어야 합니다. 해당 곡선들을 선택하고 Maya Set을 만듭니다. - **Create>Sets>Set**

여러 개의 곡선을 렌더링하는 더 빠르고 효율적인 방법은 **Curve Collector**를 사용하는 것입니다.

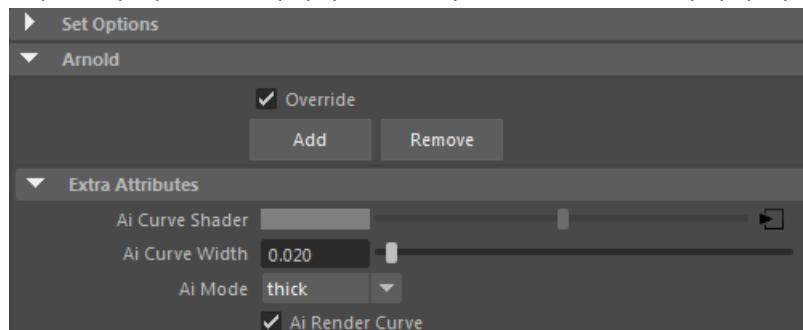
- 해당 Set을 선택하고 Attribute Editor에서 **Arnold**로 이동합니다.



- Add**를 선택하고 다음 속성들을 선택합니다.



그러면 새로 만든 속성들이 Set의 속성 편집기에서 Arnold의 Extra Attributes 아래에 나타납니다.

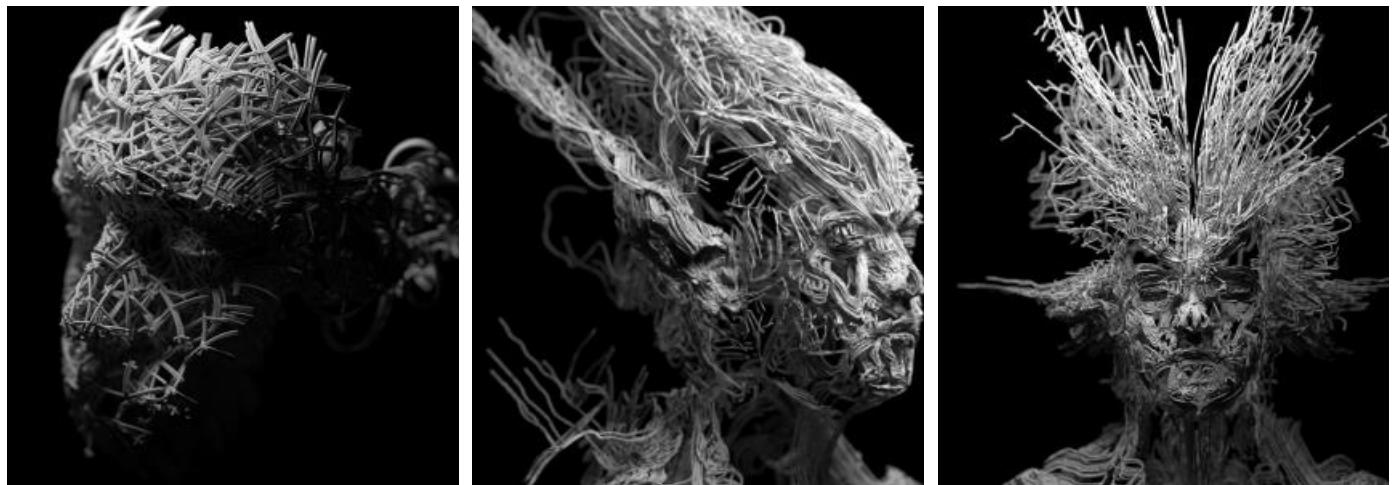


- Standard Surface** 쉐이더를 **Ai Curve Shader** 속성에 연결합니다.
- 원하는 곡선의 두께에 따라 **Ai Curve Width**를 0.02로 높입니다.
- Ai Mode**를 **thick**으로 변경하면 곡선에 원형 튜브 프로필이 생깁니다.

- **Ai Render Curve**를 선택하여 곡선을 렌더링시킵니다.



이제 다 되었습니다. 조명을 적용하고 렌더링을 시작해 보세요! 여러 Paint Effects 브러시와 deformer 설정을 즐겁게 실험해보고 흥미로운 결과를 얻을 수 있습니다.



Arnold로 유리 표면 렌더링하기



Arnold로 굴절 유리 표면을 렌더링할 때는 몇 가지 공통적인 문제에 직면하게 됩니다. 액체 대 유리 인터페이스가 있는 표면을 모델링할 때는 따라야 할 몇 가지 지침이 있습니다. 문제는 렌더링 도구에서 우리가 어떤 오브젝트에 들어오는 것인지 아니면 나가는 것인지, 그리고 들어오고 나가는 인터페이스가 무엇인지 파악하는 방법에 있습니다.

Arnold Standard Surface 쉐이더는 외부로 향하는 법선을 가정하고, 물체가 공기 중에 있으며, 물체들이 서로 겹치지 않는다고 가정합니다.

아래에서는 다양한 모델링 접근법에서 얻은 결과를 설명합니다.

- 액체와 용기가 정확히 접촉합니다.
- 액체와 용기는 사이에 틈이 있어서 서로 접촉하지 않습니다.
- 액체와 용기가 서로 겹칩니다.
- 또한 유리 표면을 렌더링할 때 사용할 재질 및 렌더링 설정에 대해서도 설명합니다.

이 자습서는 다음 3가지 부분으로 구성되어 있습니다.

- 유리와 액체 모델링하기
- 유리와 액체 쉐이딩하기
- 렌더링 설정

파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

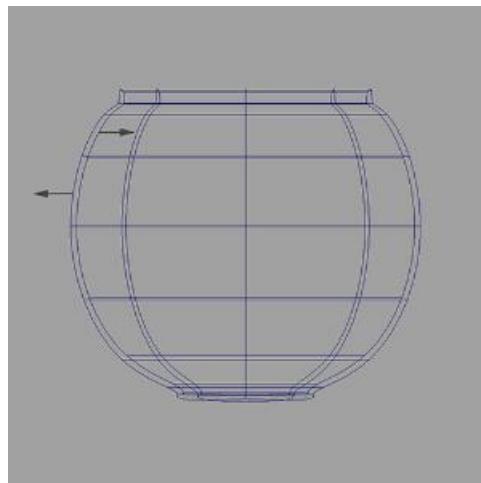
<https://support.solidangle.com/display/A5AFMUG/Rendering+Glass+Surfaces+with+Arnold>

유리와 액체 모델링하기

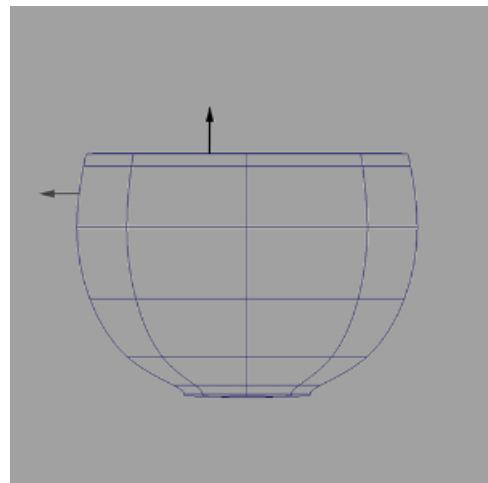
두께가 있는 유리 표면을 정확하게 모델링해야 합니다. 그렇지 않으면 유리가 올바르게 렌더링되지 않습니다.

Arnold로 굴절 표면을 정확하게 렌더링하려면 모델이 특정 지침을 따라야 합니다.

- 공기 대 유리 표면(IOR 1.5)은 공기가 유리에 직접 닿는 영역을 덮는 유리를 가리키는 법선을 가지고 있습니다.
- 유리 대 액체 표면(IOR 1.33)은 공기가 액체에 직접 닿는 영역을 덮는 액체를 가리키는 법선을 가지고 있습니다.



유리 그릇 노멀 방향

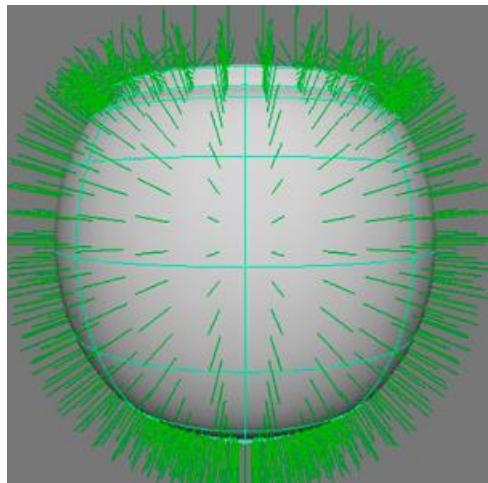
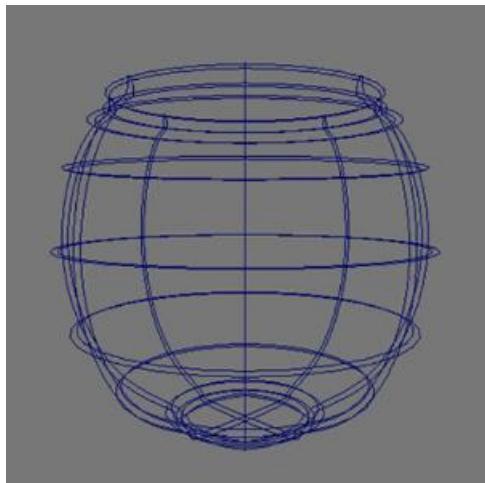


액체 노멀 방향

광선은 공기($OR = 1.0$)를 통과하여 이동해 유리에 들어가고 유리의 $IOR(1.5)$ 에 의해 굴절됩니다. 광선은 유리를 통과하여 이동한 후 유리를 떠나서 액체로 들어갑니다. 즉 IOR 가 1.5인 하나의 매체에서 IOR 가 1.33인 다른 매체로 이동합니다.

유리 표면 모델

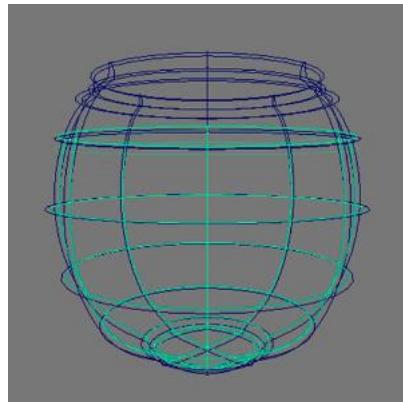
아래는 간단한 유리 그릇 모델입니다. 오른쪽 이미지에서 메시의 법선이 바깥을 올바르게 가리키고 있음을 볼 수 있습니다. Arnold로 굴절 개체를 렌더링할 때는 올바른 법선 방향이 중요합니다.



표면 노멀이 바깥쪽을 가리키고 있습니다.

액체 모델링 - 일치 표면

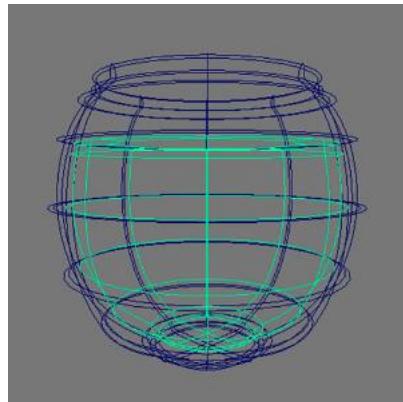
아래의 예에서는 유리와 액체의 표면이 서로 겹쳐져 있어서 서로 일치합니다. 렌더링 도구는 각 표면 사이를 식별하는 데 문제가 있으며 겹치는 인공물로 출력됩니다.



유리 및 액체 표면이 일치합니다.

액체 크기 축소

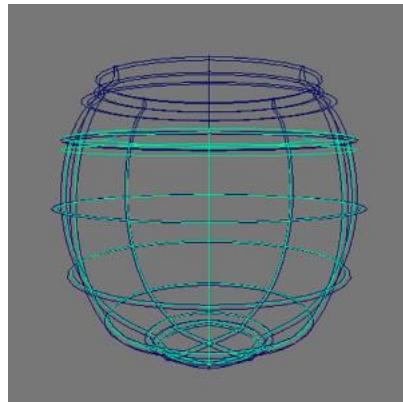
이 경우에는 액체의 크기를 약간 감소했습니다. 유리와 액체 사이의 격차가 매우 적어 렌더링이 사실적이지 않습니다.



액체의 양이 감소함(어색함)

액체 크기 증가

여기서는 액체의 크기를 증가하여 유리 그릇 지오메트리와 중첩됩니다. 액체가 유리에 닿아있는 것처럼 보여서 더 자연스럽게 연출됩니다.

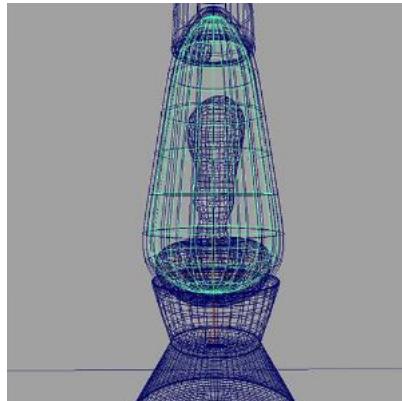


액체의 크기 증가

이 모델링 방식은 '보기'에는 좋지만 물리적으로 적합한 접근 방식은 아님에 유의하십시오. 일반적으로 Arnold을 사용하여 렌더링할 때 올바른 결과를 얻으려면 오브젝트를 가능한 사실적으로 모델링해야 합니다.

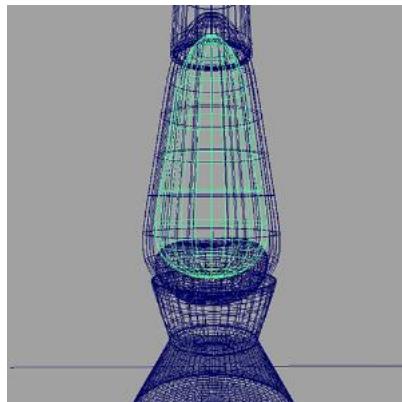
예 2

아래의 예에서 유리와 액체의 표면이 서로 겹쳐져 있어서 서로 일치합니다. 렌더링 도구는 각 표면 사이에서 문제 식별에 어려움이 있으므로 오른쪽의 이미지에서 오류를 볼 수 있습니다.



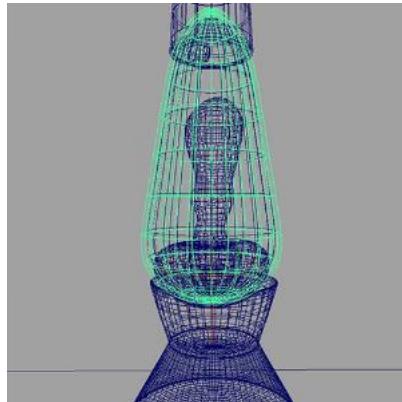
유리 및 액체 표면이 일치합니다.

이 경우에는 액체의 양이 약간 감소했습니다. 유리와 액체 사이의 격차가 매우 적어 렌더링이 사실적이지 않습니다.



액체의 크기 감소

아래에서는 액체의 양이 증가하여 유리 그릇 지오메트리와 중첩됩니다. 액체가 유리에 닿아있는 것처럼 보여서 더 자연스럽게 연출됩니다.



액체의 크기 증가

유리와 액체 쉐이딩하기

유리와 물을 정확하게 음영 처리하기 위해 사용되는 재료 설정은 다음과 같습니다.

Opaque

Arnold로 투명한 표면을 렌더링하기 전에 Opaque를 비활성화해야 합니다.

이것은 해당 메시의 Arnold 속성에 있습니다.

Standard Surface: Glass

Base Weight를 0으로 설정하고 Specular Weight를 1로 높입니다. Index of Refraction을 유리 수치로 변경합니다(1.5). Transmission Weight를 1로 높입니다.

Transmission

Transmission Weight 값을 낮추면 유리의 체적을 통과하는 빛의 양이 감소합니다(거리 기반 색조). 기본적으로 Transmission Color는 흰색이며, 이것은 거리 기반 색조가 없다는 뜻입니다. 순백색으로부터 낮춰보십시오.



Transmission Color: 회색



Transmission Color: 흰색

Standard Surface: 물

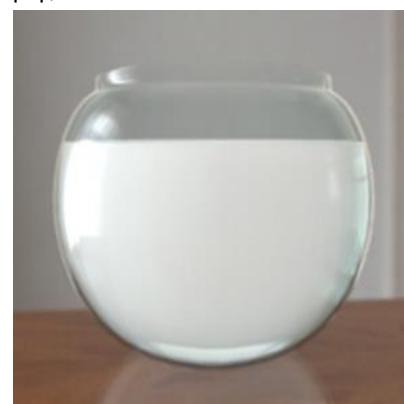
유리 재료에 사용한 설정을 반복합니다. 하지만 Index of Refraction은 물의 값인 1.33으로 설정되어 있어야 합니다.

Standard Surface: Milky liquid

milky liquid를 생성하려면 약간의 하위 표면 산란을 추가하는 것이 좋습니다. 또한 액체를 보다 사실적으로 표현하려면 약간의 방출되는 빛을 추가하여 실험해 볼 수 있습니다.



SSS Weight: 0.1



SSS Weight: 0.1. Emission 0.1

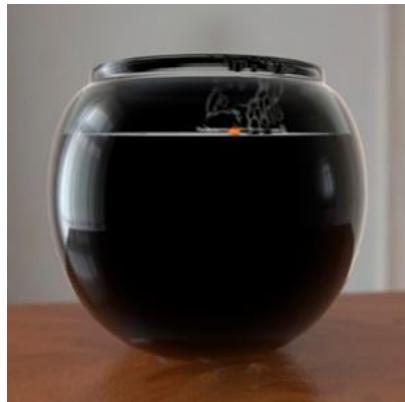
렌더링 설정

유리 표면을 사용하여 렌더링 시간이 오래 걸리는 경우 Standard Surface 쉐이더의 Advanced 섹션에서 해당 메시를 선택하고 '*enable_internal_reflections*'를 비활성화하십시오.

Transmission Depth

사실적인 유리 및 물 효과를 얻으려면 올바른 Transmission Depth 설정이 중요합니다. 값이 높을수록 렌더링 시간이 늘어난다는 점을 기억하십시오.

기본적으로 Arnold는 8의 *Transmission Depth*를 사용합니다.



2



4



8

아래 장면에는 많은 유리 표면이 들어 있습니다. Transmission 광선 깊이를 높이면 굴절된 유리에서 추가적인 디테일이 표시됩니다.



2



4



6

Total Depth

광선 깊이 설정을 높일 때는 총 깊이가 *Transmission ray depth*를 포함할 만큼 충분히 높게 설정되어 있는지 확인하십시오. 총 깊이는 장면 내 임의 광선의 총 최대 반복 깊이를 지정합니다(*Diffuse + Transmission + Specular < Total*).

최종 렌더링

다음이 최종 이미지입니다. Transmission Depth는 기본값인 8로 유지되었습니다. 렌더링이 단색 검정으로 표시되는 튀긴 물의 영역을 더 줄이려면 더 높은 값이 필요할 수 있습니다.

페인트 효과와 함께 불투명도 맵 렌더링하기

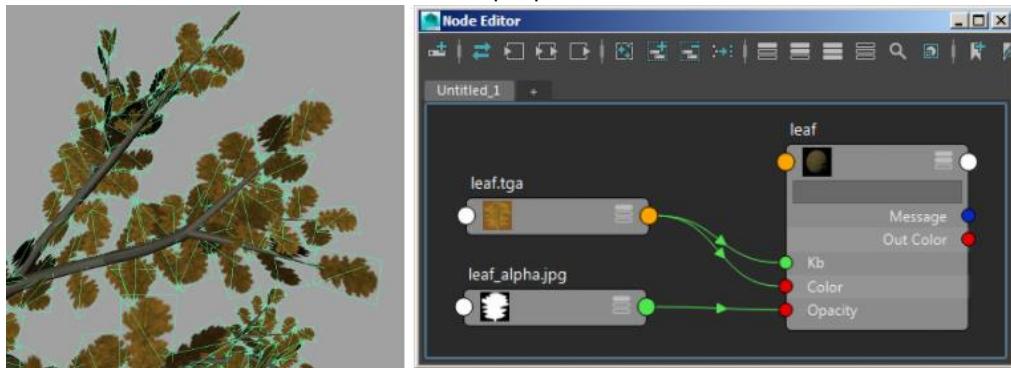


이 자습서에서는 Arnold에서 나뭇잎을 렌더링할 때 렌더링 시간을 줄이기 위해 사용되는 일부 워크플로우를 설명합니다. 이 예의 경우, 폴리곤으로 변환된 Maya Paint Effects 장면을 사용해보겠습니다. 하지만 이러한 기술은 불투명도 맵과 관련이 있는 다른 유형의 지오메트리에도 적용할 수 있습니다.

장면은 하단 링크의 사이트에서 다운로드할 수 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Rendering+Opacity+Maps+with+Paint+Effects>)

일반적인 Opacity 워크플로우는 알파 마스크를 **Standard Surface** 쉐이더의 불투명도 속성에 연결하는 것입니다 (반드시 나뭇잎 지오메트리에 대한 Arnold 속성에서 'Opaque'를 끁니다).



Standard Surface 쉐이더의 Opacity 속성에 연결된 알파 맵이 있는 나뭇잎

Ray Switch

나뭇잎의 렌더링 속도를 높이는 한 가지 방법은 **Ray Switch** 노드를 사용하는 것입니다. Ray Switch를 사용하면 렌더링하고자 하는 광선만 포함 및 제외할 수 있는 유연성이 향상됩니다. 이 상황에서는 각 잎에 대해 모든 음영 처리를 포함할 필요는 없습니다. 한 가지 워크플로우는 다음과 같습니다.

- Ray Switch 쉐이더를 나뭇잎에 할당합니다.
- Diffuse 및 불투명도 맵이 있는 Standard Surface 쉐이더를 **Ray Switch** 쉐이더의 **Camera rays**에 연결합니다.

- 다른 Standard Surface 쉐이더를 Ray Switch 쉐이더의 **Diffuse Reflection** 및 **Shadow** 광선에 연결합니다. 이렇게 하면 Camera 및 Diffuse 광선(간단한 Diffuse Standard Surface 쉐이더 사용)만 계산되어 렌더링 시간이 단축됩니다.



Ray Switch: 1분 3초



Ray Switch가 없을 경우: 1분 48초



Ray Switch: 6분 40초



Ray Switch가 없을 경우: 15분 47초

Leaf Width Scale

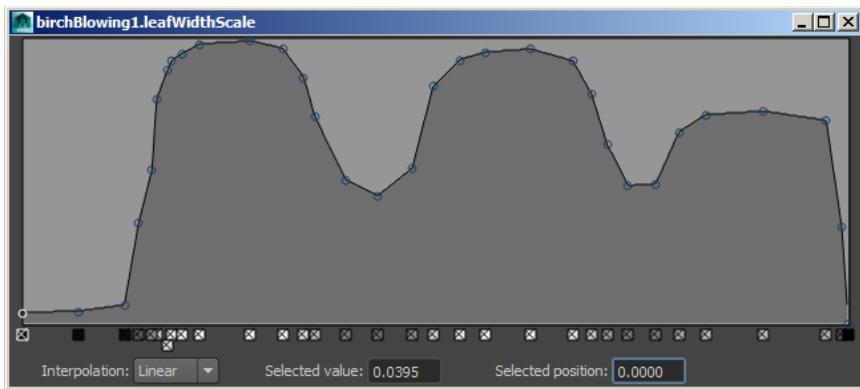
Paint Effects 안에는 이러한 불투명도 문제를 모두 피할 수 있는 해결 방법이 있습니다. 텍스쳐 기반의 투명도를 가지지 않아야 하기 때문에, Width Scale이 잎의 모양을 결정합니다. 텍스처에 대해 생각하는 방식은 Width Scale 모양의 역 엔벨로프 또는 격자 변형의 일종입니다 (나뭇잎 모양의 절반/텍스처의 절반).

다음은 관련된 워크플로우의 예입니다.

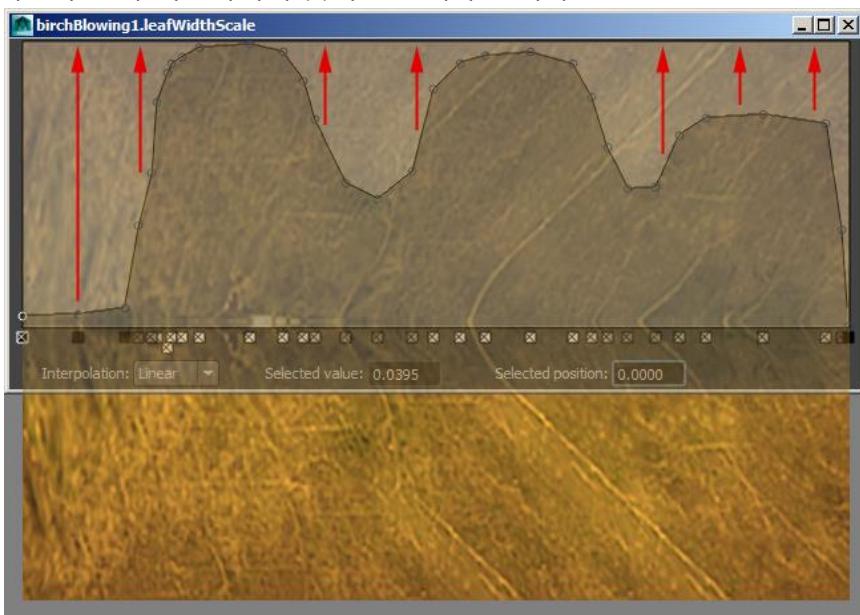
- Paint Effects 브러시 속성을 선택합니다(이 경우에는 strokeBirchBlowing). **Leaf Width Scale**로 이동합니다. 이것은 **Tubes > Growth > Leaves** 아래에 있습니다.



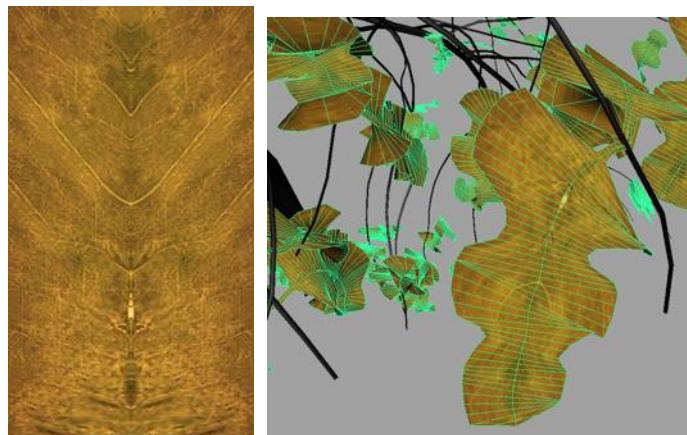
- 언더컷이 없다면, 나뭇잎의 프로필을 매우 자세하게 그립니다.



- 이미지 편집 프로그램에서 알파가 없는 텍스처를 사용하여 잎의 모양을 일치시킵니다. 나뭇잎 텍스처는 틈이 없도록 텍스처 맵의 가장자리에 맞게 조정해야 합니다.



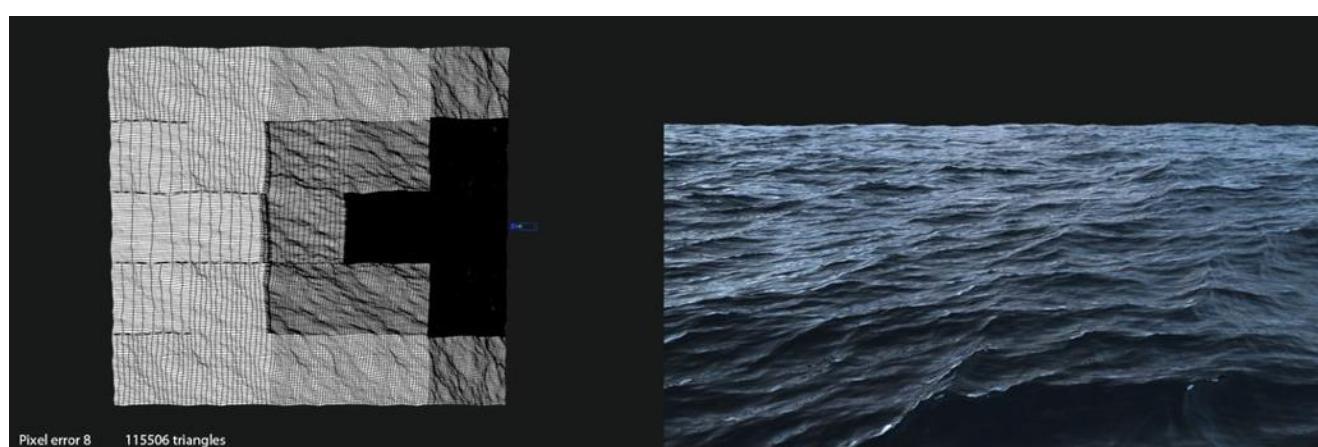
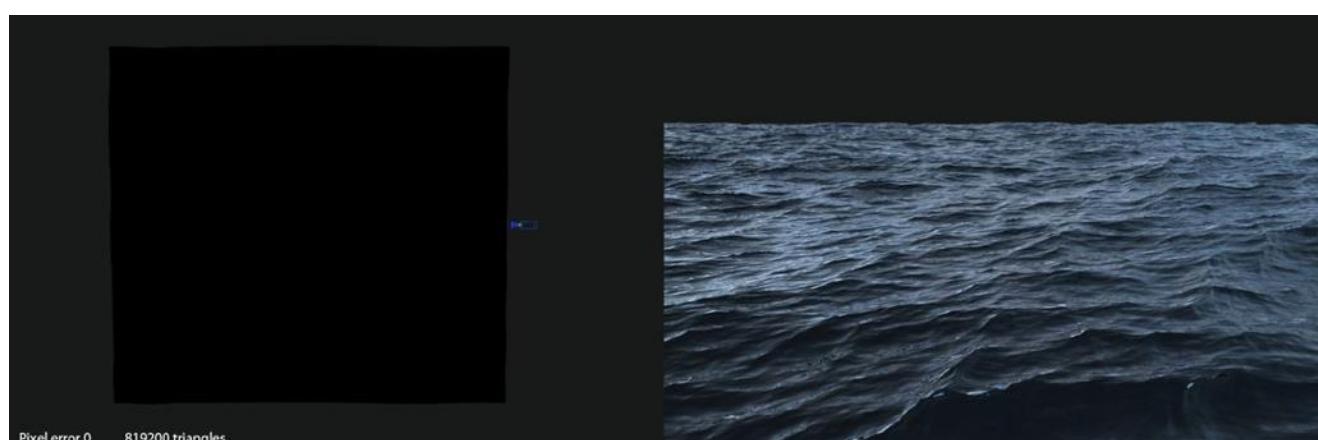
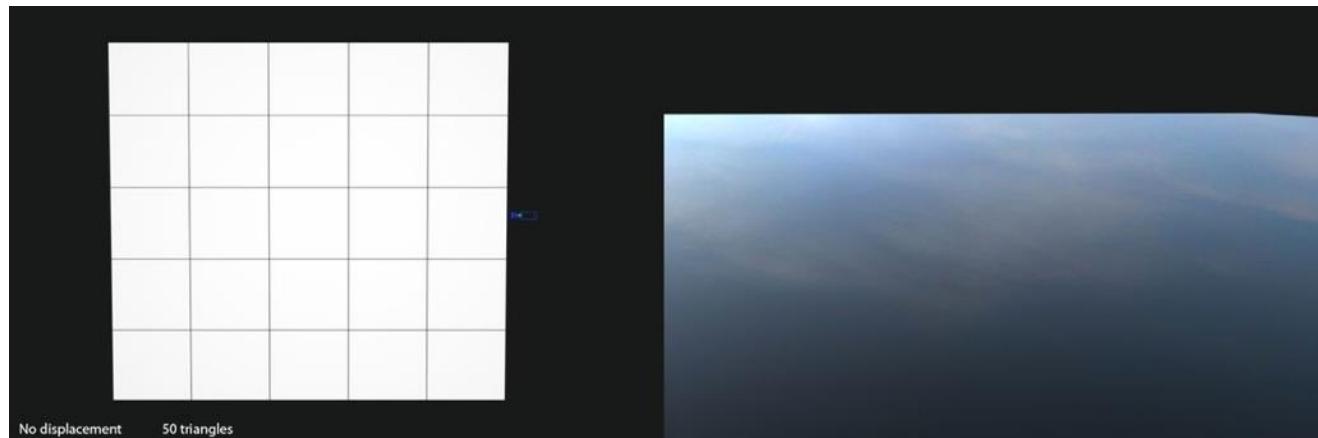
- 수정된 나뭇잎 텍스처를 표준 Standard Surface 쉐이더에 연결하고 나뭇잎 지오메트리에 할당합니다.
- Paint Effects 브러시를 폴리곤으로 변환하고 Standard Surface 쉐이더를 지정합니다.

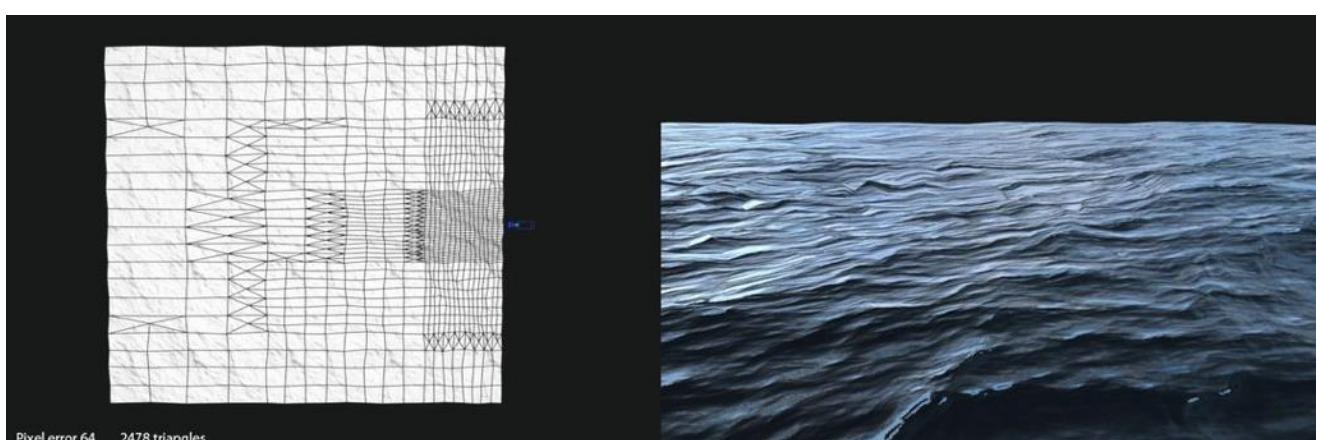
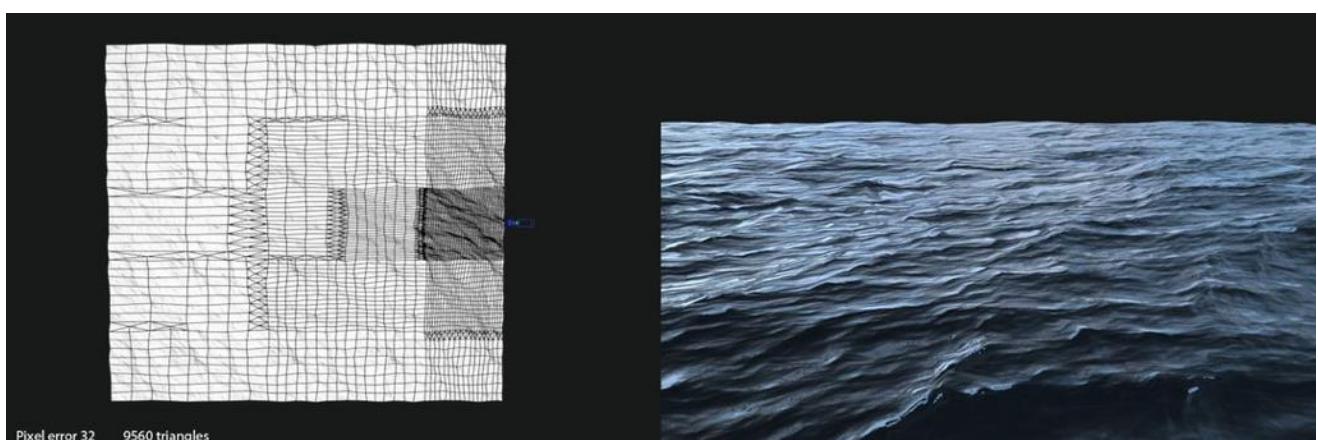
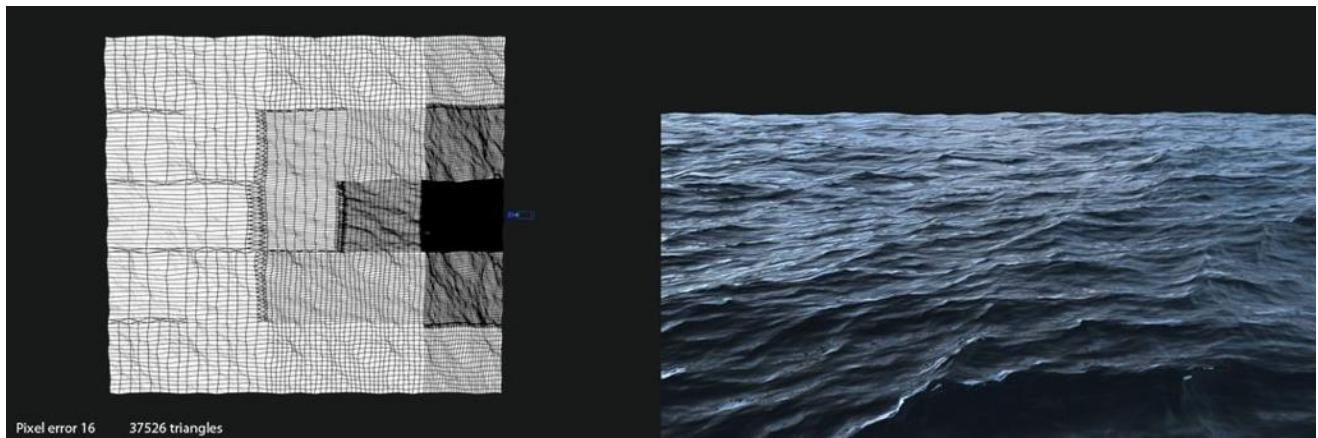


수정된 leaf width scale을 사용하여 나뭇잎 지오메트리에 지정되어 있는 수정된 나뭇잎 텍스처

Subdivision Displacement 예

다음 이미지들은 Displacement 맵핑을 사용할 때 픽셀 오류 세분화의 효과를 보여줍니다.





안개 낀 수중 장면

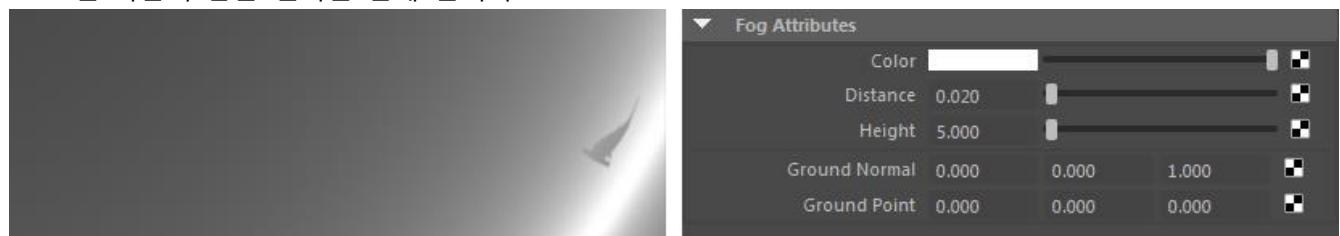


상어다! 수중 장면 주제

이 자습서에서는 상어 떼가 있는 수중 장면을 조명 처리 및 렌더링하는 방법을 알아봅니다. 안개 낀 장면을 설정하여 수중의 느낌을 표현해 보겠습니다. 또한 **gobo**를 사용하여 조명을 비춰서 가짜 커스틱 효과를 만들어 보겠습니다. 또한 **Vector displacement**를 사용하여 바다 물체들을 만들어 볼 것입니다.

Fog

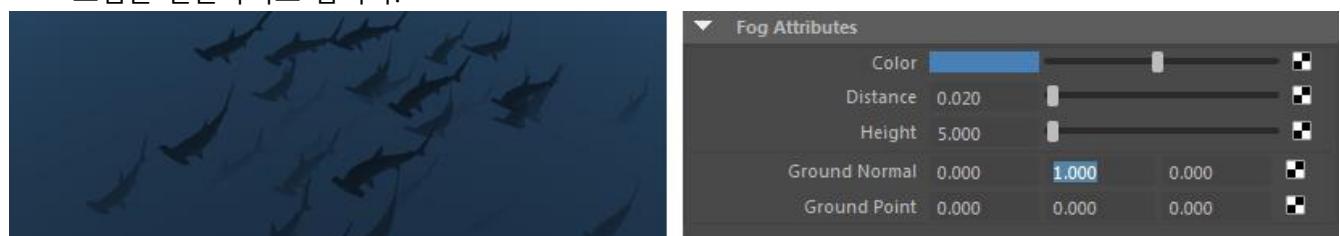
- 먼저 수중 장면에 안개를 만들어 보겠습니다. Render Settings 창을 엽니다. **Environment 속성** 아래에서 **Atmosphere**를 **none**에서 **Fog**로 변경합니다.
- 속성 편집기에 Fog 속성들이 나타납니다. **Distance**를 **0.02**와 같은 적은 양으로 올리고 장면을 렌더링하면 다음과 같은 결과를 얻게 됩니다.



기본 안개 설정. Ground point는 기본적으로 Z입니다.

Ground Normal

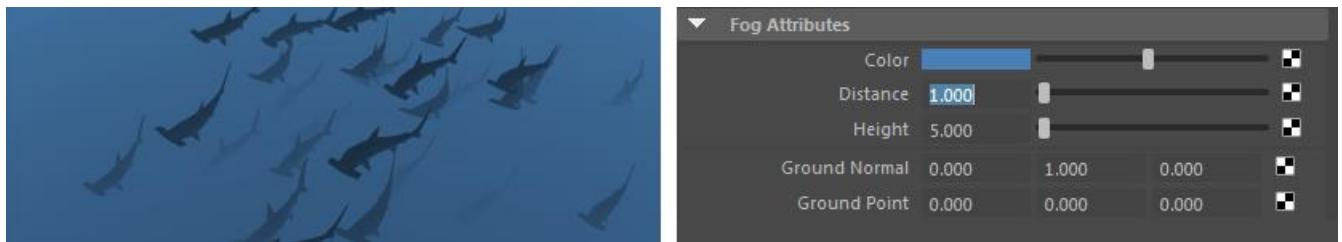
- 안개의 Color를 mid blue로 변경하고 Y축 필드에는 **1**, Z축 필드에는 **0**을 입력하여 **Ground Normal**을 **Y**로 변경합니다. 상어들은 바다 표면 근처에 있을 것으로 빛이 위에서 들어오고 Y 방향으로 감쇠하는 느낌을 연출하려고 합니다.



Ground Normal 1 (Y축)

Distance

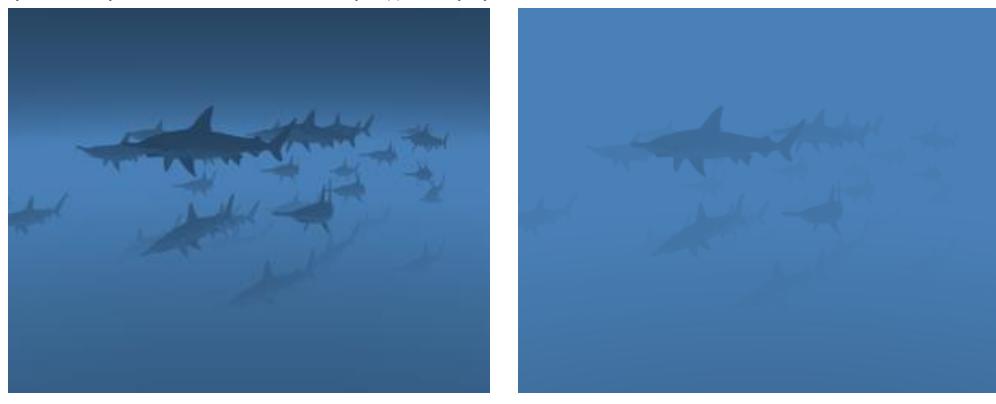
- 이제 안개의 양을 늘려보겠습니다. **Distance** 값을 1 정도로 높입니다.



1로 증가한 Distance

Height

- 위쪽의 상어 주변에는 안개가 없습니다. **Height** 값을 높입니다. 이렇게 하면 안개가 장면의 위쪽으로 이동해 보다 실감나는 장면을 연출할 수 있습니다.



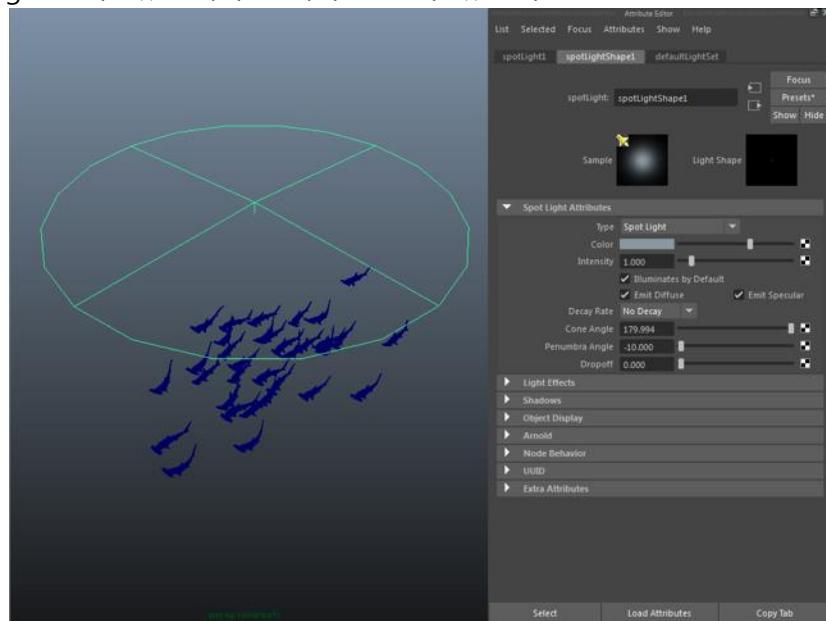
Height 5

Height 44

Spotlight lighting

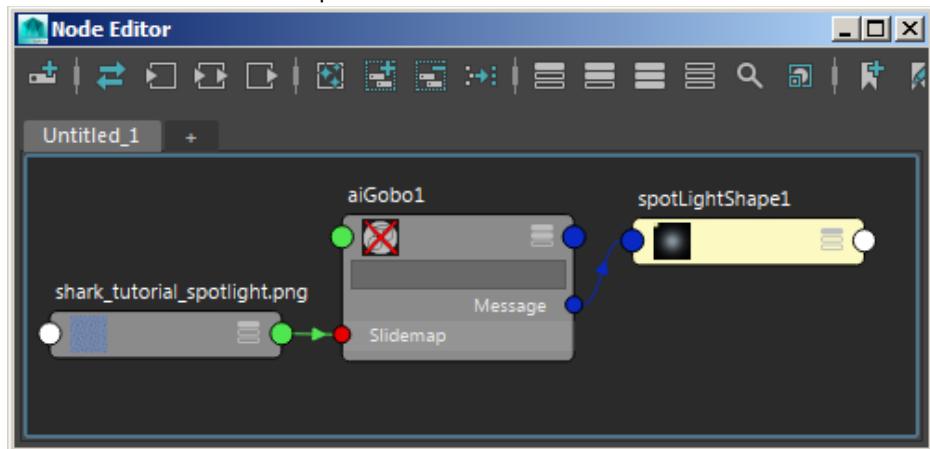
이제 장면에 빛을 비춰보겠습니다. 스포트라이트를 사용할 것입니다. 이렇게 하면 조명 필터를 추가하여 장면 내 조명을 보다 잘 제어할 수 있습니다.

- 스포트라이트를 생성하고 **cone angle**을 증가시켜서 렌더링 가능한 전체 장면을 커버합니다. 이 경우에는 175의 cone angle을 사용했습니다. 상어가 보일 수 있도록 충분한 강도 또는 노출로 증가시키십시오.



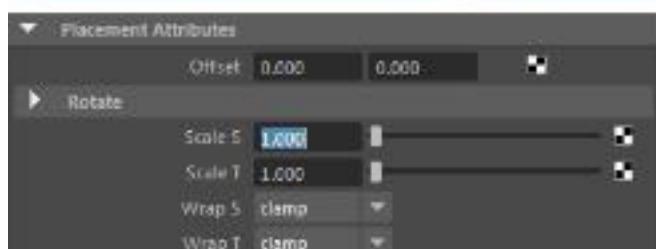
Ai Gobo

- 이제는 광원을 통해 투영되는 가짜 커스틱 효과를 만들어 보겠습니다. 이를 위해 스포트라이트에 **Ai Gobo** 조명 필터를 추가하겠습니다. 해당 스포트라이트의 Arnold 섹션 아래에서 **Light Filters**를 선택하고 Ai Gobo 필터를 추가합니다.
- Slide Map 속성에 파일 텍스처를 연결합니다. 커스틱 패턴으로 구성된 텍스처 맵을 사용할 것입니다. 아래 텍스처 맵은 Ai Gobo의 Slide Map 속성에 연결된 가짜 커스틱 효과로 사용됩니다.

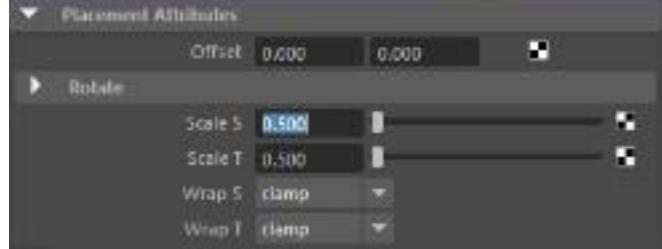


aiGobo의 'Slidemap' 속성에 연결된 커스틱 텍스처

- 장면을 테스트 렌더링합니다. 기본 설정을 사용하면 커스틱 효과의 스케일이 너무 작게 보입니다. 배율 S 및 T 값의 배율을 줄이면 Slide Map에 사용되는 커스틱 텍스처의 배율이 증가합니다.

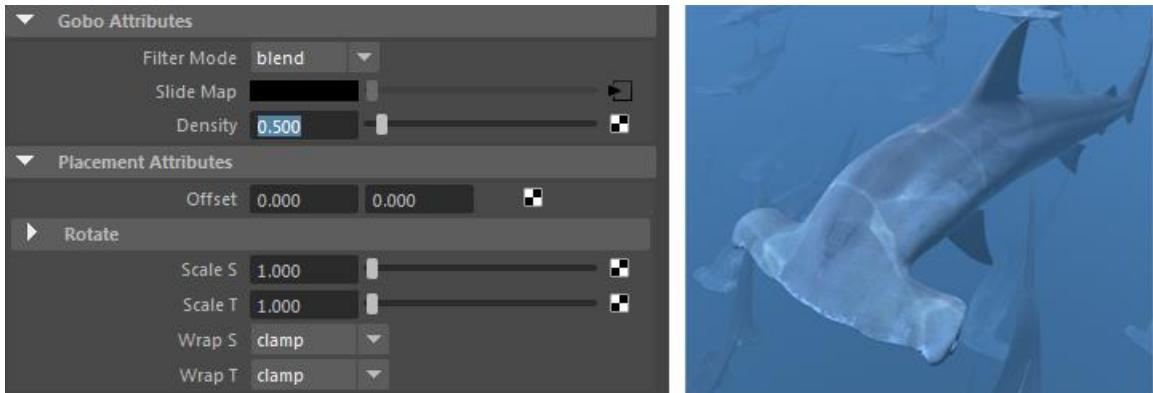


1로 설정된 Gobo Scale S T. 커스틱 패턴이 너무 작습니다.



Gobo Scale S T를 0.5로 낮추면 더 큰 커스틱 효과가 나타납니다.

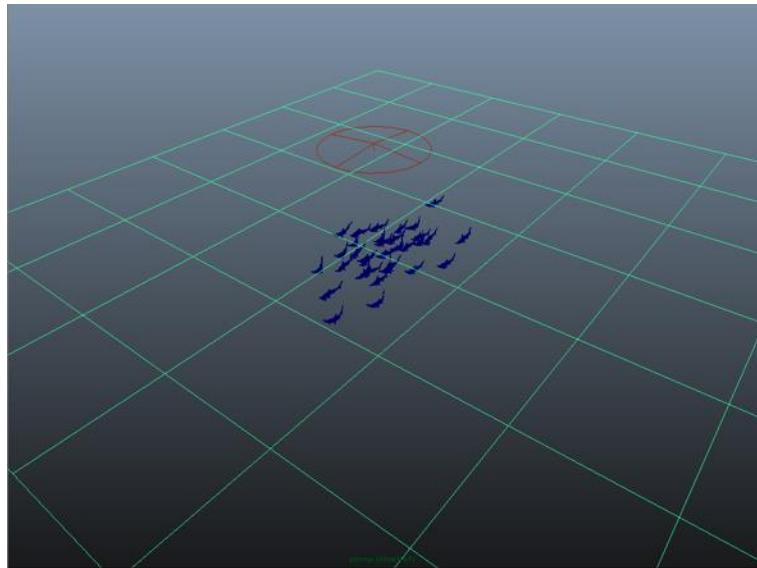
- gobo 효과가 약간 강하게 나타납니다. Density 값을 높여서 이 효과를 줄일 수 있습니다. 이것을 0.5 정도의 작은 양으로 높이면 한결 나아집니다.



0.5로 감소한 Density

바다

이제 폴리곤 면을 만들어서 전체 장면을 덮을 정도로 크기를 키워보겠습니다. 바다면은 투명해야 하므로 Arnold 속성에서 해당 면에 대한 **Opaque**을 선택 해제해야 합니다. Ai Standard 재료를 지정하고 이름을 'Sea'로 변경합니다. **Base Weight**를 0으로 줄이고, **Specular Weight**를 1로 높입니다. 그리고 **Specular Roughness**를 0으로 낮춥니다. **Transmission** 아래에서 **Transmission Weight**를 1로 높입니다. **IOR**는 1.3으로 변경합니다(공기가 물과 만나는 곳). 1.3이 물의 굴절률에 사용되는 물리적으로 정확한 값이지만 원한다면 이 값을 속일 수도 있습니다.



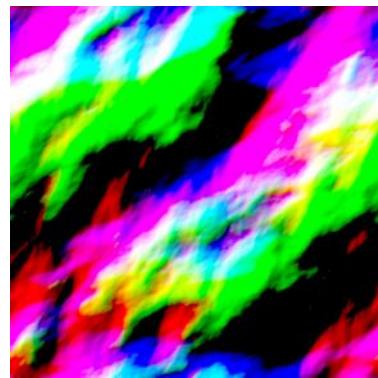
바다 표면에 사용되는 폴리곤 면

Displacement

보다 현실적인 파도 효과를 얻기 위해 **Vector Displacement** 맵을 사용하여 바다의 파도를 대체합니다. 그러면 지오메트리를 한 방향으로만 옮기는 기존의 Displacement 맵과는 다르게 바다 지오메트리가 보통과 다른 방향으로 이동됩니다. 바다 표면 생성에 대한 추가 정보는 여기에 있습니다.

(<https://support.solidangle.com/display/A5AFMUG/Rendering+an+Ocean+with+Displacement>)

- 아래의 파일 텍스처를 Maya Displacement 노드에 연결합니다. Displacement 노드를 바다에 사용하는 Ai Standard의 음영 처리 그룹에 연결합니다.



바다에 대한 Vector Displacement로 사용되는 텍스처 맵

- 바다의 폴리곤 면에 대한 Arnold 속성에서 Subdivision Type을 Catclark로 변경하고 반복 횟수를 늘립니다. 먼저 적은 횟수로 반복을 시작하고 물체가 사실적으로 보일 때까지 횟수를 높이십시오. 값이 높아지면 메모리 사용량과 렌더링 시간이 증가하므로, 이 값을 처음에 너무 높게 잡지 않도록 유의하십시오.



바다의 폴리곤 면을 위한 Arnold 세부 설정



바다가 없는 장면



바다의 위치가 옮겨진 장면

톤 매핑

아래에서 Arnold(왼쪽 이미지)의 최종 렌더링과 톤 매핑되고 색상이 보정된 이미지(오른쪽 이미지)를 볼 수 있습니다.



Arnold의 Beauty 렌더링



후 처리 소프트웨어에서 톤 편집 후

요약

지금까지 수중 안개 장면을 표현하는 방법을 알아봤습니다. 안개와 같은 환경 효과는 장면에 사실적인 분위기를 추가하는 간단한 방법입니다. 또한 Ai Gobo를 사용하여 커스틱과 같은 실제 조명 효과를 만들고 더 많은 컨트롤을 활용하며 경제적으로 렌더링할 수 있습니다.

다음은 수중 효과를 추가로 제어하기 위해 **Atmosphere Volume** 및 **Skydome** 조명 등 다른 노드를 사용하는 대체 렌더러의 일부 예제입니다.



Ai Sky 노드에 연결된 램프 텍스처



체적 산란 효과



Ai Gobo 커스틱 효과

기술적인 사항

이 자습서는 MtoA에서 사용하기 위한 쉐이더 작성법을 다루며 다음 단계로 구성됩니다.

- [Maya에서 Arnold Shader 만들기](#)
- [익스텐션\(Extensions\) 쓰기](#)
- [Oculus Rift Camera Node 만들기](#)

Maya에서 Arnold Shader 만들기

이 자습서는 Arnold 쉐이더를 작성하여 Maya에 통합하는 방법을 설명하는 초보자 가이드이며 다음과 같은 페이지로 구성되어 있습니다.

- [쉐이더 생성하기](#)
- [쉐이더 로더 생성하기](#)
- [파라미터 추가하기](#)
- [조명 필터 쉐이더 만들기](#)

쉐이더 생성하기

1.1 Arnold 쉐이더 만들기

간단한 Arnold 쉐이더의 생성 방법은 **Creating a Simple Plugin**를 참조하십시오.

(<https://support.solidangle.com/display/A5ARP/Creating+a+Simple+Plugin>)

Maya 사용자 인터페이스에서 이 쉐이더를 사용할 수 있도록 하려면 애플리케이션에 고유한 메타데이터를 추가해야 합니다.

1.2 Maya에서 쉐이더 사용하기

1.2.1 메타데이터 파일로 쉐이더에 Maya 메타데이터 추가하기

메타데이터를 추가하는 가장 좋은 방법은 쉐이더가 로드될 때 자동으로 사용될 메타데이터 파일을 만드는 것입니다. 이렇게 하면 쉐이더를 다시 컴파일할 필요 없이 수정 작업이 쉬워집니다. 또한 Maya에 고유한 메타데이터가 쉐이더 자체가 아닌 MtoA 폴더에 보관됩니다(Maya 외부에서 다시 사용할 수도 있음).

이것은 다음 코드에서 표시됩니다.

simpleShader.mtd		
1	# simple	0x00070000
2		
3	[node simple]	
4	maya.name	STRING "mySimple"
5	maya.id	INT 0x00070000
6	maya.classification	STRING "shader/surface"
7	maya.output_name	STRING "outColor"
8	maya.output_shortname	STRING "out"
9		
10	[attr color]	
11	maya.name	STRING "constantColor"

쉐이더에 maya.name을 지정하지 않으면 이 쉐이더는 ai가 접두사인 Arnold 노드 이름이 자동으로 부여됩니다. 이 경우에는 이름이 aiSimple이 됩니다.

1.2.2 쉐이더에 대해 Maya ID 선택하기

Autodesk 도움말로부터:

사이트 내부에 영원히 남게 되는 플러그인일 경우에는 서명하지 않은 하나의 int 파라미터를 취하는 생성자를 사용하십시오. 0 - 0x7fff (524288 ids)의 숫자 범위가 이러한 플러그인을 위해 예약되었습니다.

플러그인 개발 키트에서 Maya와 함께 제공되는 예제 플러그인은 0x80000 - 0x ffff f (524288 ids) 범위의 ID를

사용합니다. 이러한 예제 플러그인 중 하나를 사용자 정의한다면 추후 충돌을 피하기 위해 ID를 변경해야 합니다.

사이트간에 공유되는 플러그인은 전 세계적으로 고유한 ID가 있어야 합니다. 오토데스크 개발자 네트워크(ADN)는 256의 블럭 단위로 이러한 ID를 제공합니다. 하나 이상의 24비트 접두사가 할당됩니다. 이것을 얻은 후에는 서명 없는 2개의 int 파라미터를 취하는 MTypeId 생성자를 사용하십시오. 접두사는 첫 번째 파라미터에 들어가며 사용자는 두 번째 파라미터로 들어가는 256개의 ID 할당을 관리해야 합니다. 이 텍스트의 목적상, 0x70000부터 시작하는 ID를 사용하겠지만 귀하에게 더 적합한 ID를 자유롭게 사용하십시오. 또한 MtoA 커뮤니티와 공유하고 싶은 품질 쉐이더를 개발하고 있다면 언제든지 Autodesk에서 MtoA에게 부여한 ID를 요청하십시오.

1.2.3 쉐이더를 위해 Maya 인터페이스 만들기

이 쉐이더에 대해 Maya 인터페이스를 만들고자 한다면 다음 파일을 생성하십시오.

mySimpleTemplate.py

```

1 import pymel.core as pm
2 import mtoa.utils as utils
3 import mtoa.ui.ae.utils as aeUtils
4 from mtoa.ui.ae.shaderTemplate import ShaderAETemplate
5
6 class AEmySimpleTemplate(ShaderAETemplate):
7
8     def setup(self):
9         # Add the shader swatch to the AE
10        self.addSwatch()
11        self.beginScrollLayout()
12
13        # Add a list that allows to replace the shader for other one
14        self.addCustom('message', 'AEshaderTypeNew',
15                      'AEshaderTypeReplace')
16
17        # Begins a "Color Section"
18        self.beginLayout("Color Section", collapse=False)
19        # Add a control for the "constatColor" shader attribute
20        self.addControl("color", label="Color",
21                      annotation="Constant Color")
22
23        self.endLayout()
24
25        # include/call base class/node attributes
26        pm.mel.AEdependNodeTemplate(self.nodeName)
27

```

28	# Add Section for the extra controls not displayed before
29	self.addExtraControls()
30	self.endScrollLayout()

addControl 메소드에서 첫 번째 인수는 쉐이더 속성 이름 또는 해당 속성의 maya.name 메타데이터입니다. 이 경우 파라미터는 constantColor 또는 color입니다.

정의된 클래스 이름은 다음과 같습니다. AE <shader_name> 템플릿과 파일 이름 <shader_name> Template.py 여기서 <shader_name>은 해당 쉐이더의 maya.name 메타데이터 값입니다.

1.2.4 파일을 배치해야 하는 곳

쉐이더가 Maya에서 올바르게 작동하려면 파일의 이름을 지정하고 올바른 위치에 이 파일을 배치하는 것이 중요합니다. 다음 3가지 파일이 있습니다.

컴파일된 쉐이더

이것은 사용자가 컴파일한 쉐이더로, 쉐이더 동작을 담당하게 됩니다. 이 예에서는 Windows에서 simpleShader.dll입니다. 이것을 %MTOA_PATH%\shaders 또는 환경 변수 %ARNOLD_PLUGIN_PATH%에 포함된 다른 모든 폴더에 복사할 수 있습니다.

메타데이터 파일

이것은 모든 메타데이터 정보가 포함된 파일이며, 해당 쉐이더에 대해 Maya가 필요로 하는 정보를 추가합니다. 이 파일의 이름은 컴파일된 파일과 동일해야 하지만 .mtd 확장자를 가져야 합니다. 이 예에서 파일 이름은 simpleShader.mtd입니다. 이 파일은 컴파일된 해당 쉐이더가 위치한 폴더에 복사되어야 합니다.

템플릿 파일

이것은 이 쉐이더에 대한 Maya 템플릿이 정의된 파일입니다. 이 파일의 이름과 클래스 이름은 이전 섹션의 규칙을 따라야 합니다. 이 예제에서는 mySimpleTemplate.py입니다. 이 파일을 %MTOA_PATH%\scripts\mtoa\ui\ae 또는 환경 변수 %MTOA_TEMPLATES_PATH%에 포함된 모든 기타 폴더에 복사할 수 있습니다.

이제 모든 것이 올바른 위치에 있으면 Maya에서 aiSimple 쉐이더를 사용할 때 이 Attribute Editor를 볼 수 있습니다.

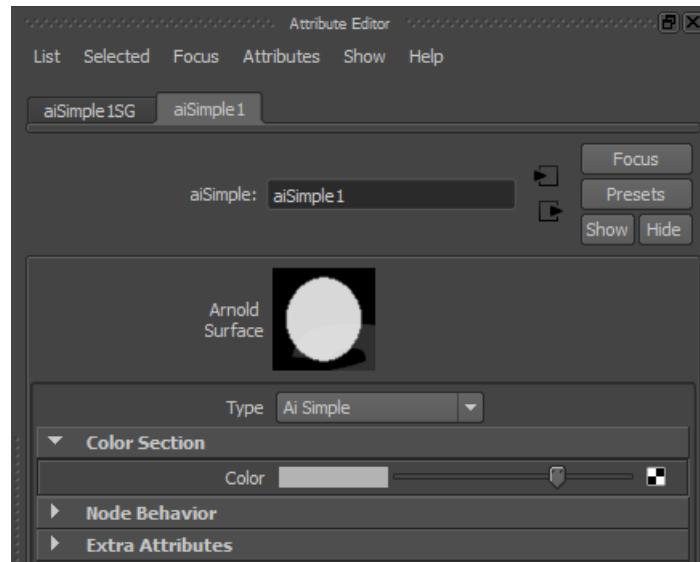


그림 3: 간단한 템플릿

1.2.5 쉐이더에 소스 코드로 Maya 메타데이터 추가하기(고급)

권장 사항은 아니지만 메타데이터 파일을 사용하는 대신, 코드에 Maya 메타데이터를 배치하는 것도 가능합니다. 다음 예제 코드는 이를 위한 구문을 보여줍니다. 그러나 별도의 메타데이터 파일이 더 유연하다는 사실을 기억하십시오.

이 방법의 좋은 용도는 특정 플러그인에 고유하지 않은 필요한 메타데이터를 추가하는 것입니다.

mayaShader.cpp

```

1 node_parameters
2 {
3     // Node metadata
4     AiMetaDataSetStr(mds, NULL, "maya.name", "aiSimple");
5     AiMetaDataSetInt(mds, NULL, "maya.id", 0x00070000);
6     AiMetaDataSetStr(mds, NULL, "maya.classification", "shader/surface");
7     AiMetaDataSetStr(mds, NULL, "maya.output_name", "outColor");
8     AiMetaDataSetStr(mds, NULL, "maya.output_shortname", "out");
9
10    AiParameterRGB("color", 0.7f, 0.7f, 0.7f);
11    AiMetaDataSetStr(mds, "color", "maya.name", "constantColor");
12 }
```

쉐이더 로더 생성하기

2.1 쉐이더 로더 생성하기

쉐이더 로더를 사용하여 단일 공유 라이브러리에 두 개의 쉐이더를 만드는 방법에 대한 예제는 [에서 Multiple Nodes in Library](#)를 참조하십시오.

(<https://support.solidangle.com/display/A5ARP/Multiple+Nodes+in+a+Library>)

2.2 메타데이터 파일 추가

MtoA가 생성된 쉐이더에 대한 메타데이터 정보를 인식하게 만들기 위해 모든 쉐이더에 대해 메타데이터 파일을 생성할 수 있습니다. 이러한 쉐이더에 대한 간단한 메타데이터 파일은 다음과 같습니다.

simple_shaders.mtd

```

# myShader1          0x00070001
# myShader2          0x00070002

[node myShader1]
    maya.name      STRING  "aiShader1"
    maya.id       INT     0x00070001
    maya.classification  STRING  "shader/surface"
    maya.output_name    STRING  "outColor"
    maya.output_shortname STRING  "out"

[attr color1]
    maya.name      STRING  "color"

[node myShader2]
    maya.name      STRING  "aiShader2"
    maya.id       INT     0x00070002
    maya.classification  STRING  "shader/surface"
    maya.output_name    STRING  "outColor"
    maya.output_shortname STRING  "out"

[attr color2]
    maya.name      STRING  "color"

```

그런 다음 이 파일을 컴파일된 쉐이더가 있는 폴더에 복사하십시오. %MTOA_PATH%\shaders\ 메타파일 이름은 컴파일된 쉐이더와 동일해야 합니다. 예를 들어, 로더가 simple_shaders.dll이라면 메타데이터 파일은 simple_shaders.mtd여야 합니다.

파라미터 추가하기

여기에서는 쉐이더에 여러 파라미터를 추가하고 Maya 인터페이스에 통합해 봅니다. 이를 통해 Maya에서 쉐이더를 쉽게 수정할 수 있습니다.

3.1 쉐이더 만들기

이전 섹션에서 작성한 이전 쉐이더 로더에 새 쉐이더를 추가할 수 있습니다. 먼저 쉐이더를 만듭니다.

parametersShader.cpp

```

1 #include <ai.h>
2
3 AI_SHADER_NODE_EXPORT_METHODS(ParametersShaderMtd);
4
5 namespace
6 {
7
8 enum ParametersShaderParams { p_int, p_uint, p_bool, p_float,
9                               p_RGB, p_vector, p_point, p_point2,
10                             p_string, p_matrix, p_enum };
11
12 const char* enum_list[] =
13 {
14     "First value",
15     "Second value",
16     "Third value",
17     NULL
18 };
19
20 };
21
22 node_parameters
23 {
24     AtMatrix id;
25     AiM4Identity(id);
26
27     AiParameterInt("IntParam", 0);
28     AiParameterUInt("UIntParam", 0);
29     AiParameterBool("BoolParam", 0);
30     AiParameterFlt("FltParam", 0.0f);
31     AiParameterRGB("RGBParam", 1.0f, 0.0f, 0.0f);

```

```

32     AiParameterVec("VecParam", 1.0f, 1.0f, 1.0f);
33     AiParameterPnt("PntParam", 0.5f, 0.5f, 0.5f);
34     AiParameterPnt2("Pnt2Param", 0.7f, 0.7f);
35
36     AiParameterStr("StrParam", "");
37     AiParameterMtx("MtxParam", id);
38     AiParameterEnum("EnumParam", 0, enum_list);
39     AiParameterArray("ArrayParam", AiArray(0, 0, AI_TYPE_RGB));
40 }
41
42 node_initialize
43 {
44 }
45
46 node_update
47 {
48 }
49
50 node_finish
51 {
52 }
53
54
55 shader_evaluate
56 {
57     AtColor color = AiShaderEvalParamRGB(p_RGB);
58     sg->out.RGB = color;
59 }

```

Enum values

enum 값에 대한 문자열은 숫자로 시작하면 안 됩니다. 따라서, "1st value"는 유효한 enum 값이 아닙니다.

다양한 유형의 노드 파라미터가 정의되어 있지만 RGBParam만 사용된다는 것을 알 수 있습니다.

이제 쉐이더를 로더에 추가하기 위해서는 이 코드를 이전 섹션의 이전 Loader.cpp 파일에 추가하기만 하면 됩니다.

loader.cpp

```
extern AtNodeMethods* ParametersShaderMtd;
```

```
enum{
```

```

MY_SHADER_1 = 0,
MY_SHADER_2,
PARAMETERS_SHADER
};

...
case PARAMETERS_SHADER:
    node->methods      = ParametersShaderMtd;
    node->output_type   = AI_TYPE_RGB;
    node->name          = "parametersShader";
    node->node_type     = AI_NODE_SHADER;
    break;

```

이전 섹션에서 설명한 대로 로더를 다시 컴파일하고 Arnold가 쉐이더를 올바르게 로드할 수 있는지 확인한 다음, 컴파일된 쉐이더를 올바른 폴더에 복사하여 Maya에서 사용할 수 있도록 하십시오. 이렇게 하면 Maya에서 사용할 수는 있지만 잘 통합되지는 않는다는 것을 알 수 있습니다.

3.2 Maya에서 쉐이더 통합하기

이 새 쉐이더를 통합하려면 첫 번째 섹션에서와 같이 메타데이터 파일에 정보를 추가하고 이 쉐이더에 대한 템플릿 스크립트를 만듭니다.

3.2.1 메타데이터 정보 추가하기

메타 데이터 정보를 추가하려면 다음 폴더에 배치한 이전 섹션에서 생성된 loader.mtd 파일에 이것을 추가해야 합니다. %MTOA_PATH%shaders%

loader.mtd

```

25 [node parametersShader]
26     desc          STRING "Arnold shader to test parameters."
27     maya.name      STRING "aiParametersShader"
28     maya.id        INT   0x00070003
29     maya.classification STRING "shader/surface"
30     maya.output_name   STRING "outColor"
31     maya.output_shortname STRING "out"
32
33 [attr IntParam]
34     desc          STRING "Integer parameter."
35     min           INT   -100
36     max           INT   100
37     softmin        INT   -10
38     softmax        INT   10

```

```

39    default      INT   -10
40    maya.shortname  STRING "int"
41    maya.name      STRING "integer"
42    [attr UIntParam]
43        desc        STRING "UInteger parameter."
44        min         INT    0
45        max         INT    100
46        softmin     INT    0
47        softmax     INT    10
48        default     INT    0
49        maya.shortname  STRING "uint"
50        maya.name      STRING "uinteger"
51    [attr BoolParam]
52        desc        STRING "Bool parameter."
53        default     BOOL   True
54        maya.shortname  STRING "bl"
55        maya.name      STRING "bool"
56    [attr FltParam]
57        desc        STRING "Float parameter."
58        min         FLOAT  0
59        max         FLOAT  10
60        softmin     FLOAT  0
61        softmax     FLOAT  1
62        default     FLOAT  0.5
63        maya.shortname  STRING "flt"
64        maya.name      STRING "float"
65    [attr RGBParam]
66        desc        STRING "RGB parameter."
67        maya.name      STRING "color"
68        maya.shortname  STRING "cl"
69    [attr VecParam]
70        desc        STRING "Vector parameter."
71        maya.name      STRING "vector"
72        maya.shortname  STRING "vec"
73    [attr PntParam]
74        desc        STRING "Point parameter."
75        maya.name      STRING "point"
76        maya.shortname  STRING "pnt"
77    [attr Pnt2Param]
78        desc        STRING "2D Point parameter."
79        maya.name      STRING "point2"

```

```

80    maya.shortname    STRING "pnt2"
81    [attr StrParam]
82        desc          STRING "String parameter."
83        maya.name      STRING "string"
84        maya.shortname STRING "str"
85        default        STRING "empty"
86    [attr MtxParam]
87        desc          STRING "Matrix parameter."
88        maya.name      STRING "matrix"
89        maya.shortname STRING "mtx"
90    [attr EnumParam]
91        desc          STRING "Enumeration parameter."
92        default        INT 1
93        maya.name      STRING "enumeration"
94        maya.shortname STRING "enum"
95    [attr ArrayParam]
96        desc          STRING "Color Array parameter."
97        maya.name      STRING "colorarray"
98        maya.shortname STRING "carray"

```

여기에는 이름, 짧은 이름, 슬라이더 제한, 설명 및 기본값과 같은 파라미터의 Maya 표현에 사용될 속성이 설명되어 있습니다.

3.2.2 Maya 템플릿 추가하기

마지막으로 해야 할 작업은 이 쉐이더에 대해 Maya 템플릿을 생성하는 것입니다. 이것은 첫 번째 섹션에서 생성한 `self.addControl(parameterName, label="Parameter Label")`와 유사합니다. 하지만 나머지 컨트롤을 추가합니다. 각 속성에 대한 올바른 컨트롤이 자동으로 생성됩니다.

`%MTOA_PATH%\scripts\mtoa\ui\ae\` 또는 `%MTOA_TEMPLATES_PATH%`에서 정의된 폴더에 `aiParametersShaderTemplate.py` 파일을 만듭니다.

aiParametersShaderTemplate.py

```

1  import pymel.core as pm
2  from mtoa.ui.ae.shaderTemplate import ShaderAETemplate
3
4  class AEaiParametersShaderTemplate(ShaderAETemplate):
5      def setup(self):
6          self.addSwatch()
7          self.beginScrollLayout()
8          self.addCustom('message', 'AEshaderTypeNew',
9                        'AEshaderTypeReplace')

```

```

10
11      self.beginLayout("Parameters list", collapse=False)
12      self.addControl("integer", label="Integer Param")
13      self.addControl("uinteger", label="Unsigned Int Param")
14      self.addControl("bool", label="Bool param")
15      self.addControl("float", label="Float param")
16      self.addControl("color", label="Color param")
17      self.addControl("vector", label="Vector param")
18      self.addControl("point", label="Point param")
19      self.addControl("point2", label="2D Point param")
20      self.addControl("string", label="String param")
21      self.addControl("matrix", label="Matrix param")
22      self.addControl("enumeration", label="Enumeration param")
23      self.addControl("colorarray", label="Color Array param")
24      self.endLayout()
25
26      pm.mel.AEdependNodeTemplate(self.nodeName)
27
28      self.addExtraControls()
29      self.endScrollLayout()

```

Maya에서 ParametersShader를 사용할 때는 이에 대해 Attribute Editor를 볼 수 있습니다.

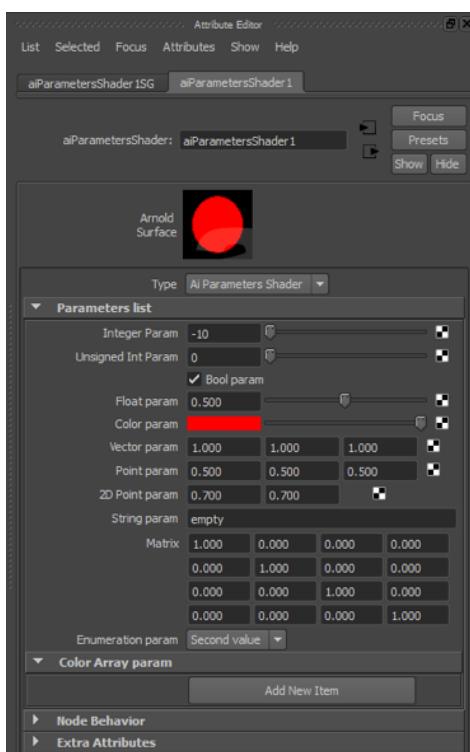


그림 4: ParametersShader 템플릿

이제 Maya로부터 쉐이더의 입력 파라미터를 구성할 수 있습니다.

3.2.3 템플릿에서 공간 최적화 피하기

Maya는 자동으로 레이아웃 공간을 최적화하려고 시도합니다. 예를 들어, 이전 코드에서 다음으로부터:

aiParametersShaderTemplate.py

```
...
self.addControl("uinteger", label="Unsigned Int Param")
self.addControl("bool", label="Bool param")
self.addControl("float", label="Float param")
...

```

다음으로 변경한 경우:

aiParametersShaderTemplate.py

```
...
self.addControl("uinteger", label="Unsigned Int Param")
self.addControl("bool", label="Bool param 1")
self.addControl("bool", label="Bool param 2")
self.addControl("float", label="Float param")
...

```

다음 결과가 나타납니다.



그림 5: 최적화된 템플릿

이를 피하기 위해서 다음 코드를 사용할 수 있습니다.

aiParametersShaderTemplate.py

```
...
self.addControl("uinteger", label="Unsigned Int Param")
self.beginNoOptimize()
self.addControl("bool", label="Bool param 1")
self.addControl("bool", label="Bool param 2")
self.endNoOptimize()
self.addControl("float", label="Float param")
...

```

그러면 다음 결과를 얻게 됩니다.

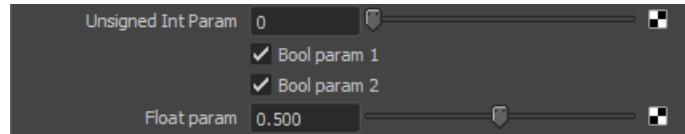


그림 6: 최적화된 템플릿

3.2.4 속성이 변경될 때 호출되는 메서드

때로는 속성이 변경될 때마다 수행될 특정 작업이 필요할 수 있습니다. 예를 들어, 도입된 값이 올바른지 확인해야 할 수 있습니다. uinteger 속성의 값이 항상 정수 속성보다 크거나 같아야 한다고 가정해 보겠습니다. 다음 코드로 이를 실행해 볼 수 있습니다.

aiParametersShaderTemplate.py

```
import pymel.core as pm
import maya.cmds as cmds
from mtoa.ui.ae.shaderTemplate import ShaderAETemplate

class AEaiParametersShaderTemplate(ShaderAETemplate):
    def checkInteger(self, nodeName):
        intAttr = self.nodeAttr('integer')
        uintAttr = self.nodeAttr('uinteger')
        intValue = cmds.getAttr(intAttr)
        uintValue = cmds.getAttr(uintAttr)
        if intValue > uintValue:
            cmds.setAttr(uintAttr, intValue)
    def setup(self):
        self.addSwatch()
        self.beginScrollLayout()
        self.addCustom('message', 'AEshaderTypeNew',
                      'AEshaderTypeReplace')
        self.beginLayout("Parameters list", collapse=False)
        self.addControl("integer", label="Integer Param",
                       changeCommand=self.checkInteger)
        self.addControl("uinteger", label="Unsigned Int Param")
        ...
...
```

이제 정수 속성이 업데이트될 때마다 uinteger 값보다 큰 값이 있으면 이 값이 증가하여 해당 값과 같아집니다. 하지만 uinteger 파라미터를 줄이면 이것이 정수 파라미터보다 낮을 수 있습니다. 이러한 경우에도 다음 코드를 쉽게 작성할 수 있습니다.

aiParametersShaderTemplate.py

```

import pymel.core as pm
import maya.cmds as cmds
from mtoa.ui.ae.shaderTemplate import ShaderAETemplate
class AEaiParametersShaderTemplate(ShaderAETemplate):
    def checkInteger(self, nodeName):
        intAttr = self.nodeAttr('integer')
        uintAttr = self.nodeAttr('uinteger')
        intValue = cmds.getAttr(intAttr)
        uintValue = cmds.getAttr(uintAttr)
        if intValue > uintValue:
            cmds.setAttr(uintAttr,intValue)
    def checkUInteger(self, nodeName):
        intAttr = self.nodeAttr('integer')
        uintAttr = self.nodeAttr('uinteger')
        intValue = cmds.getAttr(intAttr)
        uintValue = cmds.getAttr(uintAttr)
        if intValue > uintValue:
            cmds.setAttr(intAttr,uintValue)
    def setup(self):
        self.addSwatch()
        self.beginScrollLayout()
        self.addCustom('message', 'AEshaderTypeNew',
                      'AEshaderTypeReplace')
        self.beginLayout("Parameters list", collapse=False)
        self.addControl("integer", label="Integer Param",
                       changeCommand=self.checkInteger)
        self.addControl("uinteger", label="Unsigned Int Param",
                       changeCommand=self.checkUInteger)
        ...

```

3.2.5 컨트롤 활성화 및 비활성화하기

때때로 일부 파라미터는 다른 속성 값에 따라 아무 의미가 없는 경우가 있습니다. 예를 들어, float 속성이 bool 속성이 `true`일 때만 의미가 있다고 가정해 봅시다. `bool`이 `false`일 때 float 속성을 비활성화하면 사용자에게 이 점을 명확하게 전달할 수 있습니다. 이렇게 하기 위해서, 먼저 이전 섹션에서처럼 `changeCommand` 메소드를 정의할 수 있고 `arnoldDimControlIfFalse` 메소드를 사용하여 속성을 비활성화할 수 있습니다. 그 예는 다음과 같습니다.

aiParametersShaderTemplate.py

```

import pymel.core as pm
import maya.cmds as cmds
import mtoa.ui.ae.utils as aeUtils

```

```

from mtoa.ui.ae.shaderTemplate import ShaderAETemplate
class AEaiParametersShaderTemplate(ShaderAETemplate):
    def changeBool(self, nodeName):
        aeUtils.arnoldDimControlIfFalse(nodeName, "float", "bool")
    def setup(self):
        self.addSwatch()
        self.beginScrollLayout()
        self.addCustom('message', 'AEshaderTypeNew',
                      'AEshaderTypeReplace')
        self.beginLayout("Parameters list", collapse=False)
        self.addControl("integer", label="Integer Param")
        self.addControl("uinteger", label="Unsigned Int Param")
        self.addControl("bool", label="Bool param",
                       changeCommand=self.changeBool)
        self.addControl("float", label="Float param")
        ...

```

반대로, float 속성을 bool이 false인 경우에만 활성화되도록 하려면 aeUtils.arnoldDimControlIfTrue 대신 aeUtils.arnoldDimControlIfFalse를 사용할 수 있습니다. 그러나 경우에 따라 속성을 활성화 또는 비활성화하는 조건이 더 복잡할 수 있습니다. 예를 들어, 색상의 빨간색 구성 요소가 0보다 큰 경우에만 float 값을 활성화할 수 있습니다. 이 경우에는 다음과 같은 코드를 작성할 수 있습니다.

aiParametersShaderTemplate.py

```

import pymel.core as pm
import maya.cmds as cmds
import mtoa.ui.ae.utils as aeUtils
from mtoa.ui.ae.shaderTemplate import ShaderAETemplate

class AEaiParametersShaderTemplate(ShaderAETemplate):
    def changeColor(self, nodeName):
        redAttr = self.nodeAttr('colorR')
        redValue = pm.getAttr(redAttr)
        dim = not (redValue > 0)
        pm.editorTemplate(dimControl=(nodeName, "float", dim))
    def setup(self):
        self.addSwatch()
        self.beginScrollLayout()
        self.addCustom('message', 'AEshaderTypeNew',
                      'AEshaderTypeReplace')
        self.beginLayout("Parameters list", collapse=False)
        self.addControl("integer", label="Integer Param")

```

```

self.addControl("uinteger", label="Unsigned Int Param")
self.addControl("bool", label="Bool param")
self.addControl("float", label="Float param")
self.addControl("color", label="Color param",
    changeCommand=self.changeColor)

...

```

3.2.6 사용자 정의 컨트롤 만들기

MtoA는 유형에 따라 속성에 대한 컨트롤을 자동으로 만들지만 가끔은 더 많은 유연성이 있어야 특정 속성에 대한 사용자 정의 컨트롤을 만들 수 있습니다. 예를 들어, 앞의 예제에서 문자열 속성이 특정 파일을 가리키는 데 사용되었다고 가정해 보겠습니다. 그런 다음 컨트롤에 파일 탐색기를 보여주는 버튼을 원할 수 있으므로 파일 이름을 보다 직관적인 방식으로 입력할 수 있습니다. 이러한 동작을 위한 예제는 다음과 같습니다.

```

aiParametersShaderTemplate.py
import pymel.core as pm
import maya.cmds as cmds
from mtoa.ui.ae.shaderTemplate import ShaderAETemplate
class AEaiParametersShaderTemplate(ShaderAETemplate):
    def filenameEdit(self, mData):
        attr = self.nodeAttr('string')
        cmds.setAttr(attr, mData, type="string")
    def LoadFilenameButtonPush(self, *args):
        basicFilter = 'All Files (*.*)'
        ret = cmds.fileDialog2(fileFilter=basicFilter, dialogStyle=2,
                               cap='Load File', okc='Load', fm=4)
        if ret is not None and len(ret):
            self.filenameEdit(ret[0])
            cmds.textFieldButtonGrp("filenameGrp", edit=True, text=ret[0])

    def filenameNew(self, nodeName):
        path = cmds.textFieldButtonGrp("filenameGrp", label="File Name",
                                      changeCommand=self.filenameEdit, width=300)
        cmds.textFieldButtonGrp(path, edit=True, text=cmds.getAttr(nodeName))
        cmds.textFieldButtonGrp(path, edit=True, buttonLabel="...",
                               buttonCommand=self.LoadFilenameButtonPush)

    def filenameReplace(self, nodeName):
        cmds.textFieldButtonGrp("filenameGrp", edit=True,
                               text=cmds.getAttr(nodeName) )

    def setup(self):

```

```

self.addSwatch()
self.beginScrollLayout()
self.addCustom('message', 'AEshaderTypeNew',
    'AEshaderTypeReplace')
self.beginLayout("Parameters list", collapse=False)
...
self.addCustom('string', self.filenameNew, self.filenameReplace)
...

```

3.2.7 템플릿 명령 요약

여기서는 이전에 사용된 모든 템플릿 메서드를 요약하고 새로운 템플릿 메서드를 추가해 보겠습니다.

addSwatch

앞서 보았듯이, 이 명령은 Attribute Editor에서 Shader의 미리보기를 보여주는 Swatch를 생성합니다.

beginScrollLayout endScrollLayout

섹션과 속성을 배치할 수 있는 Vertical Scroll Layout을 시작 및 종료합니다.

beginLayout endLayout

레이아웃을 시작 및 종료합니다. 이름을 지정하고 기본적으로 축소할지 여부를 결정할 수 있습니다.

```
self.beginLayout("Section Name", collapse=False)
```

addControl

쉐이더 속성에 대한 컨트롤을 만듭니다. 속성의 유형에 따라 올바른 컨트롤을 자동으로 생성합니다. alabel 및 changeCommand를 정의할 수 있습니다.

```
self.addControl("attrname", label="My Attr", changeCommand=self.changeAttr)
```

addCustom

쉐이더 속성에 대한 사용자 정의 컨트롤을 만듭니다.

```
self.addCustom(attrName, newMethod, replaceMethod)
```

addSeparator

레이아웃에 구분 기호를 추가합니다.

beginNoOptimize endNoOptimize

레이아웃 공간 최적화를 중단하고 다시 시작합니다.

addBumpLayout

쉐이더에 범프 법선 매핑을 연결할 수 있게 해주는 섹션을 만듭니다.

addExtraControls

모든 추가 속성이 “Extra Controls” 섹션에서 그룹화됩니다.

addCustom('message','AEshaderTypeNew','AEshaderTypeReplace')

노드의 쉐이더 유형을 변경할 수 있는 경우 “Type” 스크롤 목록을 생성합니다.

pm.mel.AEdependNodeTemplate(self.nodeName)

쉐이더 템플릿에서 “Node Behavior” 섹션을 생성합니다.

조명 필터 쉐이더 만들기

이 쉐이더를 생성하려면 이전처럼 소스 코드 파일, 로더, 메타데이터 및 템플릿이 필요합니다. 여기서 살펴보겠습니다.

4.1 쉐이더 만들기

이전 섹션에서 작성한 이전 쉐이더 로더에 조명 필터 쉐이더를 추가할 수 있습니다. 먼저 쉐이더를 만듭니다.

simpleLightFilter.cpp

```
#include <ai.h>

AI_SHADER_NODE_EXPORT_METHODS(SimpleLightFilterMtd);

enum simpleLightFilterParams {
    p_intensity
};

node_parameters
{
    AiParameterFlt("Intensity", 0.0f);
}

node_initialize
{
}

node_update
{
}

node_finish
{
}

shader_evaluate
{
    // test if we are running as a light filter
    if (sg->light_filter)
    {
        float intensity = AiShaderEvalParamFlt(p_intensity);
```

```

    sg->light_filter->Liu = sg->light_filter->Liu*intensity;
}
}

```

이제 쉐이더를 로더에 추가하기 위해서는 이 코드를 이전 섹션으로부터 이전 Loader.cpp 파일에 추가하기만 하면 됩니다.

loader.cpp

```

...
extern AtNodeMethods* SimpleLightFilterMtd;

enum{
    SHADER_1 = 0,
    SHADER_2,
    PARAMETERS_SHADER,
    SIMPLE_LIGHT_FILTER
};

...
...

case SIMPLE_LIGHT_FILTER:
    node->methods      = SimpleLightFilterMtd;
    node->output_type   = AI_TYPE_RGB;
    node->name          = "simpleLightFilter";
    node->node_type     = AI_NODE_SHADER;
    break;
...

```

이전 섹션에서 설명한 대로 조명 필터와 로더를 컴파일하고 Arnold가 쉐이더를 올바르게 로드할 수 있는지 확인한 다음, 컴파일된 쉐이더를 올바른 폴더(일반 쉐이더인 것처럼)에 복사하여 Maya에서 사용할 수 있도록 하십시오.

4.2 Maya에 조명 필터 통합하기

Maya에 조명 필터를 통합하려면 메타데이터 파일에 정보를 추가하고 이전 섹션에서처럼 템플릿 스크립트를 만들어야 합니다.

4.2.1 조명 필터 메타데이터 정보 추가하기

메타데이터 정보를 추가하기 위해서는 컴파일한 쉐이더 파일과 동일한 폴더에 배치한 이전 섹션에서 생성된 loader.mtd 파일에 이것을 추가하기만 하면 됩니다.

loader.mtd

101 [node simplyLightFilter]

```

102    maya.name      STRING  "aiSimplyLightFilter"
103    maya.id       INT    0x00070004
104    maya.classification STRING  "light/filter"
105    maya.lights     STRING  "pointLight spotLight"
106
107    [attr Intensity]
108        maya.name      STRING  "lightIntensity"
109        softmin        FLOAT   0
110        softmax        FLOAT   10
111        default        FLOAT   1

```

여기서 `maya.classification`을 "light/filter"로 설정하는 것이 중요합니다. 또한 `Maya.lights`를 다음과 같이 이 조명 필터를 사용할 수 있는 모든 조명이 포함된 문자열로 설정합니다. `ambientLight`, `directionalLight`, `pointLight`, `spotLight`, `areaLight` 또는 `aiAreaLight`.

4.2.2 Maya 템플릿 추가하기

이전 섹션에서와 마찬가지로 일반 쉐이더와 동일한 방식으로 이 조명 필터에 대해 간단한 템플릿을 생성할 수 있습니다. 이제 조명 필터에서 적합하지 않은 일부 명령을 피해보도록 하겠습니다.

aiSimpleLightFilterTemplate.py

```

import pymel.core as pm
from mtoa.ui.ae.shaderTemplate import ShaderAETemplate

class AEaiSimpleLightFilterTemplate(ShaderAETemplate):
    def setup(self):
        self.beginScrollLayout()

        self.beginLayout("Parameters", collapse=False)
        self.addControl('lightIntensity', label="Light Intensity")

        self.endLayout()

    pm.mel.AEdependNodeTemplate(self.nodeName)
    self.addExtraControls()
    self.endScrollLayout()

```

이제 이 조명 필터에 의해 받아 들여지는 빛의 유형을 만들 때, 해당 조명의 "Arnold" 섹션에 있는 "Light Filters" 섹션에 이 조명 필터를 추가할 수 있습니다.

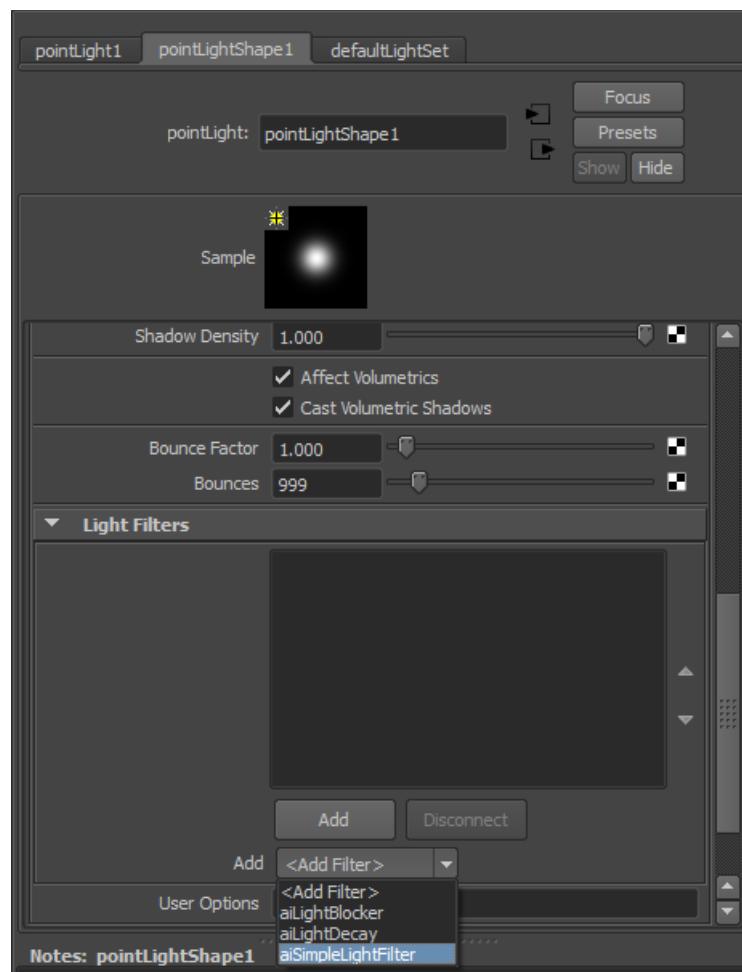


그림 7: 조명 필터 메뉴

익스텐션(Extensions) 쓰기

extension은 일반적으로 장면을 내보내거나 렌더링할 때 Arnold 노드로 변환하는 데 사용되는 Maya 노드와 관련된 변환기입니다.

실제로 이것은 다음 두 가지를 허용합니다 :

- MtoA에 의해 지원되지 않는 노드를 렌더링할 수 있음. 이것은 지원되지 않는 Maya 노드이거나 사용자 정의 Maya 노드이기 때문일 수 있습니다.
- 현재 지원되는 Maya 노드의 새로운 동작을 정의할 수 있습니다. 따라서 사용자는 지원되는 동작 또는 익스텐션 작성자가 정의한 새 동작을 선택할 수 있습니다.

거의 모든 Maya 노드를 확장할 수 있지만 Maya 노드를 확장하는 데는 몇 가지 제한이 있습니다. 간단하게 말해서 눈에 보이는 모든 DAG 노드는 내보내기 되며, 모든 DG 노드는 내보낸 노드에 연결된다고 가정할 수 있습니다. 또한 종속 노드를 가진 모든 노드가 내보내기 됩니다.

- [기본 익스텐션 만들기](#)
- [노드에 특정 Arnold 속성 추가하기](#)
- [동일한 노드에 대한 여러 변환기](#)
- [사용자 정의 노드의 변환기](#)
- [익스텐션을 MtoA 1.4.0 으로 포팅](#)

기본 익스텐션 만들기

고유한 Phong Maya 쉐이더를 표준 Arnold 쉐이더로 변환하는 매우 간단한 익스텐션을 생성할 수 있습니다. 현재 이것은 MtoA에서 지원되지 않으며 관련된 변환기가 없습니다.

먼저 우리가 필요로 하는 MtoA 변환기로부터 상속받은 새로운 클래스를 만들어야 합니다. 이것은 간단한 예제이므로 가장 단순한 CNodeTranslator로부터 상속을 시작합니다.

translator1.h

```

1 #include "translators/NodeTranslator.h"
2
3 class CTestTranslatorCmd
4     : public CNodeTranslator
5 {
6 public:
7     AtNode* CreateArnoldNodes();
8     void Export(AtNode* );
9     void Update(AtNode* );
10    static void* creator();
11 };

```

그리고 메소드를 매우 간단한 방식으로 재정의합니다.

creator

정의된 클래스의 인스턴스만 반환합니다.

CreateArnoldNodes

내보낼 Arnold 노드를 만듭니다. Maya Phong을 Arnold 표준으로 변환할 것이므로 여기에 Arnold 표준을 만듭니다.

Export

IPR 세션 중에 변경되지 않는 속성을 내보냅니다. 여기서는 실험의 목적으로 Diffuse Phong 속성을 표준 Kd 속성으로 내보내겠습니다.

Update

IPR 세션 중에 업데이트할 수 있는 속성을 내보냅니다. 여기서 색상 Phong 속성을 표준 Kd 색상 속성으로 내보낼 것입니다.

translator1.cpp

```

1 #include "translator1.h"
2
3 #include <ai_msg.h>

```

```

4 #include <ai_nodes.h>
5
6 void CTestTranslatorCmd::Export(AtNode* shader)
7 {
8     AiMsgInfo("[test extension] Exporting %s", GetMayaNodeName().asChar());
9     AiNodeSetFlt(shader, "Kd", FindMayaPlug("diffuse").asFloat() );
10    Update(shader);
11 }
12
13 void CTestTranslatorCmd::Update(AtNode* shader)
14 {
15     AiMsgInfo("[test extension] Updating %s", GetMayaNodeName().asChar());
16     AiNodeSetRGB(shader, "Kd_color",
17                  FindMayaPlug("colorR").asFloat(),
18                  FindMayaPlug("colorG").asFloat(),
19                  FindMayaPlug("colorB").asFloat());
20 }
21
22 AtNode* CTestTranslatorCmd::CreateArnoldNodes()
23 {
24     return AddArnoldNode("standard");
25 }
26
27 void* CTestTranslatorCmd::creator()
28 {
29     return new CTestTranslatorCmd();
30 }

```

이것은 하나의 예일 뿐이며 Phong 쉐이더를 유사한 표준 쉐이더로 올바르게 내보내지 않습니다.
마지막으로 Maya Phong 쉐이더용 변환기를 등록할 코드가 필요합니다.

extension1.cpp

```

1 #include "translator1.h"
2 #include "extension/Extension.h"
3
4
5 extern "C"
6 {
7
8     DLLEXPORT void initializeExtension(CExtension &plugin)

```

```

9     {
10    plugin.RegisterTranslator("phong",
11        "",
12        CTestTranslatorCmd::creator);
13 }
14
15 DLLEXPORT void deinitializeExtension(CExtension &plugin)
16 {
17 }
18
19 }

```

이것을 컴파일하려면 다음 환경 변수를 설정해야 합니다. ARNOLD_PATH, MAYA_PATH and MTOA_PATH:

```

set ARNOLD_PATH="C:\solidangle\releases\Arnold-X.X.X.X-platform"
set MTOA_PATH="C:\solidangle\mtoadeploy\20XX"
set MAYA_PATH="C:\Program Files\Autodesk\Maya20XX"

```

익스텐션을 컴파일하려면 OS에 따라 다음을 실행하십시오.

Windows

Visual Studio 명령 프롬프트를 열고 다음 명령을 실행합니다.

```
cl /c translator1.cpp /EHsc /MD /DNT_PLUGIN /DREQUIRE_IOSTREAM /I%MAYA_PATH%Winclu
/I%MTOA_PATH%Winclu /I%ARNOLD_PATH%Winclu
```

```
cl /c extension1.cpp /EHsc /MD /DNT_PLUGIN /DREQUIRE_IOSTREAM /I%MAYA_PATH%Winclu
/I%MTOA_PATH%Winclu /I%ARNOLD_PATH%Winclu
```

```
link /dll extension1.obj translator1.obj /LIBPATH:%ARNOLD_PATH%lib /LIBPATH:%MAYA_PATH%lib
/LIBPATH:%MTOA_PATH%lib ai.lib OpenGl32.lib glu32.lib Foundation.lib OpenMaya.lib OpenMayaRender.lib
OpenMayaUI.lib OpenMayaAnim.lib OpenMayaFX.lib mtoa_api.lib
```

그 다음 생성된 extension1.dll을 다음 경로에 복사해야 합니다.

MTOA_PATH\extensions

이제 Maya를 열고 임의의 오브젝트를 Phong 쉐이더에 할당하면 Lambert와 비슷한 IPR 세션에서 렌더링됩니다. 색상 Phong 속성을 변경하면 IPR에서 업데이트됩니다. 그러나 Diffuse Phong 속성을 변경하면 IPR을 중지하고 다시 시작할 때까지 업데이트되지 않습니다. 이는 Diffuse Phong 속성을 Update 메서드에서 내보내는 대신 Export 메서드에서 내보냈기 때문입니다.

장면을 내보내는 경우 Phong Maya 쉐이더가 다음과 같이 내보내진 것을 알 수 있습니다.

...

```
standard
{
    name phong1
    Kd 0.8
    Kd_color 0.5 0.5 0.5
}
```

...

여기에서 필요한 만큼 Export 및 Update 방법을 복잡하게 만들 수 있습니다.

노드에 특정 Arnold 속성 추가하기

이제 Maya 노드에 없는 Phong의 변환기에 대해 특정 속성이 필요할 수 있습니다. 이 속성을 생성하고 Arnold 속성에 대해 해당 노드의 Attribute Editor에 섹션을 추가할 수 있습니다.

Phong 변환기를 따라서 Arnold Translator에서 사용될 specular weight 속성을 추가할 것입니다. 먼저 NodeInitializer 메소드가 필요합니다.

translator2.h

```

1 #include "translators/NodeTranslator.h"
2
3 class CTestTranslatorCmd2
4   : public CNodeTranslator
5 {
6 public:
7   AtNode* CreateArnoldNodes();
8   void Export(AtNode* );
9   void Update(AtNode* );
10  static void* creator();
11  static void NodeInitializer(CAbTranslator context);
12 };

```

여기서 필요한 특성을 가진 특정 속성을 만들어보겠습니다. 부드럽고 딱딱한 범위를 정의하여 속성 값을 제한할 수 있습니다.

translator2.cpp

```

1 #include "translator2.h"
2
3 #include <ai_msg.h>
4 #include <ai_nodes.h>
5
6 void CTestTranslatorCmd2::Export(AtNode* shader)
7 {
8   AiMsgInfo("[test extension2] Exporting %s", GetMayaNodeName().asChar());
9   AiNodeSetFlt(shader, "Kd", FindMayaPlug("diffuse").asFloat() );
10  Update(shader);
11 }
12
13 void CTestTranslatorCmd2::Update(AtNode* shader)
14 {
15   AiMsgInfo("[test extension2] Making %s red", GetMayaNodeName().asChar());
16   AiNodeSetRGB(shader, "Kd_color",

```

```

17     FindMayaPlug("colorR").asFloat(),
18     FindMayaPlug("colorR").asFloat(),
19     FindMayaPlug("colorR").asFloat());
20     AiNodeSetFlt(shader, "Ks", FindMayaPlug("aiSpecularWeight").asFloat() );
21 }
22
23 AtNode* CTestTranslatorCmd2::CreateArnoldNodes()
24 {
25     return AddArnoldNode("standard");
26 }
27
28 void* CTestTranslatorCmd2::creator()
29 {
30     return new CTestTranslatorCmd2();
31 }
32
33 void CTestTranslatorCmd2::NodeInitializer(CAbTranslator context)
34 {
35     CExtensionAttrHelper helper(context.maya, "standard");
36
37     CAttrData data;
38
39     data.name = "aiSpecularWeight";
40     data.shortName = "ai_specular_weight";
41     data.hasSoftMin = true;
42     data.softMinFLT = 0.1f;
43     data.hasSoftMax = true;
44     data.softMaxFLT = 0.9f;
45     data.hasMin = true;
46     data.minFLT = 0.f;
47     data.hasMax = true;
48     data.maxFLT = 1.0;
49     data.defaultValueFLT = 0.5f;
50     helper.MakeInputFloat(data);
51
52 }

```

마지막으로 익스텐션을 등록할 때 NodeInitializer 메소드를 제공해야 합니다.

extension2.cpp

```

1 #include "translator2.h"
2 #include "extension/Extension.h"
3
4 extern "C"
5 {
6
7 DLLEXPORT void initializeExtension(CExtension &plugin)
8 {
9     plugin.RegisterTranslator("phong",
10             "",
11             CTestTranslatorCmd2::creator,
12             CTestTranslatorCmd2::NodeInitializer);
13 }
14
15 DLLEXPORT void deinitializeExtension(CExtension &plugin)
16 {
17 }
18
19 }

```

마지막으로 이 속성을 표시할 Attribute Editor에서 Arnold 섹션을 만드는 것도 좋습니다. 컴파일된 익스텐션과 동일한 이름을 가진 python 파일을 같은 폴더 MTOA_PATH\extensions에 배치하여 이 작업을 수행할 수 있습니다.

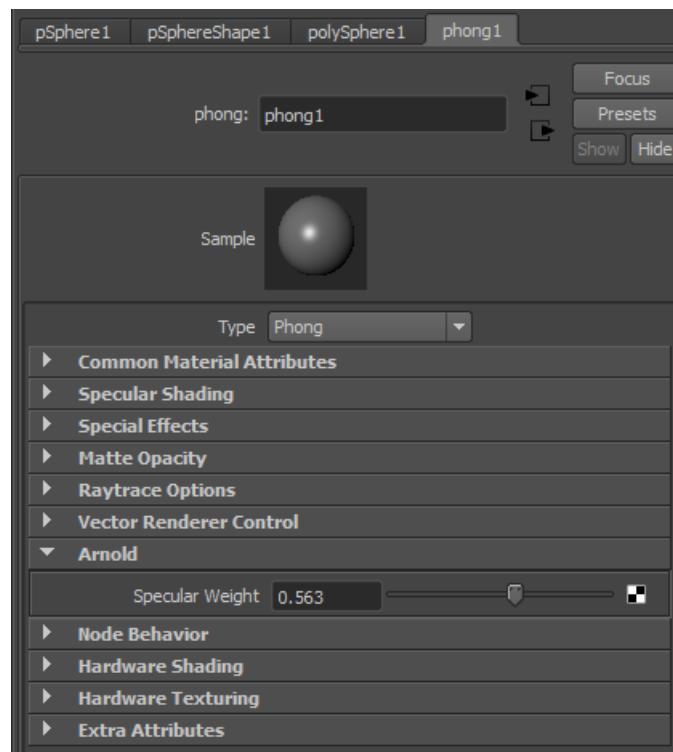
extension2.py

```

1 import mtoa.ui.ae.templates as templates
2 from mtoa.ui.ae.templates import AttributeTemplate
3
4 class Phong1Template(AttributeTemplate):
5     def setup(self):
6         self.addControl("aiSpecularWeight")
7
8 templates.registerAETemplate(Phong1Template, 'phong')

```

그러면 다음과 같이 Phong 쉐이더의 Attribute Editor에 섹션이 생성됩니다.



동일한 노드에 대한 여러 변환기

둘 이상의 변환기를 동일한 Maya 노드에 연결할 수 있습니다. 다음 단계를 따라야 합니다.

먼저 변환기를 등록할 때 이름을 지정해야 합니다. 다음과 같이 첫 번째 익스텐션을 다시 쓸 수 있습니다.

extension1.cpp

```

1 #include "translator1.h"
2 #include "extension/Extension.h"
3
4 extern "C"
5 {
6
7 DLLEXPORT void initializeExtension(CExtension &plugin)
8 {
9     plugin.RegisterTranslator("phong",
10                     "trans1",
11                     CTestTranslatorCmd::creator);
12 }
13
14 DLLEXPORT void deinitializeExtension(CExtension &plugin)
15 {
16 }
17
18 }
```

And the second like this:

extension2.cpp

```

1 #include "translator2.h"
2 #include "extension/Extension.h"
3
4 extern "C"
5 {
6
7 DLLEXPORT void initializeExtension(CExtension &plugin)
8 {
9     plugin.RegisterTranslator("phong",
10                     "trans2",
11                     CTestTranslatorCmd2::creator,
12                     CTestTranslatorCmd2::NodeInitializer);
13 }
14 }
```

```

15 DLLEXPORT void deinitializeExtension(CExtension &plugin)
16 {
17 }
18
19 }

```

추후 각각에 대해 서로 다른 Attribute Editor 템플릿을 갖기 위해 다음과 같이 이전 템플릿을 다시 작성할 수 있습니다.

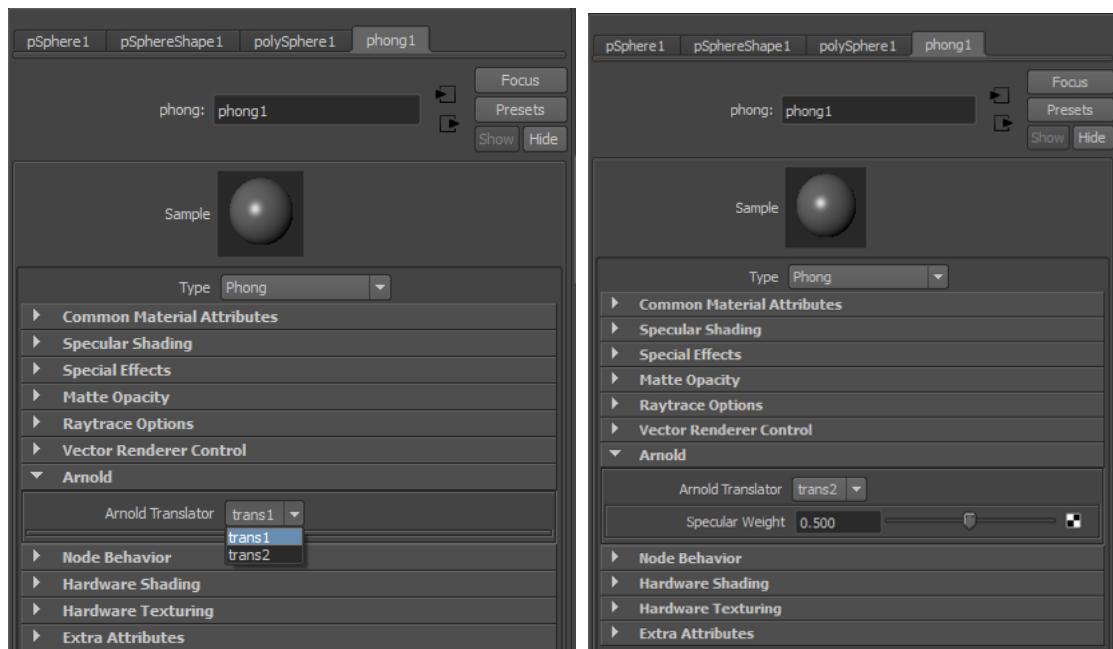
extension2.py

```

1 import mtoa.ui.ae.templates as templates
2 from mtoa.ui.ae.templates import AttributeTemplate
3
4 class PhongTemplate1(AttributeTemplate):
5     def setup(self):
6         pass
7
8 class PhongTemplate2(AttributeTemplate):
9     def setup(self):
10        self.addControl("aiSpecularWeight")
11
12 templates.registerTranslatorUI(PhongTemplate1, "phong", "trans1")
13 templates.registerTranslatorUI(PhongTemplate2, "phong", "trans2")

```

생성되는 Attribute Editor는 다음과 같습니다.



사용자 정의 노드의 변환기

먼저 사용자 정의 노드를 정의하는 매우 기본적인 Maya Plugin을 생성해 보겠습니다.

myPlugin.cpp

```

1 #include <maya/MPxLocatorNode.h>
2 #include <maya/MFnNumericAttribute.h>
3 #include <maya/MFnPlugin.h>
4 #include <maya/MTypeld.h>
5
6 class mySimpleNode : public MPxLocatorNode
7 {
8 public:
9     mySimpleNode();
10    virtual ~mySimpleNode();
11
12    static void* creator();
13    static MStatus initialize();
14    static MTypeld id;
15
16    static MObject s_value;
17 };
18
19 MTypeld mySimpleNode::id(0x00070005);
20 MObject mySimpleNode::s_value;
21
22 void* mySimpleNode::creator()
23 {
24     return new mySimpleNode();
25 }
26
27 MStatus mySimpleNode::initialize()
28 {
29     MFnNumericAttribute nAttr;
30
31     s_value = nAttr.create("my_attribute", "ma", MFnNumericData::kFloat, 0.5f);
32     nAttr.setSoftMin(0);
33     nAttr.setSoftMax(1);
34     nAttr.setMin(0);
35     nAttr.setMax(10);
36     addAttribute(s_value);

```

```

37
38     return MS::kSuccess;
39 }
40
41 MStatus initializePlugin(MObject obj)
42 {
43     MStatus status;
44     MFnPlugin plugin(obj, "My Company", "1.0", "Any");
45
46     status = plugin.registerNode("mySimpleNode",
47                                 mySimpleNode::id,
48                                 mySimpleNode::creator,
49                                 mySimpleNode::initialize,
50                                 MPxNode::kLocatorNode);
51
52     if (!status) {
53         status.perror("registerNode");
54         return status;
55     }
56
57     return status;
58 }
59
60 MStatus uninitializedPlugin(MObject obj)
61 {
62     MStatus status;
63     MFnPlugin plugin(obj);
64
65     status = plugin.deregisterNode(mySimpleNode::id);
66     if (!status) {
67         status.perror("deregisterNode");
68         return status;
69     }
70
71     return status;
72 }

```

그리고 컴파일합니다. Windows로 가정하고 Visual Studio 명령 프롬프트를 열고 다음 명령을 실행합니다.

```
cl /c myPlugin.cpp /EHsc /MD /DNT_PLUGIN /DREQUIRE_IOSTREAM /I%MAYA_PATH%\\include
```

```
link /dll myPlugin.obj /LIBPATH:%MAYA_PATH%\\lib OpenGl32.lib glu32.lib Foundation.lib OpenMaya.lib
```

```
OpenMayaRender.lib  OpenMayaUI.lib  OpenMayaAnim.lib  OpenMayaFX.lib  /export:initializePlugin
/export:uninitializePlugin /OUT:myPlugin.dll
```

Maya에서 생성된 `myPlugin.dll`을 로드하면 `mySimpleNode` 노드 인스턴스를 생성할 준비가 된 것입니다. MEL의 Script Editor에서 다음을 입력합니다.

```
createNode mySimpleNode
```

이 노드에 대한 변환기는 매우 쉽지만, 이번에는 `CNodeTranslator` 대신 `CDagTranslator`에서 변환기를 상속 받아보겠습니다. 변환 행렬을 내보내는 것처럼 DAG 노드에 고유한 메서드를 사용할 수 있습니다.

test.h

```
1 #include "translators/NodeTranslator.h"
2
3 class CTestTranslatorCmd
4     : public CDagTranslator
5 {
6 public:
7     AtNode* CreateArnoldNodes();
8     void Export(AtNode* );
9     void Update(AtNode* );
10    static void* creator();
11};
```

Arnold 구체를 내보내고, 노드에서 행렬을 받고, 노드 속성을 구체 반경에 내보내기 위한 메서드를 정의하겠습니다.

test.cpp

```
1 #include "test.h"
2
3 #include <ai_msg.h>
4 #include <ai_nodes.h>
5
6 void CTestTranslatorCmd::Export(AtNode* node)
7 {
8     AiMsgInfo("[test extension] Exporting %s", GetMayaNodeName().asChar());
9     Update(node);
10}
11
12 void CTestTranslatorCmd::Update(AtNode* node)
13 {
14     // Export the transform matrix
```

```

15     ExportMatrix(node, 0);
16
17     AiNodeSetFlt(node, "radius", FindMayaPlug("my_attribute").asFloat());
18 }
19
20 AtNode* CTestTranslatorCmd::CreateArnoldNodes()
21 {
22     return AddArnoldNode("sphere");
23 }
24
25 void* CTestTranslatorCmd::creator()
26 {
27     return new CTestTranslatorCmd();
28 }

```

그리고 나서, 익스텐션을 로드합니다.

plugin1.cpp

```

1 #include "test.h"
2 #include "myPlugin.cpp"
3 #include "extension/Extension.h"
4
5 extern "C"
6 {
7
8 DLLEXPORT void initializeExtension(CExtension &plugin)
9 {
10     plugin.RegisterTranslator("mySimpleNode",
11                             "",
12                             CTestTranslatorCmd::creator);
13 }
14
15 DLLEXPORT void deinitializeExtension(CExtension &plugin)
16 {
17 }
18 }

```

마지막으로 할 일은 익스텐션에 이 사용자 정의 Maya 노드를 추가하여 플러그인을 로드할 필요가 없도록 하는 것입니다.

먼저, initializePlugin과 uninitializePlugin 메서드를 제거하여 사용자 정의 Maya 노드 플러그인을 간소화합니다.

myNode.cpp

```
1 #include <maya/MPxLocatorNode.h>
2 #include <maya/MFnNumericAttribute.h>
3 #include <maya/MTypeld.h>
4
5
6 class mySimpleNode : public MPxLocatorNode
7 {
8 public:
9     mySimpleNode();
10    virtual ~mySimpleNode();
11
12    static void* creator();
13    static MStatus initialize();
14    static MTypeld id;
15
16    static MObject s_value;
17 };
18
19 MTypeld mySimpleNode::id(0x00070006);
20 MObject mySimpleNode::s_value;
21
22 void* mySimpleNode::creator()
23 {
24     return new mySimpleNode();
25 }
26
27 MStatus mySimpleNode::initialize()
28 {
29     MFnNumericAttribute nAttr;
30
31
32     s_value = nAttr.create("my_attribute", "ma", MFnNumericData::kFloat, 0.5f);
33     nAttr.setSoftMin(0);
34     nAttr.setSoftMax(1);
35     nAttr.setMin(0);
36     nAttr.setMax(10);
37     addAttribute(s_value);
38
39     return MS::kSuccess;
40 }
```

변환기는 수정하지 않아도 되지만 initializeExtension 메서드에서 다음을 변경해야 합니다.

plugin2.cpp

```
1 #include "test.h"
2 #include "myNode.cpp"
3 #include "extension/Extension.h"
4
5 extern "C"
6 {
7
8 DLLEXPORT void initializeExtension(CExtension &plugin)
9 {
10     plugin.RegisterNode("mySimpleNode",
11                         mySimpleNode::id,
12                         mySimpleNode::creator,
13                         mySimpleNode::initialize,
14                         MPxNode::kLocatorNode);
15
16     plugin.RegisterTranslator("mySimpleNode",
17                               "",
18                               CTestTranslatorCmd::creator);
19 }
20
21 DLLEXPORT void deinitializeExtension(CExtension &plugin)
22 {
23 }
24 }
```

이제 익스텐션이 MtoA에 로드되면 사용자 정의 노드 및 변환기가 올바르게 등록됩니다.

익스텐션을 MtoA 1.4.0으로 포팅

MtoA 1.4.0(아직 발표되지 않음)에서는 파일의 양이 현저하게 줄었으며 익스텐션을 빌드하는 데 필요한 것만 포함합니다. CMayaScene, CArnoldSession, CRenderSession과 같은 파일은 더 이상 API의 일부가 아닙니다. CSessionOptions가 여전히 존재하며 해당 장면에 대한 전역 정보가 포함되어 있습니다. Node Translator의 공용 API에서 많은 내부 메소드가 제거 또는 수정되었습니다. 이미 존재하는 익스텐션에 문제가 있다면 support@solidangle.com으로 연락주십시오.

다음은 MtoA 1.4.0에 익스텐션을 포팅할 때 가장 중요한 변경사항입니다.

- **void Export(AtNode*)** : 이전에는 2개의 함수 Export()와 Update()가 사용되었지만 대부분의 변환기는 두 함수 사이에 차이를 두지 않았습니다. 이제 Export만 필요합니다. 일부 변경사항은 첫 번째 내보내기에서만 수행되고 다음 IPR 업데이트에서는 수행되지 않으므로 변환기를 처음으로 내보내면 true를 반환하는 **IsExported()**를 호출할 수 있습니다.
- **void ExportMotion(AtNode *)** : 이전에는 ExportMotion() 및 UpdateMotion()이 있었지만 이제는 ExportMotion만 남았습니다. 일부 변경사항은 첫 번째 내보내기에서만 수행되고 다음 IPR 업데이트에서는 수행되지 않는다면 이를 확인하기 위해 IsExported()를 호출할 수 있습니다.

인수 "step"은 많은 여러 기능을 통해 전송되는 전역 옵션이므로 사라졌습니다. 언제든지 **GetMotionStep()**을 호출하여 현재 step을 얻을 수 있습니다. 총 모션 스텝의 수를 **GetNumMotionSteps()**에 의해 얻을 수 있습니다.

즉,

ExportMotion(node, step);

은 정확히 다음으로 교체될 수 있습니다.

ExportMotion(node);

int step = GetMotionStep();

인수 step은 또한 ExportMatrix(AtNode *node, int step)에서도 사라졌습니다. 이제 **ExportMatrix(AtNode *node)**가 되었습니다.

- **Motion Blur의 중요한 변화:** MtoA 1.4.0은 항상 현재 프레임을 먼저 내보낼 것이므로 모션 스텝 내보내기 순서가 변경되었습니다. 따라서 애니메이션 데이터의 Export() 도중에 배열이 할당되면 인덱스 0이 아닌 GetMotionStep으로 채워져야 합니다.

예:

Export(node)

*AtArray *array = AiArrayAllocate("my_animated_data", GetNumMotionSteps());*

AiArraySetFlt(array, 0, 1.f);

는 다음으로 대체되어야 합니다.

Export(node)

*AtArray *array = AiArrayAllocate("my_animated_data", GetNumMotionSteps());*

```
AiArraySetFlt(array, GetMotionStep(), 1.f);
```

현재 프레임이나 모션 스텝을 내보내고 있는지 파악하기 위한 `step > 0`에 대한 테스트는 더 이상 유효하지 않으며 `IsExportingMotion()` 함수 호출로 대체해야 합니다.

- 연결된 노드를 내보낼 때 `ExportNode` 함수가 `ExportConnectedNode(const MPlug & outputPlug)`로 대체되었습니다.
- **AddArnoldNode** : 이미 존재했지만 일부 MtoA 익스텐션은 이것을 사용하지 않았습니다. 생성되는 노드를 추적하려면 이들을 제대로 삭제하도록 하십시오. `AiNode()`를 수동으로 생성하는 대신 항상 이 함수를 사용해야 합니다.
- **RequiresShaderExport()** . 쉐이더를 내보내기 전에 여러 형상 변환기가 다음 테스트를 수행했습니다.

```
if (( CMayaScene::GetRenderSession()->RenderOptions()->outputAssMask() & AI_NODE_SHADER) ||
CMayaScene::GetRenderSession()->RenderOptions()->forceTranslateShadingEngines() )
```

이제 `CMayaScene`과 `CRenderSession`이 더 이상 공용 API에 없기 때문에 다음으로 대체할 수 있습니다.

```
If (RequiresShaderExport())
```

- `ManageUpdateCallback`의 이름이 `RegisterUpdateCallback`으로 변경되었습니다.

다음과 같이 몇몇 새로운 가상 메소드가 추가되었습니다.

- **virtual void Init()**

노드 초기화 시 호출됩니다(부모 함수를 호출하는 것을 잊지 마십시오).

- IPR 업데이트 중에 실행되는 작업을 사용자 정의하려면:

virtual NodeChanged(MObject &object, MPlug &plug)

지정된 플러그에 대해 maya에서 속성 "먼지" 신호를 전송할 때마다 호출됩니다.

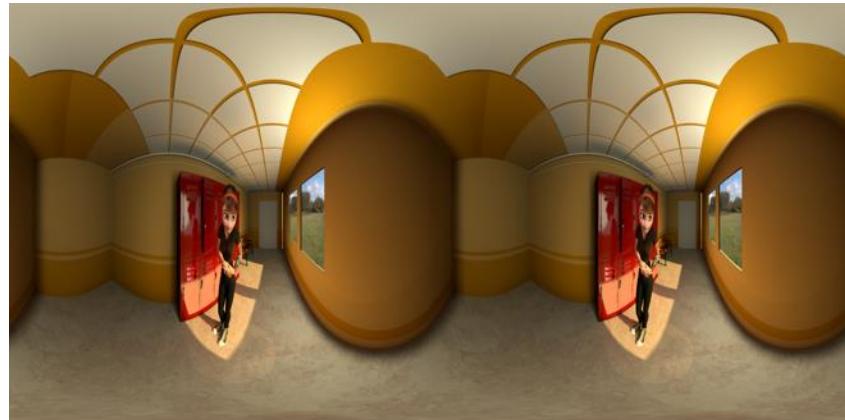
이 속성을 사용하여 변경 중인 속성을 기반으로 다르게 동작할 수 있습니다.

부모 함수를 호출하지 않으면 업데이트가 실행되지 않습니다. 또한 `SetUpdateMode()`를 호출하여 여러 업데이트 모드를 설정할 수도 있습니다.

Oculus Rift Camera Node 만들기

이 자습서에서는 나중에 Oculus Rift에서 시각화할 수 있는 입체 파노라마 이미지를 만드는 데 사용할 수 있는 MtoA용 맞춤형 카메라를 제작해 봅니다.

Oculus Rift는 가상 현실 내부에 한 걸음 진입할 수 있게 해주는 가상 현실 헤드셋입니다.



위의 고해상도 이미지를 Oculus Rift에서 볼 수 있습니다.

- [기본 구체 카메라 만들기](#)
- [MtoA에 카메라 통합하기](#)
- [입체 파노라마 카메라 만들기](#)
- [Oculus Rift에서 결과 시각화하기](#)

기본 구체 카메라 만들기

먼저 기본 구체 카메라를 만들기 위해 다음 3가지 단계를 따라야 합니다.

- 화면 좌표로부터 구체의 좌표 구하기
- 카메라 방향으로 변환하기
- x와 y 미분 계산

OculusCamera.cpp

```

1 #include <ai.h>
2 #include <ai_cameras.h>
3 #include <ai_nodes.h>
4 #include <cstring>
5 #include <ai_metadata.h>
6 AI_CAMERA_NODE_EXPORT_METHODS(OculusCameraMethods);
7
8 node_parameters
9 {
10 }
11 node_initialize
12 {
13     AiCameraInitialize(node, NULL);
14 }
15 node_update
16 {
17     AiCameraUpdate(node, false);
18 }
19 node_finish
20 {
21     AiCameraDestroy(node);
22 }
23 camera_create_ray
24 {
25     // Calculate spherical angles
26     float theta = AI_PI      * input->sx;
27     float phi   = AI_PIOVER2 * input->sy;
28
29     const float sin_theta = sinf(theta);
30     const float cos_theta = cosf(theta);
31     const float sin_phi = sinf(phi);
32     const float cos_phi = cosf(phi);

```

```

33 // normalized direction
34 output->dir.x = sin_theta * cos_phi;
35 output->dir.y = sin_phi;
36 output->dir.z = -cos_theta * cos_phi;
37 // derivative with respect to x
38 output->dDdx.x = cos_theta;
39 output->dDdx.y = 0.f;
40 output->dDdx.z = sin_theta;
41 output->dDdx *= input->dsx * AI_PI * cos_phi;
42
43 // derivative with respect to y
44 output->dDdy.x = -sin_theta * sin_phi;
45 output->dDdy.y = cos_phi;
46 output->dDdy.z = cos_theta * sin_phi;
47 output->dDdy *= input->dsy * AI_PIOVER2;
48 }
49 node_loader
50 {
51 if (i > 0)
52     return false;
53 node->methods      = OculusCameraMethods;
54 node->output_type  = AI_TYPE_NONE;
55 node->name         = "oculusCamera";
56 node->node_type    = AI_NODE_CAMERA;
57 strcpy(node->version, AI_VERSION);
58 return true;
59 }

```

또한 Node Loader를 생성하고 "node_type"을 "AI_NODE_CAMERA"로 설정할 것입니다.
이 카메라를 컴파일하기 위해 쉐이더인 것처럼 컴파일합니다.

Windows:

환경 변수 ARNOLD_PATH를 Arnold SDK가 있는 폴더로 설정합니다. 다음 폴더에 Arnold를 설치한 경우:

C:\solidangle\releases\Arnold-X.X.X.X-platform

다음 명령을 사용하여 ARNOLD_PATH를 해당 폴더로 설정합니다.

```
set ARNOLD_PATH="C:\solidangle\releases\Arnold-X.X.X.X-platform"
```

그린 다음 Visual Studio 명령 프롬프트를 열고 다음 명령을 실행합니다.

```
cl /LD /I %ARNOLD_PATH%\include /EHsc OculusCamera.cpp /link /LIBPATH:%ARNOLD_PATH%\lib
ai.lib
```

OSX:

다음 폴더에 Arnold를 설치했다고 가정합니다.

```
/Users/userName/solidangle/Arnold-X.X.X.X-darwin
```

다음 명령을 사용하여 터미널을 열고 ARNOLD_PATH를 해당 폴더로 설정합니다.

```
export ARNOLD_PATH=/Users/userName/solidangle/Arnold-X.X.X.X-darwin
```

그런 다음 아래 명령을 실행하여 카메라 쉐이더를 컴파일합니다.

```
gcc -I$ARNOLD_PATH/include -L$ARNOLD_PATH/bin -lai -dynamiclib OculusCamera.cpp -o OculusCamera.dylib
```

이제 Arnold가 카메라를 인식할 수 있는지 확인하려면 다음 명령을 실행합니다.

```
%ARNOLD_PATH%#bin#kick -l <path_to_shader> -info oculusCamera
```

다음 결과가 나타납니다.

node:	oculusCamera	
type:	camera	
output:	(null)	
parameters:	19	
filename:	C:#solidangle#mtoadeploy#2014#shaders/oculusCamera.dll	
version:	4.2.1.1	
Type	Name	Default
-----	-----	-----
POINT[]	position	0, 0, 0
POINT[]	look_at	0, 0, -1
VECTOR[]	up	0, 1, 0
MATRIX[]	matrix	
FLOAT	near_clip	0.0001
FLOAT	far_clip	1e+030
FLOAT	shutter_start	0
FLOAT	shutter_end	0
ENUM	shutter_type	box
POINT2[]	shutter_curve	(empty)
ENUM	rolling_shutter	off
FLOAT	rolling_shutter_duration	0
NODE	filtermap	(null)
ENUM	handedness	right
FLOAT[]	time_samples	(2 elements)
POINT2	screen_window_min	-1, -1
POINT2	screen_window_max	1, 1
FLOAT	exposure	0
STRING	name	

이제 구체 카메라가 완성됩니다.

MtoA에 카메라 통합하기

카메라를 MtoA에 통합하려면 템플릿과 메타데이터 파일을 만들어서 MtoA 설치의 올바른 폴더에 복사해야 합니다.

메타데이터 파일 만들기

메타데이터 정보를 추가하려면 이것을 oculusCamera.mtd 파일에 작성해야 합니다.

oculusCamera.mtd		
25	# oculusCamera	0x00070007
26	[node oculusCamera]	
27	maya.name	STRING "camera"
28	maya.id	INT 0x00070007
29	maya.translator	STRING "oculusCamera"

그런 다음 이 파일과 "oculusCamera.dll" 파일을 %MTOA_PATH%\shaders 폴더에 넣습니다.

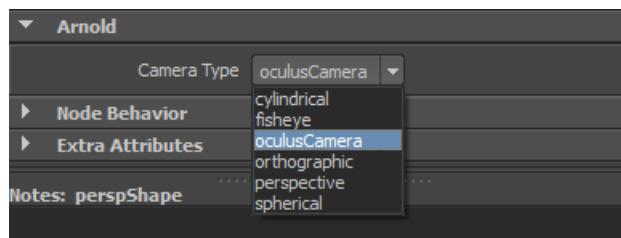
템플릿 파일 만들기

이제 이 카메라에 대해 Maya 템플릿을 만들어야 합니다.

%MTOA_PATH%\scripts\mtoa\ui\ae\ 또는 %MTOA_TEMPLATES_PATH%에서 지정된 폴더에 aiOculusCameraTemplate.py 파일을 만듭니다.

aiOculusCameraTemplate.py	
1	import mtoa.ui.ae.templates as templates
2	class aiOculusCameraTemplate(templates.AttributeTemplate):
3	def setup(self):
4	pass
5	templates.registerTranslatorUI(aiOculusCameraTemplate, "camera", "oculusCamera")

이제 Maya에서 "oculusCamera"를 카메라 변환기 하나로 선택해야 합니다.



입체 파노라마 카메라 만들기

Oculus Camera의 경우에는 다음 두 가지 속성이 필요합니다.

- 모드: 이것은 눈 이미지를 Side by Side, Over/Under, 원쪽 눈만, 아니면 오른쪽 눈만 렌더링할 것인지 여부를 지정합니다.
- Eye Separation: 두 카메라 사이의 거리입니다.

카메라의 최종 코드는 다음과 같습니다.

OculusCamera.cpp

```

1 #include <ai.h>
2 #include <ai_cameras.h>
3 #include <ai_nodes.h>
4 #include <cstring>
5 #include <ai_metadata.h>
6 AI_CAMERA_NODE_EXPORT_METHODS(OculusCameraMethods);
7 namespace
8 {
9 #define _mode (params[0].INT )
10 #define _eyeSeparation (params[1].FLT )
11 enum mode
12 {
13     M_SBS = 0,
14     M_OB,
15     M_LE,
16     M_RE
17 };
18 const char* mode_list[] =
19 {
20     "Side by Side",
21     "Over Under",
22     "Left Eye",
23     "Right Eye",
24     NULL
25 };
26 enum eye
27 {
28     E_RIGHT_EYE = 0,
29     E_LEFT_EYE
30 };

```

```

31 };
32 node_parameters
33 {
34     AiParameterEnum("mode", 0, mode_list);
35     AiParameterFlt("eyeSeparation", 0.65f);
36 }
37 node_initialize
38 {
39     AiCameralInitialize(node, NULL);
40 }
41 node_update
42 {
43     AiCameraUpdate(node, false);
44 }
45 node_finish
46 {
47     AiCameraDestroy(node);
48 }
49 camera_create_ray
50 {
51     const AtParamValue* params = AiNodeGetParams(node);
52
53     int mode = _mode;
54     float eyeSeparation = _eyeSeparation;
55
56     int currentEye = E_RIGHT_EYE;
57     float sx = input->sx;
58     float sy = input->sy;
59
60     if(mode == M_SBS)
61     {
62         if(input->sx < 0)
63         {
64             currentEye = E_LEFT_EYE;
65             sx = 2 * (input->sx + 0.5f);
66         }
67         else
68         {
69             currentEye = E_RIGHT_EYE;
70             sx = 2 * (input->sx - 0.5f);
71         }
72     }
73 }

```

```

72 }
73 else if (mode == M_OB)
74 {
75     if(input->sy < 0)
76     {
77         currentEye = E_LEFT_EYE;
78         sy = 2 * (input->sy + 0.5f);
79     }
80 else
81 {
82     currentEye = E_RIGHT_EYE;
83     sy = 2 * (input->sy - 0.5f);
84 }
85 }
86 else if (mode == M_LE)
87 {
88     currentEye = E_LEFT_EYE;
89 }
90 else if (mode == M_RE)
91 {
92     currentEye = E_RIGHT_EYE;
93 }

94
95 // Calculate spherical angles
96 float theta = AI_PI * sx;
97 float phi = AI_PIOVER2 * sy;

98
99 const float sin_theta = sinf(theta);
100 const float cos_theta = cosf(theta);
101 const float sin_phi = sinf(phi);
102 const float cos_phi = cosf(phi);
103 // normalized direction
104 output->dir.x = sin_theta * cos_phi;
105 output->dir.y = sin_phi;
106 output->dir.z = -cos_theta * cos_phi;
107 // derivative with respect to x
108 output->dDdx.x = cos_theta;
109 output->dDdx.y = 0.f;
110 output->dDdx.z = sin_theta;
111 output->dDdx *= input->dsx * AI_PI * cos_phi;
112

```

```

113 // derivative with respect to y
114     output->dDdy.x = -sin_theta * sin_phi;
115     output->dDdy.y = cos_phi;
116     output->dDdy.z = cos_theta * sin_phi;
117     output->dDdy *= input->dsy * AI_PIOVER2;
118
119     if(currentEye == E_LEFT_EYE)
120     {
121         output->origin.x = -0.5*eyeSeparation*cos_theta*cos_phi;
122         output->origin.z = -0.5*eyeSeparation*sin_theta*cos_phi;
123     }
124     else
125     {
126         output->origin.x = 0.5*eyeSeparation*cos_theta*cos_phi;
127         output->origin.z = 0.5*eyeSeparation*sin_theta*cos_phi;
128     }
129 }
130 node_loader
131 {
132     if (i > 0)
133         return false;
134     node->methods      = OculusCameraMethods;
135     node->output_type  = AI_TYPE_NONE;
136     node->name         = "oculusCamera";
137     node->node_type    = AI_NODE_CAMERA;
138     strcpy(node->version, AI_VERSION);
139     return true;
140 }

```

자세한 생성 방법은 여기를 참조하십시오.

<http://paulbourke.net/papers/vsmm2006/vsmm2006.pdf>

또한 메타데이터 파일과 Maya 템플릿도 업데이트해야 합니다.

oculusCamera.mtd 파일을 업데이트하여 두 가지 새로운 파라미터에 대한 정보를 제공하겠습니다.

oculusCamera.mtd

```

25 # oculusCamera          0x00070007
26 [node oculusCamera]
27     maya.name           STRING  "camera"
28     maya.id              INT    0x00070007
29     maya.translator       STRING  "oculusCamera"

```

```

30
31 [attr mode]
32     maya.name      STRING  "aiMode"
33     default      INT    0
34
35 [attr eyeSeparation]
36     maya.name      STRING  "aiEyeSeparation"
37     softmin        FLOAT   0
38     softmax        FLOAT   1
39     default      FLOAT   0.65

```

Maya 템플릿에는 다음 새로운 파라미터가 포함됩니다.

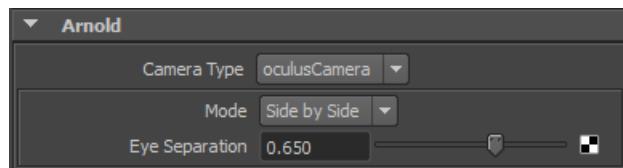
aiOculusCameraTemplate.py

```

1 import mtoa.ui.ae.templates as templates
2 class aiOculusCameraTemplate(templates.AttributeTemplate):
3     def setup(self):
4         self.addControl("aiMode")
5         self.addControl("aiEyeSeparation")
6     templates.registerTranslatorUI(aiOculusCameraTemplate, "camera", "oculusCamera")

```

이제, Maya에서 새로운 파라미터가 있는 "oculusCamera"를 선택할 수 있습니다.



Oculus Rift에서 결과 시각화하기

OculusCamera를 사용하여 비디오를 만든 후에는 비디오가 다음과 같아야 합니다.



렌더링 설정

구체 투영에서 잘 작동하는 이미지를 얻으려면 다음 설정을 지정해야 합니다.

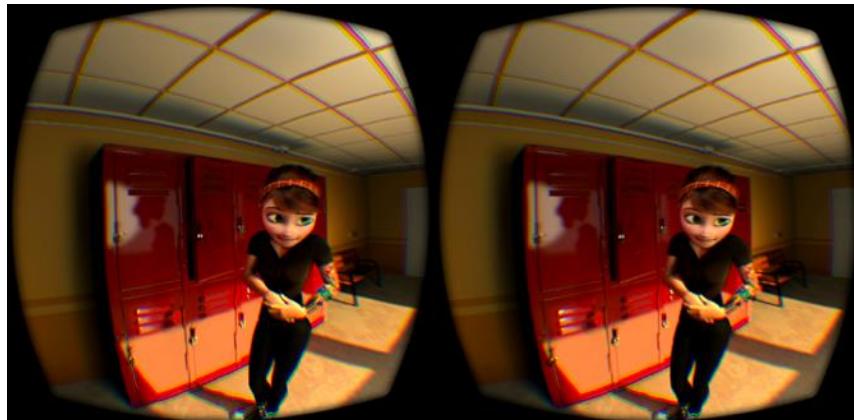
- 높이의 두 배인 너비
- 장치 종횡비: 1, 픽셀 종횡비: 0.5

그러나 몰입할 수 있는 경험을 느끼고 싶다면 Oculus Rift 헤드셋을 사용해야 합니다.

이를 수행할 수 있는 몇 가지 프로그램이 있습니다. 이 경우 MaxVR이 사용되었지만 입체 및 구체 카메라의 재생산을 지원하는 모든 Oculus Rift 비디오 플레이어가 잘 작동할 것입니다.

선택한 프로그램에서 비디오를 열고 구체 투영법 및 올바른 입체 설정을 선택하면 렌더링한 장면 내에서 어떤 방향으로도 볼 수 있습니다.

실제로 보이는 모습은 다음과 같습니다.



Volume 튜토리얼



이 자습서에서는 MtoA로 체적을 렌더링하는 방법을 설명합니다.

- [Polymesh를 Volume으로 변환](#)
- [Volume Shader를 사용하여 구름 렌더링하기](#)
- [Volume – Displacement](#)
- [Volume Sample RGB 예](#)

Polymesh를 Volume으로 변환



Polymesh 오브젝트들을 Volume으로 렌더링을 할 수 있습니다. 이것은 슬리드 오브젝트를 다양하고 흥미로운 방식으로 렌더링하는 많은 창조적인 기회를 제공합니다. 이 짧은 튜토리얼에서는 Standard Volume 쉐이더의 displacement 속성을 사용하여 쉐이딩의 결합을 통해 볼륨과 Polymesh를 결합하는 것을 보여줍니다.

Polymesh는 반드시 닫혀있어야 하며, 서로 겹쳐있지 말아야 합니다. 이런 경우 렌더링이 잘 되지 않을 수 있습니다.

Step Size

- 우선, 머리 메쉬를 볼륨으로 변환해야합니다. 메쉬를 선택하고 Arnold 속성에서 **Step Size**를 **0.1**로 증가시킵니다. Step Size가 양수이면 Polymesh를 볼륨으로 바꾸고 해당 값을 볼륨 raymarching의 기본 단계로 사용합니다.

Standard Volume

- Standard Volume** 쉐이더를 폴리메쉬에 적용하고 **Density**를 **20**로 설정합니다. 이렇게하면 displacement 텍스처링이 나중에 더 명확하게 정의됩니다.



Step Size 값을 반드시 입력해야 하며, 메쉬를 볼륨으로 렌더하기 위해 Standard Volume 쉐이더를 적용해야 합니다.

Volume Padding

다음 단계는 Padding을 사용하여 볼륨을 확대하는 것입니다. 이것은 텍스처로 볼륨을 이동하려는 경우에 유용합니다. 바운딩 박스뿐만 아니라 바운딩 박스 내부의 광선 범위에 사용되는 복셀에도 적용됩니다. 이 경우 아주 작은 값인 0.05를 사용합니다.

최상의 성능을 위해서는 가능한 낮게 설정해야 하며 displacement 량의 최대 값은 반드시 포함해야 합니다.

Volume

다음은 Arnold Volume을 머리 뒤쪽에 놓고 동일한 Standard Volume 쉐이더를 적용합니다. 또는 다른 매쉬를 사용하여 머리와 동일한 쉐이더를 적용할 수 있습니다. 목표는 displacement 텍스처링 과정에서 복잡해 보이는 것을 갖는 것입니다.



동일한 Standard Volume 쉐이더로 볼륨 렌더

Displacement Texture (Standard Volume Shader)

Standard Volume 쉐이더의 displacement 속성에 파일 텍스처를 연결하십시오. 이 경우 카메라 프로젝션이 사용됩니다. 텍스처의 효과(대부분 머리 부분)를 볼 수 있지만 효과를 더 두드러지게 만들 필요가 있습니다.



Displacement 텍스처가 보인다.

Range Shader

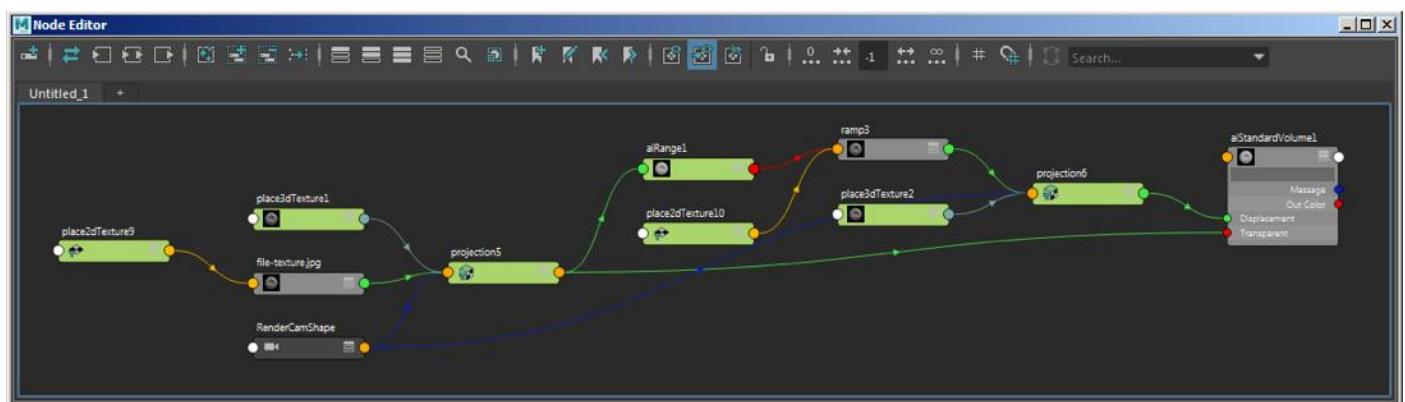
- Range 쉐이더를 사용하여 파일 텍스처의 입력값을 다시 매핑하여 효과를 과장할 수 있습니다. 파일 텍스처를 Range 쉐이더 입력에 연결하고 Range 쉐이더를 Standard Volume 쉐이더의 displacement에 연결하세요. Range 쉐이더의 **Output Max** 값을 **8**로 높이세요. 이제 파일 텍스처가 볼륨 이동에 미치는 영향을

실제로 확인해야 합니다.

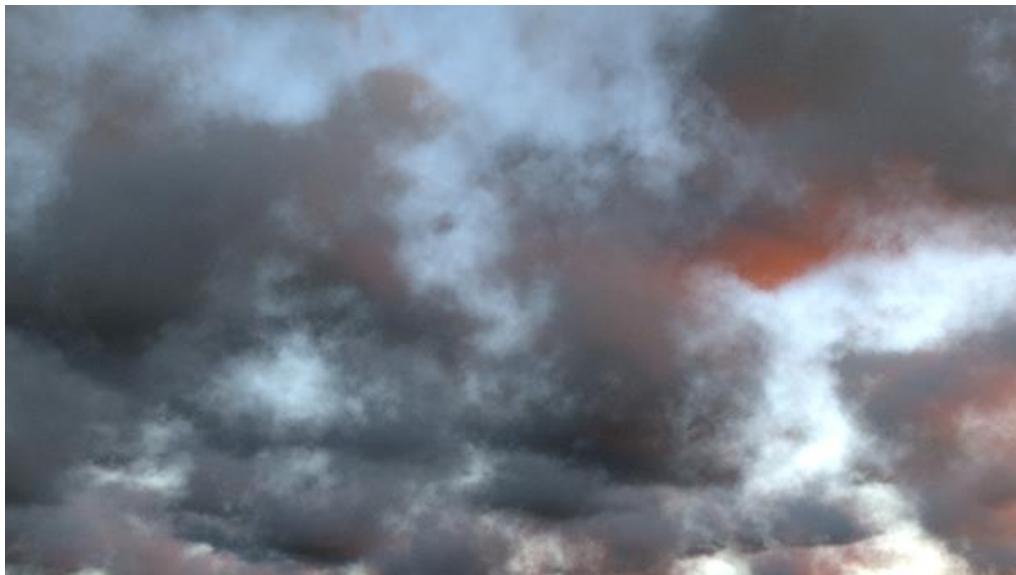


Displacement 쉐이더로 인해 Range 쉐이더가 더 분명하게 보임

마지막으로, 일부 색상을 추가하기 위해 동일한 파일 텍스처를 Standard Volume 쉐이더의 Scatter Color 또는 Transparent Weight 속성에 연결해 보십시오.



Volume Shader를 사용하여 구름 렌더링하기

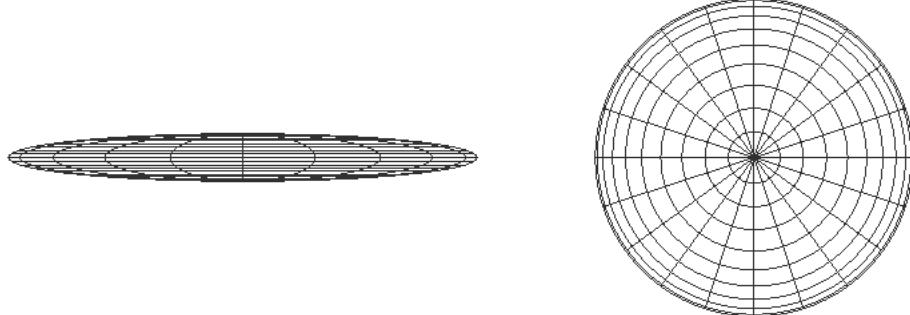


이 간단한 자습서에서는 **Volume Shader**와 Maya **Cloud 3d texture**의 조합을 사용하여 구름 효과를 만드는 방법을 설명합니다. 이 설정을 사용하여 여러 가지 구름이 있는 하늘을 표현할 수 있습니다. 또한 스포트라이트를 추가하여 햇빛을 재현하고 Atmosphere Volume을 추가하여 '신이 내린 빛'과 같은 효과를 연출할 수 있습니다.

Maya 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Rendering+Clouds+using+the+Volume+Shader>

- 구름 층을 표현하려면 먼저 Volume 쉐이더를 할당할 수 있는 객체를 만들어야 합니다. 구체나 큐브를 만들고 Y 축에서 축소합니다.



- 구체의 **Step Size**를 **0.1** 정도로 설정합니다. Step Size가 양수 값일 때 폴리메쉬는 볼륨으로 변환됩니다.

Step Size는 고정된 크기이므로 컨테이너가 클수록 더 많은 단계가 수행되고 렌더링이 더 오래 걸립니다. 또한 스텝 크기가 너무 작으면 렌더링 시간이 증가하므로 유의해야 합니다.

- Standard Surface** 쉐이더를 구체에 적용합니다.

- Maya **Cloud 3d texture**를 만들고 **Standard Volume** 쉐이더의 **Transparent Weight**에 연결합니다. **Cloud texture** 대한 Maya 3d 텍스처 노드의 크기를 줄이면 더 자세히 볼 수 있습니다.
- 마지막으로 **Skydome** 조명을 만들고 **Physical Sky** 쉐이더를 해당 Color 속성에 연결합니다. 체적에서 노이즈를 줄이려면 Skydome 조명에서 Volume Samples 수를 늘려야 합니다.

Volume - Displacement

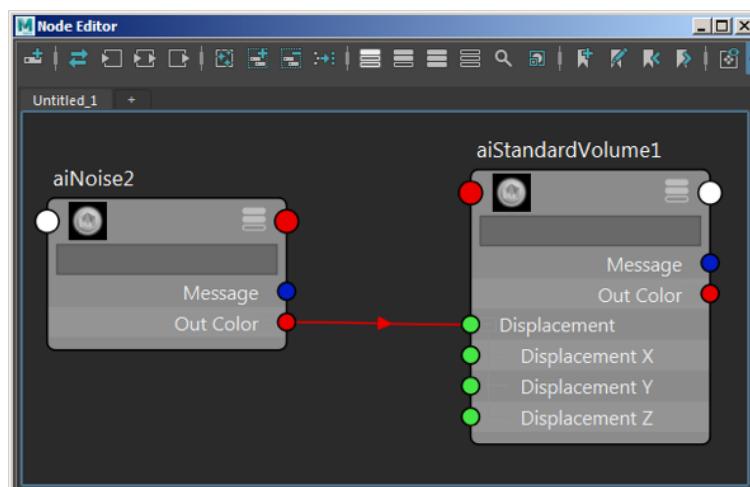


이 자습서에서는 Standard Volume 쉐이더의 Displacement 속성에 연결된 노이즈 쉐이더를 사용하여 체적을 샘플링하는 위치를 오프셋하는 방법을 보여줍니다. 이 기술을 사용하여 몇 가지 재미있는 효과를 만들 수 있습니다. 예를 들어, 흐르는 듯한 가스 연기를 표현하기 위해 노이즈 쉐이더에 애니메이션 효과를 적용할 수 있습니다. 이 예제에서는 노이즈 쉐이더를 사용하지만 텍스처 맵을 사용하여 체적을 오프셋할 수 있습니다.

Maya 파일은 하단 링크의 사이트에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Volume---Displacement>

- 먼저 open vdb 다운로드 페이지에서 토끼 볼륨 파일을 다운로드합니다.
- Arnold Volume을 만들고 체적 vdb 파일 - **bunny_cloud.vdb**를 엽니다.
- Standard Volume** 쉐이더와 **aiNoise** 쉐이더를 만듭니다. aiNoise 쉐이더를 **Standard Volume** 쉐이더의 **Displacement** 속성에 연결합니다. 토끼의 체적을 보려면 **Density**를 높여야 할 수도 있습니다.



Standard Volume의 'Displacement'에 연결된 노이즈 쉐이더

노이즈 유형 예

아래 이미지는 Standard Volume의 Displacement 속성에 연결된 다양한 노이즈 유형을 사용했을 때의 효과를 보여줍니다.



Arnold 노이즈



alCellNoise (alShaders)



Perlin 노이즈

지금까지 **Standard Volume** 쉐이더의 **Displacement** 속성에 대해 알아봤습니다.

Volume Sample RGB 예



이 자습서에서는 VolumeSampleRGB 쉐이더를 사용하여 체적의 외관을 두껍고 '기름지고' 스모키한 체적으로 완전히 변신시키는 방법을 설명하겠습니다. 이 쉐이더에는 'Volume Sample Float' 쉐이더와 함께 'contrast' 및 'gamma'와 같은 'post-production' 스타일의 속성이 있습니다. 이러한 속성들은 체적의 외관을 '미세하게 조정'하는 데 매우 유용합니다. 이 예제에서는 VolumeSampleRGB 쉐이더를 사용하여 일반적인 화재 체적을 보다 두꺼운 연기로 변환해 보았습니다. 여기서는 디스크에 체적 VDB로 작성된 Houdini의 Pyro FX 헬프에 있는 기본 화재 시뮬레이션을 사용합니다.

이 자습서에 있는 VDB를 사용할 것이므로 먼저 **Writing Volumes To VDB**

(<https://support.solidangle.com/display/AFHUG/Writing+Volumes+To+VDB>) 자습서를 따를 것을 권장합니다. 원본 Houdini 파일과 VDB 파일은 하단 링크 사이트에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Volume+Sample+RGB+Example>

Volume

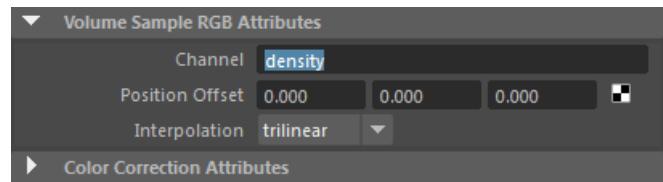
- 먼저 Volume - **Arnold > Volume**을 만들겠습니다.
- Volume의 **Filename**에서 **default_flame.114.vdb** 파일을 엽니다.
- Grids 필드 유형 **'density'**가 있어야 합니다. 이것은 Volume VDB가 함께 저장된 동적 필드 그룹의 이름입니다.

Standard Volume

- Standard Volume** 쉐이더를 생성하고 이것을 체적에 할당합니다. **Scattering Color**를 **blue**로 변경하고 **Transparency Color**를 **orange**로 변경합니다. 장면의 조명에 따라 **Scattering Intensity**를 조절하는 것이 좋습니다.
- VolumeSampleRGB** 쉐이더를 만들고 **VolumeSampleRGB** 쉐이더의 **output**(MtoA에서 **R**) 속성을 Standard Volume의 **Density** 속성에 연결합니다.

Volume Sample RGB

- 지금까지 기술적인 세부 사항을 배웠으므로 Volume Sample RGB 쉐이더의 다양한 속성을 사용하여 재미있는 시간을 보낼 수 있습니다. 그러나 계속 진행하기 전에 Volume Sample RGB 쉐이더의 채널 속성에 'density'라는 단어를 입력해야 합니다. 그렇지 않으면 체적의 밀도 채널을 볼 수 없습니다.



채널 속성에 'density' 입력

다음은 Volume Sample RGB 쉐이더의 속성을 변경하는 몇 가지 추가 예입니다.

Contrast(대비)



1



2



10

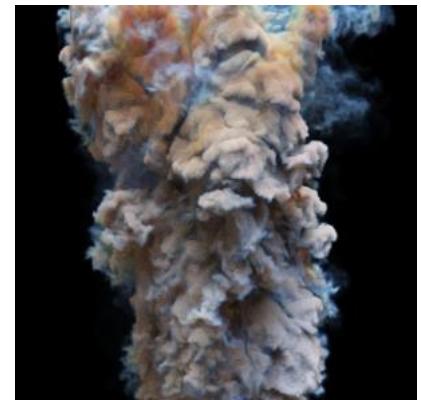
Exposure(노출)



0



1



8

Multiply(증식)



1



10



25

Add(추가)



-5

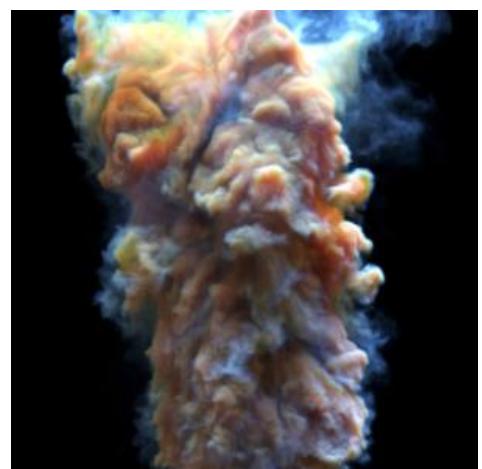
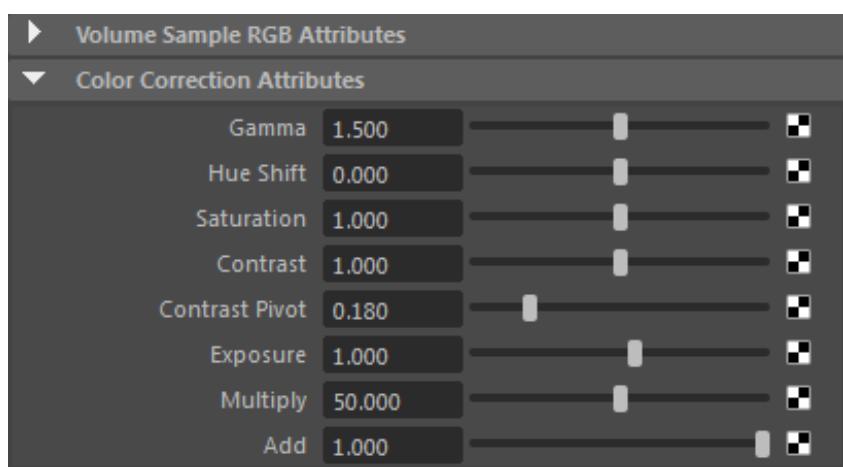


0



5

이 예제에서 사용된 최종 설정에는 아래 이미지와 같이 *Gamma*, *Exposure*, *Multiply*, *Add* 속성의 조정이 포함됩니다.



Volume Sample RGB의 최종 속성

볼륨 광선 깊이

볼륨 광선 깊이를 높여도 체적의 모양이 크게 달라질 수 있습니다. 노이즈를 보상하려면 반드시 Volume Samples의 개수를 높이십시오. 이 예에서는 6개의 Volume Sample이 사용되었습니다.

볼륨 광선 깊이를 늘리면 해당 체적에서 조명의 추가적인 다중 산란을 보상하기 위해 장면 조명의 노출을 줄여야 할 수 있습니다.



0



0

XGen 튜토리얼



XGen은 많은 프리미티브를 빠르고 쉽게 만들어주는 강력한 인스턴스화 도구입니다. 예를 들어, 잔디밭, 숲, 경관과 같은 큰 규모의 환경을 쉽고 빠르게 채울 수 있습니다. 스플라인, 구체 또는 폴리곤 오브젝트를 포함한 간단한 기본 지오메트리 유형을 인스턴스화할 수 있습니다.

브러쉬 기반의 손질 도구를 사용하여 머리카락과 털을 스타일링 할 수 있습니다. 또한 PTEX 맵, 수식 및 속성 슬라이더를 사용하여 프리미티브의 배치, 모양 및 동작을 제어할 수도 있습니다.

아래 자습서는 XGen을 사용하여 장면을 렌더링하는 방법을 알아봅니다.

- [XGen 및 Arnold와 렌더링 가이드](#)
- [아카이브 \(Archives\)](#)
- [색상 지정하기](#)
- [스플라인 \(Splines\)](#)
- [수식을 사용하여 구체의 크기를 균일하게 조절하기](#)

XGen 및 Arnold와 렌더링 가이드



이 자습서에서는 XGen에서 10만 개 이상의 삼각형이 있는 프리미티브의 인스턴스를 5만개 이상 사용하여 복잡한 장면을 만들어봅니다. 최종 장면은 총 30억 개의 가시적인 삼각형을 포함하며, 1.2GB의 메모리만 사용하여 렌더링합니다.

장면은 하단 링크를 통해 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Guide+to+Rendering+with+XGen+and+Arnold>

1 - 프리미티브 아카이브 (Primitive Archive)



먼저 XGen 아카이브로 사용하고자 하는 지오메트리를 로딩합니다. 나중에 XGen **Size** 속성을 사용하여 치수를 조절할 것입니다. 나중에 수정할 수 있기 때문에 어떠한 초기 쉐이더나 모델에도 적용할 수 있습니다. 아카이브를 내보내기 전에 Arnold 형식으로 내보내는지 확인하십시오(예: MtoA 플러그인을 로드해야 함). 또한 Arnold Render Settings를 열어 장면에 대한 MtoA 설정을 초기화해야 합니다. 계속 진행하기 전에 장면을 저장합니다.

지오메트리를 선택한 채로 **Generate>Export Selection as Archive(s)...**로 이동합니다. **Archive Name**을 **stalagmite**로 변경하고 아카이브를 Local Archives 폴더에 저장하십시오.

2 - XGen Description

폴리곤 평면을 만듭니다. 이 평면을 사용하여 XGen 설명을 할당하겠습니다. 폴리곤 평면을 선택하고 **Generate>Create Description...**으로 이동합니다. 그러면 **Create XGen Description**이 열립니다. **Custom Geometry/Archives**를 선택합니다. 다른 기본 설정은 그대로 두고 **Create**을 클릭합니다.

Archive Files 섹션에서 이전에 생성한 아카이브를 추가합니다. 원시 크기가 장면에 맞도록 **Size** 속성을

수정해야 할 수도 있습니다.



이제는 프리미티브가 무작위로 평면 위에 흩어져 있지만 크기와 방향은 같습니다. 방향에 임의성을 추가하기 위해 **Twist** 속성에서 XGen 샘플 표현식 중 하나를 사용합니다.

Primitive Attributes 아래에서 **Twist** 슬라이더 컨트롤 오른쪽에 있는 아래쪽 화살표를 클릭합니다. 메뉴에서 **Load Expression>samples>Geometry>flatten_Room**을 선택하면, **Twist** 아래에 **Width** 및 **Depth** 속성이 나타납니다. **Twist Width**를 최소 200으로 높입니다.

3 - Density

표현식 편집기에서 사용한 값들에 따라 **Density** 값을 변경해야 할 수도 있습니다. 석순 아카이브가 너무 밀도가 높다면 밀도 값을 줄이는 것이 좋습니다. 반대로 너무 밀도가 낮다면 밀도를 높이거나 최대 **width** 크기를 줄여야 합니다. 아래 이미지는 밀도를 높였을 때 XGen에 의해 생성되는 프리미티브의 수가 어떤 영향을 받는지 보여줍니다.



흔한 문제는 장면에 너무 많은 프리미티브가 있을 수 있으며 Maya 뷰포트의 성능이 저하된다는 것입니다. XGen **Preview/Output** 탭에서 **Preview Settings** 섹션의 **Percent** 속성을 줄일 수 있습니다. 그러면 뷰포트에 표시되는 프리미티브 수를 줄이는 동시에 Arnold를 호출할 때 여전히 모든 뷰를 렌더링 할 수 있습니다.

4 - 텍스처 맵으로 속성 수정

텍스처 맵과 표현식의 조합을 사용하여 프리미티브의 길이(높이)를 제어할 수 있습니다. 먼저 텍스처 맵을 만들어서 길이를 조절하겠습니다. **Length** 오른쪽에 있는 아래 방향 화살표를 클릭합니다. **Create Map...**을 선택합니다. (이것은 평면에 Maya 쉐이더가 할당되어 있는 경우에만 작동한다는 점을 유의하십시오). **Map Resolution**을 200 정도로 높입니다. 그 다음 **Create**를 클릭합니다.

Hypershade 또는 **Node Editor** 창에서 파일 텍스처 노드가 해당 평면에 연결되어 있어야 합니다. 파일 텍스처를 선택하고 프리미티브의 길이를 구동하는 데 사용할 텍스쳐 맵을 엽니다. XGen에서 파일 텍스처를 업데이트하려면 길이 속성 컨트롤의 오른쪽에 있는 디스크 아이콘을 클릭해야 합니다.

이제 높은 값이 0에서 1로 됩니다. 이 범위를 변경하고 기본 높이에 임의성을 추가하려고 합니다. 현재 **Length**

수식을 수정하여 이것을 실행할 수 있습니다.



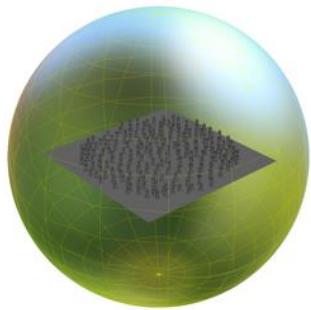
Length 슬라이드 컨트롤 오른쪽에 있는 **Expression** 아이콘을 클릭합니다. **Expression Editor**에 다음 길이 수식 텍스트가 나타납니다.

```
$a=map('${DESC}/paintmaps/length');#3dpaint,200.0.  
$a
```

Expression Editor에서 마지막 행 전에 다음 행을 추가합니다. **\$a = 2*\$a + rand(0,0.75);** 이제 프리미티브의 길이 범위가 0에서 2까지이고, 0에서 0.75 사이의 임의 값이 추가됩니다. 완료하면 **Accept**를 클릭합니다.

5 - Lighting 및 Camera DOF

AiSkyDome 조명을 사용하여 장면을 비춥니다. Arnold 메뉴에서 **Arnold>Lights>SkyDome Light**를 선택합니다. SkyDome 조명의 **Color** 속성에 파일 텍스처를 연결하고 적합한 HDR 맵을 선택하십시오. sky dome 조명의 Arnold 설정에서 **Exposure**를 조절하여 장면이 고르게 비춰지도록 하십시오.



장면에서 그림자 노이즈를 줄이려면 조명 **Sample**을 3으로 늘립니다. 매크로 사진 장면을 시뮬레이션하기 위해 카메라에 피사계 심도를 추가하겠습니다. 카메라를 XGen 설명에 가깝게 배치합니다. 또한 카메라의 초점 거리를 늘려야 합니다. 최종 이미지에서는 65mm가 사용되었습니다. 해당 카메라를 선택하고 Arnold 설정으로 이동합니다. **Enable DOF**를 선택하고 초점 거리를 선택합니다. 카메라에 얇은 피사계 심도를 추가하려면 **Aperture Size**를 늘리십시오. 장면의 크기에 따라 이 값을 조정해야 합니다.

6 - 렌더링



장면의 렌더링을 테스트하려면 IPR를 사용해야 합니다. MtoA에서 IPR 렌더링을 사용하면 빠른 피드백으로 장면의 조명, 카메라 및 쉐이더를 대화식으로 업데이트할 수 있습니다. XGen 설명을 수정한 후 IPR를 새로고침해야 할 수도 있습니다. 결과에 만족하면 최종 렌더링 준비가 완료된 것입니다. **Camera(AA)** 샘플을 7이나 8로 높입니다. 카메라 DOF가 포함된 이미지 영역에 나타나는 노이즈의 양에 따라 이 값을 조절해야 할 수도 있습니다. 간접적으로 밝게 빛난 부분에서 노이즈가 발생하면 **Diffuse(GI)** 샘플을 2 또는 3으로 높여보십시오.

아카이브 (Archives)



- [XGen에서 속성 패턴 애니메이트하기](#)
- [텍스처 맵으로 프리미티브 제어하기](#)
- [장난감 병사 만들기](#)
- [수식으로 도미노 배치하기](#)

XGen을 사용하여 아카이브를 내보내기 전에 MtoA를 로드해야 하고 적어도 한 번 Arnold 렌더링 설정을 열어서 아카이브를 올바르게 내보낼 수 있어야 합니다.

XGen에서 속성 패턴 애니메이트하기



장면은 XGen '아카이브' 큐브 프리미티브로 구성됩니다. 텍스처 맵은 'length' 속성을 유도하고 수식은 'twist' 속성을 유도합니다.

XGen은 무작위로 또는 균일하게 배치된 임의 개수의 프리미티브를 배치할 수 있는 많은 창조적인 기회를 제공하는 강력한 도구입니다. XGen Description과 함께 텍스처 맵을 사용하여 프리미티브 생성의 위치와 밀도를 지정하고 프리미티브 속성의 수를 제어할 수 있습니다.

이 자습서는 텍스처 맵을 사용하여 XGen 프리미티브의 길이를 쉽게 '구동'하는 방법을 보여줍니다. 이러한 프리미티브는 구체, 스플라인 또는 아래 예와 같이 아카이브로 내보낸 큐브 지오메트리까지 범위가 다양합니다. 그 다음 **\$frame** 수식을 사용하여 아카이브의 길이를 애니메이트합니다. 마지막으로 XGen 및 MtoA를 사용하여 애니메이션을 일괄 렌더링하는 방법에 대해 설명하겠습니다.

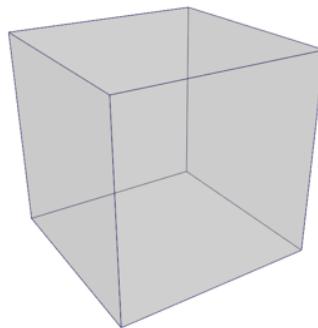


이 프로세스를 안내하는 동영상은 하단 링크에 있습니다.

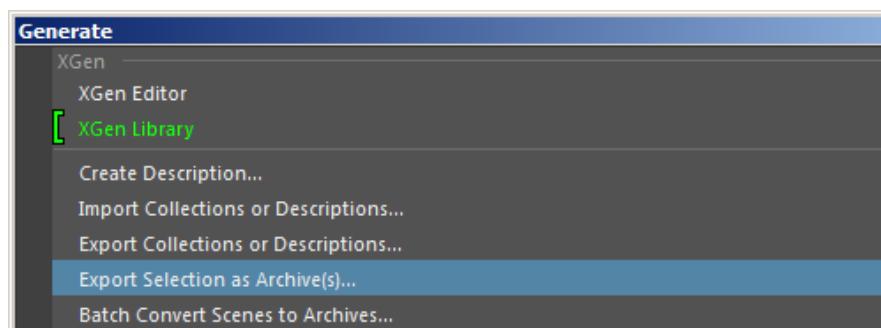
<https://www.youtube.com/watch?v=L7XZxffWork#t=579>

프리미티브 아카이브

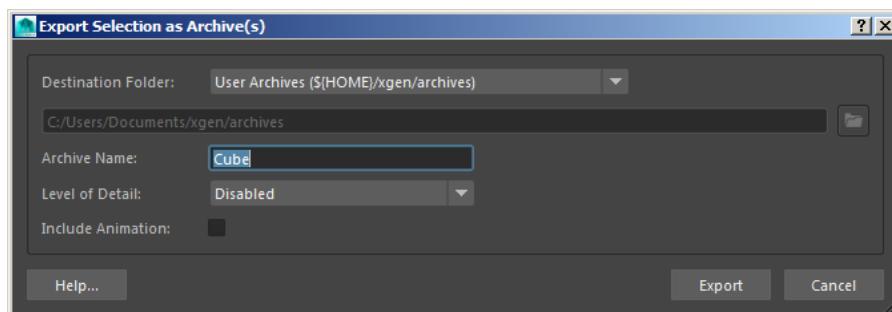
- 먼저 XGen 아카이브로 사용하고자 하는 폴리곤 큐브를 만듭니다. 큐브의 scale은 1,1,1로 두십시오. 나중에 XGen Size 속성으로 조절할 것입니다.



- 큐브를 선택한 채로 **Generate>Export Selection as Archive(s)...**로 이동합니다.

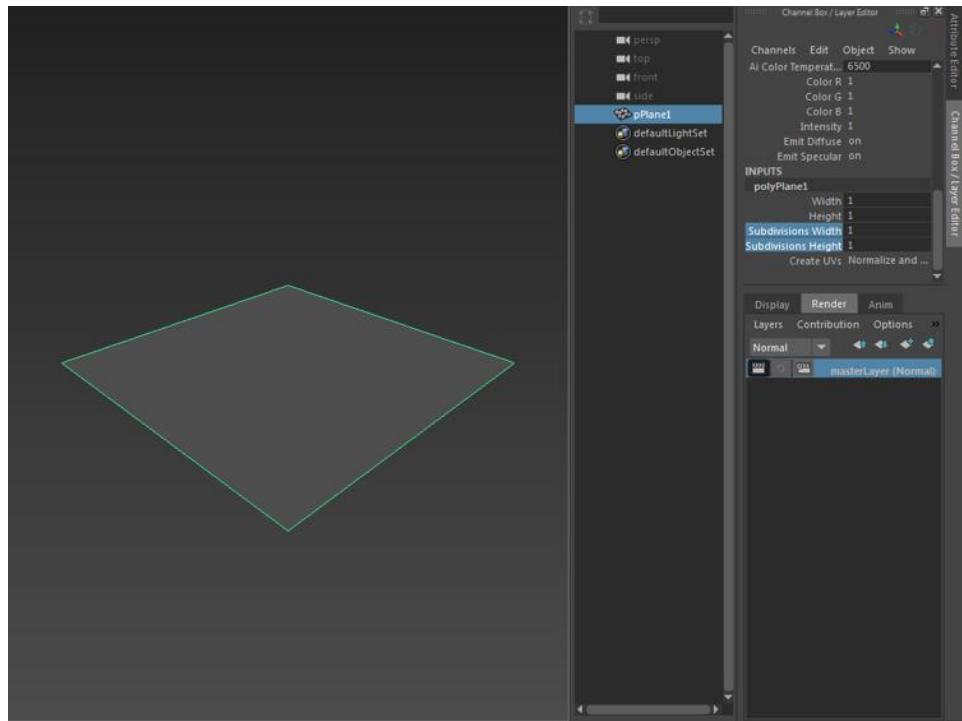


- Archive Name을 **cube**로 변경하고 아카이브 파일을 저장할 위치를 선택하십시오.

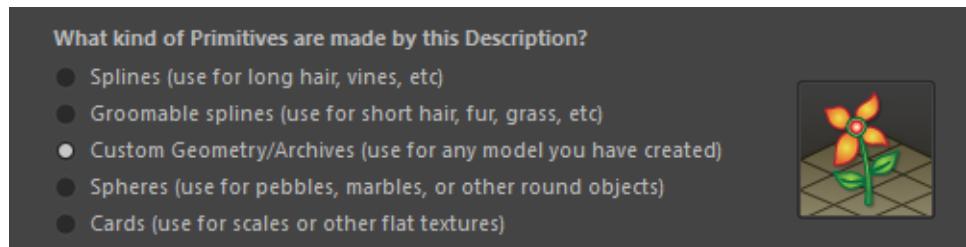


XGen 설명

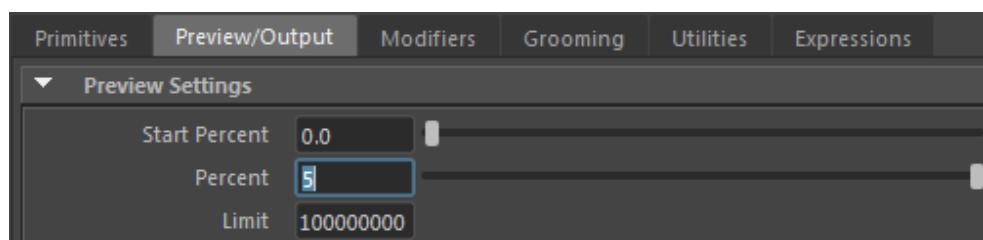
- 폴리곤 평면을 만듭니다. 이 평면을 사용하여 XGen 설명을 할당하겠습니다. XGen 프리미티브의 형체를 원형으로 만들려면 **Subdivisions Width**와 **Subdivisions Height**가 1로 설정되어 있는지 확인합니다. 형체를 정사각형으로 만들려면 **Subdivisions Width**와 **Subdivisions Height**를 10으로 증가시킵니다. 이 예제에서는 1로 설정했습니다.



- 폴리곤 평면을 선택하고 **Generate> Create Description...**으로 이동하면 아래와 같이 **Create XGen Description**이 열립니다.
- **Custom Geometry/Archives**를 선택합니다. 다른 기본 설정은 그대로 두고 **Create**를 클릭합니다.

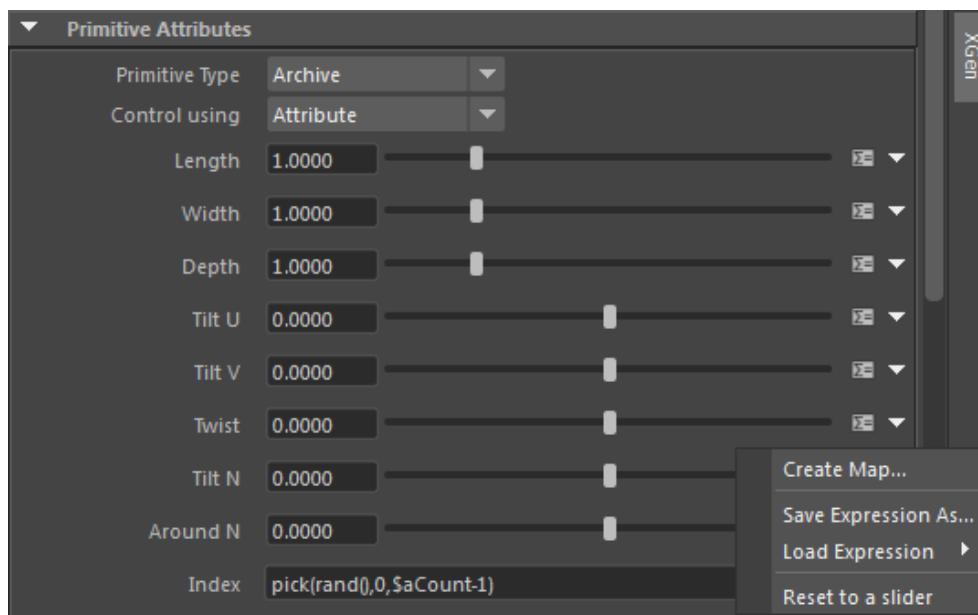


- 계속 진행하기 전에 **Preview/Output** 탭에서 **Percent** 값을 낮추는 것이 좋습니다. **Percent** 값이 너무 높으면 생성된 많은 수의 프리미티브로 인해 컴퓨터가 멈출 수도 있습니다.



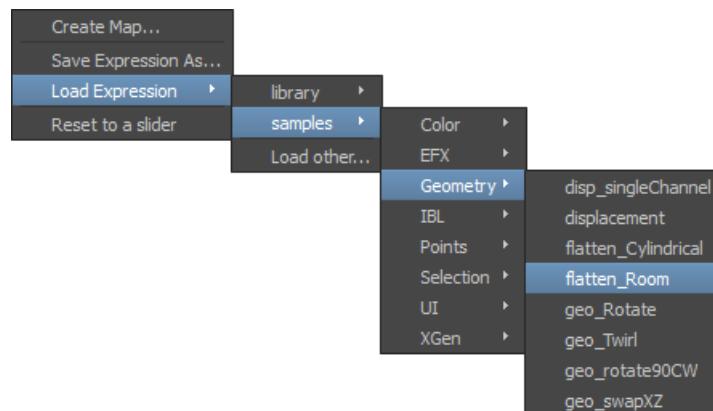
Twist Expression

- **Primitive Attributes** 아래에서 **Twist** 슬라이더 컨트롤 오른쪽에 있는 아래쪽 화살표를 클릭합니다.

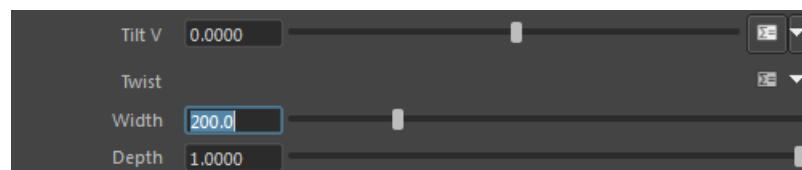


- 메뉴에서 다음을 선택합니다.

Load Expression>samples>Geometry>flatten_Room



- Twist 아래에 **Width** 및 **Depth** 속성이 나타납니다. **Twist Width**를 최소 200으로 높입니다.



Density

Expression Editor에서 사용한 값들에 따라 **Density** 값을 변경해야 할 수도 있습니다. 큐브 아카이브의 밀도가 너무 높다면 **Density** 값을 줄이는 것이 좋습니다. 반대로 큐브의 밀도가 너무 낮다면 밀도를 높이거나 최대 **width** 크기를 줄여야 할 수 있습니다. 아래 이미지는 Density를 높였을 때 XGen에 의해 생성되는 프리미티브의 수가 어떤 영향을 받는지를 보여줍니다.



Density 35000.0

- **Density** 값을 70000.0으로 두 배 높이면 아래와 같아집니다.

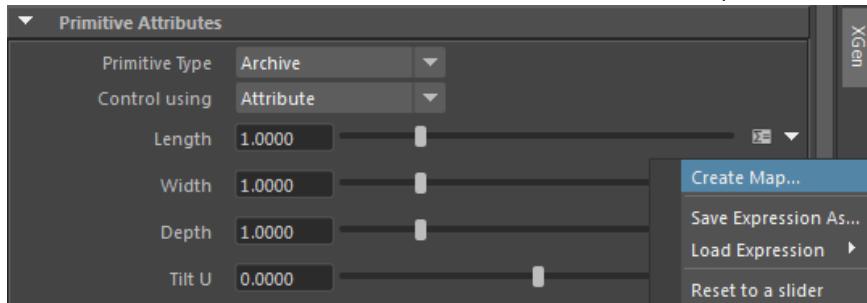


Density 70000.0

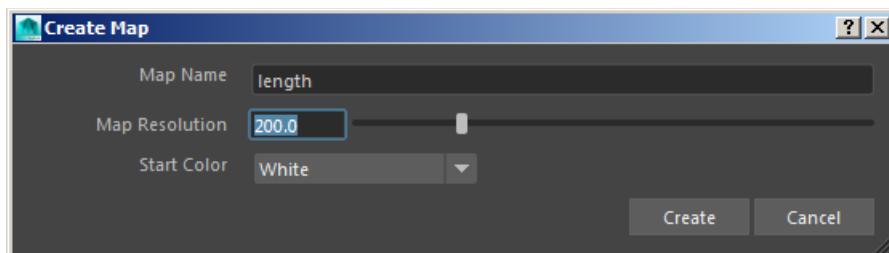
Length

텍스처 맵과 수식의 조합을 사용하여 큐브의 길이(높이)를 제어할 수 있습니다. 먼저 텍스처 맵을 만들어서 길이를 조절하겠습니다.

- Length 오른쪽에 있는 아래 방향 화살표를 클릭합니다. **Create Map...**을 선택합니다(이것은 평면에 Maya 쉐이더가 할당되어 있는 경우에만 작동한다는 점을 유의하십시오).

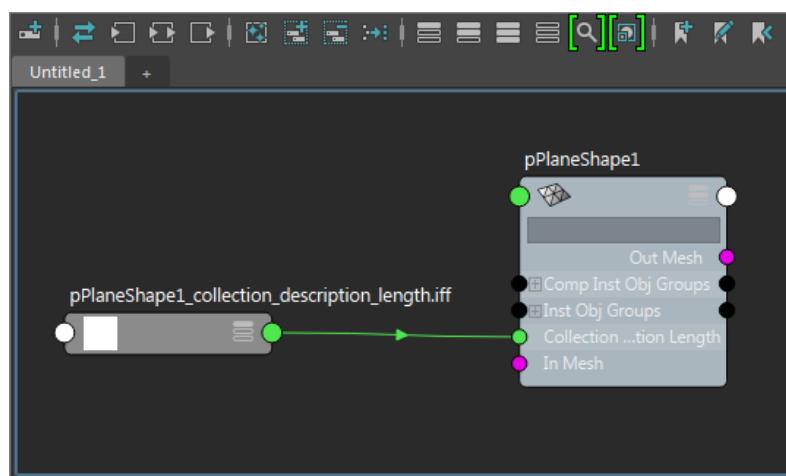


- Map Name은 **length**여야 합니다. Map Resolution을 200 정도로 높입니다. 그러면 PTEX 맵의 해상도가 해당 텍셀로 설정됩니다. 고해상도 텍스처를 사용할 때는 더 큰 Map Resolution 값을 사용하십시오. 이것이 완료되면 **Create**를 클릭합니다.



텍스처 맵 생성

이제 텍스처 맵을 추가하겠습니다. 슬라이드 컨트롤 오른쪽에 있는 페인트 아이콘을 클릭합니다. Hypershade 및 Node Editor 창에서 파일 텍스처가 해당 평면에 연결되어 있어야 합니다.



- 파일 텍스처를 선택하고 큐브 프리미티브의 길이를 구동하는 데 사용할 텍스처 맵을 엽니다. 대비가 높고 작은 크기에서도 인식할 수 있는 텍스처 맵을 선택하십시오. 이미지에 부수적인 디테일이 너무

많으면 큐브로 잘 전송되지 않습니다.



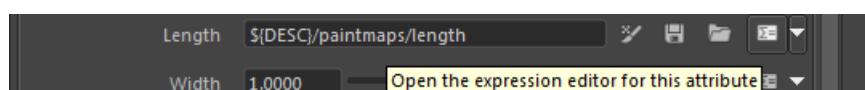
큐브 아카이브(상단 페이지 이미지에 사용되는 맵)의 '길이'를 구동하는 데 사용되는 흑백 텍스처.

- 장면을 렌더링하기 전에 먼저 XGen에 파일 텍스처를 업데이트할 것을 지시합니다. 이것은 각 `length`, `width`, `depth` 속성에 대해 디스크 아이콘을 클릭하여 실행할 수 있습니다. 조명을 만들고 장면을 렌더링합니다.

원시 속성을 정의하는 데 사용되는 파일 텍스처를 업데이트한 후 디스크 아이콘을 선택해야 합니다. 그렇지 않으면 XGen이 이것을 업데이트하지 않고 렌더링도 변경되지 않습니다.

Length Expression

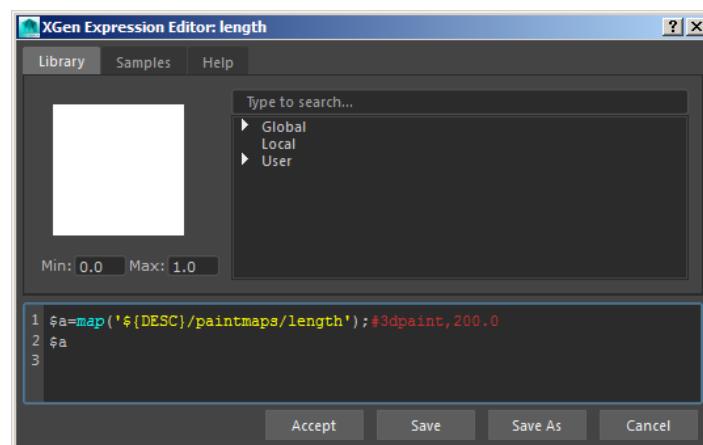
- Length 슬라이더 컨트롤 오른쪽에 있는 **Expression** 아이콘을 클릭합니다.



- Expression Editor**에 다음 `length` 수식 텍스트가 나타납니다. **Expression Text Field**에 다음 텍스트가 나타납니다.

```
$a=map('${DESC}/paintmaps/length');#3dpaint,200.0
```

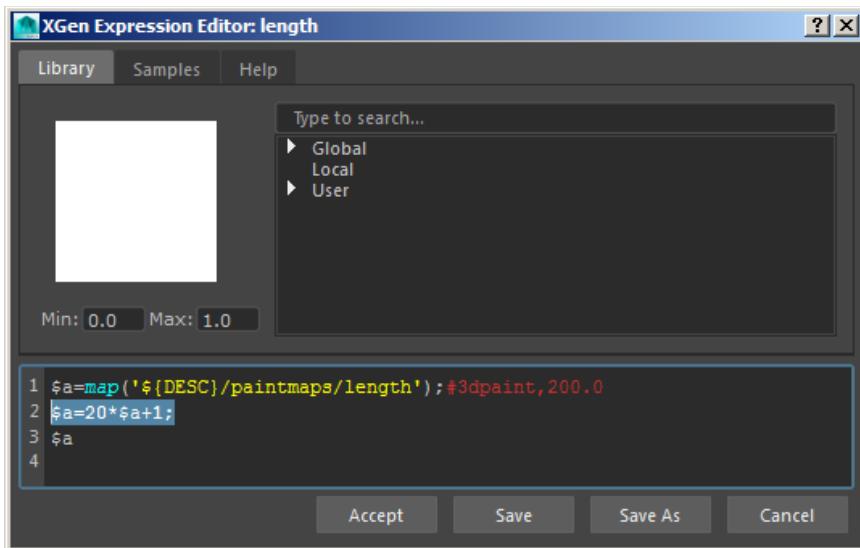
이것은 텍스처 맵을 사용하여 길이 속성을 정의하는 데 사용되며 200은 앞에서 정의한 Map Resolution입니다.



- Expression Editor**에서 다음 행을 추가합니다.

```
$a=20*$a+1;
```

여기서 20은 큐브 아카이브의 최대 길이이며 1은 최소 크기입니다. 길이 Expression Editor는 다음과 같습니다. 수식 텍스트를 복사합니다. 완료 후 Accept를 클릭합니다.



- 여기에 사용한 **Length** 것과 동일한 수식을 붙여 넣습니다. **Depth** 속성에도 동일한 것을 실행합니다.

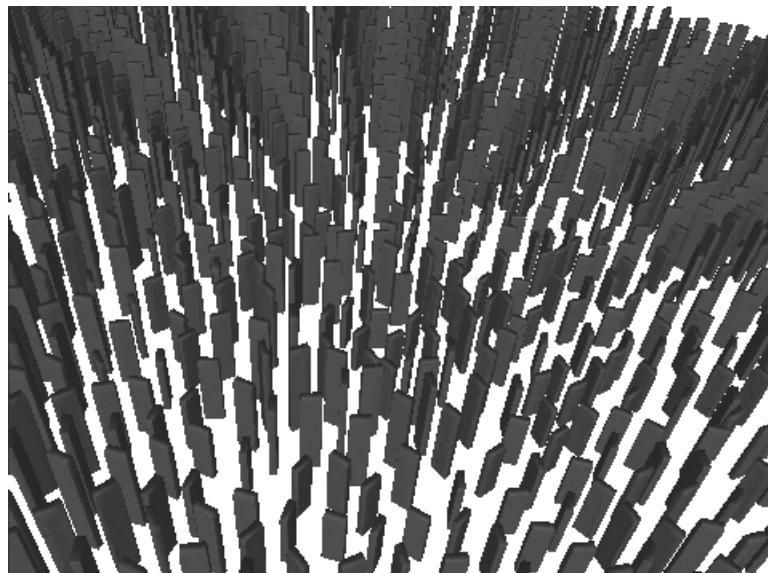
애니메이션

XGen 큐브 아카이브의 **length** 속성에 애니메이션을 추가하기 위해 **\$frame** 수식을 사용할 수 있습니다.

- 길이 속성에 대해 **Expression Editor**를 열고 Editor 창에 다음을 추가합니다.

```
$a=map('${DESC}/paintmaps/length1');#3dpaint,200.0
$a=20*$a+$frame+1;
$a
```

- Length** 속성의 크기를 늘리려면 **20**을 **40**과 같이 더 높은 값으로 높이는 것이 좋습니다. **Accept**를 클릭합니다. **Frame**이라는 슬라이더가 생성된 것을 볼 수 있습니다.
- Time Slider에서 프레임 1로 이동합니다. **Frame**을 마우스 오른쪽 버튼으로 클릭하고 **Set Key**를 선택합니다. 큐브 길이가 0으로 시작되도록 음수 값을 사용해야 할 수도 있습니다.
- 끝 프레임(예: 100)으로 이동하여 프레임에 대해 다른 키 프레임(25와 비슷한)을 만듭니다. 애니메이션을 보려면 Time Slider를 움직입니다. 애니메이션 중에 큐브의 길이가 늘어나는 것처럼 보일 것입니다.
- Twist Width** 속성을 키프레임화하여 큐브가 흥미로운 형태로 왜곡되는 것도 확인할 수 있습니다. 아래 예에서는 200과 600 사이의 값들을 사용합니다.



Batch Render

일괄 렌더링 전에 다음 단계를 수행해야 합니다. 그렇지 않으면 XGen 설명이 렌더링되지 않습니다.

- 장면을 저장합니다.
- 지오메트리 및 XGen 설명이 선택된 채로 XGen description 메뉴에서 **File>Export Patches for Batch Render**로 이동합니다. XGen description에 애니메이션이 있으면 해당 애니메이션을 활성화하고 **Frame Range**를 선택하십시오. **AbcExport.mll**을 Plug-in Manager에서 로드해야 합니다. 그렇지 않으면 애니메이션이 제대로 내보내지지 않습니다.
- 장면을 저장합니다.
- 애니메이션이 마음에 들면 일괄 렌더링 애니메이션을 시작할 수 있습니다.



결론

지금까지 텍스처 맵과 수식을 통해 XGen 프리미티브 아카이브의 형태를 애니메이트하는 방법을 살펴봤습니다. 이것은 이 강력한 도구가 할 수 있는 것들 중 한 가지일 뿐입니다. 여러 텍스처 맵과 스플라인 및 구체와 같은 기본 유형에 대해 실험해보십시오. 다음은 길이 속성에 연결된 다양한 텍스처를 사용하는 추가적인 예입니다.

텍스처 맵으로 프리미티브 제어하기



텍스처 맵에 의해 구동되는 XGen '아카이브' 큐브 프리미티브로 구성된 장면

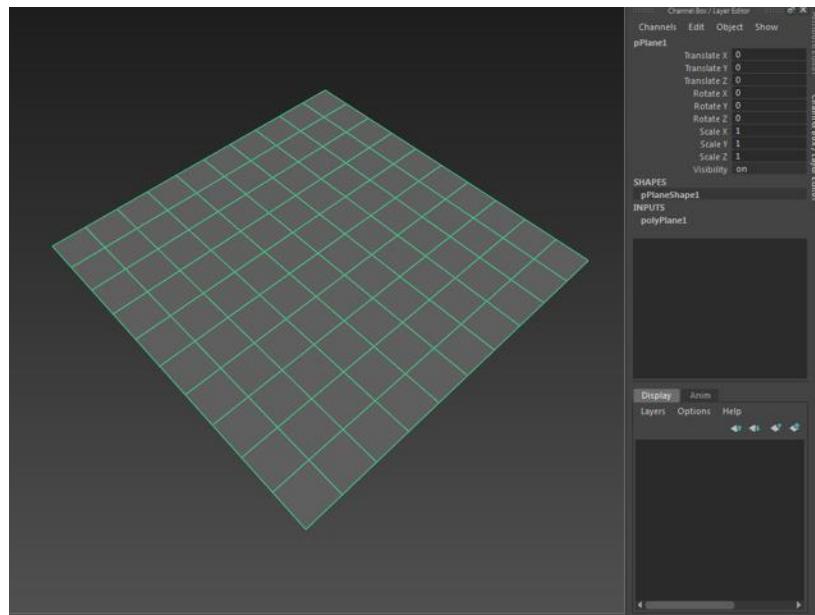
XGen은 무작위로 또는 균일하게 배치된 임의 개수의 프리미티브를 배치할 수 있는 많은 창조적인 기회를 제공하는 강력한 도구입니다. XGen Description과 함께 텍스처 맵을 사용하여 프리미티브 생성의 위치와 밀도를 지정하고 프리미티브 속성의 수를 제어할 수 있습니다. 이 자습서는 텍스처 맵을 사용하여 XGen 프리미티브의 길이, 너비, 깊이를 쉽게 '구동'하는 방법을 보여줍니다. 이러한 프리미티브는 구체, 스플라인 또는 아래 예와 같이 아카이브로 내보낸 지오메트리까지 범위가 다양합니다.



텍스처 맵에 의해 구동되는 '피라미드 형상' 아카이브 분포

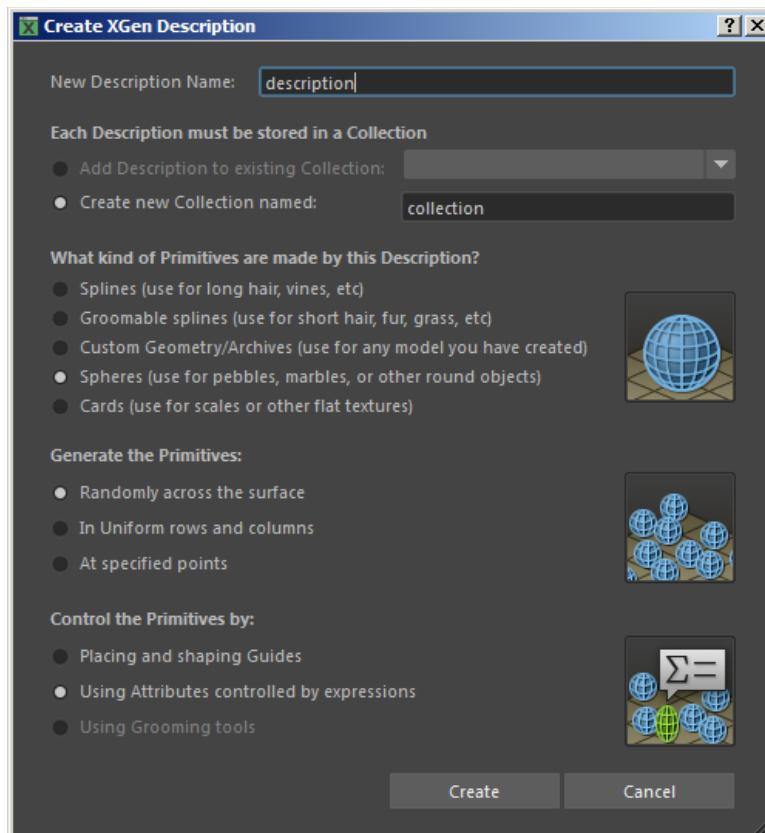
이 방법으로 할 수 있는 일은 무궁무진합니다. 이 자습서는 맵을 사용하여 풍경에 있는 나무의 위치를 유도하거나 이 경우와 같이 우스꽝스러운 얼굴을 만드는 데 사용할 수 있습니다.

- 먼저 폴리곤 평면을 만들고 **Subdivisions Width** 및 **Subdivisions Height**를 10으로 늘립니다. 이 평면을 사용하여 XGen 설명을 할당할 것입니다.

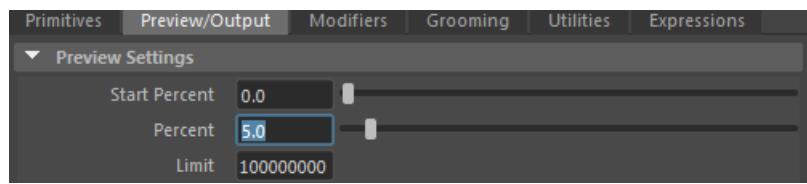


- 폴리곤 평면을 선택하고 **Generate> Create Description...**으로 이동합니다. 그러면 아래와 같이 **Create XGen Description**이 열립니다.

- Spheres** primitive를 선택합니다. 다른 기본 설정은 그대로 두고 **Create**를 선택합니다.

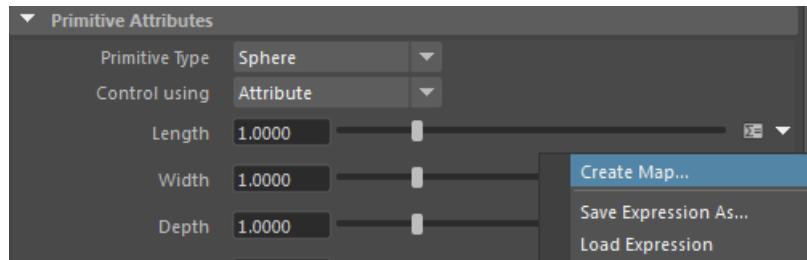


- 계속 진행하기 전에 **Preview/Output** 탭에서 **Percent** 값을 낮추는 것이 좋습니다. **Percent** 값이 너무 높으면 생성된 많은 수의 프리미티브로 인해 컴퓨터가 멈출 수도 있습니다.

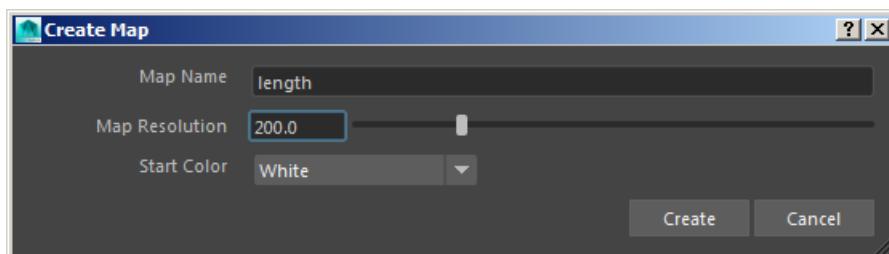


- Length 오른쪽에 있는 아래 방향 화살표를 클릭합니다. **Create Map...**을 선택합니다.

원래 Maya 쉐이더가 XGen 설명에 할당되어 있는지 확인하십시오. 그렇지 않으면 이 단계가 작동하지 않습니다! 맵을 만든 후에는 다른 쉐이더로 변환할 수 있습니다.

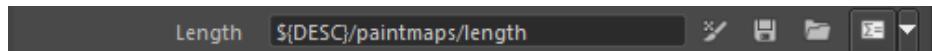


- Map Name은 'length'여야 합니다. Map Resolution을 200 정도로 높이면 PTEX 맵의 해상도가 면당 텍셀로 설정됩니다. 고해상도 텍스처를 사용할 때는 더 큰 Map Resolution 값을 사용하십시오. 이것이 완료되면 **Create**를 클릭합니다.



Expressions

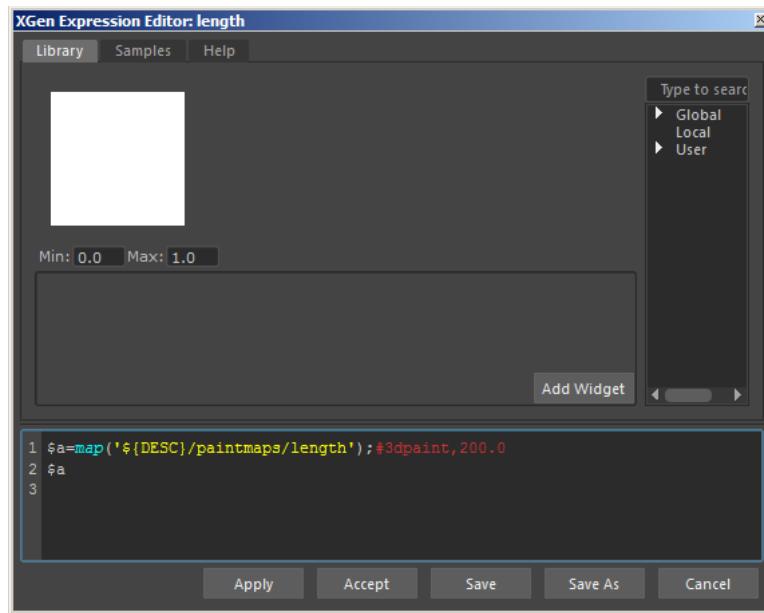
- Length 슬라이더 컨트롤 오른쪽에 있는 Expression 아이콘을 클릭합니다.



- Expression Editor에 다음 **length** 수식 텍스트가 나타납니다. Expression Text Field에 다음 텍스트가 나타납니다.

`$a=map('${DESC}/paintmaps/length');#3dpaint,200.0`

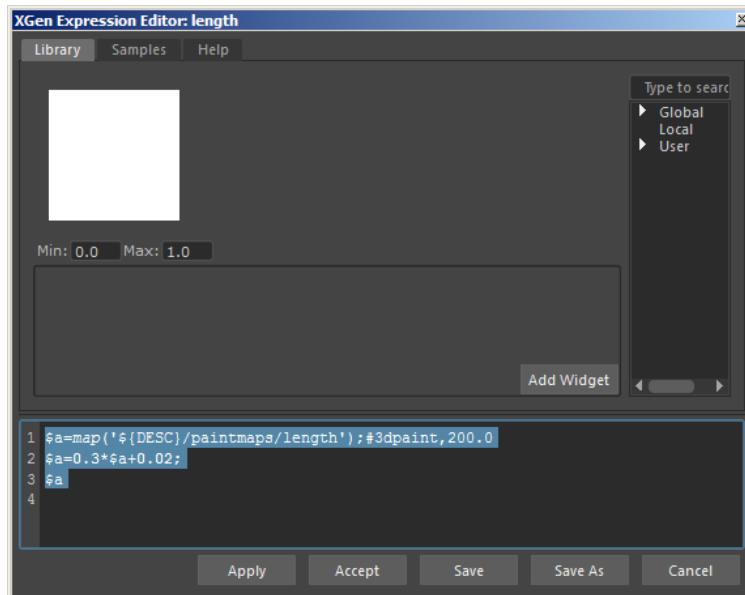
이것은 텍스처 맵을 사용하여 길이 속성을 정의하는 데 사용되며 200은 앞에서 정의한 Map Resolution입니다.



- **Expression Editor**에서 다음 행을 추가합니다.

```
$a=0.3*$a+0.02;
$a
```

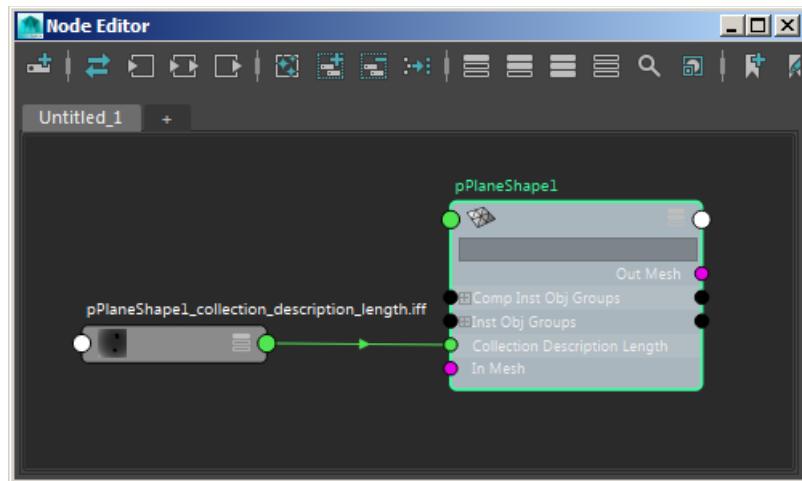
여기서 0.5(0.3 + 0.2)는 구체의 최대 길이이며 0.02는 최소 크기입니다. 길이 Expression Editor는 다음과 같습니다. 수식 텍스트를 복사합니다. 구체 프리미티브의 Width 및 Depth 설정에 대해 동일한 수식을 사용합니다. 완료 후 Accept를 클릭합니다.



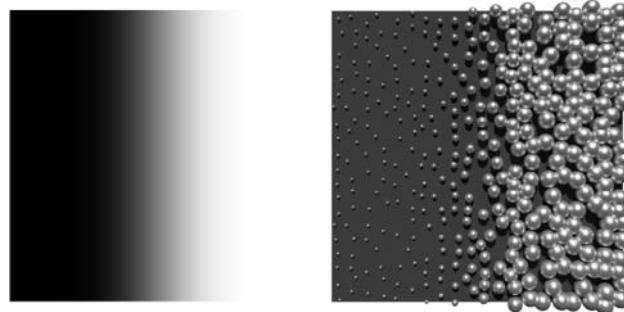
- 그런 다음 **Width** 슬라이더 컨트롤의 오른쪽에 있는 **Expression** 아이콘을
- 클릭하면 XGen Expression Editor에서 PTEX 맵에 의해 생성된 수식을 볼 수 있습니다.
- 여기에 **length**를 위해 사용한 것과 동일한 수식을 붙여 넣습니다. **Depth** 속성에도 동일한 것을 실행합니다.

텍스처 맵 생성

이제 텍스처 맵을 추가하겠습니다. 슬라이드 컨트롤  오른쪽에 있는 페인트 아이콘을 클릭합니다. Hypershade 및 Node Editor 창에서 파일 텍스처가 해당 평면에 연결되어 있어야 합니다.



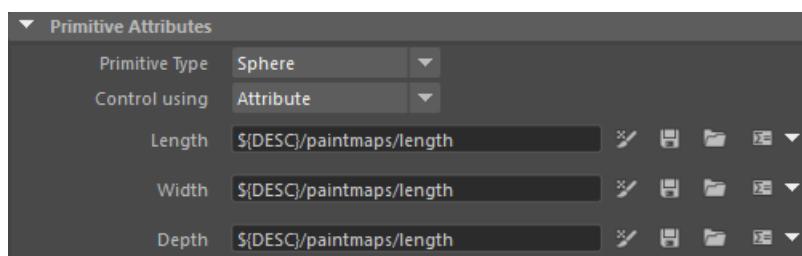
- 파일 텍스처를 선택하고 구체 프리미티브의 크기를 구동하는 데 사용할 텍스처 맵을 엽니다. 대비가 높고 작은 크기에서도 인식할 수 있는 텍스처 맵을 선택하십시오. 이미지에 부수적인 디테일이 너무 많으면 구체로 잘 전송되지 않습니다. 아래 예에서 구형 프리미티브의 길이, 너비 및 깊이 스케일에 대한 회색조 램프의 효과를 볼 수 있습니다.



회색조 램프 기반의 구체 분산

- 장면을 렌더링하기 전에 먼저 XGen에 파일 텍스처를 업데이트할 것을 지시해야 합니다. 이것은 각 length, width, depth 속성에 대해 디스크 아이콘  클릭하여 실행할 수 있습니다. 조명을 만들고 장면을 렌더링합니다.

원시 속성을 정의하는 데 사용되는 파일 텍스처를 업데이트한 후 디스크 아이콘  선택해야 합니다. 그렇지 않으면 XGen이 이것을 업데이트하지 않고 렌더링이 변경되지 않습니다.

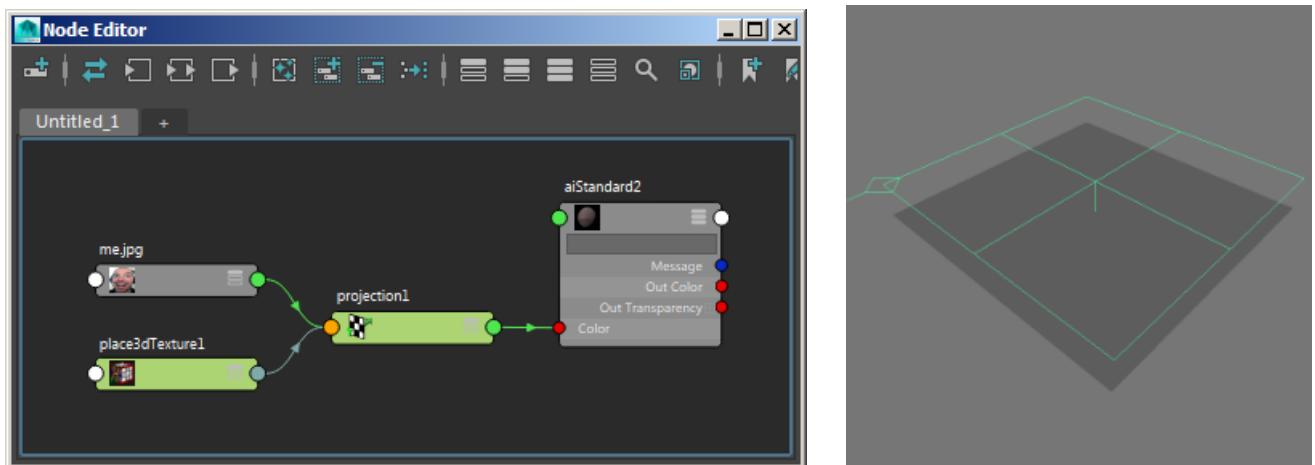




텍스처 맵 없이 렌더링된 구체 프리미티브

텍스처링

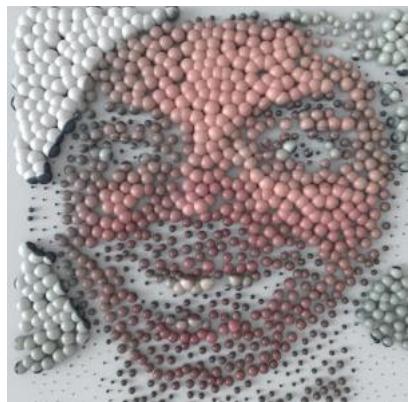
- 다음 단계는 색상 텍스처 파일을 추가하는 것입니다. **Standard Surface** 쉐이더를 XGen 컬렉션에 할당합니다. 색상 파일 텍스처를 **Diffuse Color** 속성에 할당합니다. 폴리곤 평면과 정렬할 수 있는 투영된 파일 텍스처 맵인지 확인하십시오. **place3dTexture** 노드를 선택하고 **Fit to Group BBox**를 선택합니다. **place3dTexture**가 폴리곤 평면의 크기에 맞춰집니다.



얼굴 이미지는 3d projection 텍스처 맵을 사용합니다.

밀도

- Expression Editor에서 사용한 값들에 따라 **Density** 값을 변경해야 할 수도 있습니다. 구체의 밀도가 너무 높다면 **Density** 값을 줄이는 것이 좋습니다. 반대로 구체들의 밀도가 너무 낮다면 **Expression Editor**에서 밀도를 높이거나 **최대 length** 크기를 높여야 할 수 있습니다. 아래 이미지는 Density를 높였을 때 XGen에 의해 생성되는 프리미티브의 수가 어떤 영향을 받는지를 보여줍니다.



Density 20



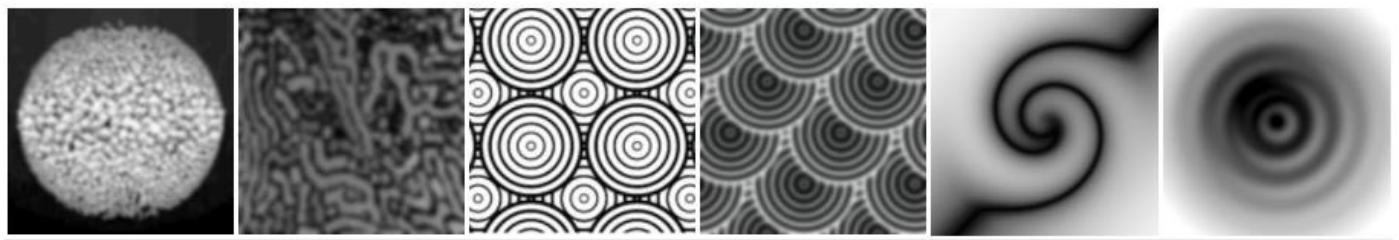
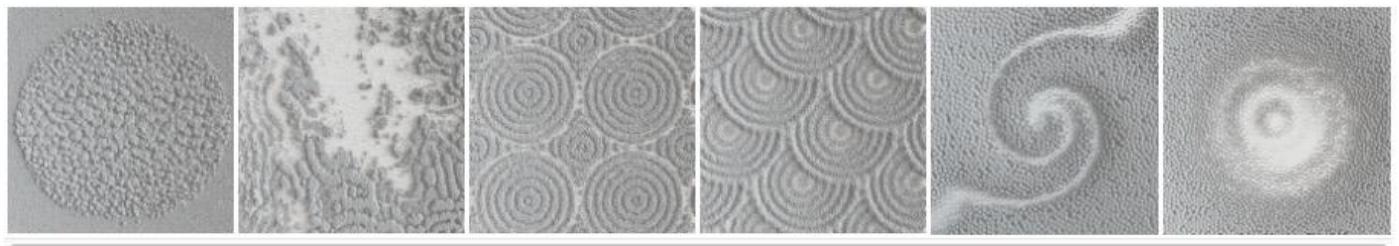
Density 40



Density 80

추가 예

지금까지 텍스처 맵으로 XGen 프리미티브를 제어하는 방법을 살펴봤습니다. 이것은 이 강력한 도구가 할 수 있는 것들 중 한 가지일 뿐입니다. 다음은 프리미티브 배포를 위해 사용되는 여러 이미지의 몇 가지 예입니다. 여러 텍스처 맵을 실험해보고 스플라인 및 아카이브 등 다른 기본 유형을 사용해 보십시오.



구체 프리미티브를 구동하는 데 사용되는 텍스처 맵

장난감 병사 만들기



이 자습서에서는 장난감 병사들과 놀던 즐거웠던 유년기로 돌아가보고자 합니다. 하지만 어린 시절과는 달리 XGen을 사용하여 수식과 텍스처 맵을 마스크로 사용하여 병사들을 흥미로운 구성으로 배열할 수 있습니다. 또한 수식과 텍스처 맵을 사용하여 개별 병사에게 색을 추가하는 방법에 대해서도 설명합니다. 실제 세계에서 영감을 얻으려면 이 아티스트, 그리고 이 아티스트의 작품에 대해서도 살펴봐야 합니다. 이 두 아티스트는 모두 재미있는 예술 작품을 만들기 위해 장난감 병사들을 사용합니다.

이 자습서는 다음 장들로 구성되어 있습니다.

- [아카이브 파일 만들기](#)
- [XGen 설명 작성하기](#)
- [Generator Attributes](#)
- [수식을 사용하여 병사에 임의 색상 지정하기](#)
- [수식을 마스크로 사용하기](#)
- [마스크로 텍스처 맵 사용하기](#)
- [워크플로우 문제 - Primitive Bound](#)

장난감 병사 모델링 및 관련 동영상은 하단 링크에서 다운로드 할 수 있습니다.

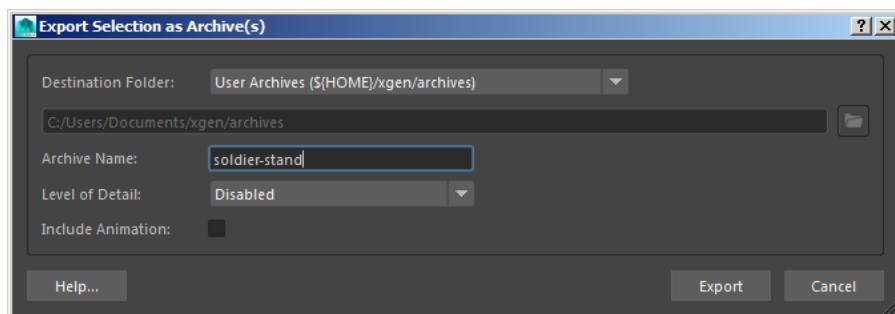
<https://support.solidangle.com/display/A5AFMUG/Create+an+Army+of+Toy+Soldiers>

아카이브 파일 만들기

- 먼저 Maya에서 장난감 병사들을 엽니다. 우선 개별 병사 모델을 XGen 아카이브로 변환해야 XGen에서 이들과 함께 놀 수 있습니다.



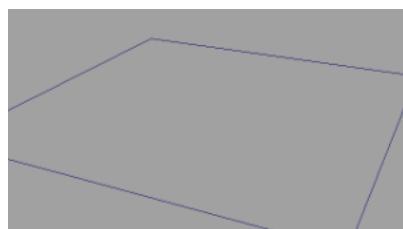
- 각 병사를 개별적으로 선택하고 **Generate>Export Selection as Archive(s)...**로 이동합니다. 아카이브 이름과 대상 폴더를 선택하고 **Export**를 선택하십시오.



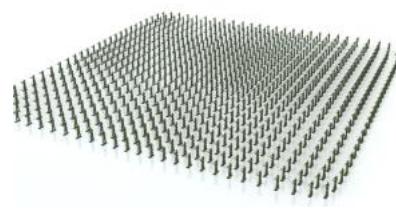
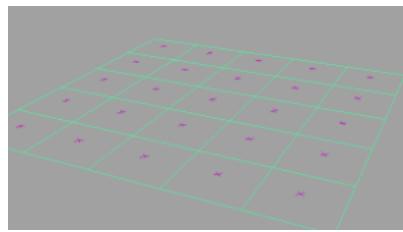
XGen 설명 작성하기

- 폴리곤 평면을 만들고 병사들로 채우고자 하는 크기로 설정하십시오. **Height** 및 **Width**에서 폴리곤 평면의 **subdivision**의 수를 최소 5로 높입니다.

XGen 아카이브 프리미티브의 수는 폴리곤 평면 내의 구획 수와 관련이 있습니다. 병사들이 평면의 전체 크기를 채우지 못한다면 폴리곤 평면에 더 많은 구획을 추가해야 합니다.

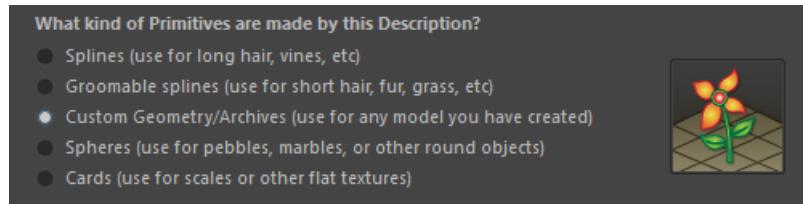


구획이 1개인 폴리곤 평면 병사들이 그런 것이 아닙니다.

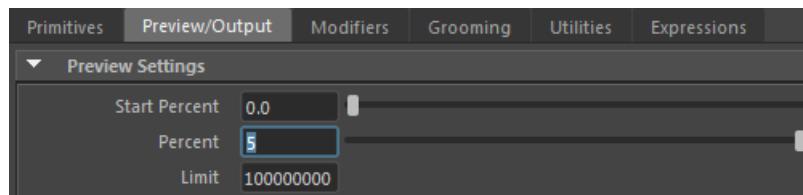


구획이 5개인 폴리곤 평면 병사들이 평면의 크기를 채웁니다.

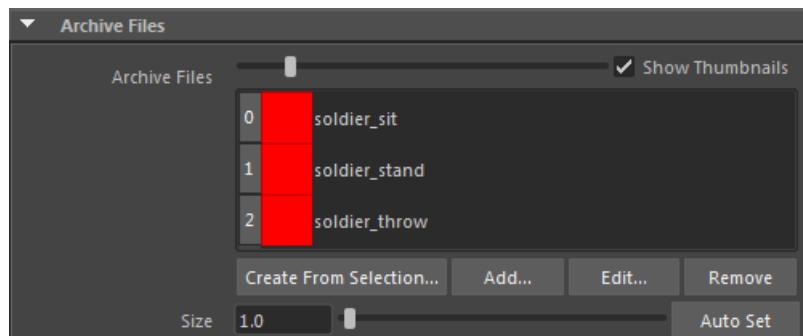
- 폴리곤 평면이 선택된 상태로 **Generate> Create XGen Description**으로 이동합니다. **Custom Geometry/Archives**를 선택합니다(생성한 모든 모델에 해당).



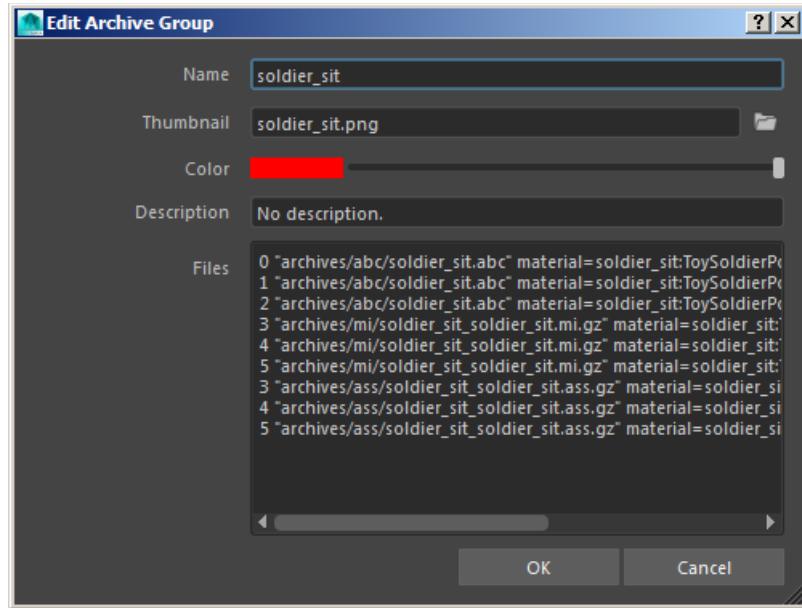
- 계속 진행하기 전에 **Preview/Output** 탭에서 **Percent** 값을 낮추는 것이 좋습니다. **Percent** 값이 너무 높으면 생성된 많은 수의 프리미티브로 인해 컴퓨터가 멈출 수도 있습니다.



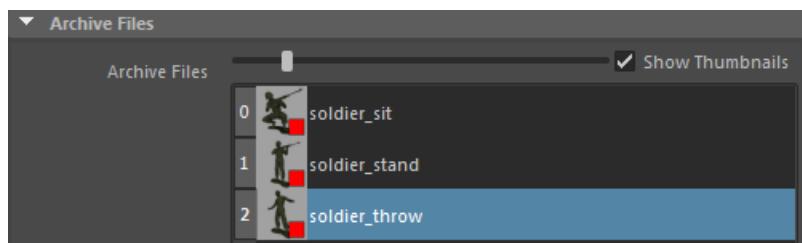
- **Archive Files** 아래에서 **Add**를 선택하고 앞서 병사 아카이브를 저장한 위치로 이동합니다.



- 각 병사에 대한 아이콘의 색상을 변경하고 각 아카이브를 나타내는 이미지 아이콘까지 추가할 수 있습니다. 이렇게 하려면 아카이브 파일 중 하나에 대한 빨간색 사각형을 마우스 오른쪽 버튼으로 클릭하고 **Edit**을 선택하십시오.

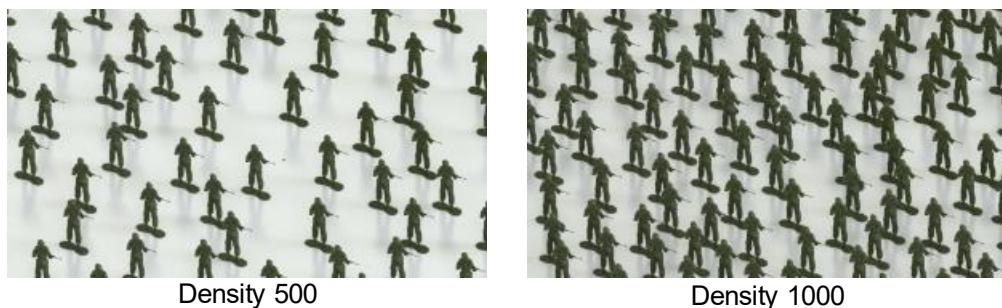


- Thumbnail 폴더 아이콘을 클릭하고 각 병사를 나타내는 비트맵을 선택합니다. 아래 예제에서 각 병사에 대한 Maya 뷰포트를 스크린에 캡처하고 이들 이미지를 파일로 저장했다는 것을 알 수 있습니다. 또한 위의 슬라이더를 조정하여 아이콘의 색상을 변경하고 크기를 늘릴 수도 있습니다.



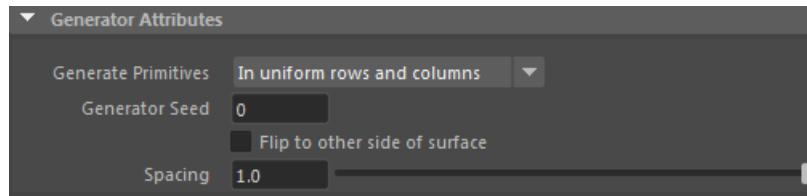
밀도

- Density를 500 정도로 높입니다. 그러면 폴리곤 평면에서 병사 프리미티브의 수가 증가합니다.



Generator Attributes

병사 배치하기



Generate Primitives가 **In uniform rows and columns**로 설정되어 있으면 **Spacing** 속성이 **Density**를 교체합니다. **Spacing** 속성을 높이면 프리미티브 사이의 거리가 늘어납니다.



일반 프리미티브 'Randomly across the surface'

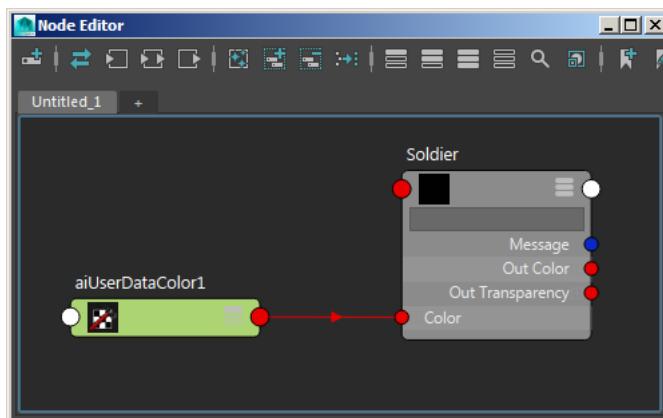


일반 프리미티브 'In uniform rows and columns'

수식을 사용하여 병사에 임의 색상 지정하기

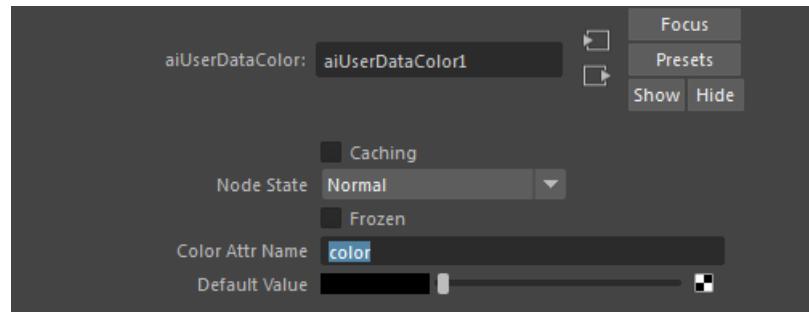
이를 수행하는 자세한 방법은 '[프리미티브에 임의 색상 지정하기](#)' 자습서를 참조하십시오.

- **Ai UserData Color** 노드를 생성하고 병사 XGen 설명에 지정된 **Standard Surface** 쉐이더의 **Color** 속성에 연결합니다.



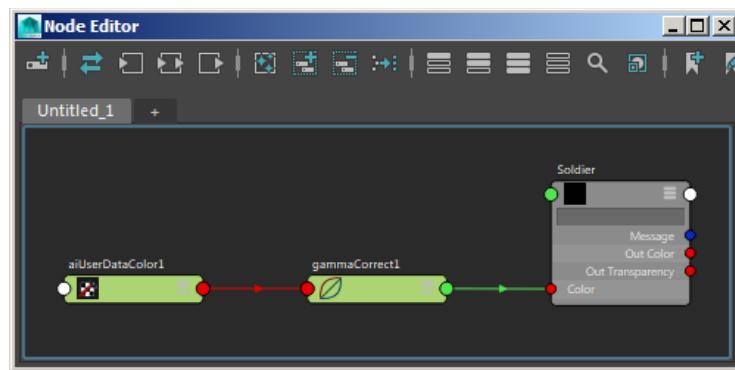
Standard Surface 쉐이더의 Color 속성에 연결된 Ai UserData Color 노드

- Ai UserData 색상 노드의 **Color Attr Name**에 **color**을 입력합니다. XGen 설명에서 동일한 "**Color**" 이름을 사용하겠습니다.

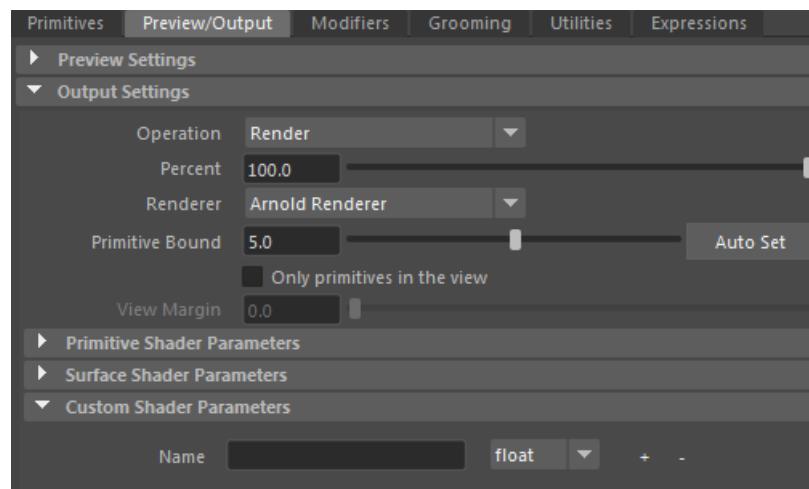


Ai UserData Color 노드에서 Color Attr Name에 'color'라는 이름을 지정합니다.

- MtoA는 **Ai UserData Color** 노드를 감마 보정하지 않으므로 아래 그림과 같이 Ai UserData Color 노드와 Standard Surface Shader의 Color 속성 사이에 **Maya Gamma Correct** 노드를 추가해야 합니다. Gamma RGB 값을 0.454로 변경합니다.



- 장면을 렌더링합니다. Ai UserData Color Default Value가 black이기 때문에 군인들이 검은색입니다. XGen의 **Custom Shader Parameters**를 사용하여 XGen 설명에 이를 연결해야 합니다.
- XGen에서 **Preview/Output** 탭을 클릭하고 **Output Settings**를 엽니다. 그 아래 **Custom Shader Parameters**가 나타납니다. 여기서 식을 추가할 것입니다.



Custom Shader Parameters (아래)

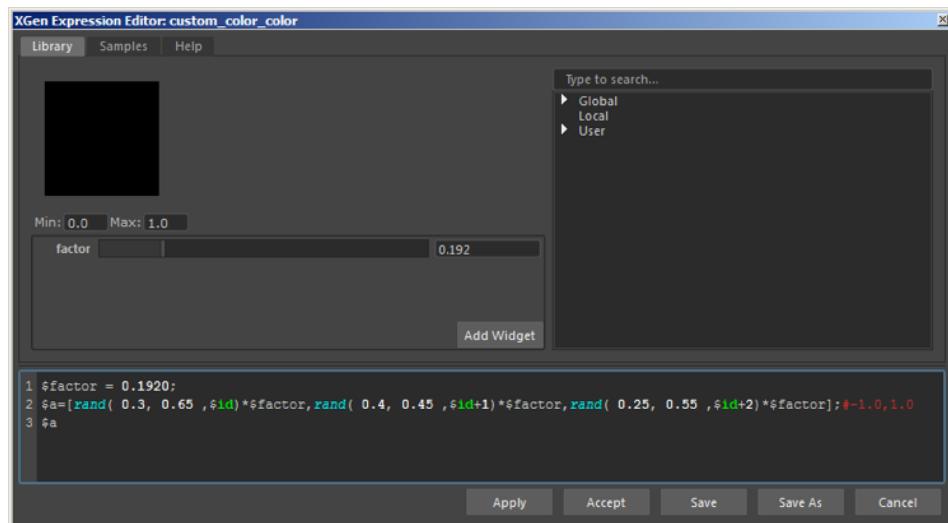
- Name 텍스트 필드에 Ai UserData Color 노드가 있는 **Color Attr Name**에 사용한 이름을 입력합니다. 이 경우에는 'color'라는 단어를 사용했습니다. **Float**를 클릭하고 **Color**로 변경합니다. 이것이 변경하고자 하는 속성이기 때문입니다.



Color로 설정된 Custom Shader Parameters

- 새로운 'color color' 파라미터의 오른쪽에 있는 XGen Expression Editor 아이콘을 클릭하고 Expression Editor에 다음 텍스트를 추가하십시오.

```
$factor = 0.1920;
$a=[rand( 0.3, 0.65 ,$id)*$factor,rand( 0.4, 0.45 ,$id+1)*$factor,rand( 0.25, 0.55 ,$id+2)*$factor];#-1.0,1.0
$a
```



사용자 정의 색상을 위한 XGen Expression Editor

- 장면을 렌더링합니다. 이제 여러분은 위에서 사용한 값을 기반으로 병사들의 색상이 무작위로 생성되었음을 알 수 있습니다.



마스크로 작업하기

수식을 마스크로 사용하기

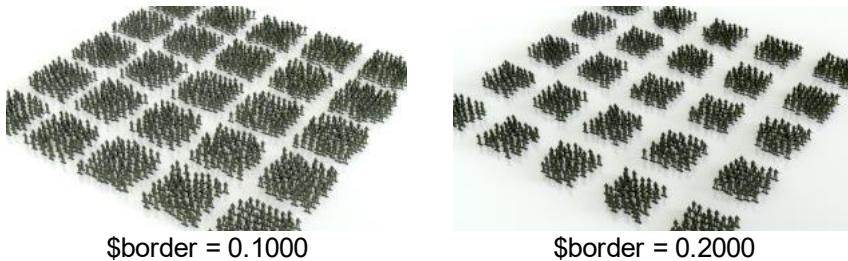
또한 수식을 사용하여 각 폴리곤 얼굴에 나타나는 병사의 수를 정의할 수도 있습니다.

- Mask 슬라이더 컨트롤의 오른쪽에 있는 Expression 아이콘을 클릭하고 XGen Expression Editor에

다음 텍스트를 입력하십시오.

```
$border = 0.1150;
$u > $border && $u < (1-$border) && $v > $border && $v < (1-$border)
```

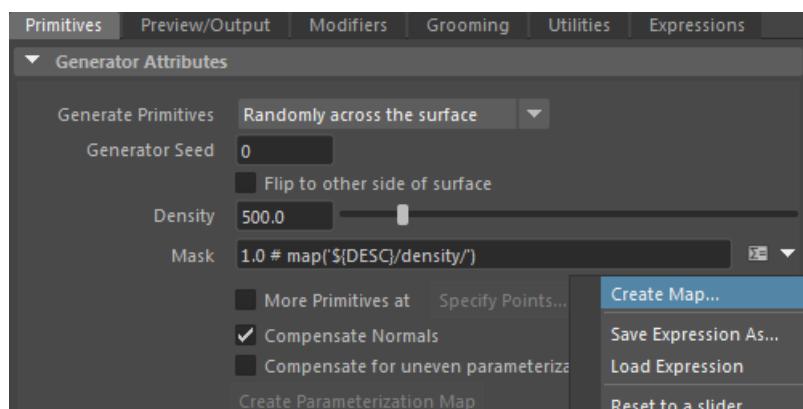
각 얼굴에 대해 \$u 및 \$v로 UV 좌표를 얻을 수 있습니다. 이 수식을 사용하면 U가 0.2보다 크고 0.8보다 작고 V가 0.2보다 크고 0.8보다 작은 장소만 채웁니다. 이 방법으로 폴리곤 평면의 각 면 가운데에 정사각형이 생깁니다.



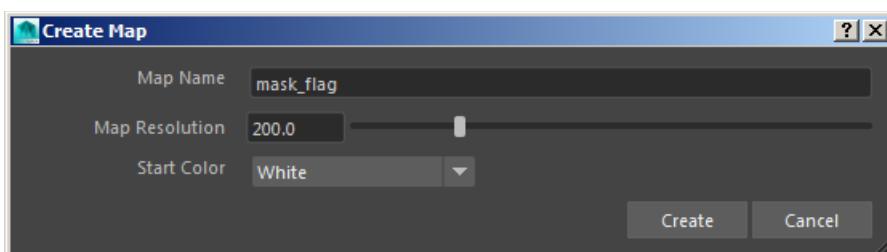
마스크로 텍스처 맵 사용하기

또한 흑백 격자 텍스처를 사용하여 아군을 위한 형태를 매핑할 수 있습니다.

- Mask 오른쪽에 있는 아래 방향 화살표를 클릭합니다. **Create Map...**을 선택합니다. (이것은 평면에 Maya 쉐이더가 할당되어 있는 경우에만 작동한다는 점을 유의하십시오).



- Map Name은 'mask'여야 합니다. Map Resolution을 200 정도로 높입니다. 그러면 PTEX 맵의 해상도가 해당 텍셀로 설정됩니다. 고해상도 텍스처를 사용할 때는 더 큰 Map Resolution 값을 사용하십시오. 이것이 완료되면 **Create**를 클릭합니다.



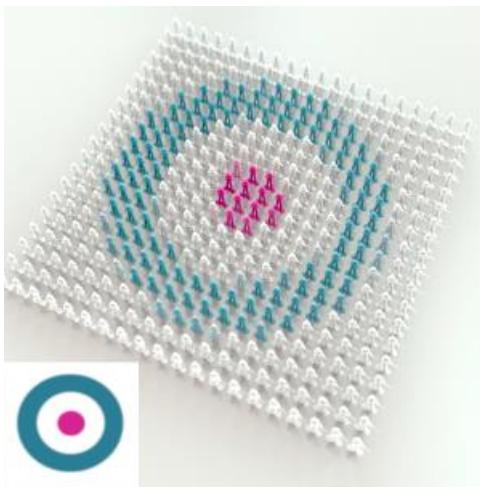


Density Mask로 사용되는 격자 텍스처 맵



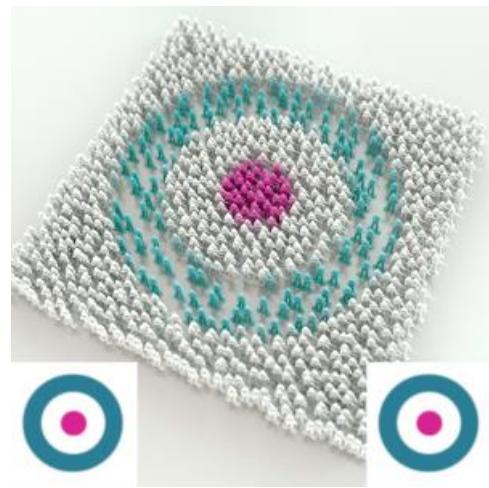
Density Mask로 사용되는 체커 텍스처 맵

아래 예제는 색상 텍스처를 사용하여 밀도 마스크를 구동할 때의 또 다른 예를 보여줍니다.



색상에 대해서만 사용된 원형

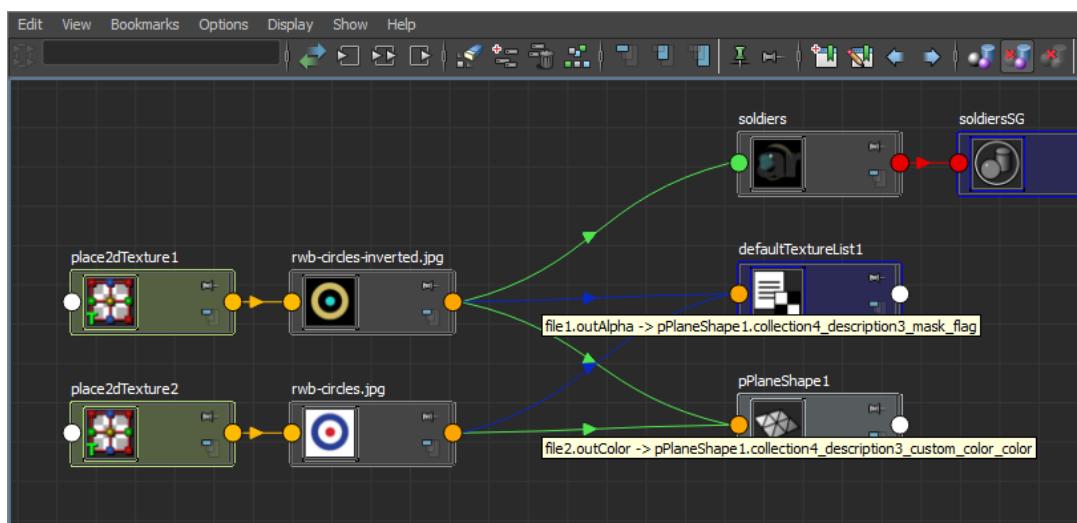
맵(프리미티브가 배치된 'In uniform rows and columns')



색상과 밀도 모두에 사용되는 원형

맵(프리미티브가 배치된 'In uniform rows and columns')

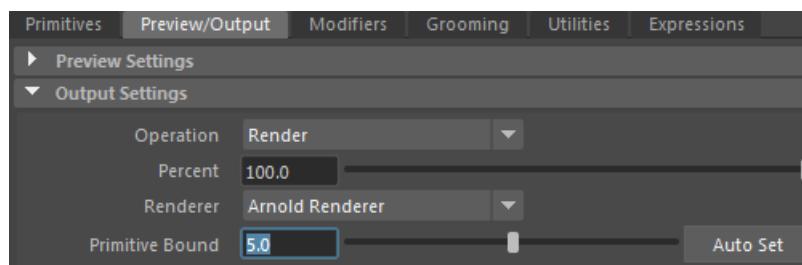
- 마스크에 사용된 맵을 반전하면 해당 색상 맵에 사용된 흰색 영역에서 투명도를 얻을 수 있습니다.



마스크에 대해 사용된 맵이 반전되므로 병사들이 검은 색 영역에 나타나지 않습니다.



워크플로우 문제 - Primitive Bound



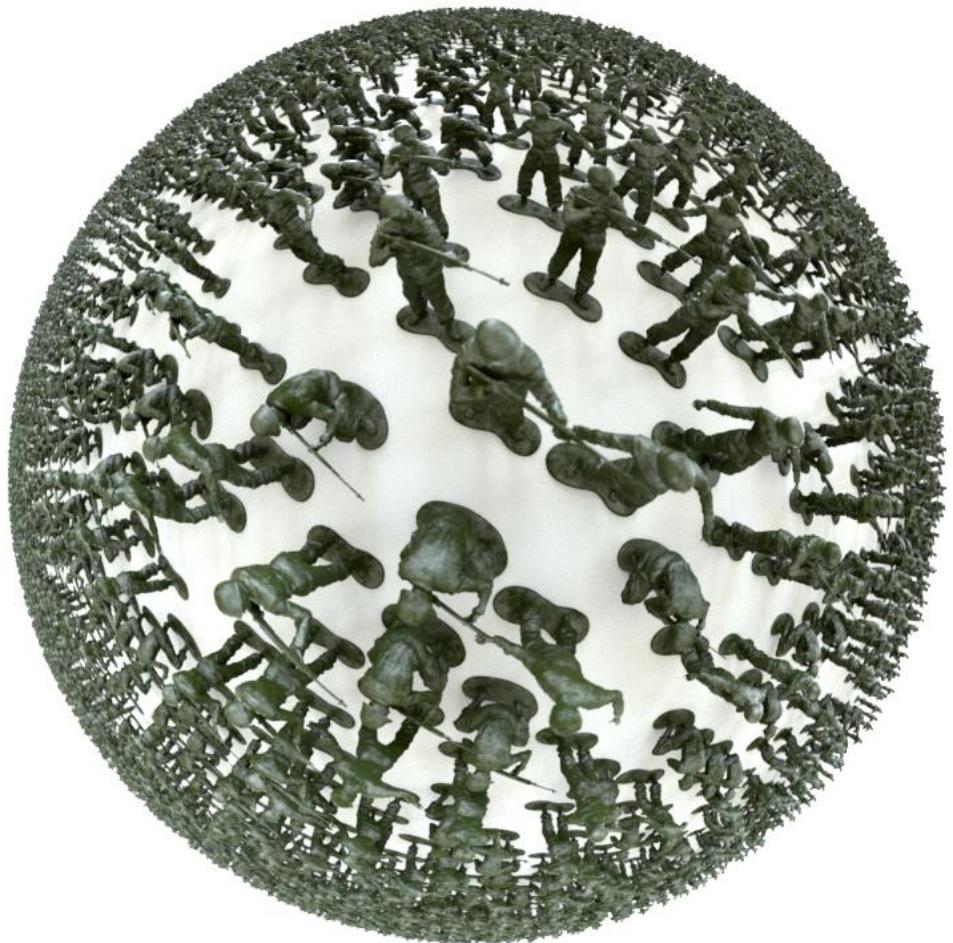
아래의 렌더링을 보면 오른쪽 상단 모서리에 약간의 '잘림'이 있습니다. 이것은 경계 상자 영역이 너무 낮게 설정되어 병사들이 이 영역 외부로 렌더링되지 않기 때문입니다. **Primitive Bound**를 높이면 이 문제가 해결됩니다. 이것은 **Preview/Output** 탭 아래의 **Output Settings**에서 볼 수 있습니다.



Primitive Bound가 XGen 아카이브 프리미티브에 대해 충분히 크게 설정되지 않았습니다.

결론

지금까지 XGen에서 마스크를 만들고 원시 아카이브 장난감 병사 모델을 색칠하는 방법을 알아봤습니다. 흥미로운 시간이 되셨기를 바랍니다. 하지만 이제는 장난감 병사들을 치워놓고 방 청소를 해야 할 시간입니다!



수식으로 도미노 배치하기



이 자습서에서는 XGen을 사용하여 다양한 형태로 도미노를 설정 및 정렬하는 방법을 보여줍니다. 프리셋 수식 중 하나를 사용하여 **Twist** 속성을 구동해 흥미로운 패턴을 만들어 보겠습니다. 이 자습서를 따라하면 개별적으로 도미노를 배열하는 데 필요한 모든 수고와 번거로움을 피할 수 있으며 훨씬 쉬워진 작업에 놀라게 될 것입니다.



이 비디오는 병사들의 원시 아카이브가 포함된 XGen 장면의 IPR 렌더링을 보여줍니다. 도미노들이 움직이는 모습은 하단 링크를 통해 볼 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Positioning+Dominoes+with+Expressions>

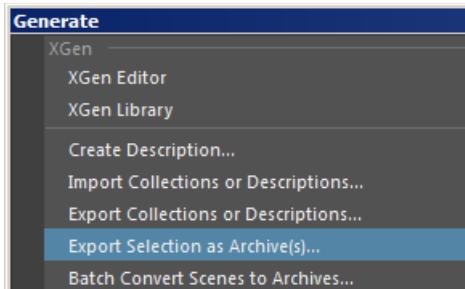
- 먼저 도미노 모델은 하단 링크를 이용하여 Arnold 페이지에서 다운로드 할 수 있습니다.
<https://support.solidangle.com/display/A5AFMUG/Positioning+Dominoes+with+Expressions>



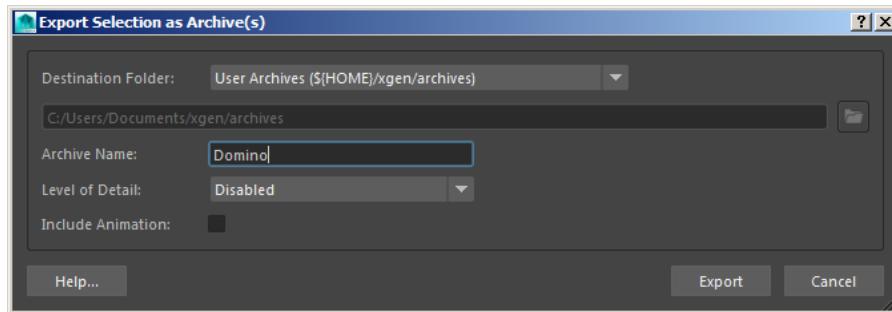
도미노 파일을 디스크에 저장하는 것이 중요합니다. 그렇지 않으면 XGen이 다음 오류를 표시합니다.

// Error: line 1: xgmArchiveExportUI failed because the current scene isn't saved yet. //

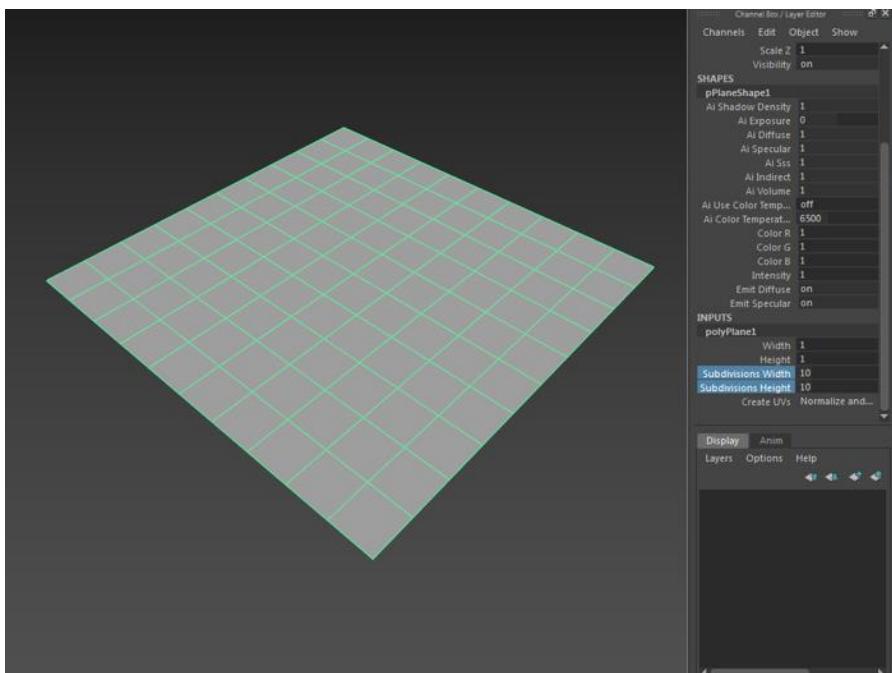
- 도미노 모델을 선택한 채로 **Generate>Export Selection as Archive(s)...**로 이동합니다.



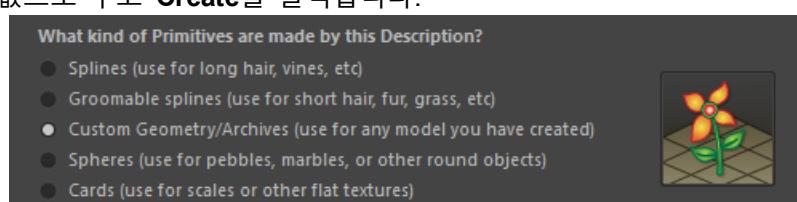
- Archive Name을 **domino**로 변경하고 아카이브 파일을 저장할 위치를 선택합니다.



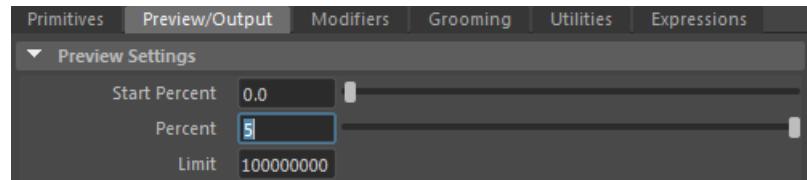
- 이제 XGen 도미노 아카이브를 적용할 면이 필요합니다. 폴리곤 평면을 만들고 **Subdivisions Width** 및 **Subdivisions Height**를 10으로 늘립니다. 이 평면을 사용하여 XGen 설명을 할당할 것입니다.



- 폴리곤 평면을 선택하고 **Generate> Create Description...**으로 이동합니다. 그러면 아래와 같이 **Create XGen Description**이 열립니다. **Custom Geometry/Archives**를 선택합니다(생성한 모든 모델에 해당). 다른 설정은 기본값으로 두고 **Create**를 클릭합니다.

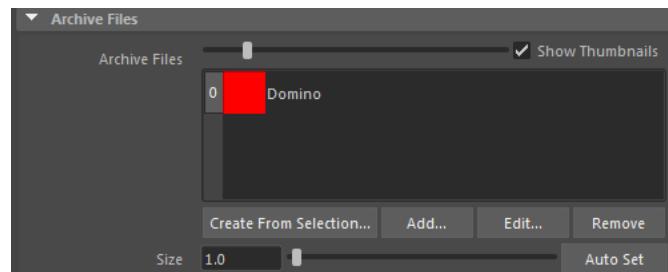


- 계속 진행하기 전에 **Preview/Output** 탭에서 **Percent** 값을 낮추는 것이 좋습니다. **Percent** 값이 너무 높으면 생성된 많은 수의 프리미티브로 인해 컴퓨터가 멈출 수도 있습니다.

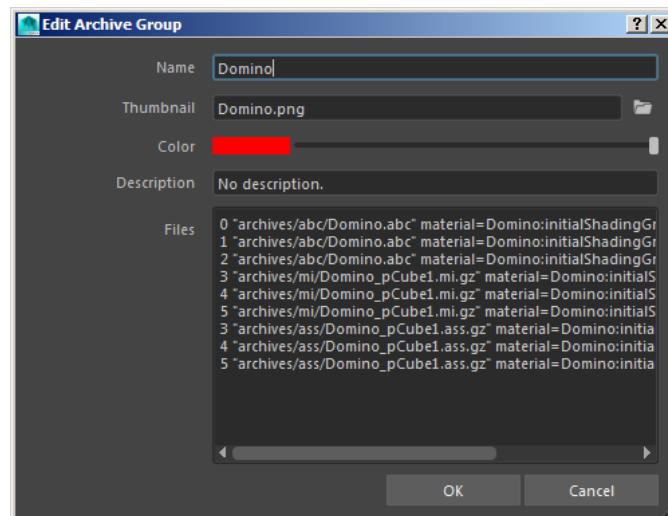


아카이브 추가

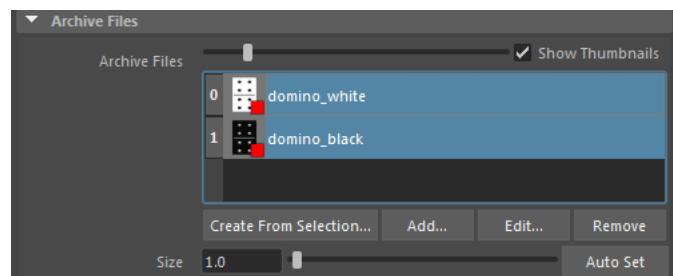
Primitives 탭으로 이동하여 **Archive Files** 아래에서 **Add**를 선택하고 앞서 도미노 아카이브를 저장한 위치로 이동하십시오. Yes를 클릭하여 저장된 도미노 아카이브 파일과 함께 내보낸 Material을 가져오십시오.



- 각 도미노에 대한 아이콘의 색상을 변경하고 각 아카이브를 나타내는 이미지 아이콘까지 추가할 수 있습니다. 이렇게 하려면 아카이브 파일 중 하나에 대한 빨간색 사각형을 마우스 오른쪽 버튼으로 클릭하고 **Edit**을 선택하십시오.



- Thumbnail 폴더 아이콘**을 클릭하고 각 도미노를 나타내는 비트맵을 선택합니다. 아래 예제에서 각 도미노에 대한 Maya 뷰포트를 스크린에 캡처하고 이를 이미지를 파일로 저장했다는 것을 알 수 있습니다. 또한 위의 슬라이더를 조정하여 아이콘의 색상을 변경하고 크기를 늘릴 수도 있습니다.

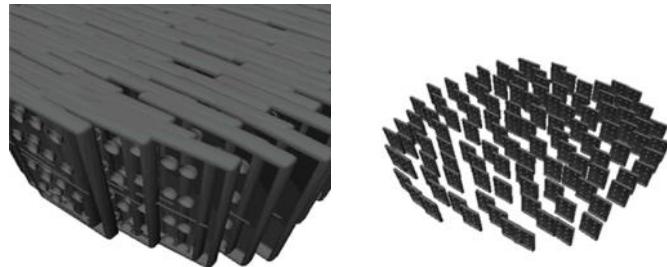


Density

- **Density**를 1000 정도로 높입니다. 그러면 폴리곤 평면에서 도미노 프리미티브의 수가 증가합니다. 아카이브가 바닥 평면에 잘 닿지 않는다면 XGen으로 내보내기 전에 중심점이 도미노의 베이스에 있는지 확인하십시오.

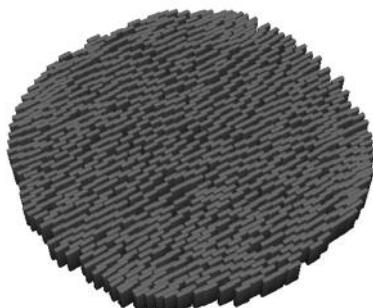
Size

이 값을 키우면 아카이브가 작게 표시되고 줄이면 아카이브가 커집니다. 이 값은 XGen 설명을 할당하는 데 사용한 바닥 평면 지오메트리의 크기에 따라 달라집니다. 도미노가 서로 교차하지 않도록 적절하게 **Size**를 조정하십시오.



1에서 20으로 커진 Size 값

최종적으로 다음과 같은 모습이어야 합니다(미리보기 퍼센트 100).



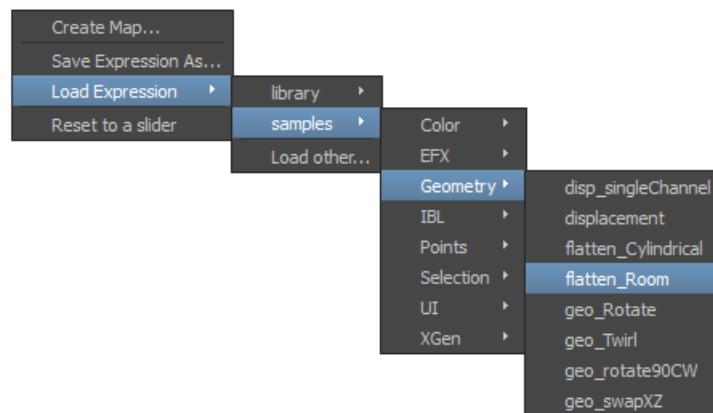
Twist Expression

- **Primitive Attributes** 아래에서 **Twist** 슬라이더 컨트롤 오른쪽에 있는 아래쪽 화살표를 클릭합니다.



- 메뉴에서 다음을 선택합니다.

Load Expression>samples>Geometry>flatten_Room.

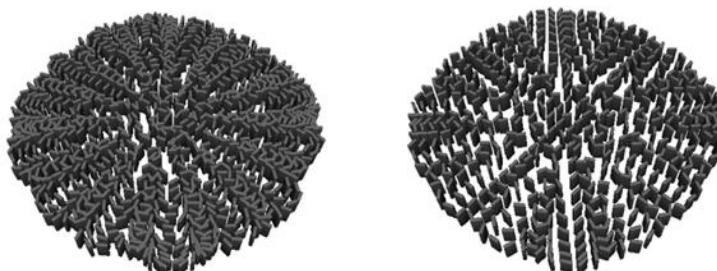


Twist 아래에 **Width** 및 **Depth** 속성이 나타납니다.



Twist Width

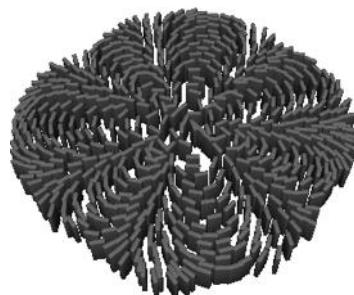
아래의 애니메이션에서 Twist Width 슬라이더를 높이면 효과를 볼 수 있습니다(Twist Width 값은 1에서 1000까지 움직입니다).



Randomly across the surface (왼쪽). In uniform rows and columns (오른쪽).

Around N

도미노 사이에서 교차점이 발견되었다면 도미노의 **Density** 또는 **Size**를 줄이면 됩니다. 또는 **Around N** 값을 조절해 보십시오. 이 속성은 표면 벡터 주위로 아카이브를 회전시킵니다(이 벡터는 Bump, Tilt U, Tilt V 등에 영향을 받지 않음).



-180과 180 사이에서 키프레임화된 'Around N' 값

아래의 이미지는 **Randomly across the surface**와 **In uniform rows and columns**로 생성된 프리미티브가 있는 다양한 **Twist Width** 값을 보여줍니다.

Twist Width: 47



Randomly across the surface

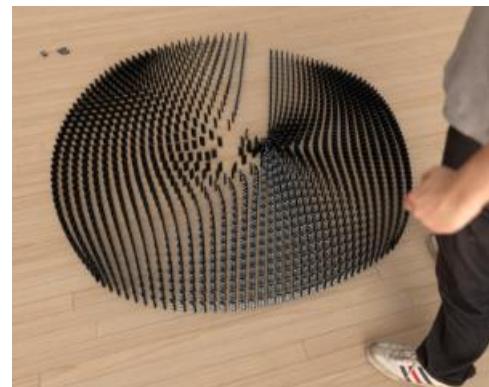


In uniform rows and columns

Twist Width: 87



Randomly across the surface

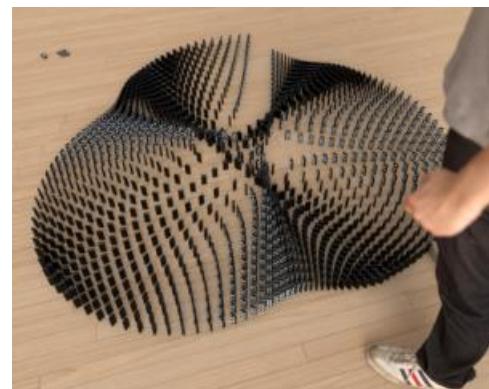


In uniform rows and columns

Twist Width: 205



Randomly across the surface

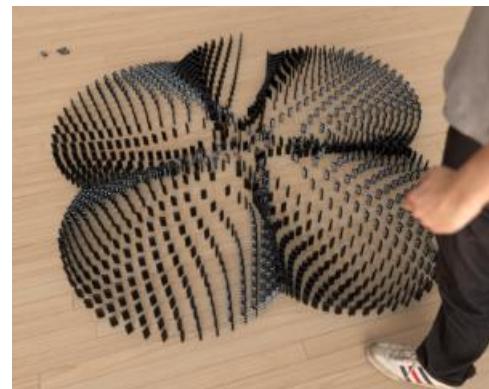


In uniform rows and columns

Twist Width: 313



Randomly across the surface



In uniform rows and columns

Twist Width: 630



Randomly across the surface



In uniform rows and columns

Twist Width: 1000



Randomly across the surface



In uniform rows and columns

Twist Width: 2000



Randomly across the surface



In uniform rows and columns

지금까지 수식을 사용하여 도미노를 배치하는 방법을 설명했습니다. 다양한 값을 사용하여 도미노의 다양한 형태를 만들어보세요. 넘어뜨리지 않게 주의하세요!



<https://youtu.be/046fktemC4>

도미노 아카이브가 넘어지는 효과를 내기 위해 애니메이트됨(Tilt U 및 V 속성) 영상 제공: Pedro Fernando Gómez

색상 지정하기



아래 자습서들은 XGen 프리미티브에 색상을 지정하는 방법에 대해 설명합니다.

- [프리미티브에 임의 색상 지정하기](#)
- [프리미티브 아카이브에 색상 지정하기](#)
- [구체 프리미티브에 색상 지정하기](#)
- [스플라인에 색상 지정하기](#)

프리미티브에 임의 색상 지정하기

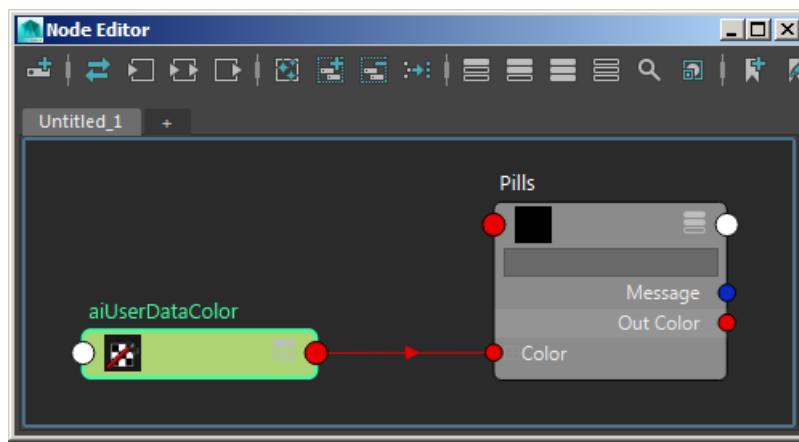


이 자습서에서는 수식과 사용자 데이터 노드의 조합을 사용하여 여러 XGen 프리미티브에 임의의 색상을 지정하는 방법에 대해 설명합니다.

이 자습서를 시작하기 전에, XGen 프리미티브의 스케일을 정의하는 방법을 보여주는 '[‘텍스처 맵으로 프리미티브 제어하기’](#) 자습서를 읽을 것을 권장합니다.

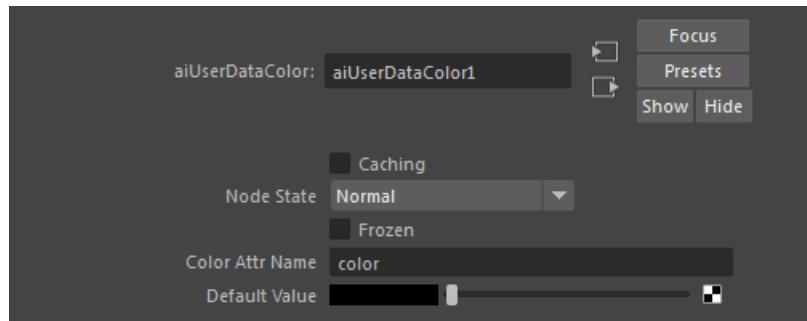
장면 설정

- 우선, 모델을 **XGen 아카이브**로 내보내야합니다.
- XGen 아카이브 설명을 만들고 해당 아카이브 파일을 추가합니다. **Standard Surface** 쉐이더를 이 아카이브 컬렉션에 할당합니다.
- User Data** 노드를 만들고 Standard Surface 쉐이더의 **Color attribtue**에 연결합니다.



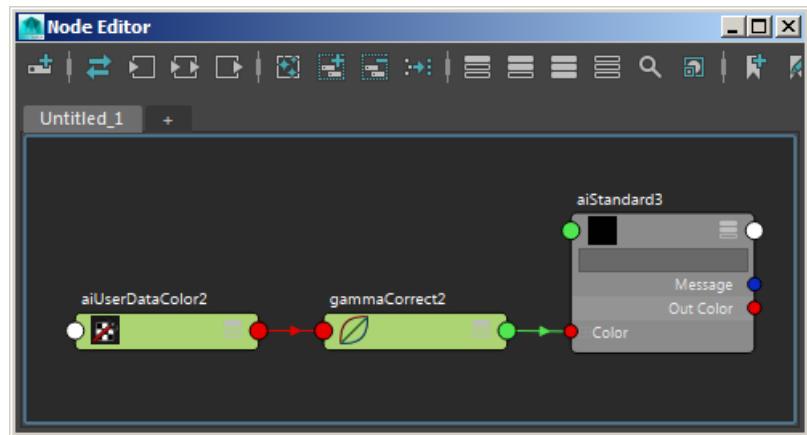
Standard Surface 쉐이더의 Color 속성에 연결된 Ai UserData Color 노드

- Ai UserData 색상 노드의 **Color Atr Name**에 **color**을 입력합니다. XGen 설명에서 동일한 "Color" 이름을 사용하겠습니다.



Ai UserData Color 노드에서 Color Attr Name에 'color'라는 이름을 지정합니다.

- MtoA는 Ai UserData Color 노드를 감마 보정하지 않으므로 아래 그림과 같이 **Ai UserData Color** 노드와 Standard Surface Shader의 **Color attribute** 사이에 **Maya Gamma Correct** 노드를 추가해야 합니다. Gamma RGB 값을 0.454로 변경합니다.



- 장면을 렌더링합니다. **Ai UserData Color Default Value**가 black이기 때문에 알약들이 검은색입니다. XGen의 **Custom Shader Parameters**를 사용하여 XGen 설명에 이를 연결해야 합니다.

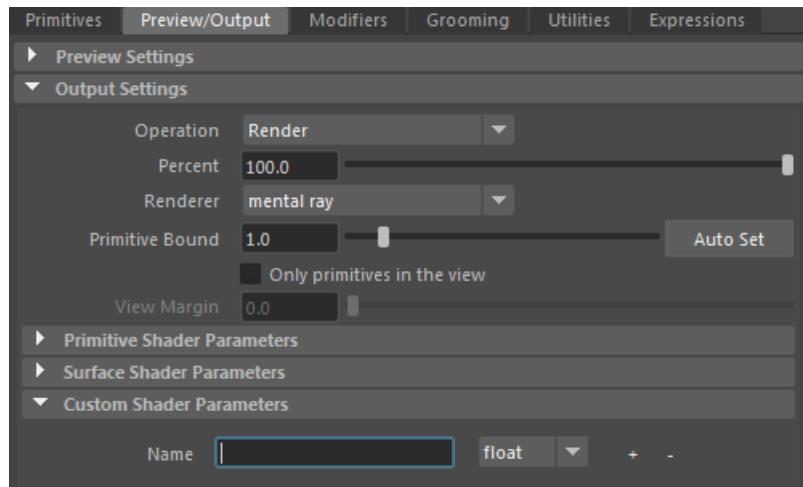


알약 아카이브들은 Ai UserData Color 노드의 기본색인 검정을 사용하고 있습니다.

- Density**를 약 20으로 높이면 XGen 원시 아카이브 알약 모델의 개수가 증가합니다.

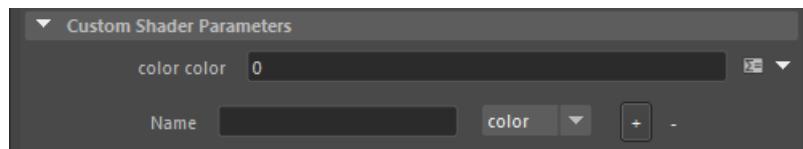
사용자 정의 쉐이더 파라미터

- XGen에서 **Preview/Output** 탭을 클릭하고 **Output Settings**를 엽니다. 그 아래 **Custom Shader Parameters**가 나타납니다. 여기서 식을 추가할 것입니다.



Custom Shader Parameters (아래)

- Name 텍스트 필드에 **Ai UserData Color** 노드가 있는 **Color Attr Name**에 사용한 이름을 입력합니다. 이 경우에는 'color'라는 단어를 사용했습니다. **Float**를 **Color**로 변경하십시오. 이것이 우리가 변경하고자 하는 속성이기 때문입니다.



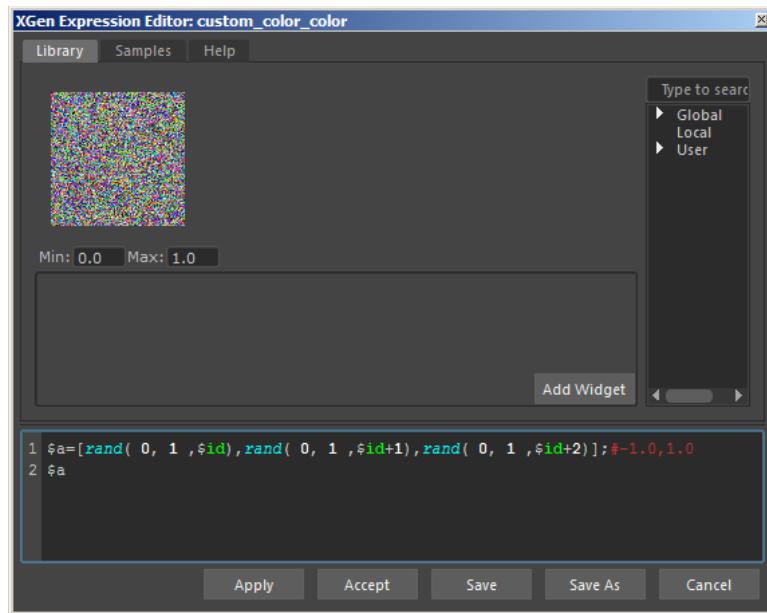
Color로 설정된 Custom Shader Parameters

Expression

- 새로운 'color color' 파라미터의 오른쪽에 있는 **XGen Expression Editor** 아이콘을  클릭하고 **Expression Editor**에 다음 텍스트를 추가하십시오.

```
$a=[rand( 0, 1 , $id),rand( 0, 1 , $id+1),rand( 0, 1 , $id+2)];#-1.0,1.0
$a
```

여기서 **rand**는 0과 1 사이의 RGB 값을 정의합니다.



RGB 값을 0에서 1까지 정의하는 데 사용되는 XGen Expression Editor 안의 수식

- 장면을 렌더링하면 다음과 같습니다.



- rand** 값을 다음으로 변경한다면:

```
$a=[1,rand( 0, 0.5 , $id+1),rand( 0, 0.5 , $id+2)]; #-1.0,1.0
$a
```

RGB 값을 가진 무엇인가가 생성되지만 붉은색은 많아지고 녹색과 청색은 감소합니다.



- rand** 값을 다음으로 변경한다면:

```
$a=[rand( 0.5, 1 , $id),rand( 0, 0.8 , $id+1),rand( 0, 0.8 , $id+2)]; #-1.0,1.0
$a
```

빨간색 값이 0.5와 1 사이이기 때문에 알약의 색상이 보다 다양합니다.

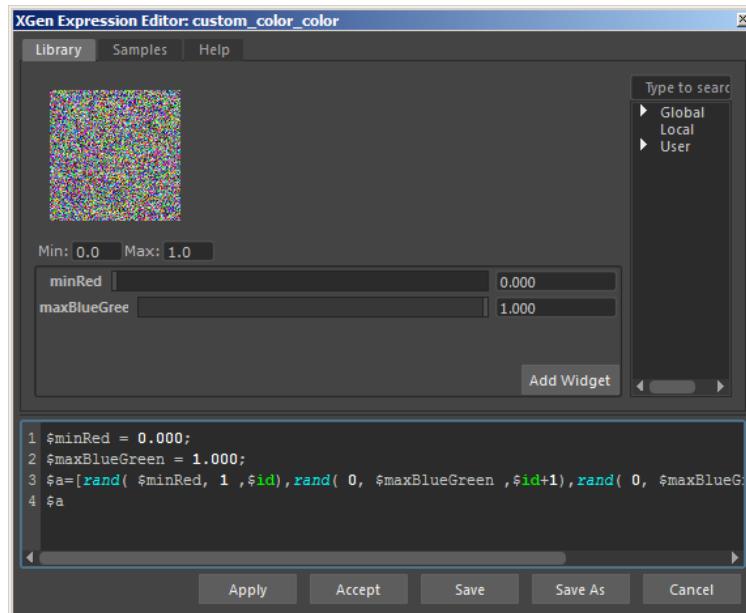


Sliders

- XGen Expression Editor에서 값을 입력하지 않고 RGB 색상을 보다 쉽게 조정할 수 있는 슬라이더를 만들 수 있습니다. XGen Expression Editor에서 다음 텍스트를 추가합니다.

```
$minRed = 0.000;
$maxBlueGreen = 1.000;
$a=[rand( $minRed, 1 , $id),rand( 0, $maxBlueGreen , $id+1),rand( 0, $maxBlueGreen , $id+2)];#-1.0,1.0
$a
```

두 개의 슬라이더가 나타납니다. 이들을 사용하여 생성된 색상들을 대화식으로 미리볼 수 있습니다. 따라서 이제 변경 사항이 있을 때마다 장면을 테스트 렌더링할 필요가 없습니다!



새로운 슬라이더들이 있는 XGen Expression Editor

프리미티브 아카이브에 색상 지정하기



이 자습서에서는 텍스처 맵과 함께 **User Data** 노드를 사용하여 XGen 프리미티브에 색상을 적용하는 방법에 대해 설명합니다. 그 결과는 **Pointillism**이라는 기술과 유사합니다. 여기서는 작은 컬러 점이 패턴으로 적용되어 이미지를 형성합니다. 실제 세계에서 영감을 얻으려면 이 아티스트와 이 아티스트의 작품 중 일부뿐만 아니라 많은 작은 물체를 사용하여 큰 예술 작품을 만드는 이 아티스트도 살펴봐야 합니다.

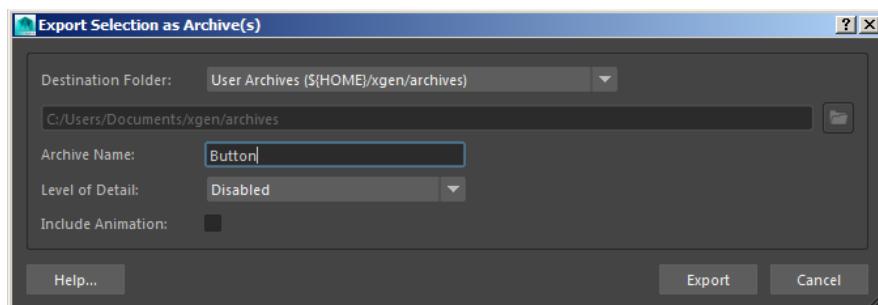
이 자습서에 사용된 버튼 모델은 하단 사이트에서 다운로드할 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Assigning+Color+to+Primitive+Archives>



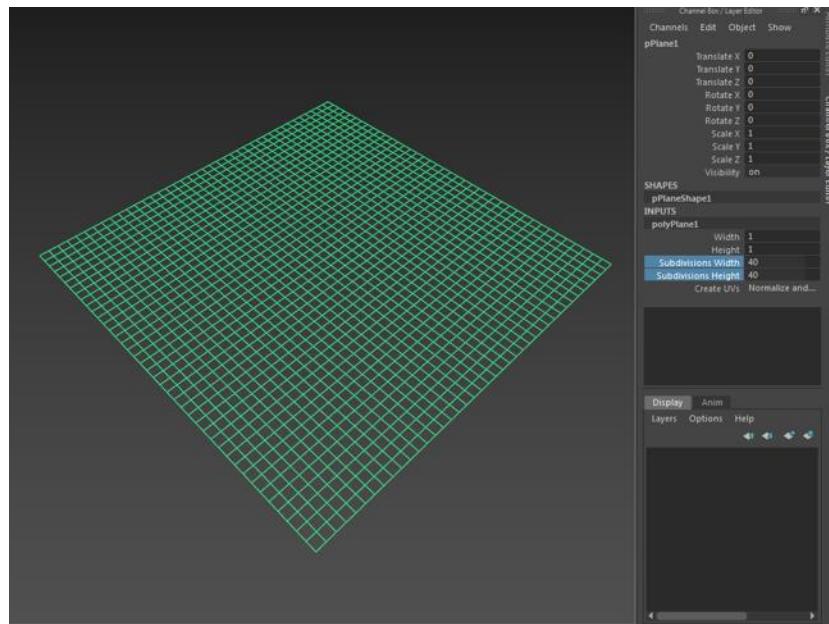
이 자습서를 시작하기 전에, XGen 프리미티브의 스케일을 정의하는 방법을 보여주는 “[텍스처 맵으로 프리미티브 제어하기](#)” 자습서를 읽을 것을 권장합니다.

- 먼저 위 링크에서 제공되는 대로 Maya에서 버튼 모델을 엽니다. 해당 버튼 지오메트리를 선택하고 **Generate>Export Selection as Archive(s)...**로 이동합니다. 아카이브 이름과 대상 폴더를 선택하고 **Export**를 선택합니다.

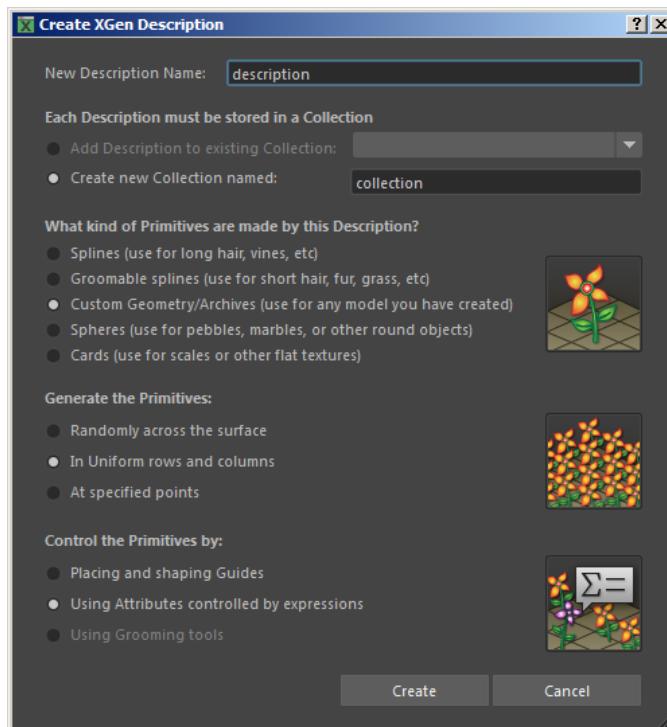


- 폴리곤 평면을 사용하여 XGen 설명을 할당하겠습니다. 먼저 폴리곤 평면을 생성하고 **Subdivisions Width** 및 **Subdivisions Height**를 40으로 증가시킵니다. 우리는 프리미티브를 'In uniform rows and

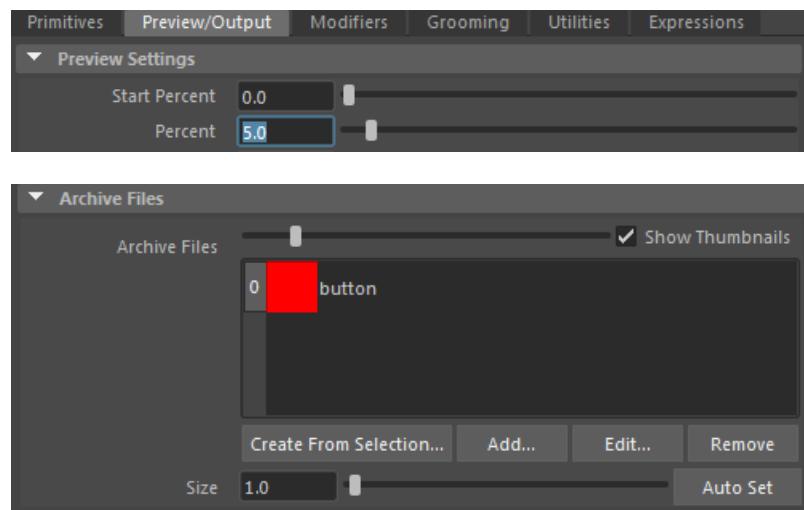
columns'에 정렬할 것이므로 충분한 수의 구획이 필요합니다(XGen은 각 폴리곤 면의 중심에 프리미티브를 배치합니다).



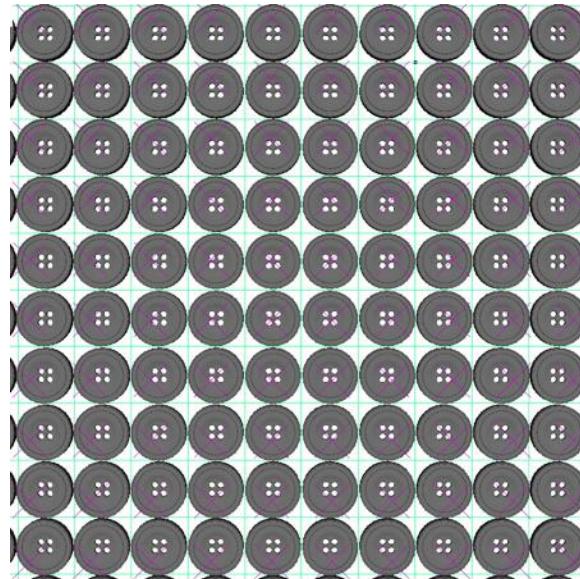
- 해당 폴리곤 평면을 선택하고 **XGen>Create Description...**으로 이동합니다. 그러면 아래와 같이 **Create XGen Description**이 열립니다.
- Custom Geometry/Archives** 프리미티브를 선택합니다. **Generate the Primitives**를 *In Uniform rows and Columns*로 변경한 다음 **Create**를 클릭합니다.



- 계속 진행하기 전에 **Preview/Output** 탭에서 **Percent** 값을 낮추는 것이 좋습니다. **Percent** 값이 너무 높으면 생성된 많은 수의 프리미티브로 인해 컴퓨터가 멈출 수도 있습니다.

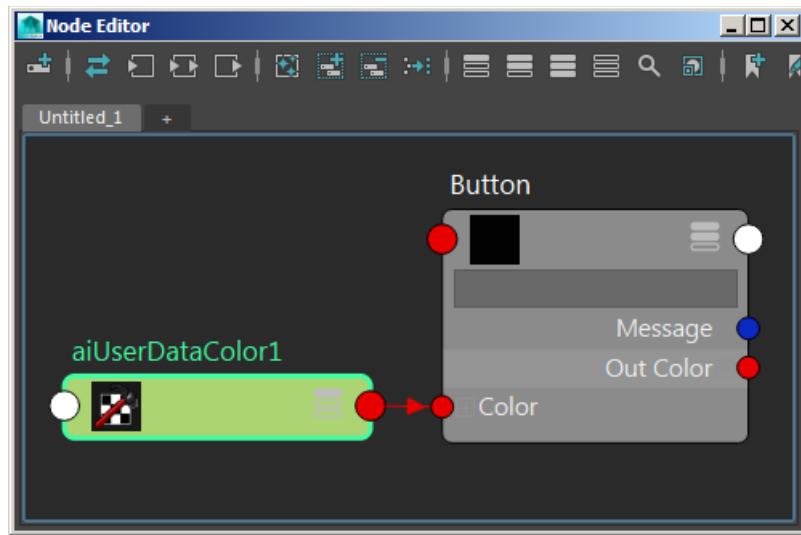


- 버튼들이 거의 맞닿을 때까지 **Spacing**과 **Size**를 조절합니다. 이 경우에는 **Spacing**은 1, **Size**는 1230입니다.



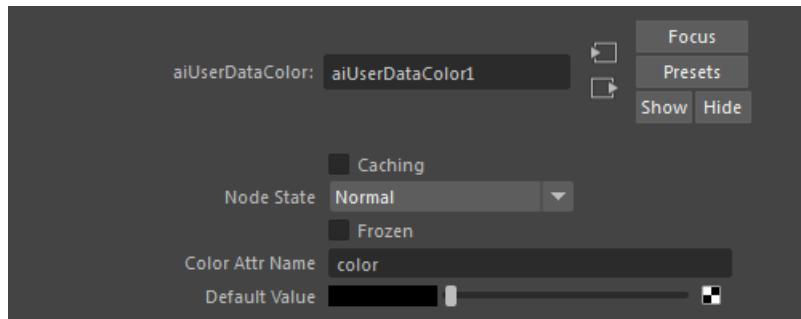
버튼들이 거의 닿도록 **Size**와 **Spacing**을 조절합니다.

- Standard Surface** 쉐이더를 새로운 XGen 컬렉션에 할당합니다.
- Ai UserData Color** 노드를 만들고 이것을 **Standard Surface** 쉐이더의 **Color** 속성에 연결합니다.



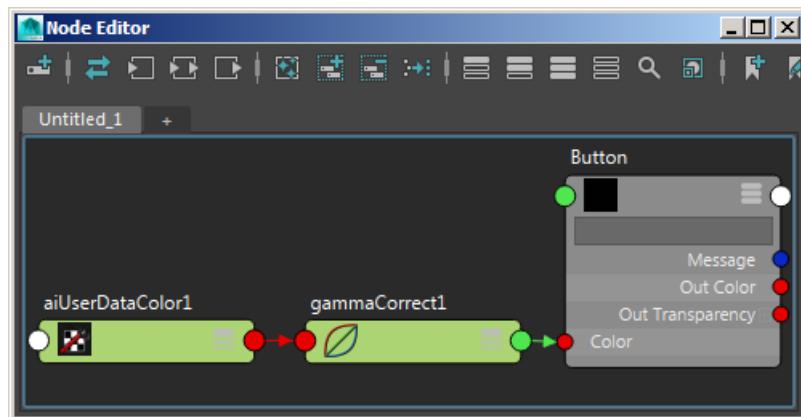
Standard Surface 쉐이더의 Color 속성에 연결된 Ai UserData Color 노드

- Ai UserData 색상 노드의 **Color Attr Name**에 **color**을 입력합니다. XGen 설명에서 동일한 'Color' 이름을 사용하겠습니다.

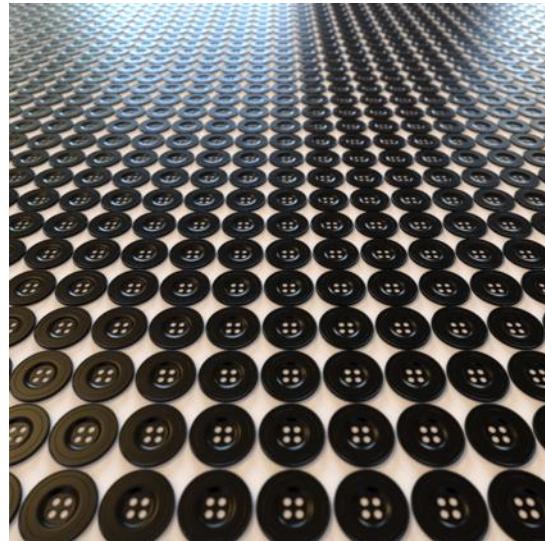


Ai UserData Color 노드에서 Color Attr Name에 'color'라는 이름을 지정합니다.

- MtoA는 Ai UserData Color 노드를 감마 보정하지 않으므로 아래 그림과 같이 **Ai UserData Color** 노드와 Standard Surface Shader의 **Color** 속성 사이에 **Maya Gamma Correct** 노드를 추가해야 합니다. Gamma RGB 값을 0.454로 변경합니다.



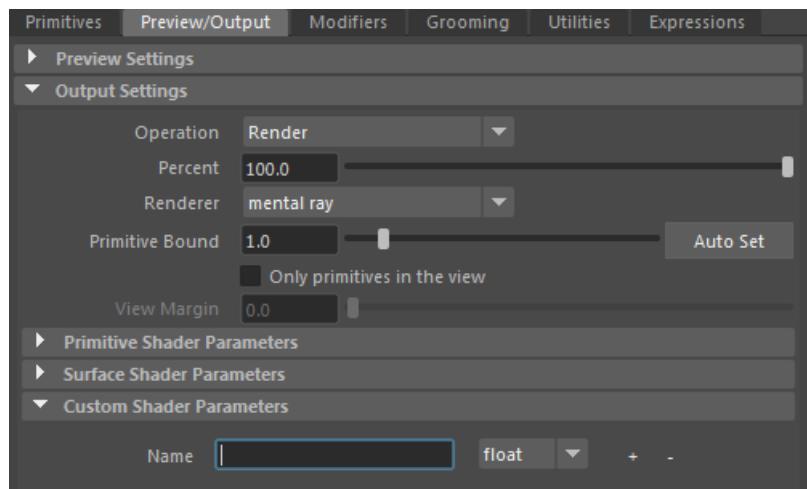
- 장면을 렌더링합니다. **Ai UserData Color Default Value**가 black이기 때문에 버튼들이 검은색입니다. XGen의 **Custom Shader Parameters**를 사용하여 XGen 설명에 이를 연결해야 합니다.



버튼 아카이브 프리미티브들은 Ai UserData Color 노드의 기본색인 검정을 사용하고 있습니다.

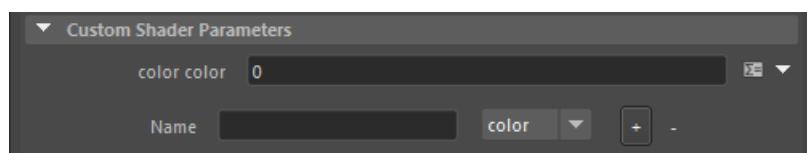
사용자 정의 쉐이더 파라미터

- XGen에서 **Preview/Output** 탭을 클릭하고 **Output Settings**를 엽니다. 그 아래 **Custom Shader Parameters**가 나타납니다.



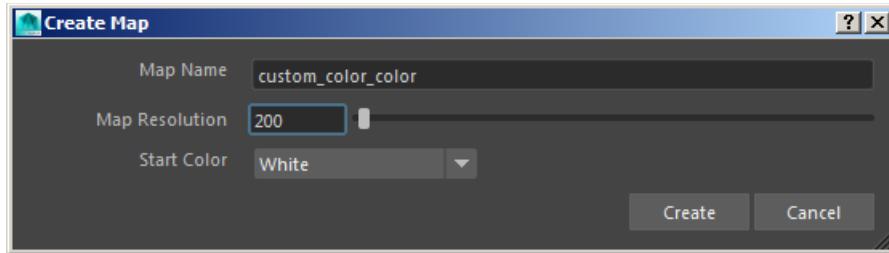
사용자 정의 쉐이더 파라미터

- Name** 텍스트 필드에 **Ai UserData Color** 노드가 있는 **Color Attr Name**에 사용한 이름을 입력합니다. 이 경우에는 'color'라는 단어를 사용했습니다. **Float**를 클릭하고 **Color**로 변경합니다. 이것이 변경하고자 하는 속성이기 때문입니다.

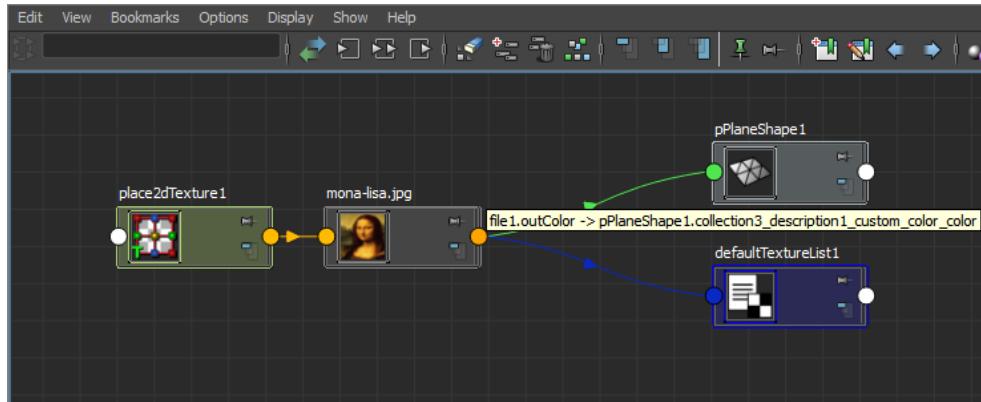


Color로 설정된 Custom Shader Parameters

- 해당 색상 텍스트 필드 오른쪽에 있는 아래 방향 화살표를 클릭합니다. **Create Map...**을 선택합니다(이것은 평면에 Maya 쉐이더가 할당되어 있는 경우에만 작동한다는 점을 유의하십시오). **Map Resolution**을 높입니다.

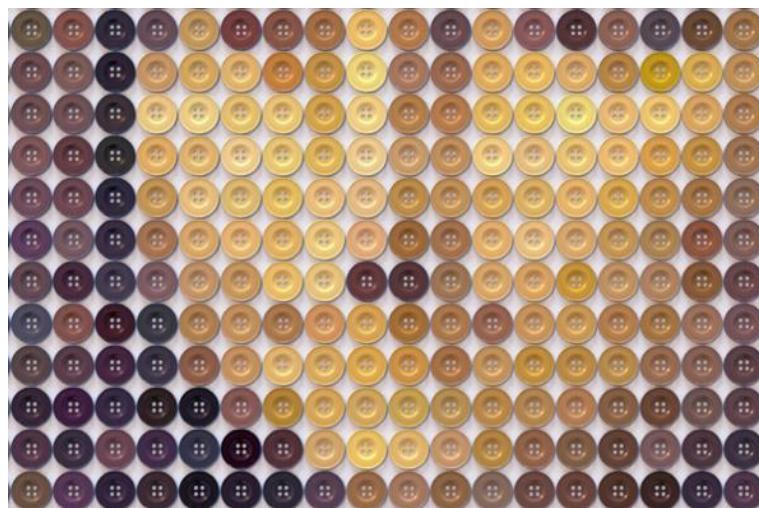


- Hypershade 및 Node Editor 창에서 파일 텍스처가 해당 평면에 연결되어 있어야 합니다. 해당 파일 텍스처를 선택하고 버튼 아카이브 프리미티브의 색상을 구동하는 데 사용할 텍스처 맵을 엽니다.



프리미티브 속성을 정의하는 데 사용되는 파일 텍스처를 업데이트한 후 디스크 아이콘을 선택해야 합니다. 그렇지 않으면 XGen이 디스크에 ptex 맵을 굽지 않으며 렌더링이 진행되지 않습니다.

- 장면을 렌더링합니다. 기본 아카이브는 파일 텍스처 맵과 동일한 색상이어야 합니다. 이제 완료되었습니다!



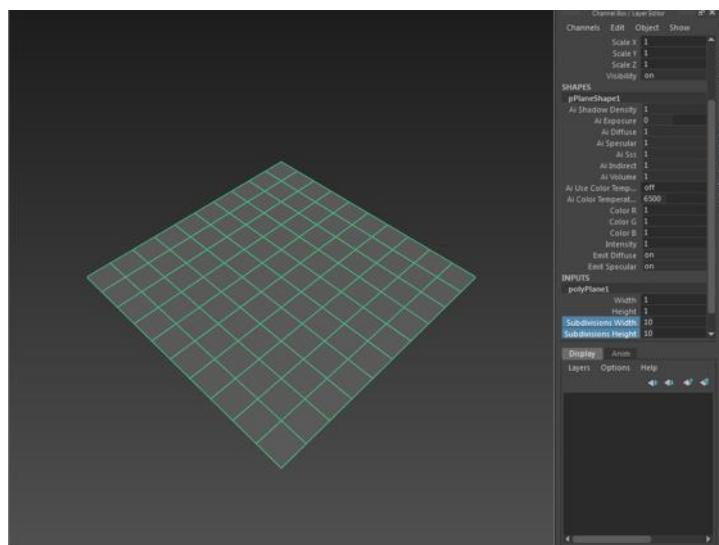
구체 프리미티브에 색상 지정하기



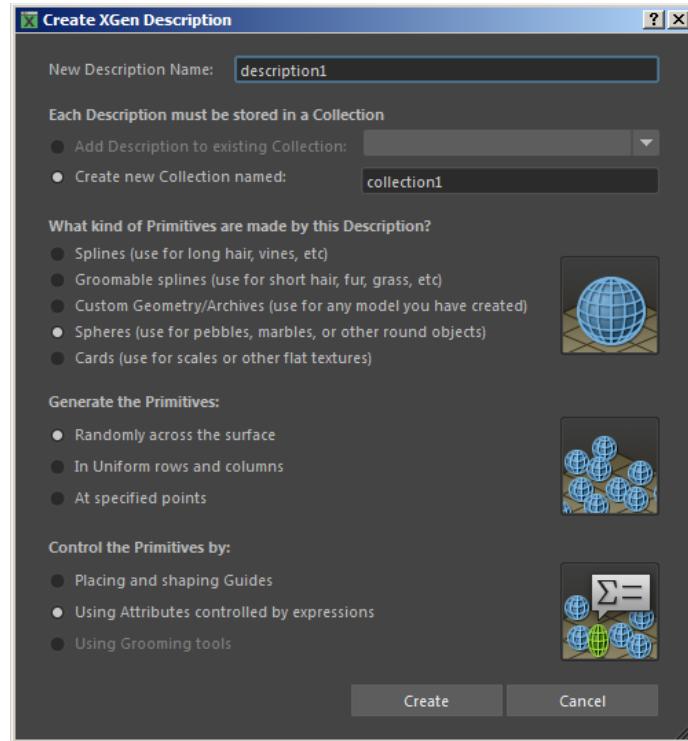
이 자습서에서는 텍스처 맵과 함께 사용자 데이터 노드를 사용하여 XGen 프리미티브에 색상을 적용하는 방법에 대해 설명합니다. 그 결과는 **Pointillism**이라는 기술과 유사합니다. 여기서는 작은 컬러 점이 패턴으로 적용되어 이미지를 형성합니다.

이 자습서를 시작하기 전에, XGen 프리미티브의 스케일을 정의하는 방법을 보여주는 “[텍스처 맵으로 프리미티브 제어하기](#)” 자습서를 읽을 것을 권장합니다.

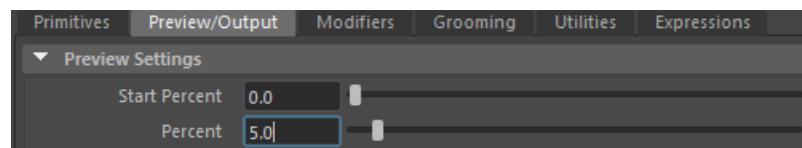
- 먼저 폴리곤 평면을 만들겠습니다. **Subdivisions Width** 및 **Subdivisions Height**를 **10**으로 늘립니다. 이 평면을 사용하여 XGen 설명을 할당할 것입니다.



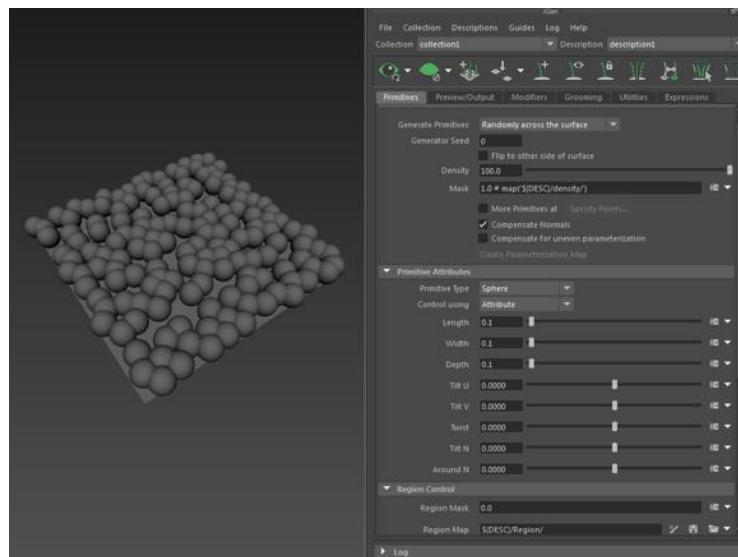
- 해당 폴리곤 평면을 선택하고 **Generate>Create Description**으로 이동합니다. 그러면 아래와 같이 **Create XGen Description**이 열립니다.
- Spheres** primitive를 선택합니다. 다른 기본 설정은 그대로 두고 **Create**를 클릭합니다.



- 계속 진행하기 전에 **Preview/Output** 탭에서 **Percent** 값을 낮추는 것이 좋습니다. **Percent** 값이 너무 높으면 생성된 많은 수의 프리미티브로 인해 컴퓨터가 멈출 수도 있습니다.

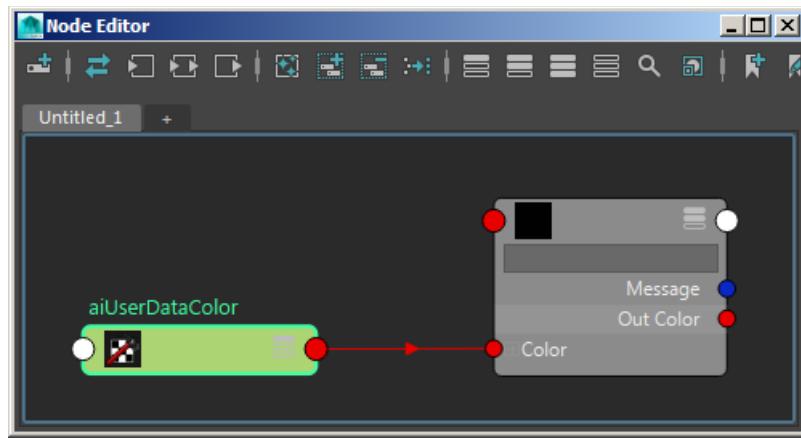


- Density**를 100 정도로 높이고 Sphere 프리미티브의 **Length**, **Width**, **Depth**를 0.1 정도로 낮춥니다.



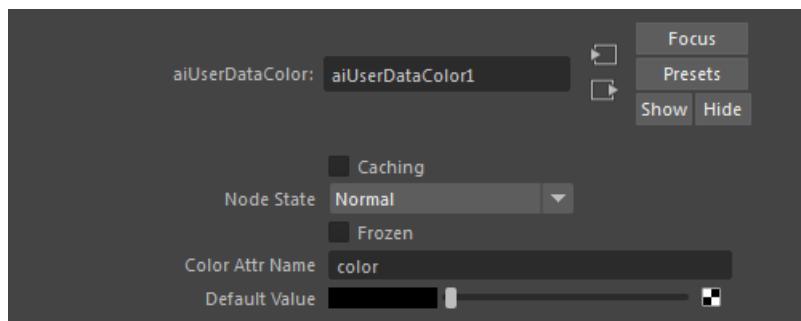
Sphere primitive Density는 100으로 Length, Width, Depth는 0.1로 설정됨

- Standard Surface 쉐이더를 새로운 XGen 컬렉션에 할당합니다.
- Ai UserData Color** 노드를 만들고 이것을 Standard Surface 쉐이더의 **Color** 속성에 연결합니다.



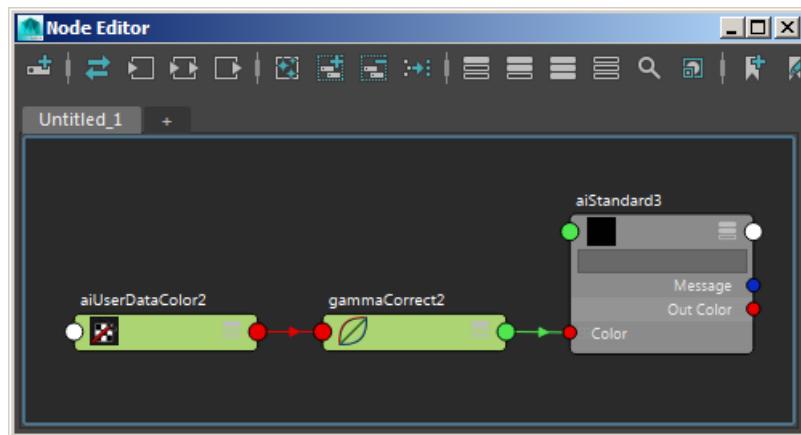
Standard Surface 쉐이더의 Color 속성에 연결된 Ai UserData Color 노드

- Ai UserData 색상 노드의 **Color Attr Name**에 **Color**을 입력합니다. XGen 설명에서 동일한 '**Color**' 이름을 사용하겠습니다.

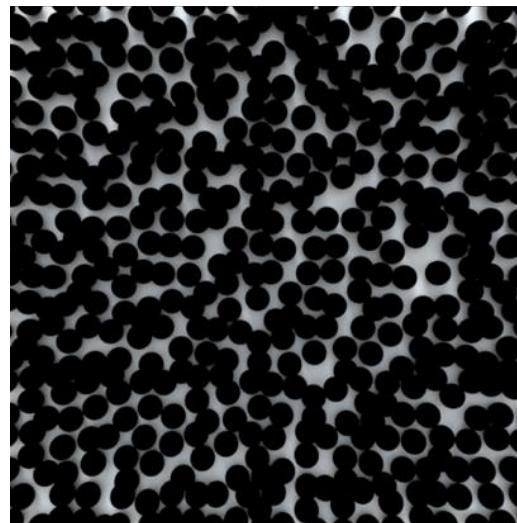


Ai UserData Color 노드에서 Color Attr Name에 'color'라는 이름을 지정합니다.

- MtoA는 Ai UserData Color 노드를 감마 보정하지 않으므로 아래 그림과 같이 **Ai UserData Color** 노드와 Standard Surface Shader의 **Color** 속성 사이에 **Maya Gamma Correct** 노드를 추가해야 합니다. Gamma RGB 값을 0.454로 변경합니다.



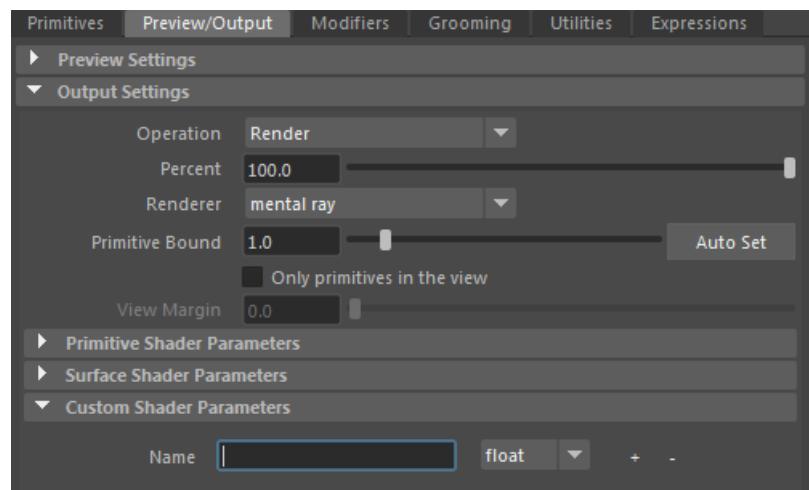
- 장면을 렌더링합니다. Ai UserData Color **Default Value**가 black이기 때문에 구체들이 검은색입니다. XGen의 **Custom Shader Parameters**를 사용하여 XGen 설명에 이를 연결해야 합니다.



구체 프리미티브들은 Ai UserData Color 노드의 기본색인 검정을 사용하고 있습니다.

사용자 정의 쉐이더 파라미터

- XGen에서 **Preview/Output** 탭을 클릭하고 **Output Settings**를 엽니다. 그 아래 **Custom Shader Parameters**가 나타납니다. 여기서 식을 추가할 것입니다.



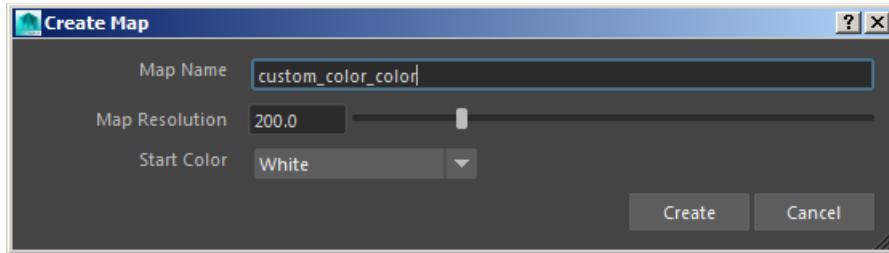
사용자 정의 쉐이더 파라미터

- Name** 텍스트 필드에 **Ai UserData Color** 노드가 있는 **Color Attr Name**에 사용한 이름을 입력합니다. 이 경우에는 'color'라는 단어를 사용했습니다. **Float**를 클릭하고 **Color**로 변경합니다. 이것이 변경하고자 하는 속성이기 때문입니다.

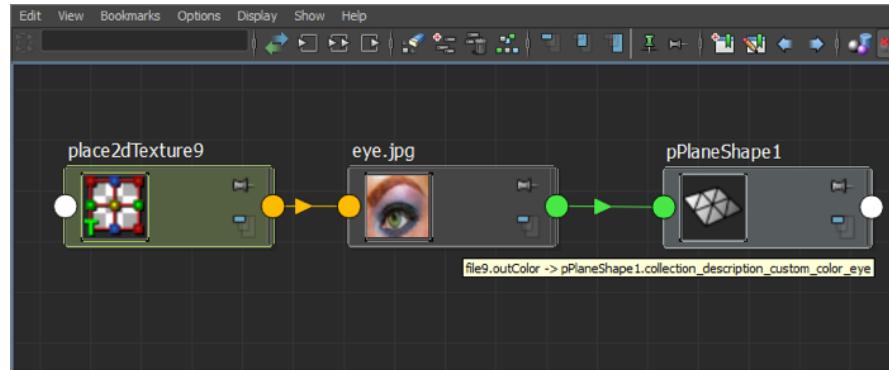


Color로 설정된 Custom Shader Parameters

- 해당 색상 텍스트 필드 오른쪽에 있는 아래 방향 화살표를 클릭합니다. **Create Map...**을 선택합니다. (이것은 평면에 Maya 쉐이더가 할당되어 있는 경우에만 작동한다는 점을 유의하십시오). **Map Resolution**을 200 정도로 높입니다.



- Hypershade 및 Node Editor 창에서 파일 텍스처가 해당 평면에 연결되어 있어야 합니다. 파일 텍스처를 선택하고 구체 프리미티브의 색상을 구동하는 데 사용할 텍스처 맵을 엽니다.



프리미티브 속성을 정의하는 데 사용되는 파일 텍스처를 업데이트한 후 디스크 아이콘을  선택해야 합니다. 그렇지 않으면 XGen이 디스크에 ptex 맵을 굽지 않으며 렌더링이 진행되지 않습니다.

- 장면을 렌더링합니다. 구체 아카이브는 이제 파일 텍스처 맵과 동일한 색상이어야 합니다. 이제 Sphere 프리미티브의 크기와 밀도를 조정하면 모두 완료됩니다!

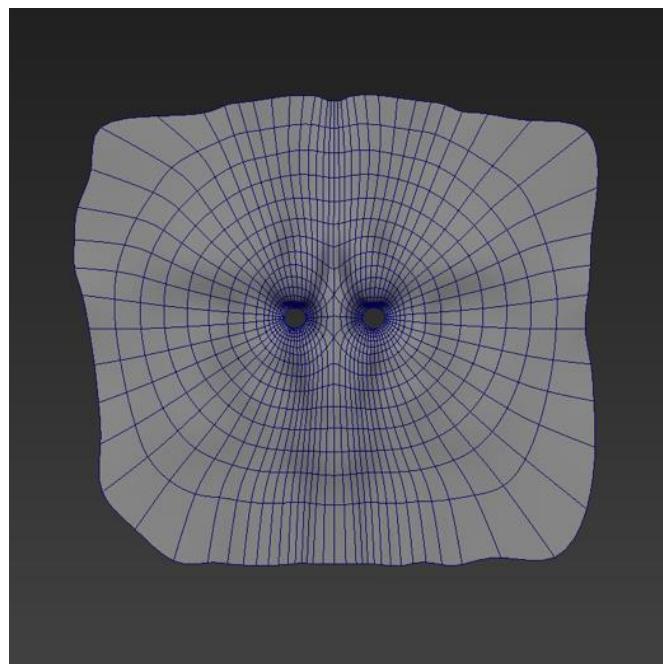


스플라인에 색상 지정하기



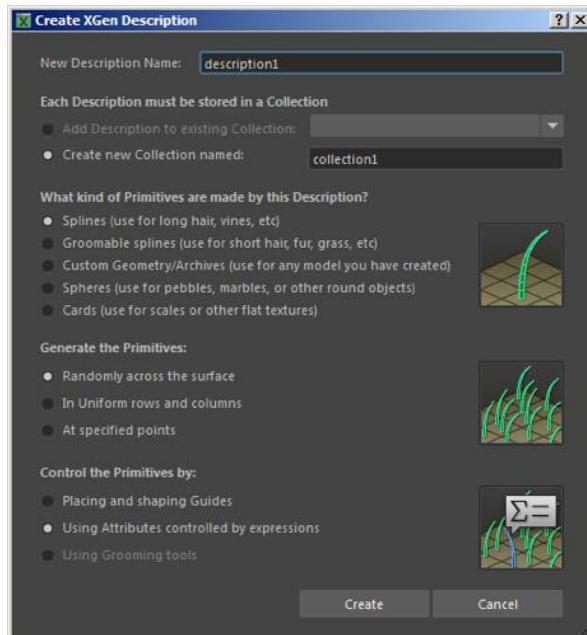
이 자습서에서는 텍스처 맵과 함께 User Data 노드를 사용하여 XGen 스플라인에 색상을 적용하는 방법에 대해 설명합니다. XGen 설명을 위한 기반으로 간단한 폴리곤 메시를 사용합니다.

- 먼저 첨부한 Maya scene 파일을 엽니다. 이것은 XGen 설명을 할당하는 데 사용하게 될 폴리곤 메시로 구성됩니다.

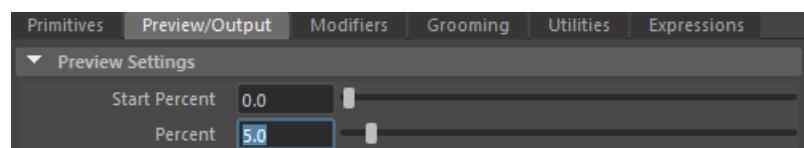


- 폴리곤 평면을 선택하고 **Generate> Create Description...**으로 이동합니다. 그러면 아래와 같이 **Create XGen Description**이 열립니다.
- Splines**를 선택합니다(긴 머리카락, 덩굴 등에 사용). 다른 기본 설정은 그대로 두고 **Create**를

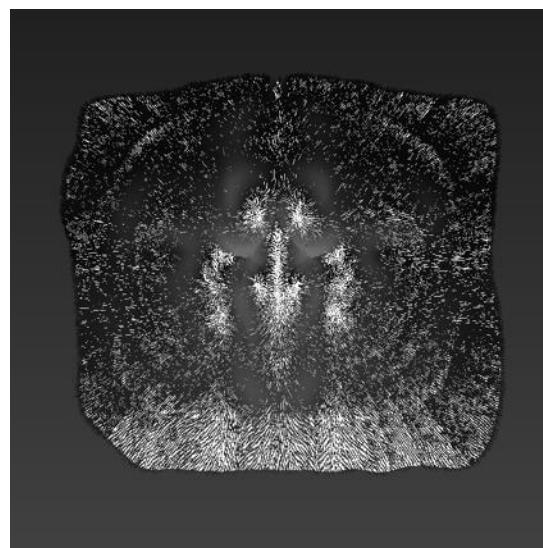
클릭합니다.



- 계속 진행하기 전에 **Preview/Output** 탭에서 **Percent** 값을 낮추는 것이 좋습니다. **Percent** 값이 너무 높으면 생성된 많은 수의 프리미티브로 인해 컴퓨터가 멈출 수도 있습니다.

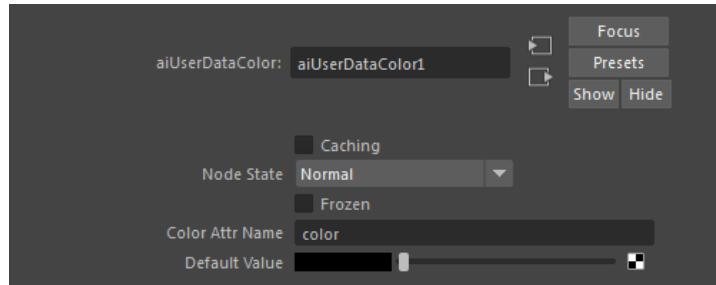


- Density**를 30 정도로 높입니다.



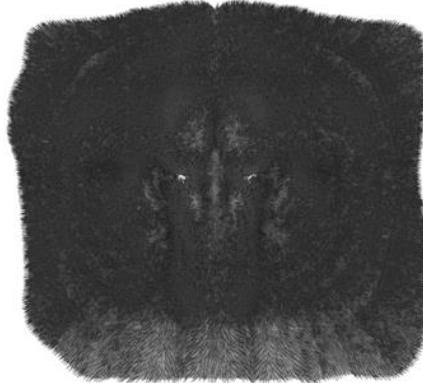
Spline Density가 30으로 설정됨

- Standard Hair 쉐이더를 새로운 XGen 컬렉션에 할당합니다.
- Ai UserData Color** 노드를 만들고 이것을 Standard Hair 쉐이더의 **Tip Color** 속성에 연결합니다.
- Ai UserData 색상 노드의 **Color Attr Name**에 **color**을 입력합니다. XGen 설명에서 동일한 '**Color**' 이름을 사용하겠습니다.



Ai UserData Color 노드에서 Color Attr Name에 'color'라는 이름을 지정합니다.

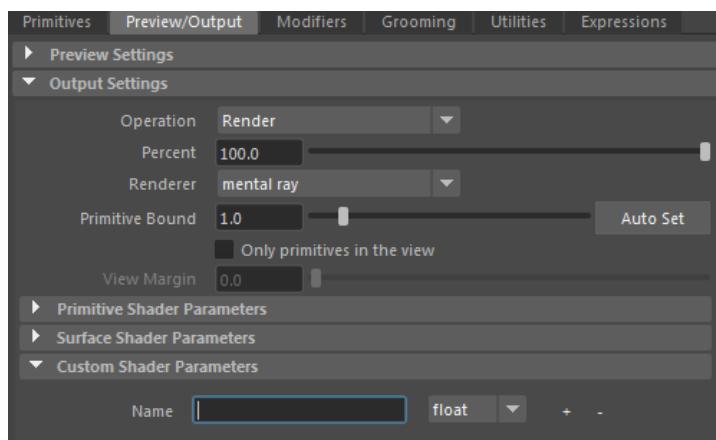
- MtoA는 Ai UserData Color 노드를 감마 보정하지 않으므로 **Ai UserData Color** 노드와 Standard Hair Shader의 **Tip Color** 속성 사이에 **Maya Gamma Correct** 노드를 추가해야 합니다.
- 장면을 렌더링합니다. **Ai UserData Color Default Value**가 black이기 때문에 구체들이 검은색입니다. XGen의 **Custom Shader Parameters**를 사용하여 XGen 설명에 이를 연결해야 합니다.



스플라인들은 Ai UserData Color 노드의 기본색인 검정을 사용하고 있습니다.

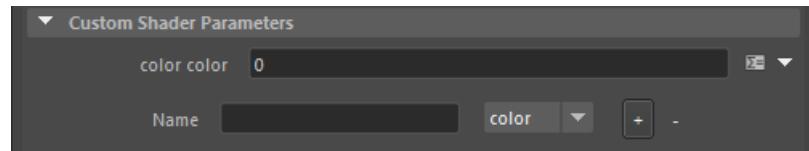
사용자 정의 쉐이더 파라미터

- XGen에서 **Preview/Output** 탭을 클릭하고 **Output Settings**를 엽니다. 그 아래 **Custom Shader Parameters**가 나타납니다. 여기서 식을 추가할 것입니다.



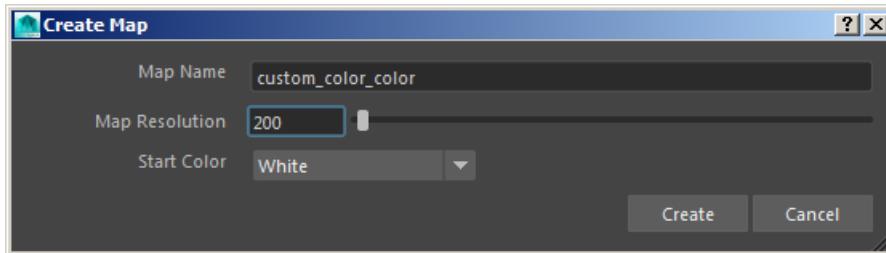
사용자 정의 쉐이더 파라미터

- Name** 텍스트 필드에 **Ai UserData Color** 노드가 있는 **Color Attr Name**에 사용한 이름을 입력합니다. 이 경우에는 'color'라는 단어를 사용했습니다. **Float**를 클릭하고 **Color**로 변경합니다. 이것이 변경하고자 하는 속성이기 때문입니다.

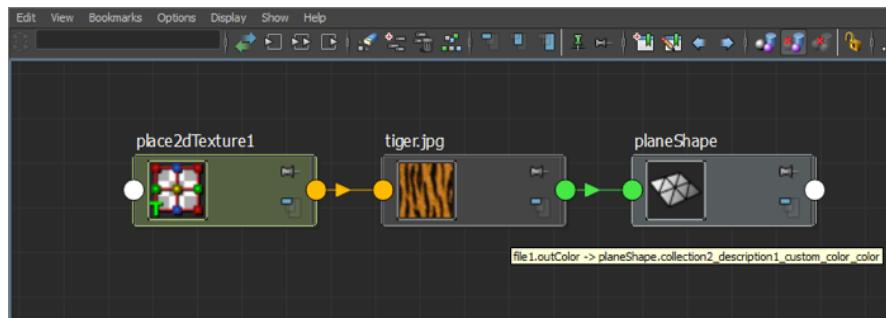


Color로 설정된 Custom Shader Parameters

- 해당 색상 텍스트 필드 오른쪽에 있는 아래 방향 화살표를 클릭합니다. **Create Map...**을 선택합니다. (이것은 평면에 Maya 쉐이더가 할당되어 있는 경우에만 작동한다는 점을 유의하십시오). Map Resolution을 200 정도로 높입니다.



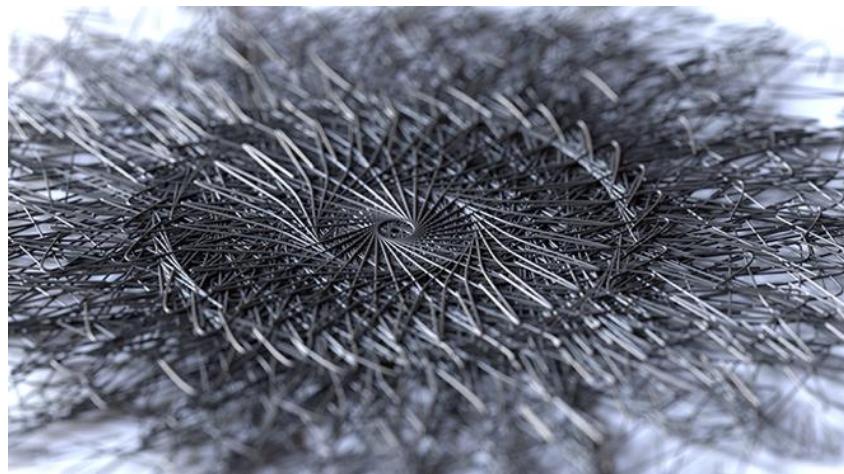
- Hypershade 및 Node Editor 창에서 파일 텍스처가 해당 지오메트리에 연결되어 있어야 합니다. 파일 텍스처를 선택하고 구체 프리미티브의 색상을 구동하는 데 사용할 텍스쳐 맵을 엽니다.



해당 파일 텍스처를 업데이트한 후 디스크 아이콘을 선택해야 합니다. 그렇지 않으면 XGen이 디스크에 ptex 맵을 굽지 않으며 렌더링이 진행되지 않습니다.

- 장면을 렌더링합니다. 스플라인들이 이제 파일 텍스처 맵과 동일한 색상이어야 합니다. 이제 완료되었습니다!

스플라인 (Splines)



- [스플라인으로 잔디 만들기](#)
- [접프하는 거미](#)
- [XGen 스플라인 패턴](#)

스플라인으로 잔디 만들기



이 자습서에서는 XGen 스플라인을 사용하여 잔디밭을 매우 쉽고 빠르게 만드는 방법을 보여줍니다. 또한 잔디의 동작과 모습을 제어하기 위해 몇 가지 수식을 사용할 것입니다.

이 자습서는 다음 장들로 구성되어 있습니다.

잔디 만들기

잔디 굽히기

수식을 사용하여 잔디 색상을 무작위로 변경하기

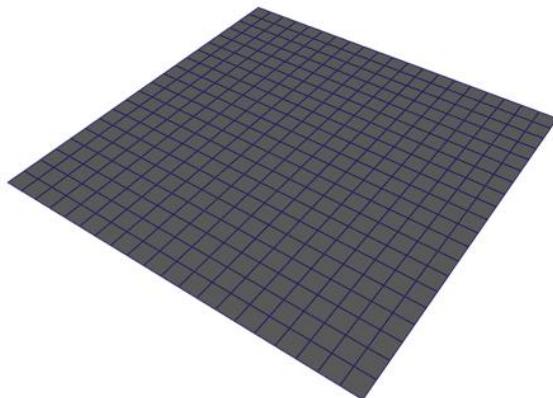
수식으로 길이 변경하기

이 장면을 설정하는 방법을 보여주는 영상과 Maya 파일은 하단 링크의 사이트에서 볼 수 있습니다.

<https://support.solidangle.com/display/A5AFMUG/Creating+Grass+with+Splines>

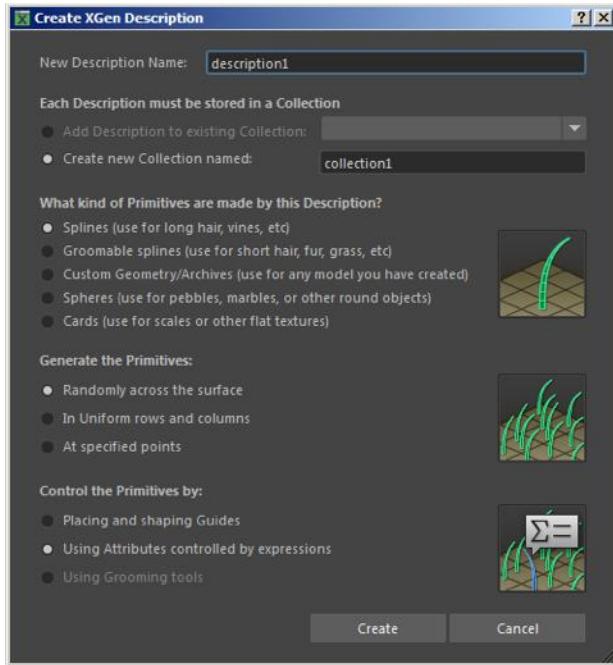
잔디 만들기

- **Height**와 **Width**가 모두 **20**개의 구획으로 이루어진 폴리곤 평면을 만듭니다. 이것은 평면 안에서 잔디를 위한 다양한 기반이 될 수 있습니다. 평면을 **X**와 **Z**에서 최대 **10** 정도로 지정합니다.

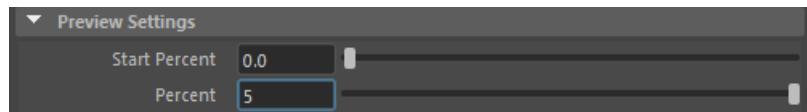


- 해당 폴리곤 평면을 선택하고 **XGen>Create Description...**으로 이동합니다. 그러면 아래와 같이 **Create XGen Description**이 열립니다.

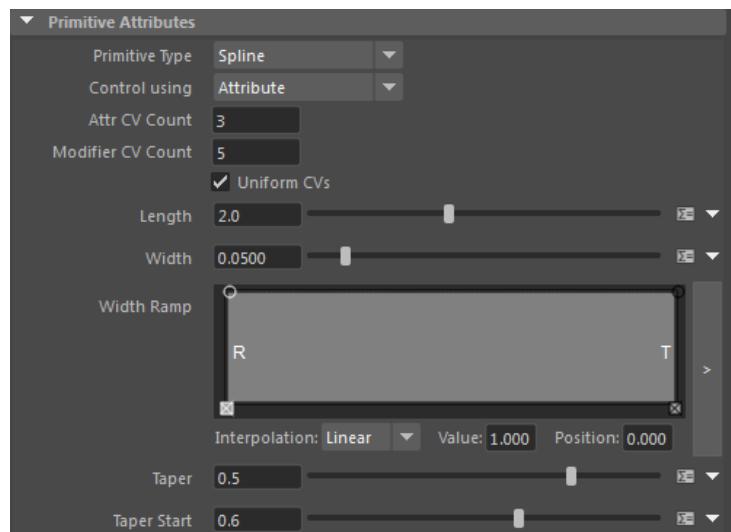
- Splines를 선택합니다(긴 머리카락, 덩굴 등에 사용). 다른 기본 설정은 그대로 두고 Create를 클릭합니다.



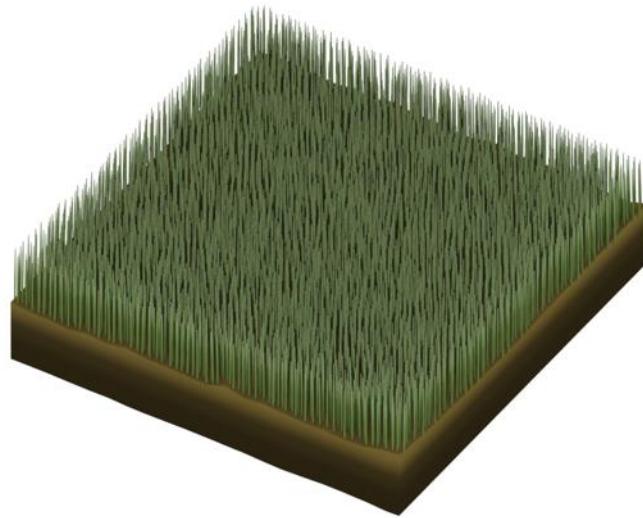
- 계속 진행하기 전에 Preview/Output 탭에서 **Percent** 값을 낮추는 것이 좋습니다. **Percent** 값이 너무 높으면 생성된 많은 수의 프리미티브로 인해 컴퓨터가 멈출 수도 있습니다.



- 평면을 잔디로 덮으려면 Density를 5000 정도로 높입니다. Width를 0.05 정도로 낮추면 잔디가 얇아집니다. spline Length를 2로 높입니다. Taper를 0.5로, Taper Start를 0.5로 조절하여 스플라인을 보다 풀잎처럼 보이게 합니다.



- Standard Surface 쉐이더를 XGen 설명에 할당합니다. **Base Color**를 녹색으로 변경합니다. Skydome 조명을 만들고 장면을 렌더링합니다.



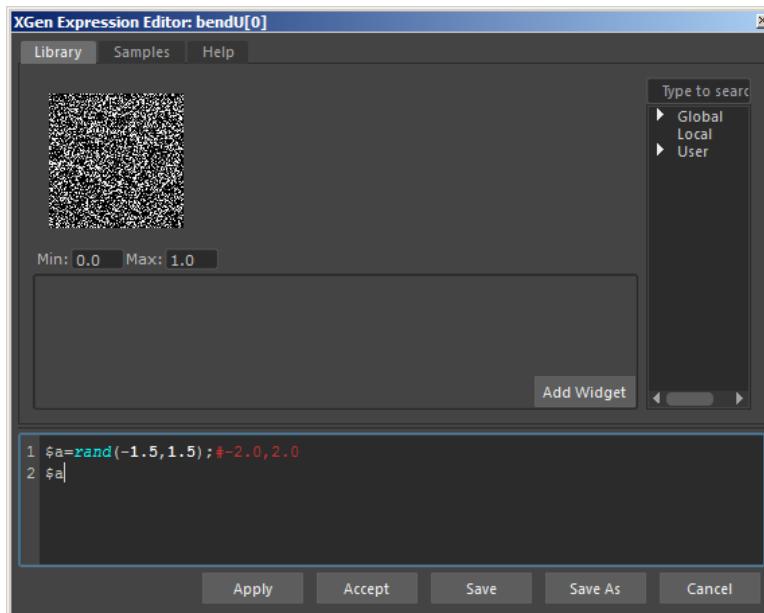
XGen 잔디 스플라인이 있는 평면(평면 아래에 램프가 배치된 큐브).

잔디 굽히기

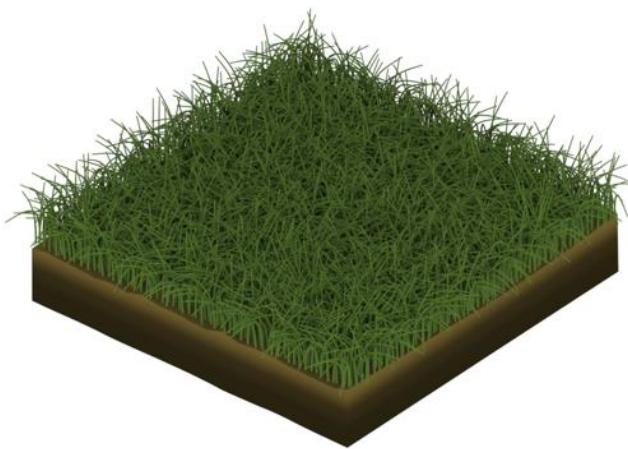
잔디를 한 방향으로 균일하게 구부리기보다는 수식을 사용하여 임의의 방향으로 구부릴 수 있습니다. Bend U와 V는 해당 CV에서 스플라인을 얼마나 구부릴 것인지를 제어합니다. 0은 굽힘이 없는 것이고 1.0은 90도 굽힘입니다.

Bend U[0] 및 **Bend V[0]** 옆에 있는 Expression 아이콘을  클릭하고 XGen Expression Editor에서 다음 텍스트를 입력합니다.

```
$a=rand(-1.5,1.5);#-2.0,2.0
$a
```



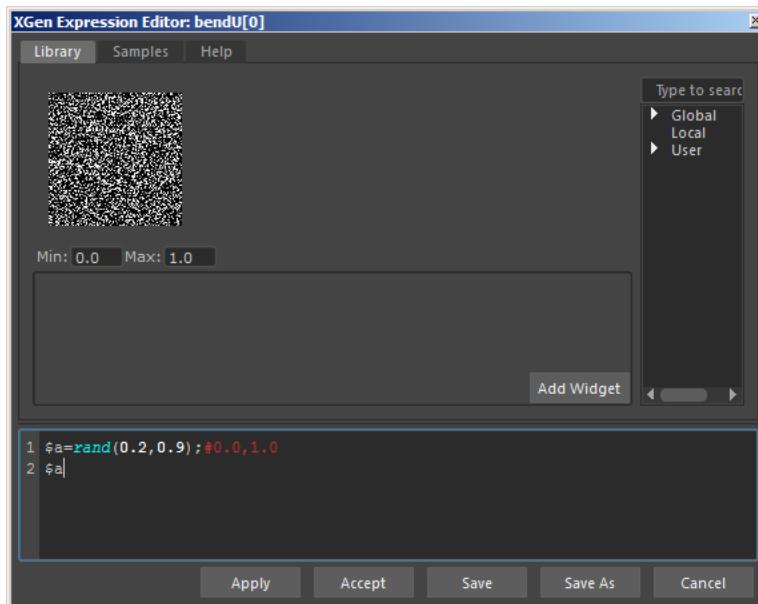
잔디 스플라인은 무작위적으로 굽혀져 보여야 합니다.



잔디가 여전히 약간 균일해 보입니다. **Bend Param[0]**을 사용하여 다양성을 추가하겠습니다. 이 속성은 굽힘이 발생할 스플라인 위치를 제어합니다. 이 값은 0.0(스플라인의 기준)에서 1.0까지(끝)입니다.

- **Bend U[0]** 옆에 있는 Expression 아이콘을  클릭하고 XGen Expression Editor에서 다음 텍스트를 입력합니다.

```
$a=rand(0.2,0.9);#0.0,1.0
$a
```

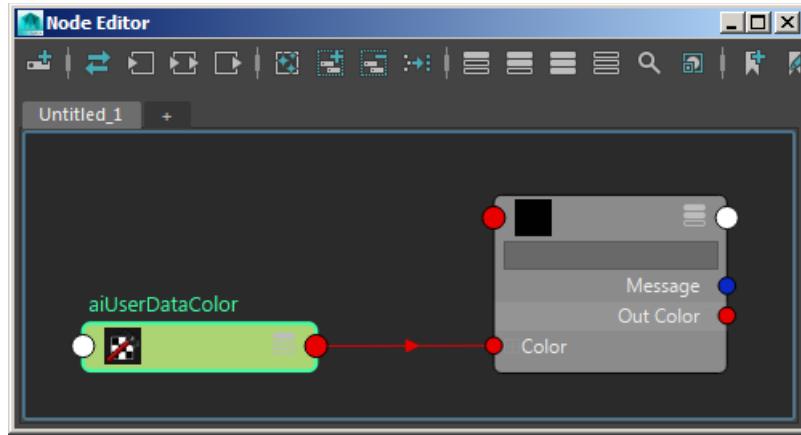


잔디 굴곡이 Bend Param [0] 덕분에 더 무작위로 보입니다.

수식을 사용하여 잔디 색상을 무작위로 변경하기

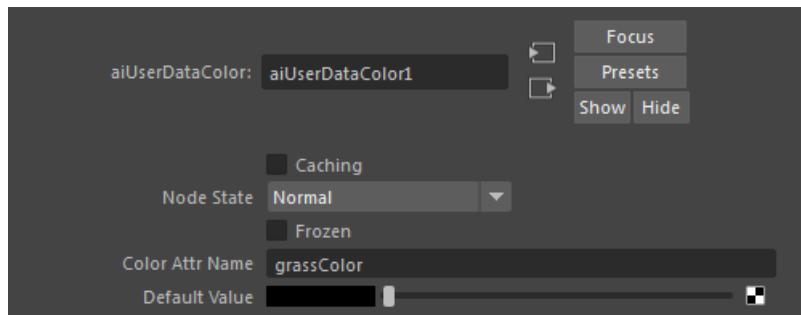
Ai UserData Color 노드와 함께 XGen 수식을 사용하여 잔디의 색조를 임의로 변경할 수 있습니다.

- **Ai UserData Color** 노드를 생성하고 잔디 XGen 설명에 지정된 Standard Surface 쉐이더의 **Color** 속성에 연결합니다.



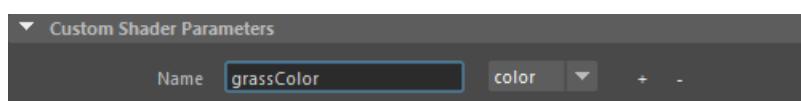
Standard Surface 쉐이더의 Color 속성에 연결된 Ai UserData Color 노드

- Ai UserData 색상 노드의 **Color Attr Name**에 **grassColor**을 입력합니다. XGen 설명에서 동일한 **grassColor** 이름을 사용하겠습니다.



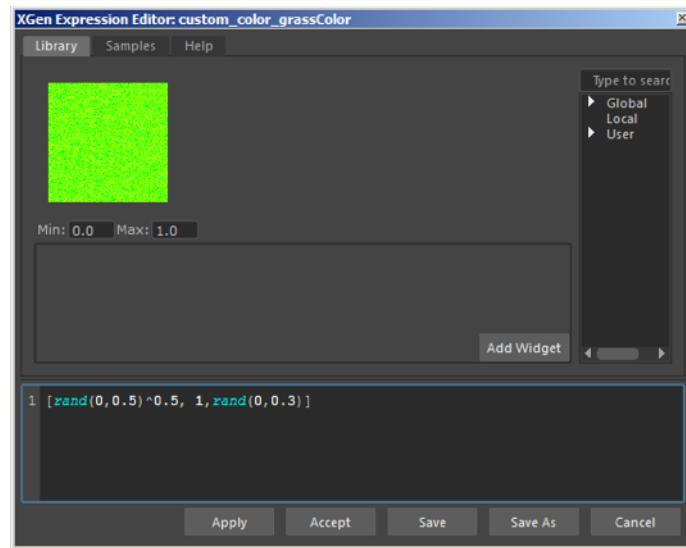
Ai UserData Color 노드에서 Color Attr Name에 'grassColor'라는 이름을 지정합니다.

- XGen에서 **Preview/Output** 탭을 클릭하고 **Output Settings**를 엽니다. 그 아래 **Custom Shader Parameters**가 나타납니다. **Name** 텍스트 필드에 Ai UserData Color 노드가 있는 **Color Attr Name**에 사용한 이름을 입력합니다. 이 경우에는 '**grassColor**'라는 단어를 사용했습니다. **Float**를 클릭하고 **Color**로 변경합니다. 이것이 변경하고자 하는 속성이기 때문입니다.

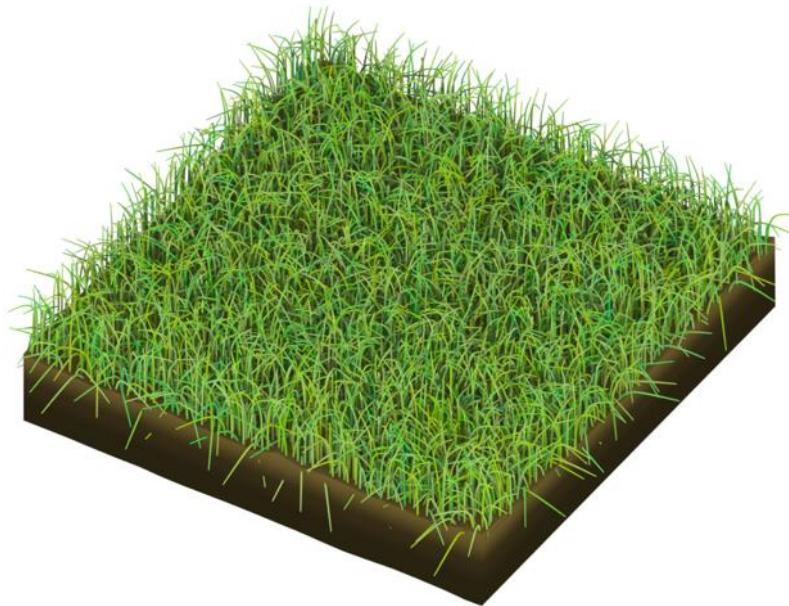


- 새로 정의된 grassColor 색상 속성 옆에 있는 Expression 아이콘을  클릭하고 다음 수식을 입력하십시오.

```
[rand(0,0.5)^0.5, 1,rand(0,0.3)]
```



- 장면을 렌더링합니다. 잔디의 녹색 색조 음영이 이제 무작위로 나타납니다.

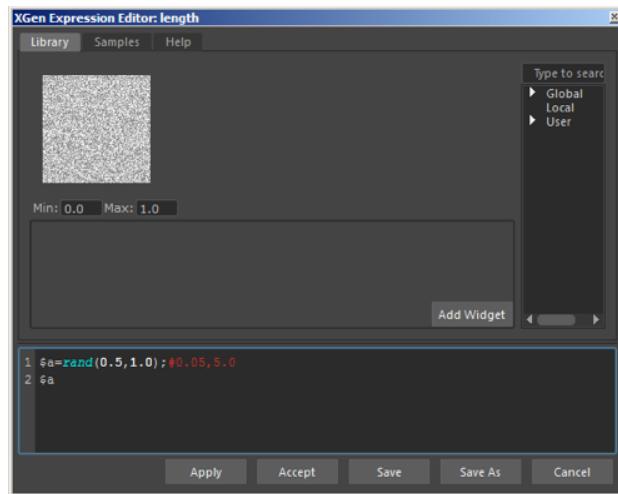


수식으로 길이 변경하기

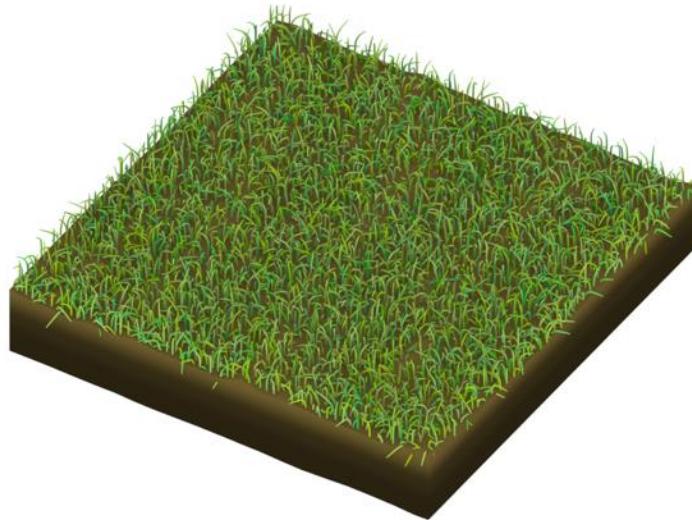
이제 Expression을 사용하여 잔디의 길이를 무작위로 변경할 수 있습니다.

- **Length** 속성 옆에 Expression 아이콘을  클릭하고 다음 수식을 입력하십시오.

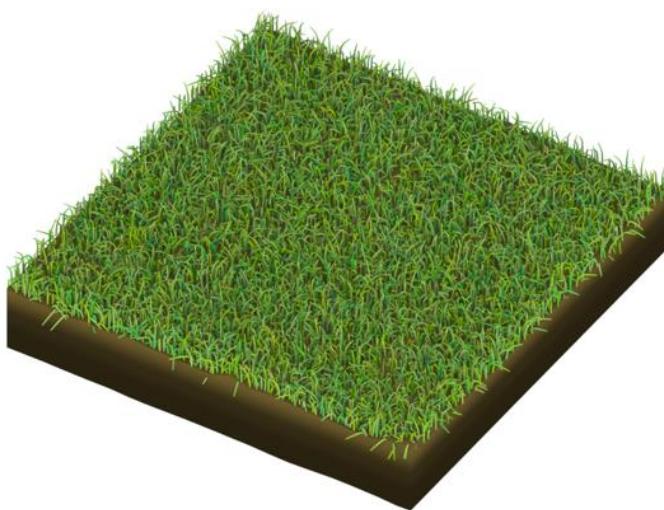
```
$a=rand(0.5,10.0);#0.050.5.0
$a
```



잔디가 다음과 유사하게 보입니다.



- 마지막으로 **Density**를 10000으로 높이면 최종 렌더링 준비가 된 것입니다. 잔디 셰이더에 Backlighting과 Specularity를 추가하여 좀더 사실적으로 만들 수도 있습니다.



점프하는 거미

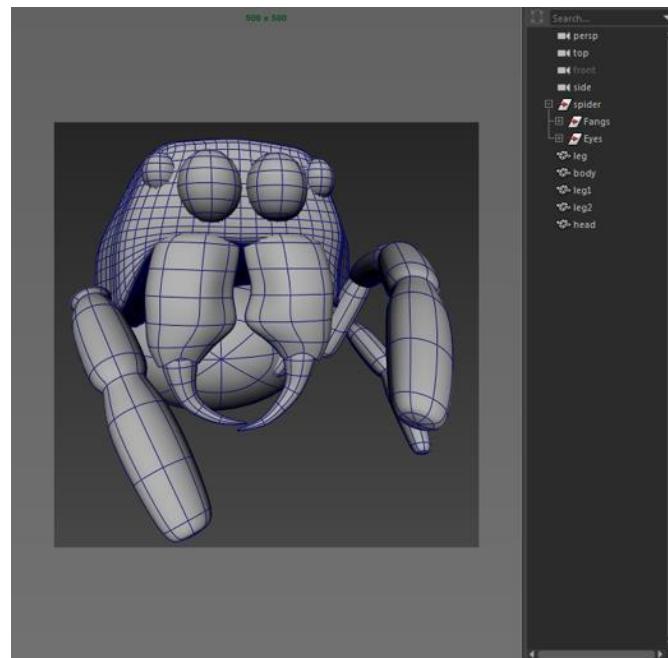


이 자습서는 XGen과 MtoA에서 스플라인을 사용하는 초보자 가이드입니다. XGen Splines과 Maya의 Paint Effects 및 Arnold 렌더링 곡선의 변형 도구를 조합하여 점프하는 거미 모델의 툴을 만드는 방법을 보여줍니다. 또한 곡선을 형성 및 변형하여 보다 사실적인 결과를 얻는 방법에 대해서도 설명합니다. 마지막으로, 믿을 만한 '전자 현미경' 사진 장면을 얻기 위해 거미 모델 자체에 음영 처리를 하고 빛을 비출 것입니다.

- 먼저 거미 모델을 로드합니다. 이것은 모발 스플라인을 적용할 수 있는 별도의 폴리곤 모델로 구성되어 있습니다.

거미 모델은 하단 링크의 사이트에서 다운로드하실 수 있습니다.

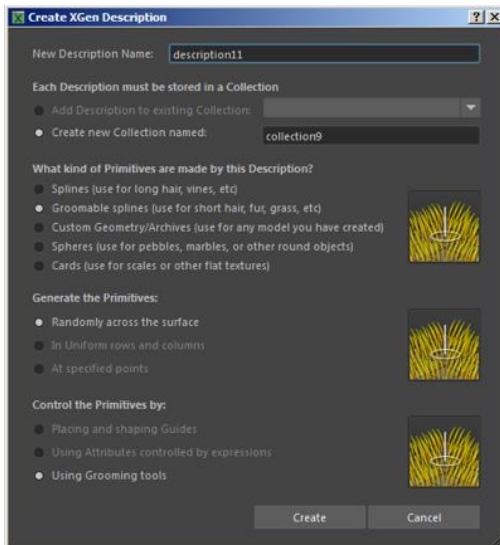
<https://support.solidangle.com/display/A5AFMUG/Jumping+Spider>



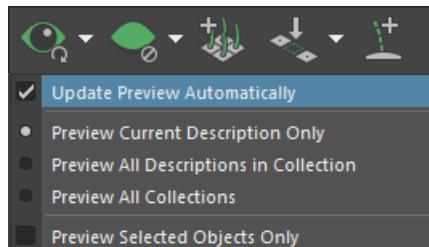
XGen 스플라인

각 신체 부위에 대해 다양한 설명을 만들어 보겠습니다. 다리부터 시작하겠습니다.

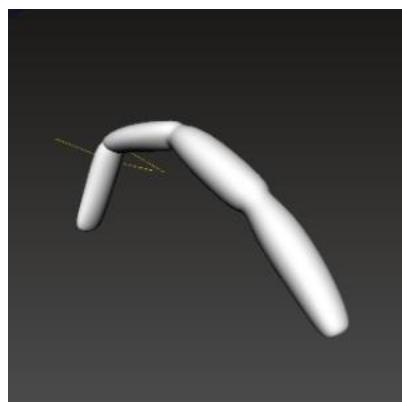
- 다리 지오메트리를 선택하고 **Generate> Create Description**으로 이동합니다. Create XGen Description 창이 나타납니다. **Groomable splines**를 선택하고 Create를 클릭합니다.



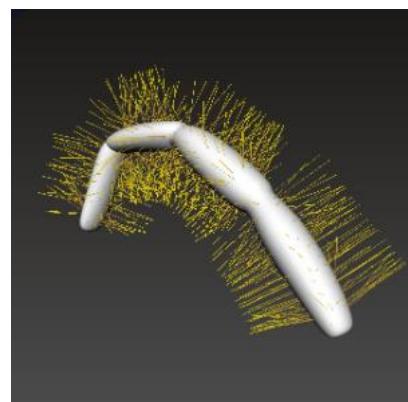
- 뷰포트에서 XGen 설명을 자동으로 업데이트하도록 하려면 'Update Preview Automatically'를 활성화하십시오. 이렇게 하면 해당 설명에 대한 모든 변경사항이 Maya의 뷰포트에 실시간으로 표시되며, 변경이 있을 때마다 미리보기(눈 모양 아이콘)를 계속 업데이트할 필요가 없습니다.



- 모발의 수를 늘리려면 **Density** 값을 늘려야 합니다. **Grooming** 탭으로 이동하고 **Settings** 아래에서 **Density** 값을 **80**으로 높입니다.



기본 Density 1



Density 80

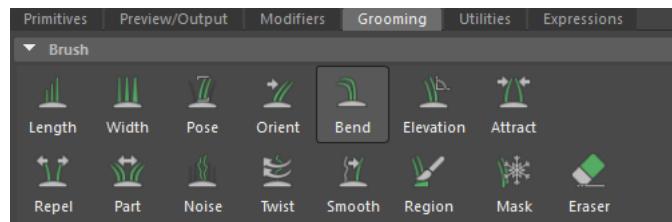
Grooming

다리

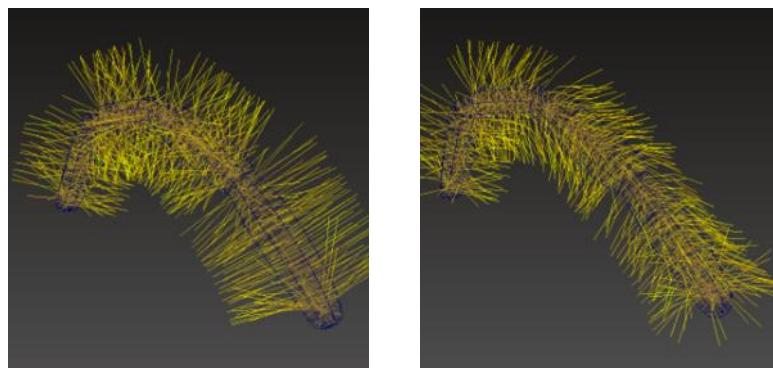
XGen의 브러쉬 기반 손질 도구를 사용하여 다리 털을 스타일링할 수 있습니다.

Blend

먼저 다리의 모양을 따라 곡선을 아래쪽으로 구부리기 시작하겠습니다.



- **Grooming** 탭으로 이동하고 **Bend** brush를 선택합니다. 뷔포트에서 마우스를 다리 위로 가져갑니다. 그러면 커서가 브러시 도구로 변합니다. 이 브러시를 클릭하여 다리를 따라 위에서 아래로 끌어옵니다. 다리를 따라 브러싱하면 털이 구부러지는 것을 볼 수 있습니다.

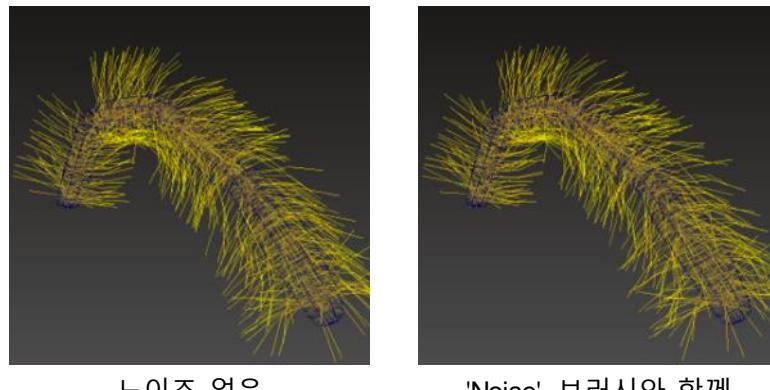


'Bend' 브러시가 없는 기본 길이

'Bend' 브러시

노이즈

- 털이 너무 균일합니다. 노이즈를 추가하여 이러한 균일성을 줄일 수 있습니다. Noise 브러시를 선택하고 털을 따라 반복해서 브러싱하십시오.



노이즈 없음

'Noise' 브러시와 함께

다리 쉐이딩

- 곡선을 손질한 후에는 다리 설명을 선택하고 Standard Surface 쉐이더를 지정하십시오. 렌더링에 '전자 현미경' 모양을 얻기 위해 반사가 없는 쉐이더를 원하기 때문에 Standard Hair 대신 Standard Surface를 사용하려고 합니다.

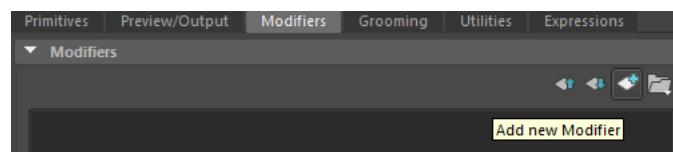


- 다른 다리들에 동일한 다리 설명을 추가하려면 다른 다리 메시를 선택하고 **Descriptions>Bind Patches>Add Selected Faces**로 이동하십시오.

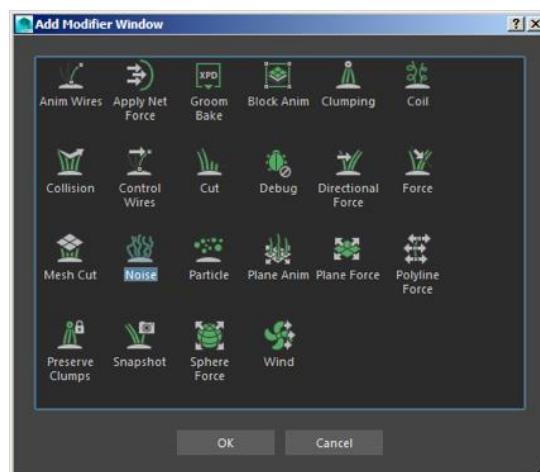
몸

다리 설명을 복제하여 몸에도 사용할 수 있습니다.

- 다리 설명을 선택하고 **Descriptions>Duplicate Description...**으로 이동합니다.
- 다리 지오메트리를 추가할 때 사용했던 위 단계를 설명에 반복합니다. 몸 지오메트리를 선택하고 **Descriptions>Bind Patches>Add Selected Faces**로 이동합니다.
- 몸의 설명에 약간의 변형을 신속하게 추가하기 위해 수정자를 추가할 수 있습니다. **Modifiers** 탭으로 이동하고 **Add new Modifier**를 선택합니다.

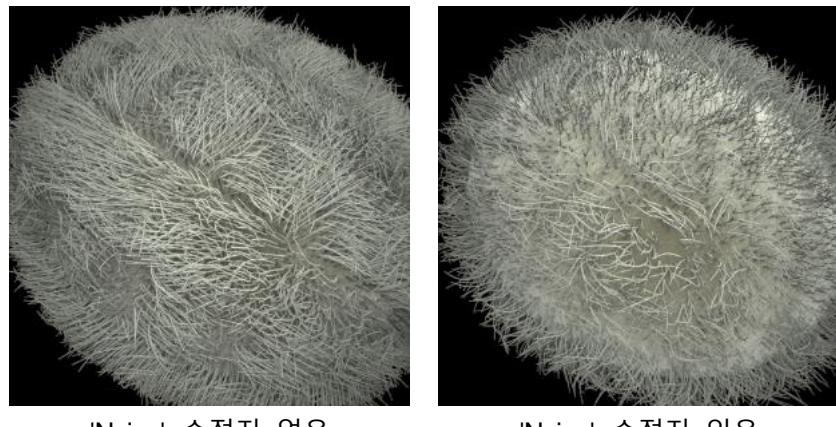


- Noise** 아이콘을 선택하고 OK를 클릭합니다.



Noise modifier

- Frequency**와 **Magnitude**를 높여서 원하는 양의 노이즈를 얻습니다. 이 경우 Frequency는 4, Magnitude는 1.3입니다.



'Noise' 수정자 없음

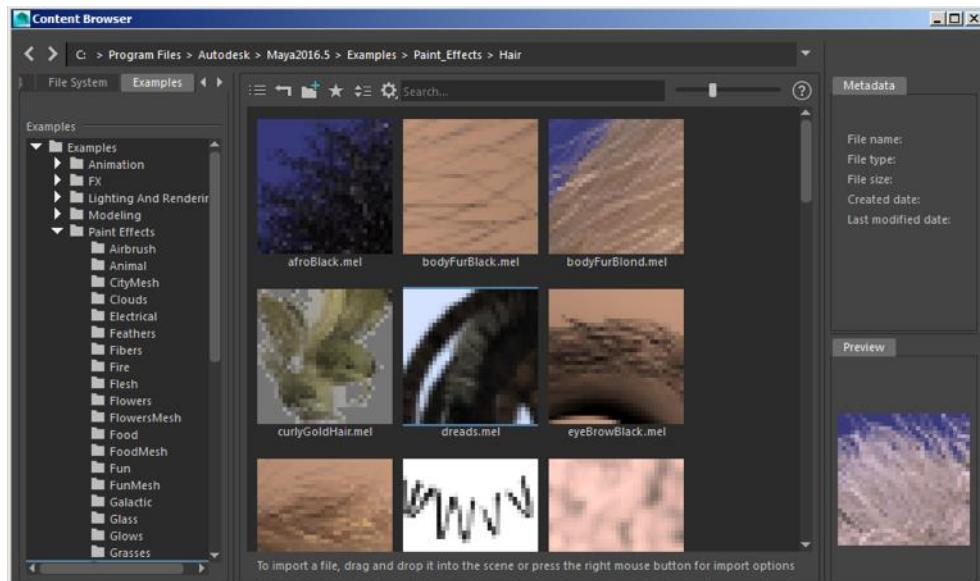
'Noise' 수정자 있음

머리

머리에 대해 Paint Effects를 사용한 다음 곡선으로 변환해 보겠습니다. 이렇게 하면 곡선으로 변환했을 때 더 많은 도구 옵션이 제공됩니다. 따라서 개별 헤어 곡선을 '형성'할 때 보다 많은 제어력을 확보할 수 있습니다.

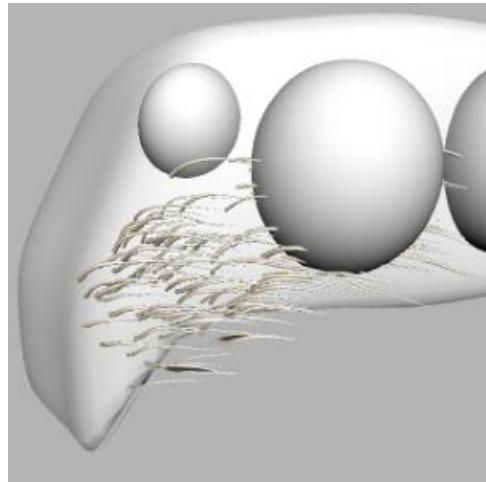
Paint Effects 브러시

- 먼저 머리 지오메트리를 선택하고 **Paint Effects>Make Paintable**로 이동합니다. 그러면 Paint Effects 스트로크를 머리의 표면에 직접 페인팅할 수 있습니다.
- Content Browser**를 엽니다. Paint Effects 탭을 선택하고 hair 폴더 아래에 있는 **furBunny.mel** 브러시를 선택합니다.



Hair 폴더에서 'furBunny' 스트로크를 선택합니다.

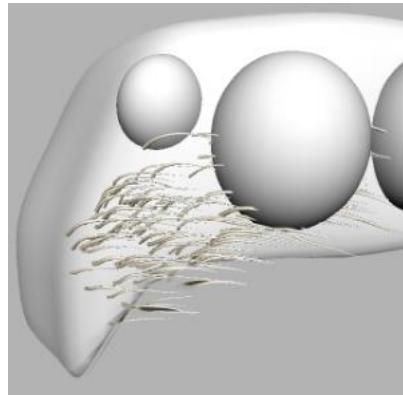
- 아래 이미지와 같이 머리를 가로질러 스트로크를 긋습니다.



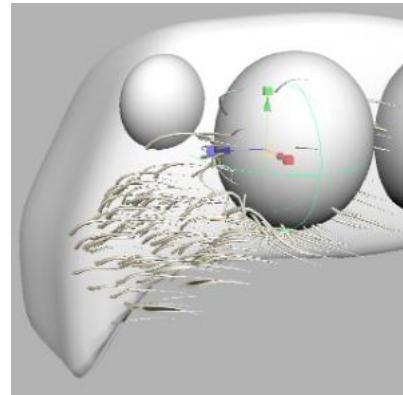
Paint Effects 수정자

현재 스트로크가 눈의 지오메트리를 교차시키고 있습니다. 우리는 스트로크가 안구를 감싸는 효과를 연출하려고 합니다. **Paint Effects Modifier**를 사용하여 이것을 시도할 수 있습니다.

- **furbunny** Paint Effects 스트로크를 선택하고 **Paint Effects>Create Modifier**를 선택합니다. 수정자를 눈알과 동일한 크기로 조정하고 똑같은 위치에 배치합니다. **Force** 값을 높입니다. 아래 예는 2 값을 사용합니다.



Line Modifier 없음

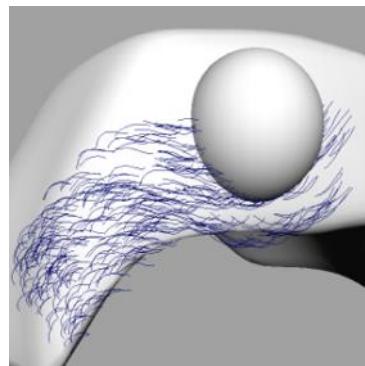


Line Modifier 있음

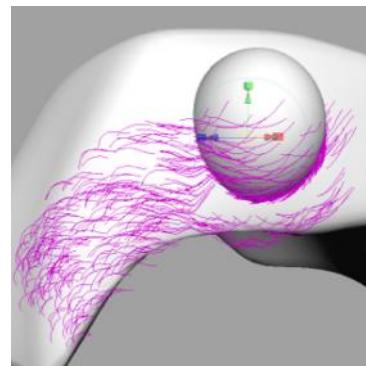
Sculpt Deformer

'Paint Effects Modifier'는 우리가 원하는 효과를 내지 못하고 너무 무작위로 느껴집니다. 그 대신 **sculpt deformer**를 사용하여 형상을 변형할 수 있습니다. 하지만 Paint Effects 스트로크를 먼저 곡선으로 변환하고 Paint Effects Modifier를 삭제해야 합니다.

- **furbunny** 스트로크를 선택하고 **Modify>Convert>Paint Effects to Curves**로 이동합니다.
- 해당 곡선들을 선택하고 **Create Deformers>Sculpt Deformer**로 이동합니다('Project' 모드로 설정됨).



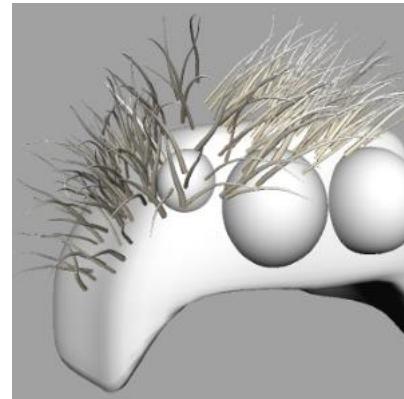
Sculpt Deformer 없음.
스트로크가 눈을 통과합니다.



Sculpt Deformer 있음. 눈
주위에서 곡선이 변형됩니다.

머리 윗부분

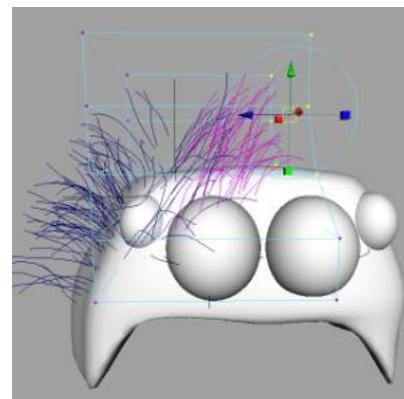
- 머리 윗부분을 또 한 번 페인팅합니다. 선택하고 **Global Scale**을 6으로 높여서 아래 이미지처럼 스트로크의 크기를 키웁니다.



Paint Effects 스트로크 - 유리 조절 6

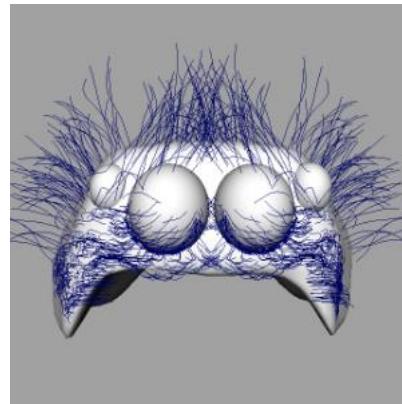
곡선들을 다시 뒤로 잡아 당겨 다른 쪽과 겹치지 않도록 하기 위해(나중에 다시 복제할 예정) **Lattice Deformer**를 사용할 수 있습니다.

- 끝 곡선을 선택하고 **Create Deformers>Lattice**를 선택합니다. 격자가 선택된 상태에서 구성 요소 모드(F8)로 들어가 격자의 끝 cv를 선택하고 머리 중심에 겹치지 않도록 다시 이동하십시오. 또한 이 격자를 사용하여 곡선을 추가적으로 형상화할 수 있습니다.



Lattice Deformer를 사용하여 곡선 형상화

- 곡선의 모양이 마음에 든다면 하나로 그룹화하여 Z 방향으로 머리의 다른 쪽까지 가로질러 복제합니다.

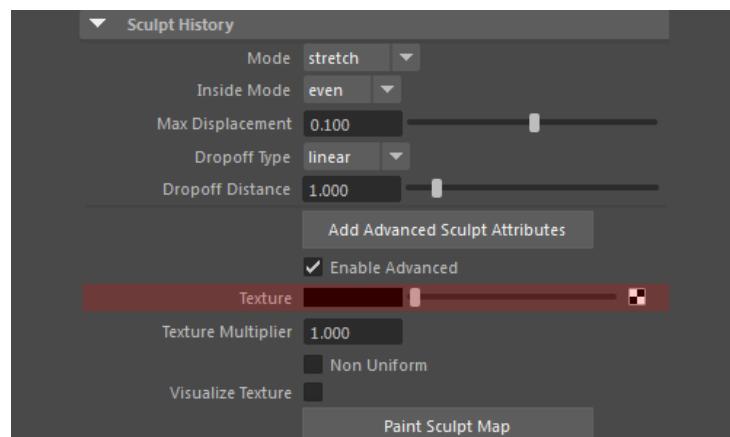


곡선을 머리의 다른 쪽까지 복제(-Z 방향).

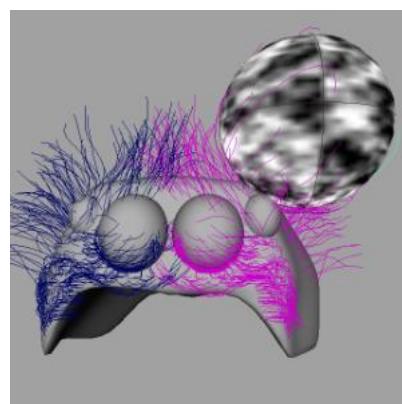
곡선에 노이즈 추가

현재 곡선의 한 그룹을 간단히 다른 쪽으로 복제했기 때문에 곡선이 상당한 대칭을 이루고 있습니다. 대칭성을 줄이는 한 가지 방법은 곡선에 약간의 노이즈 변형을 추가하는 것입니다. **sculpt deformer**는 텍스처 맵을 사용하여 곡선을 변형하는 옵션이 있습니다.

- 곡선의 오른쪽 그룹을 선택하고 **Deform> Sculpt**로 이동합니다.



- **Sculpt Deformer**의 **Texture** 속성에 노이즈 텍스처를 추가합니다. 뷔포트에서 해당 텍스처를 보려면 hardware texturing - **Shading>Hardware Texturing**을 활성화해야 합니다.

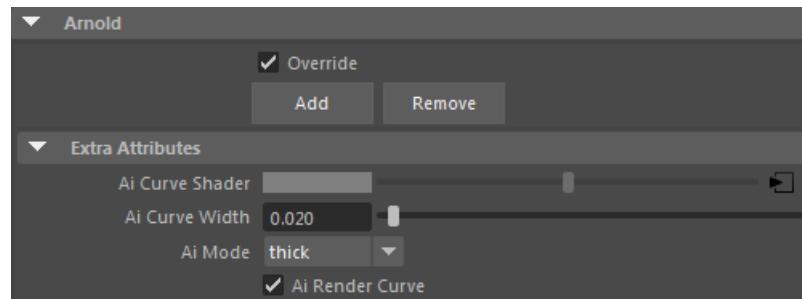
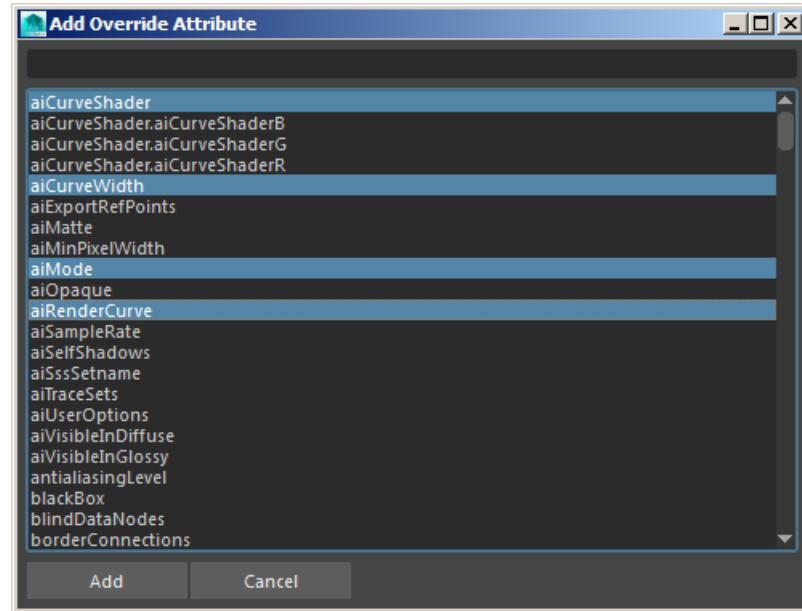


Override Set

- 해당 곡선들을 선택하고 세트를 만듭니다. - **Create>Sets>Set**
- 속성 편집기에서 해당 세트를 엽니다. **Arnold** 아래에서 **Add**를 선택합니다. 'Add Override Attribute' 창이

나타납니다. 세트에 다음 속성을 추가합니다.

1. Ai Curve Shader
2. Ai Curve Width
3. Ai Mode
4. Ai Render Curve



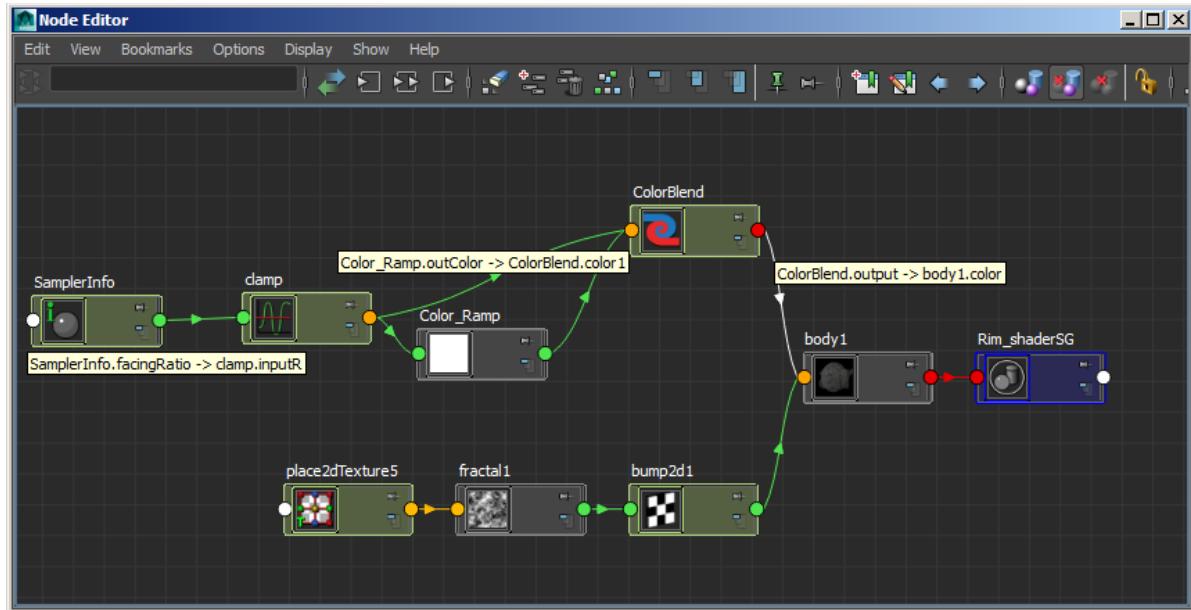
세트에 재정의 속성 추가

- **Ai Hair** 쉐이더를 **Ai Curve Shader** 속성에 연결합니다.
- 원하는 곡선의 두께에 따라 Ai Curve Width를 0.02로 높입니다.
- Ai Mode를 **thick**으로 변경합니다. 그러면 곡선에 원형 튜브 프로필이 생깁니다.
- **Ai Render Curve**를 선택하여 곡선을 렌더링시킵니다.

Shading

몸

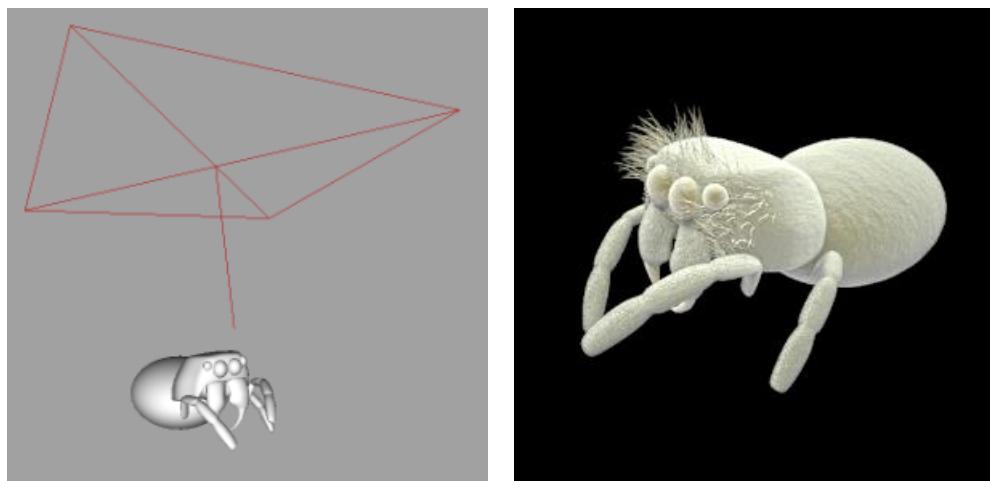
거미의 몸에는 전자 현미경을 통해 거미가 촬영되었다는 느낌을 주는 "rim shader"가 지정되어 있습니다. 샘플러 정보 노드는 Standard Surface 쉐이더의 Diffuse 속성에 할당된 램프 텍스처에 연결됩니다.



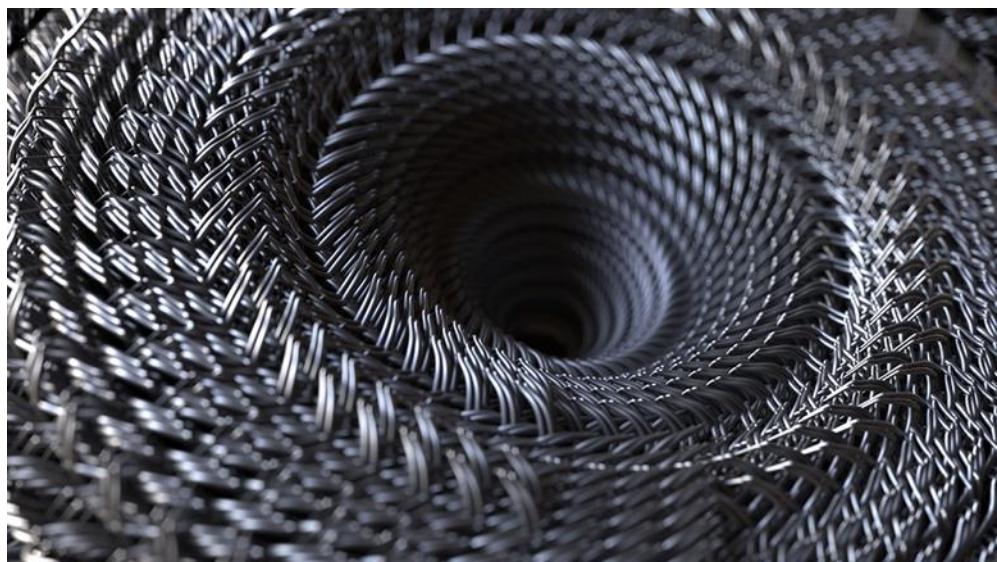
거미의 몸 쉐이더를 보여주는 노드 편집기

조명

장면은 커다란 사진 소프트박스를 나타내는 '쿼드(quad)'로 설정된 큰 Ai Area 조명으로 간단히 빛이 켜집니다. 최종 렌더링을 위해 샘플을 3으로 높였습니다.



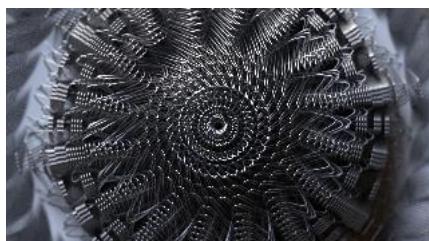
XGen 스플라인 패턴



이 짧은 자습서에서는 **XGen**을 사용하여 만달라를 연상시키는 스플라인 패턴을 만드는 방법을 설명합니다. XGen의 Spline Primitive 속성 중 일부를 사용하여 아래 예제 이미지와 같은 다양한 패턴을 만들어 보겠습니다. 이러한 스플라인 속성은 최면적인 애니메이션을 생성하기 위해 키프레임화될 수 있습니다. 대부분의 예는 여기에 있습니다.

더 많은 예는 하단 링크를 참조 바랍니다.

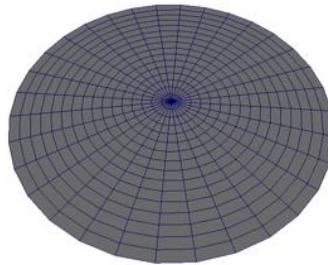
<http://www.leegriggs.com/xgen-splines>



Base Geometry

- 먼저 NURBS 실린더를 만들겠습니다. 상단 뚜껑을 벗기고 기본 실린더 형상을 삭제하십시오. 꼭대기 형상에 XGen 설명을 적용하기만 하면 됩니다.

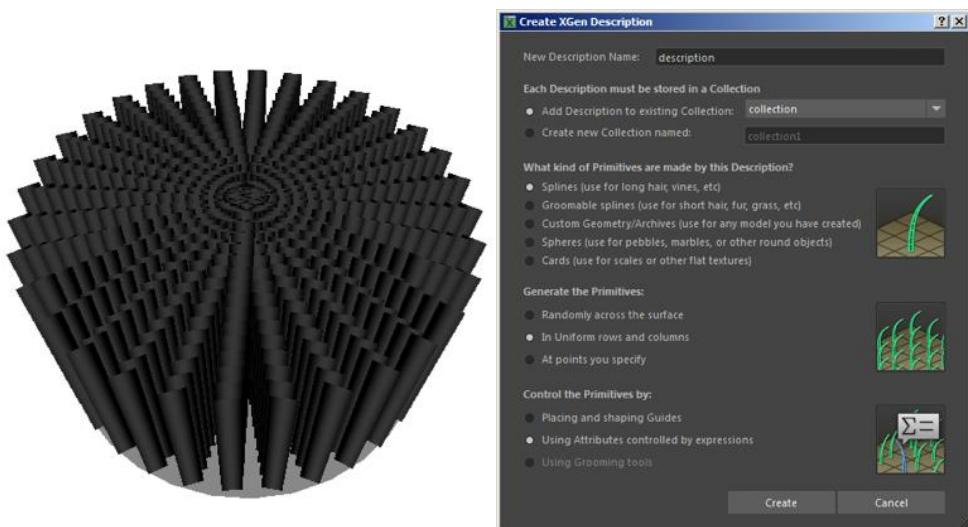
- NURBS 상단 표면을 폴리곤 지오메트리로 변환합니다(XGen은 폴리곤 지오메트리에서만 작동). **Modify -> Convert NURBS to Polygons**. 다음과 같이 변환된 폴리곤 오브젝트가 남도록 원본 NURBS 표면을 삭제합니다.



폴리곤으로 변환된 NURBS 실린더 상단 뚜껑

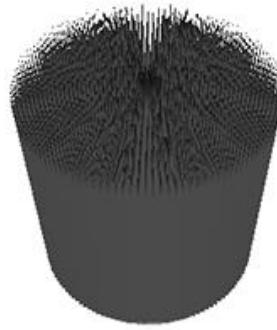
Create 설명

- 실린더 위쪽 뚜껑 지오메트리를 선택한 상태에서 **Generate -> XGen -> Create Description**으로 이동합니다.
- Splines**를 선택합니다(긴 머리카락, 덩굴 등에 사용). **Generate the Primitives**를 *In Uniform rows and columns*로 변경하여 아래 이미지처럼 보이게 합니다.



스플라인 속성

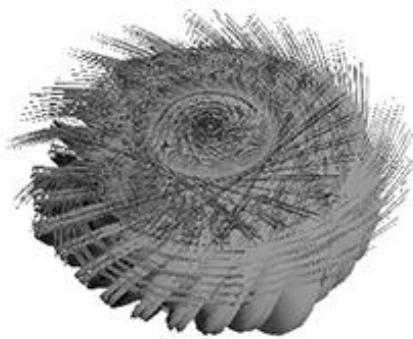
- XGen Editor에서 **Spacing**을 **0.05**로 줄입니다. 그러면 topCap 지오메트리의 표면에 실린더에 대한 더 많은 스플라인이 생성됩니다.
- Splines**의 길이를 **2.0** 정도로 높입니다.
- Spline Width**을 **0.1** 정도로 낮춥니다.
- Taper**을 1 정도로 높입니다. Taper는 베이스와 비교하여 각 스플라인의 끝에서 너비를 제어합니다. 양수 값은 끝을 베이스보다 좁게 만들고 음수 값은 끝을 베이스보다 넓게 만듭니다.
- Bend U** 값을 약 **-1**로 줄이십시오. Bend는 해당 CV에서 스플라인이 구부러지는 정도를 제어합니다. 0은 굽힘이 없는 것이고 1.0은 90도 굽힘입니다. 다음과 같은 모습을 보입니다.



하단 링크 이미지 참조

<https://support.solidangle.com/display/A5AFMUG/XGen+Spline+Patterns>

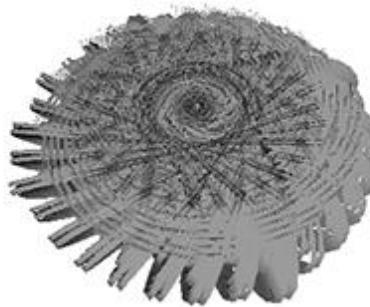
- 다음에는 **Tilt** 속성을 조절해 보겠습니다. 이 속성은 스플라인을 표면 법선으로부터 멀리 기울입니다.



하단 링크 이미지 참조

<https://support.solidangle.com/display/A5AFMUG/XGen+Spline+Patterns>

- 이제 **Bend V** 속성을 조절해 보겠습니다. 이것은 스플라인이 해당 CV에 직각으로 구부러지는 정도를 제어합니다. 0은 굽힘이 없는 것이고 1.0은 90도 굽힘입니다.



하단 링크 이미지 참조

<https://support.solidangle.com/display/A5AFMUG/XGen+Spline+Patterns>

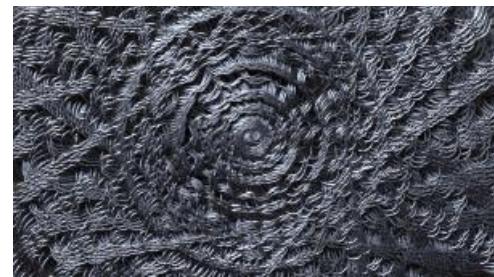
- 이제 흥미로운 패턴이 발생하고 있습니다. 흥미롭고 독창적인 결과를 얻기 위해 이러한 Spline 속성 설정을 여러 조합으로 시도해 보십시오!

In Uniform Rows and Columns와 Randomly Across the Surface 비교

아래 이미지는 프리미티브 생성 방법을 비교한 것입니다. 비록 **In Uniform Rows and Columns**가 일반적으로 더 나아 보이지만 두 방법 모두 서로 잘 작동합니다. 두 방법을 모두 시도하여 어떤 쪽이 더 나은지 확인해 보십시오.



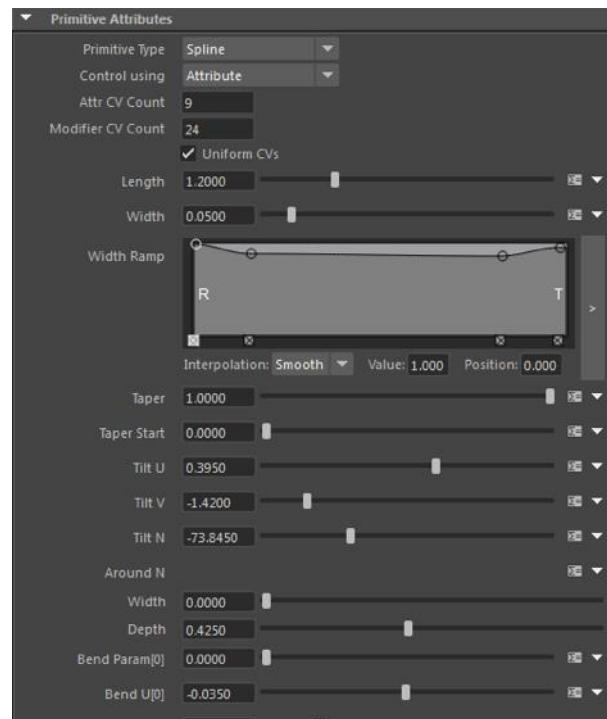
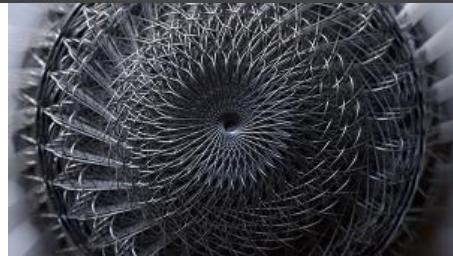
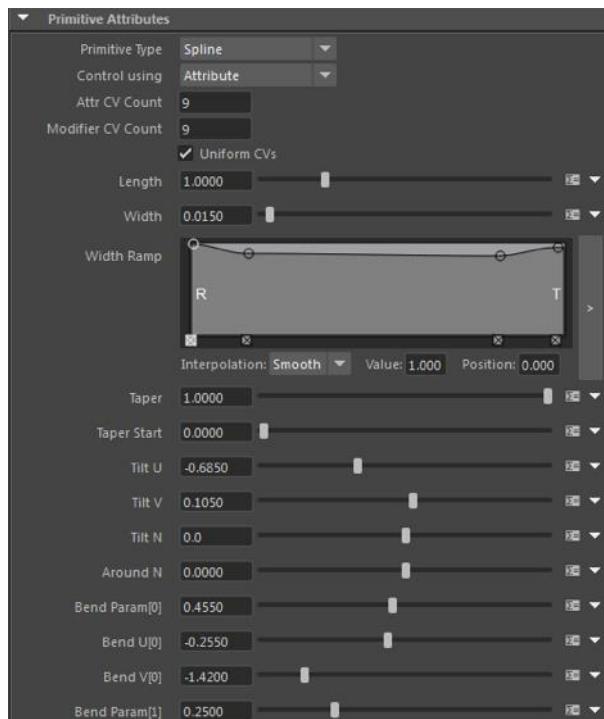
In uniform rows and columns

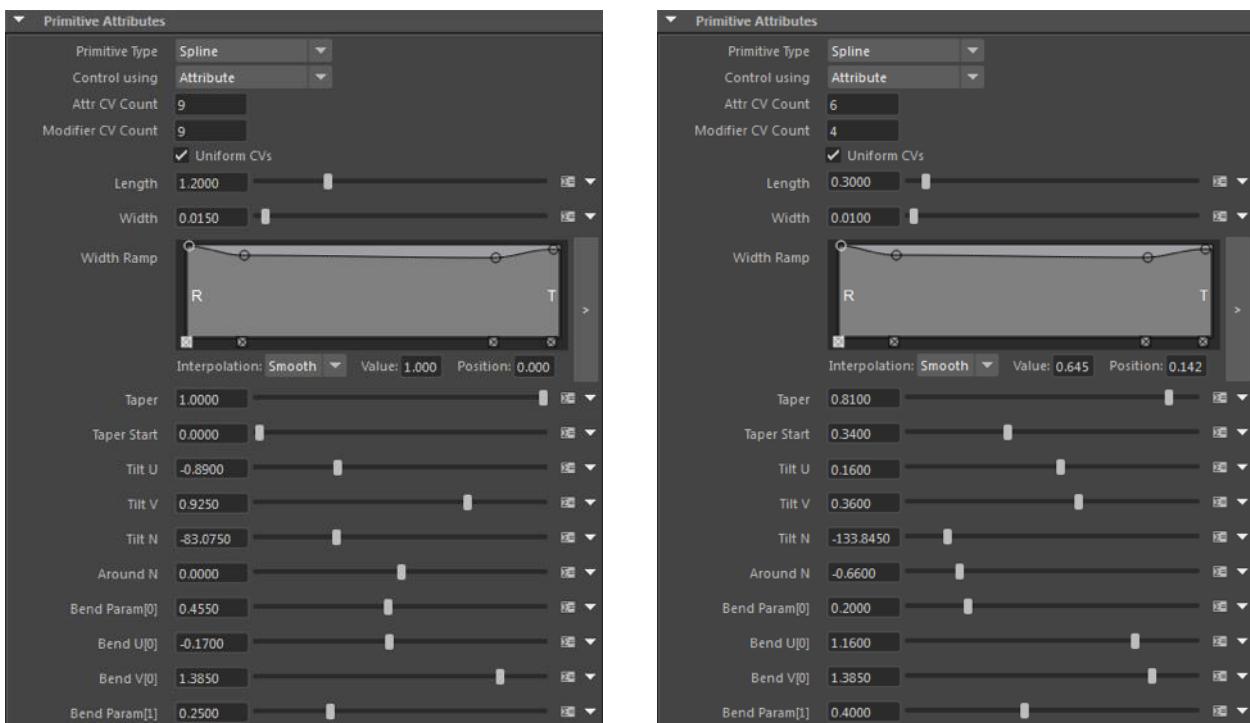


Randomly across the surface

스플라인 설정 예

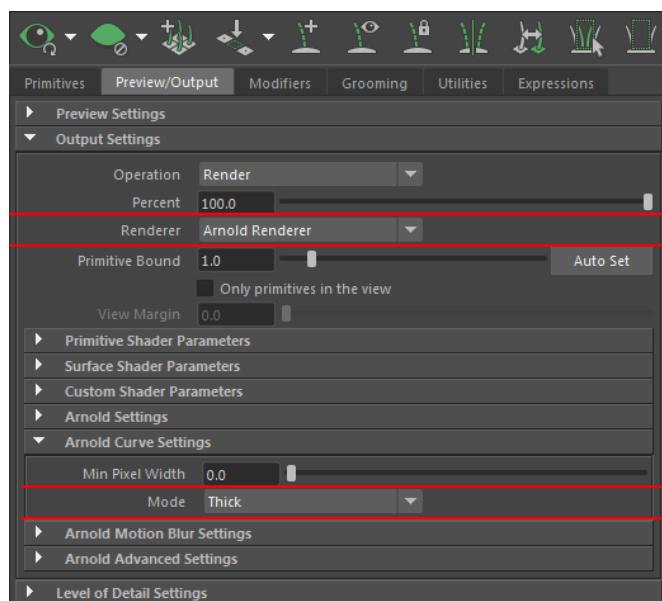
아래는 XGen Spline Primitive 속성 설정을 가진 완성된 렌더링의 몇 가지 예입니다. **Bend** 및 **Tilt** 스플라인 속성을 조정하면 매우 다양한 것들을 볼 수 있습니다.





Rendering

- 렌더링하기 전에 **Arnold Renderer**가 **Renderer**로 설정되어 있고 **Mode**가 **Thick**으로 설정되어 있는지 확인하십시오(리본은 우리가 원하지 않는 평면 곡선입니다). 이를 설정은 XGen Preview/Output 탭 아래에 있습니다. 쉐이더를 XGen 콜렉션에 할당합니다. 이제 렌더링 준비가 다 되었습니다!



Batch Render

일괄 렌더링 전에 다음 단계를 수행해야 합니다. 그렇지 않으면 XGen 설명이 렌더링되지 않습니다.

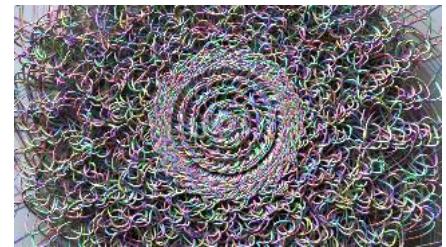
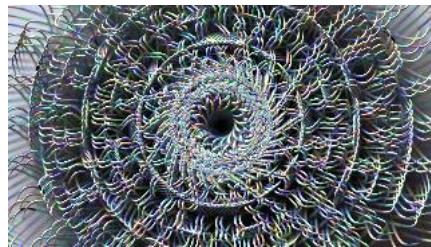
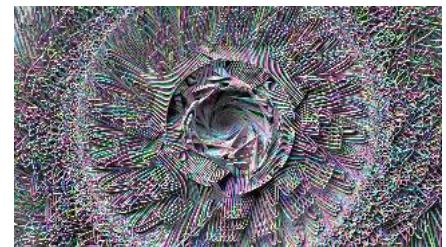
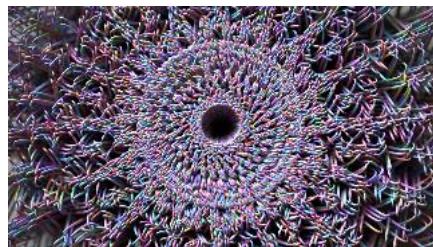
- 장면을 저장합니다.
- 지오메트리 및 XGen 설명이 선택된 채로 XGen description 메뉴에서 **File>Export Patches for Batch Render**로 이동합니다. XGen description에 애니메이션이 있으면 해당 애니메이션을 활성화하고 **Frame Range**를 선택하십시오. **AbcExport.mll**을 Plug-in Manager에서 로드해야 합니다. 그렇지 않으면 애니메이션이 제대로 내보내지지 않습니다.

- 장면을 저장합니다.
- 애니메이션이 마음에 들면 일괄 렌더링 애니메이션을 시작할 수 있습니다.



결론

지금까지 XGen 스플라인으로 패턴을 렌더링하는 방법을 알아봤습니다. 여러 가지 폴리곤 형상을 사용하여 설명을 첨부하고 수식과 색상을 스플라인 속성에 추가해보는 것도 시도해 볼만 합니다. 즐겁게 XGen 스플라인에 도전해보세요!

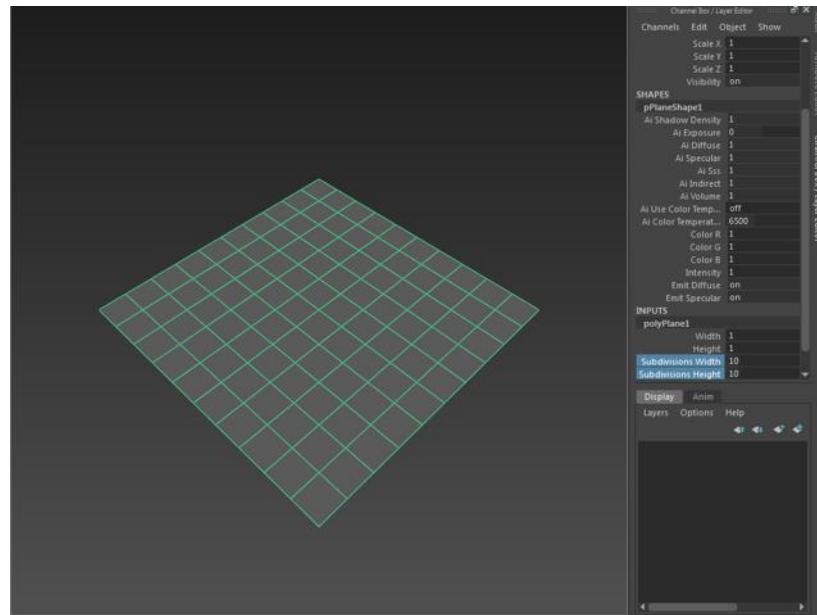


수식을 사용하여 구체의 크기를 균일하게 조절하기

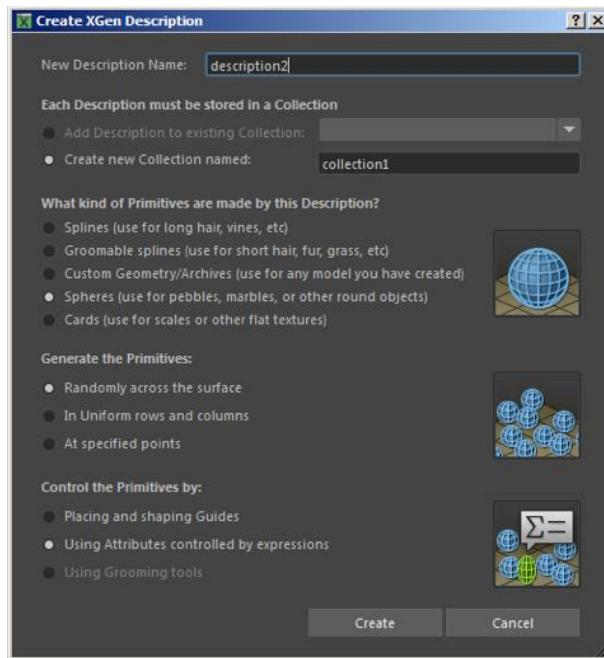


이 짧은 자습서에서는 Expressions를 사용하여 Sphere Primitives의 크기를 무작위로 조절하는 방법을 설명합니다.

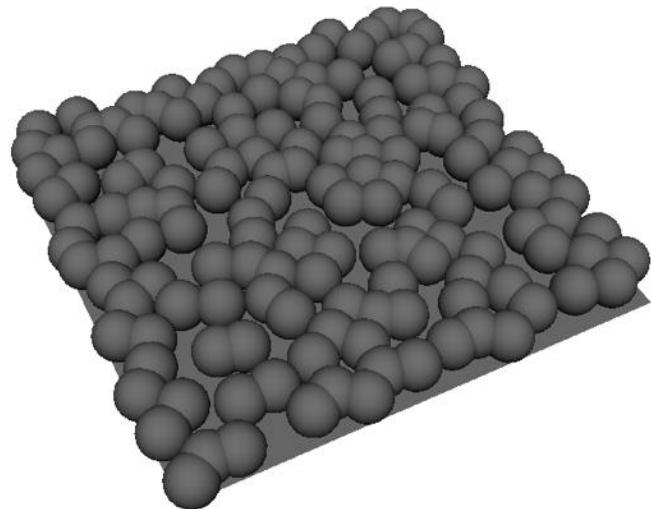
- 먼저 XGen 설명을 할당할 수 있는 폴리곤 평면을 만들겠습니다. 평면을 **X**, **Y**, **Z**에서 **10**으로 조절합니다. **Width** 및 **Height**에서 평면의 **subdivision**을 **10**으로 높입니다.



- 폴리곤 평면이 선택된 상태로 **Generate> Create Description...**으로 이동합니다.
- Create XGen Description이 나타나서 **What kind of Primitives are made by this Description?** 질문합니다. **Spheres**와 **Generate the Primitives - Randomly across the surface**를 선택합니다.

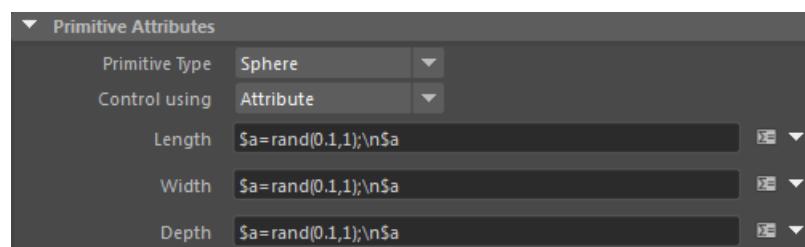


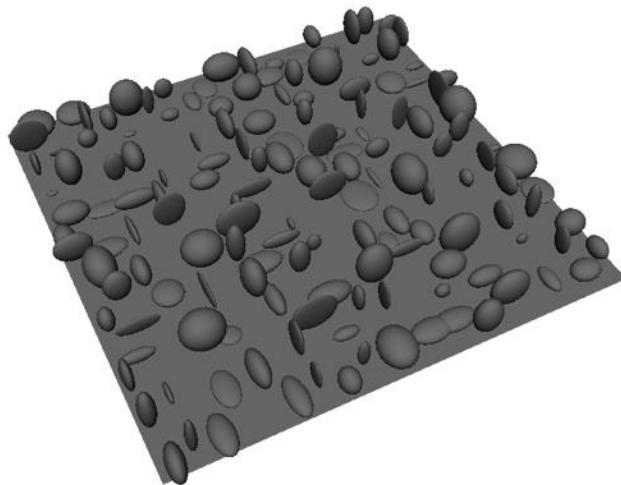
- **Density**를 100 정도로 높이면 아래 이미지처럼 구체들로 채워집니다.



Length, Width, Depth 속성 각각에 다음과 같은 일반적인 임의의 수식을 추가하려는 경우 균일하지 않은 방향으로 구체의 크기가 임의로 조정되는 것을 알 수 있습니다.

```
$a=rand(0.1,1);
$a
```

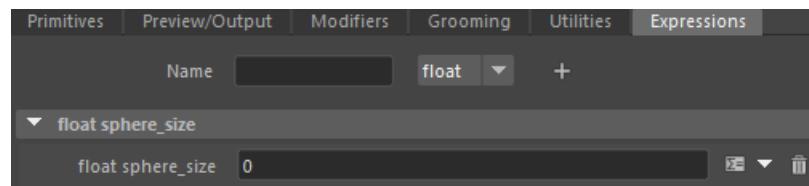




구체의 길이, 너비 및 깊이가 개별적으로 임의 조정됩니다.

모든 방향에서 구체의 크기를 균일하게 조정하는 방법은 **Expressions** 탭에서 **Primitive Attributes**로부터 호출할 수 있는 속성을 만드는 것입니다.

- **Expressions** 탭으로 이동하고 **Name** 속성에 **sphere_size**를 입력합니다. **float**로 설정되어 있는지 확인하고 아이콘을 클릭하여 속성을 생성합니다.

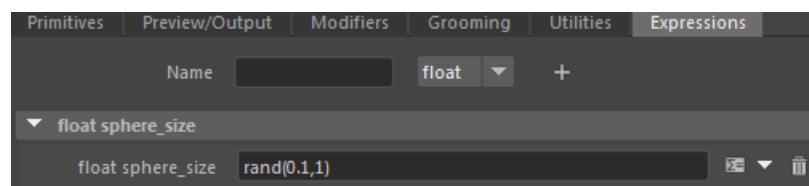


새로운 float 속성 **sphere_size**가 아래에 생성된 것을 볼 수 있습니다.

- 텍스트 필드에 다음 수식을 입력합니다.

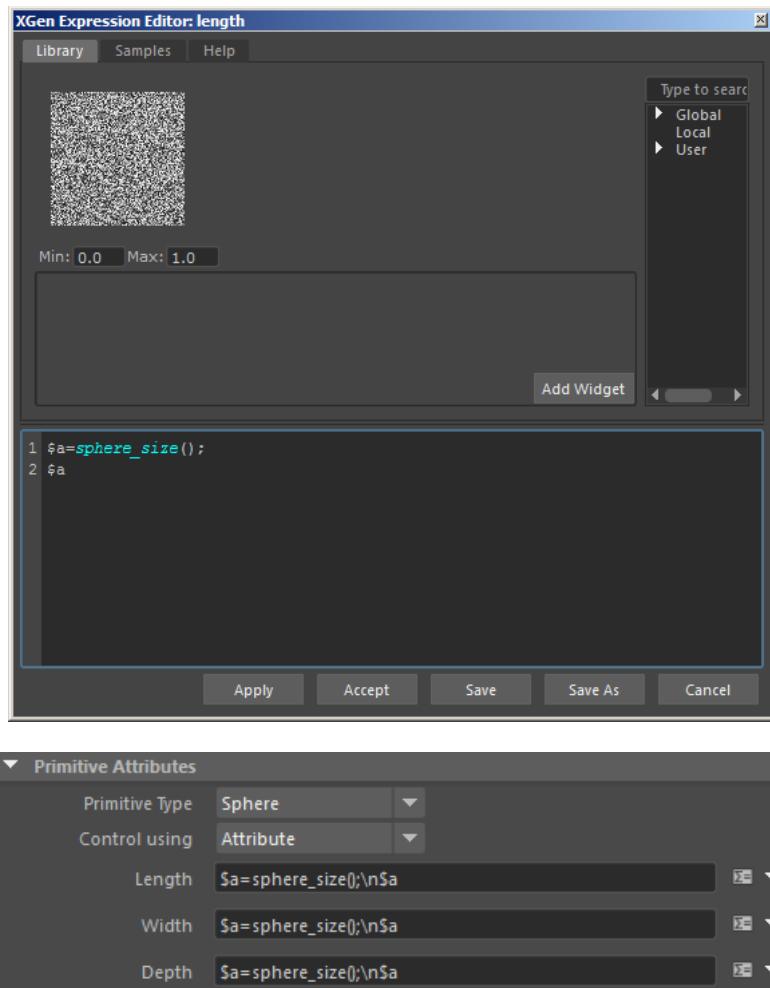
```
rand(0.1,1)
```

이들 값은 구체 크기의 전역 범위를 결정합니다.

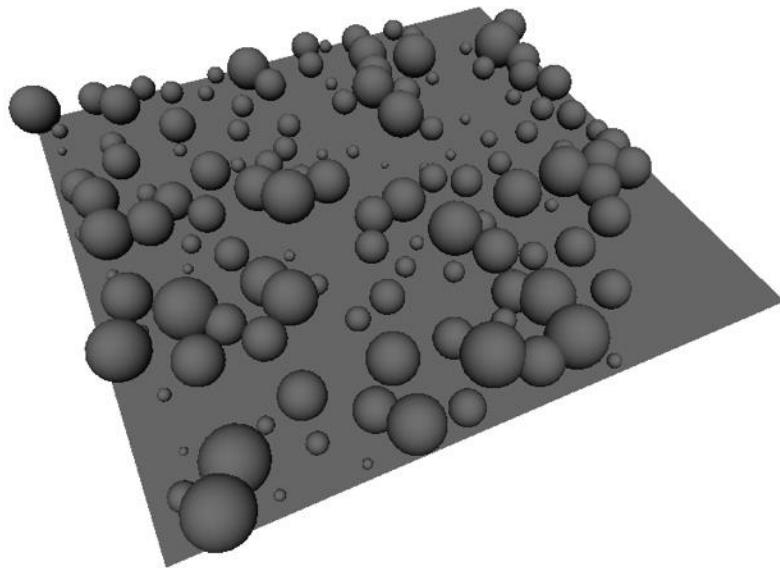


- 이제 이전 수식을 **Length**, **Width** 및 **Depth** 속성에서 다음으로 대체하십시오.

```
$a=sphere_size();
$a
```



이제 구체의 크기가 모든 방향으로 불규칙하게 조절된 것을 알 수 있습니다.



지금까지 수식을 사용하여 구체의 크기를 균일하게 조절하는 방법을 살펴봤습니다. 예제 이미지에서와 같이 구체의 색상을 임의로 변경하려면 프리미티브에 무작위 색상 지정하기를 참조하십시오. 구체에 색상 텍스처 맵을 지정하는 방법을 배우려면 이 자습서 - 기본 아카이브에 색상 지정하기를 참조하십시오.

