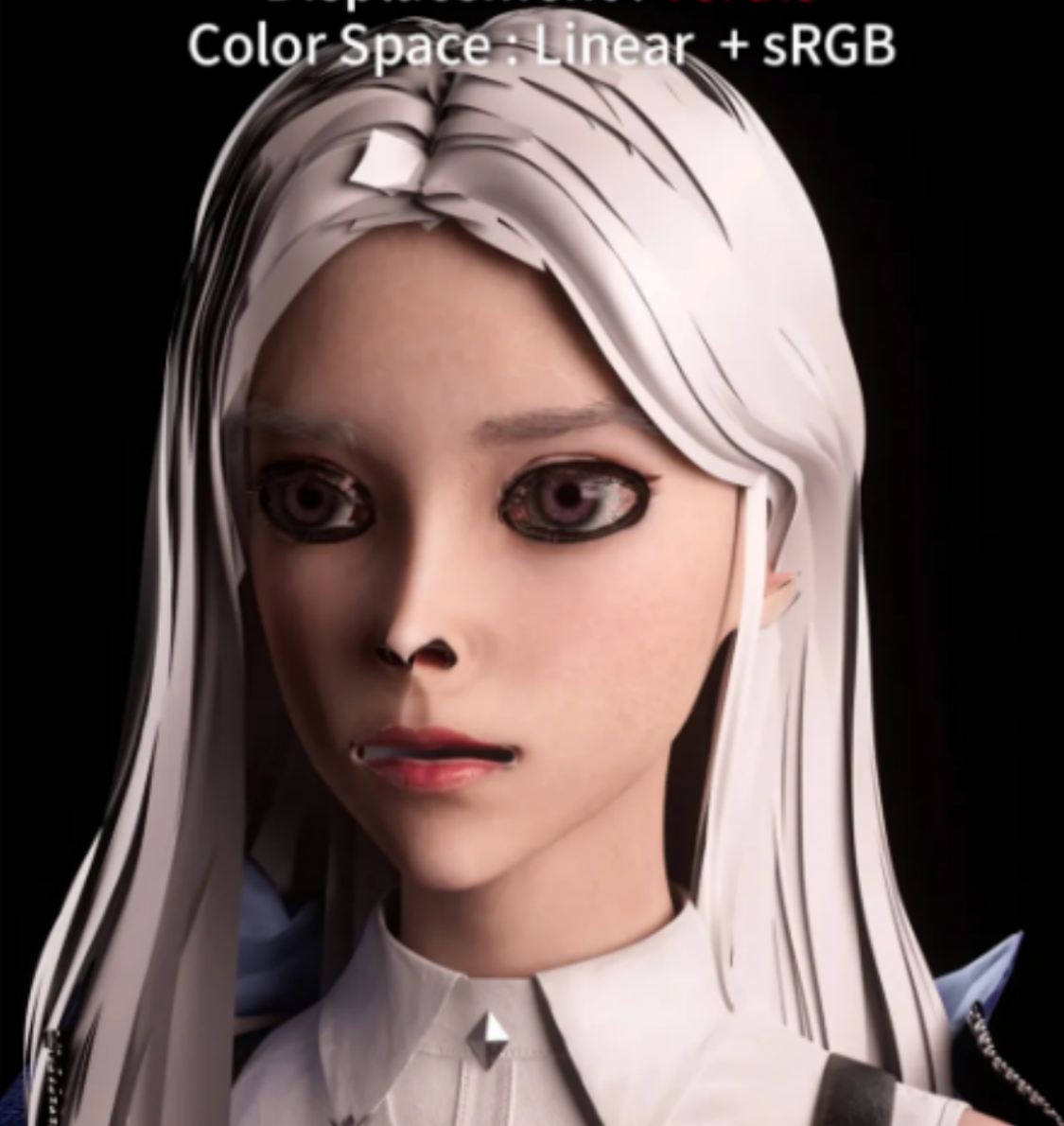


Octane (Image Texture) Color space

Displacement : **Vertxe**
Color Space : Linear + sRGB



Displacement : **Texture**
Color Space : Linear + sRGB

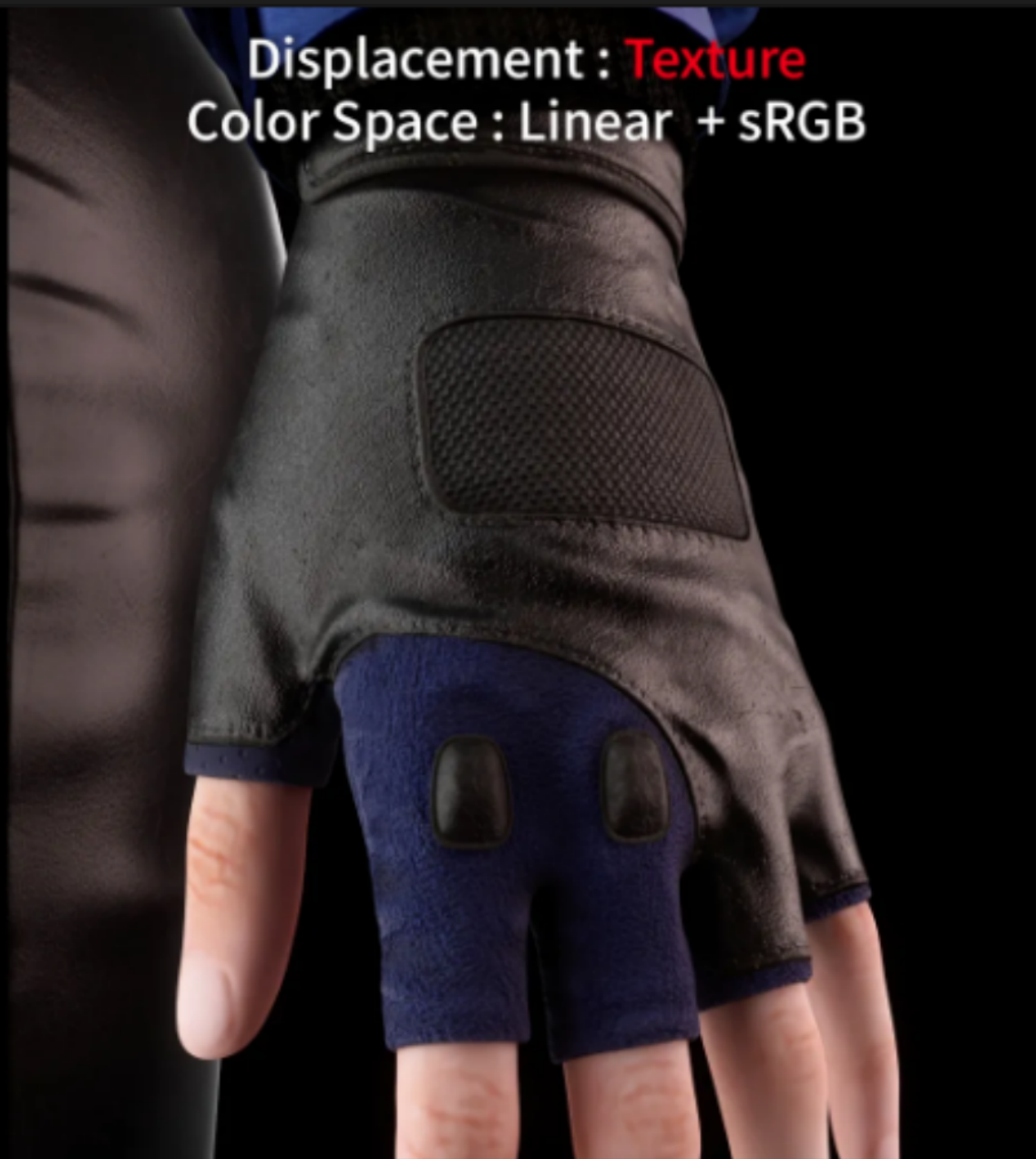


디스맵 옵션중 기본 텍스처 모드를 버텍스 모드로 바꾸자 모션블러가 적용 되는 것을 확인 하지만 디스맵 의 영향이 비정상적으로 강해지거나 찌그러지는 현상

Displacement : **Vertxe**
Color Space : Linear + sRGB



Displacement : **Texture**
Color Space : Linear + sRGB



마찬가지로 디스맵의 일그러짐 기본 수치로는 의도된 디스맵의 현상을 유지하기 어려움 기존 작업시 디스맵도 모션블러가 작동 되었는데 왜 안 될까?

이유는 카메라 모션블러의 영향으로 오브젝트가 블러를 먹은거였던거고 정확한 오브젝트의 모션블러 수치는 아니었다

- 1. 텍스처 모드 : 셰이더 레벨에서 시각적으로 변형된 것처럼 보여주는것 빠르고 가볍지만 실제 지오메트리에는 변형 없음
- 2. 버텍스 모드 : 실제 지오메트리 변형 > 모션블러 완전 지원 (카메라 + 오브젝트) 하지만 고해상도 메쉬에만 효과적 (샵디비전 or 디스 노드에서 샵디 추가) 메모리 과다 사용

Displacement : **Vertex**
Color Space : **Non- Color Data**



=

Displacement : **Texture**
Color Space : **Non- Color Data**

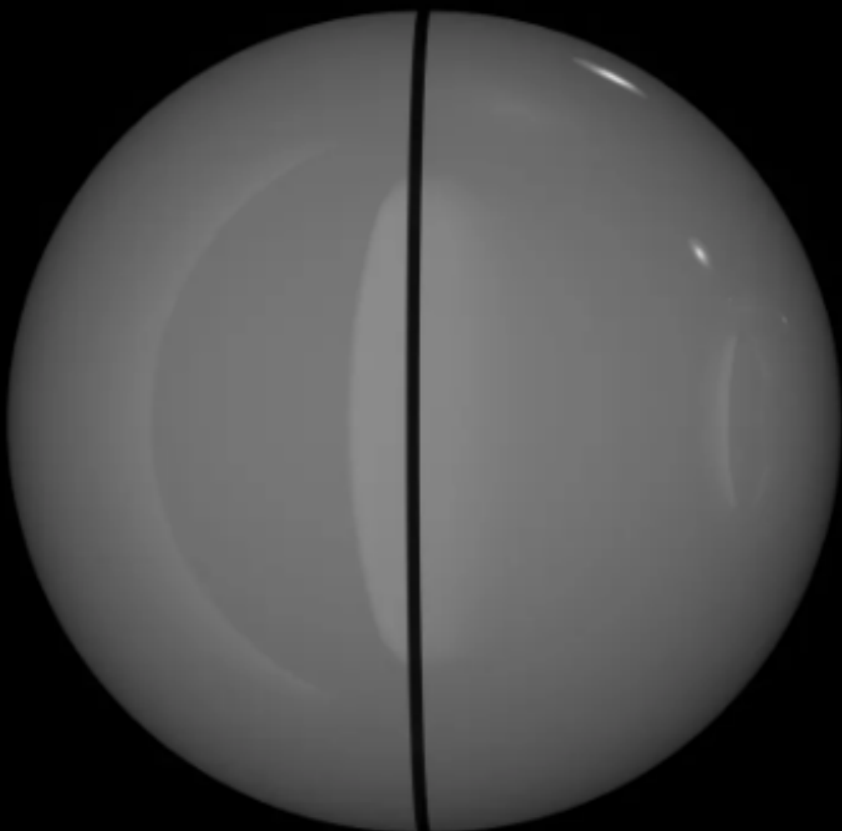


색공간 변경 Lineare +sRGB >>>> Non color data

- 1. 결과 : 기존수치의 모델링 현상 유지/ 디스맵 수치가 의도대로 작동됨 모션블러 작동한다
- 2. 단점 : Vram 용량을 조금더 많이 소모

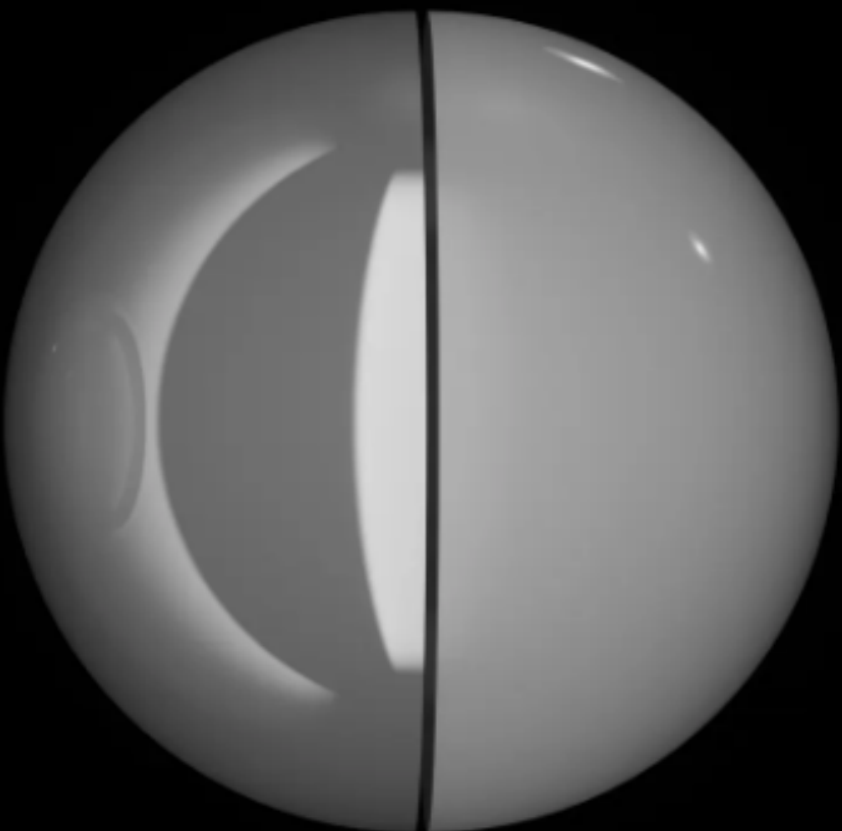
근본적인 원인 분석

왜 Linear sRGB (Gamma: 1) 와 Non Data는 선형적인 Y=1 인 데이터 값인데 왜 수치가 다른 걸까?
우리가 보기엔 같은 선형적인 그래프지만 옥테인은 '컬러값으로 받아들이냐', '데이터 값으로 받아들이냐'에서 부터 형식이 달라짐



Albedo

RgbSpectrum : 0.5 Float Texture : 0.5
R:128, G:128, B:128



Metallic

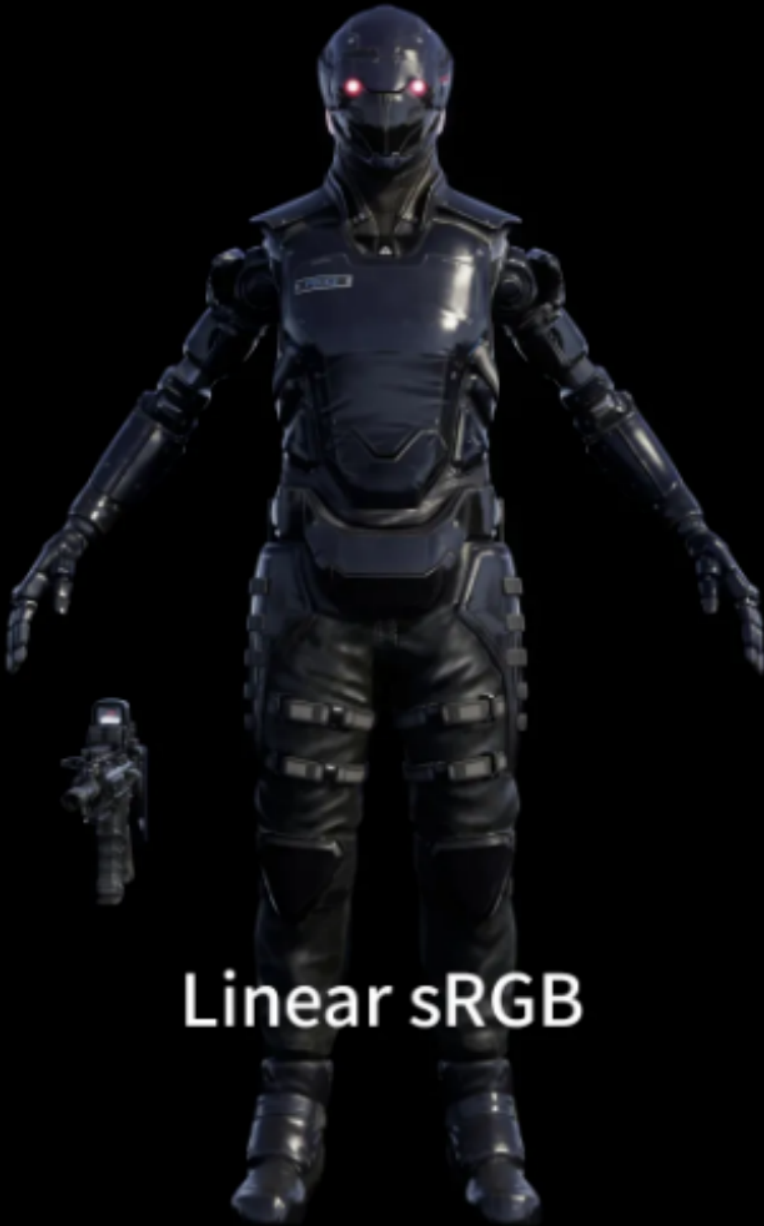
RgbSpectrum : 0.5 Float Texture : 0.5
R:128, G:128, B:128

위처럼 알베도 채널에 컬러값과 수치값을 0.5로 맞췄을땐 같은 중간지점 회색으로 보이기 때문에 차이를 느끼기 어렵지만 수치값으로 측정되는 메탈릭 채널에서는 서로 다른결과를 보여줌

- 1. Float : 정확히 0.5로 해석되어 Octane이 의도한 물리기반 수치로 작동
- 2. RGB Spectrum : 컬러 텍스처로 간주되며 내부적으로 RGB 각각을 평가하거나 평균내는 방식으로 사용해 정확하게 0.5로 동작하지않음

Linear (Gamma 1)로 두면 픽셀 밝기는 Non Color와 동일하게 보임 하지만 여전히 Color Texture 로 분류돼서 색공간 (Texture Data) 로 인식됨 반면 Non Color Data로 하면 내부적으로 “이건색이 아니라 수치!” 라고 완전히 분리해서 렌더파이프라인에서 색관련 연산(감마,톤매핑)을 완전히 제외

여러가지 수치값 (Non Color Data) 로 바꿔야 하는 채널이 많은데 눈으로 보기엔 어떤것이 정답인지 몰라서 더 연구를 해야함 하지만 디스맵은 일관적으로 Non Data Color 하는것이 정확한 수치값이다 (버텍스 모드나 텍스처 모드 둘다 동일하게)



Linear sRGB



sRGB, Non Color Data

기존 메터리얼 작업은 잘못된 정보여서 항상 텍스처 노드를 연결하고 그라디언트나 수치를 수정해야했었음



Linear sRGB



sRGB, Non Color Data

결론

- 1. Albedo,Color = sRGB
- 2. Roghness , Specular, Nomal map 등 = Non Color Data

Legacy Gamma는 구식 방식으로 , 색상 오류 유발 가능성 있음.

✓ **각 설정 요약**

항목	처리 방식	VRAM 사용량	주로 쓰는 곳
Normal	RGB 풀컬러	가장 많음	컬러 텍스처 (피부, 옷 등)
Float	Grayscale	적음	Roughness, Bump, AO 등
Alpha	Alpha 채널만 사용	매우 적음	투명도 마스크

⚠ **실무에서 주의할 점**

- 노멀맵은 **RGB** 데이터를 갖기 때문에 **Normal** 로 설정해야 합니다.
→ 단, **Height Map**이나 **Roughness Map**은 Float로 설정하세요.
→ 이렇게 하면 최대 수십~수백 MB의 VRAM을 절약할 수 있습니다.
- **Float** 은 자동으로 **RGB** → 흑백 변환해주기 때문에 별도 변환 없이 쓸 수 있습니다.
- **Alpha** 는 투명도 정보만 쓸 때 유용하며, 컬러 데이터는 완전히 무시됩니다.

✓ **사용 예시 정리**

텍스처 종류	Color Space	Image Type 설정
알베도(Base Color)	sRGB	Normal
노멀맵(Normal Map)	Non-Color Data	Normal
러프니스(Roughness)	Non-Color Data	Float
범프맵(Bump)	Non-Color Data	Float
컷아웃 마스크	Non-Color Data	Alpha

PBR 파이프라인 기준

물리 기반 렌더링(PBR)에서는 컬러 텍스처(알베도, Base Color)는 감마가 적용된 sRGB 이미지로 넣는 것이 표준입니다.

Roughness, Metalness, Normal 등은 선형 모드로 불러와야 하고요.

따라서 컬러 텍스처는 "sRGB" 로 설정하는 것이 올바른 방식입니다.

Legacy Gamma는 구 버전 호환용

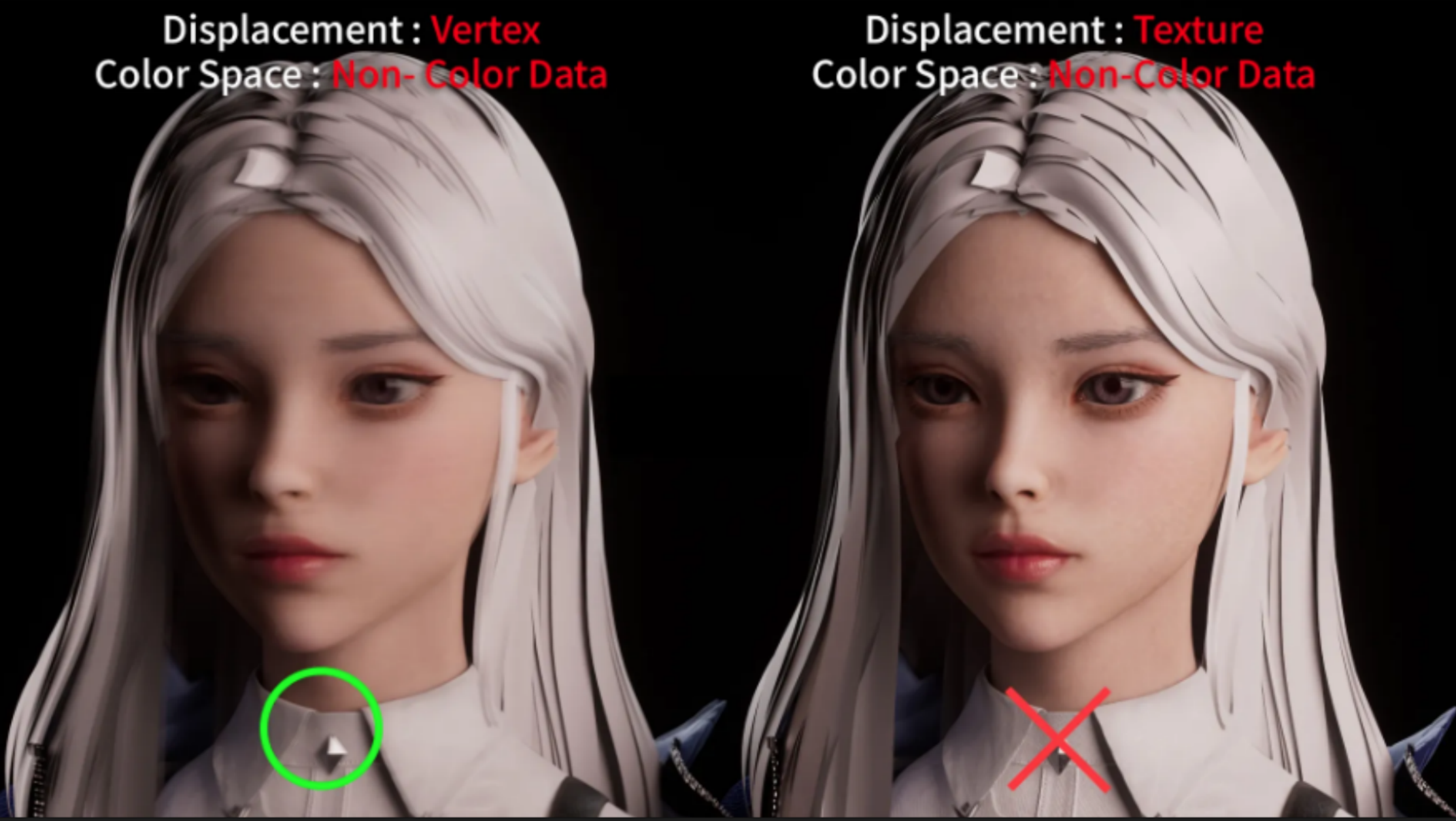
"Linear sRGB + Legacy Gamma"는 예전 Octane 버전에서 감마 보정을 수동으로 맞췄던 방식입니다.

지금은 비표준이며, 새로운 프로젝트에는 사용 비권장입니다.

사용 시 색이 이중 감마 처리되어 어둡거나 밝게 깨질 수 있음.

직접 감마 2.2를 곱해서 저장하는 방식보다는, 그냥 sRGB 모드에 맡기는 방식이 더 안전하고 표준에 맞는 처리입니다.

즉, 눈으로 보기에 큰 차이가 없어도, 렌더링의 정확성이나 후속 작업에서 오류를 줄이기 위해선 표준 설정을 따르는 게 중요합니다. (GPT)



결과

1. Displacement Map Color Space는 항상 Non Color Data
2. 지형이나 움직이지 않는 오브제에게는 Texture 모드! (Vram 절감)
3. 움직이거나 변형되는 물체는 버텍스 모드로!