DEVOCEAN

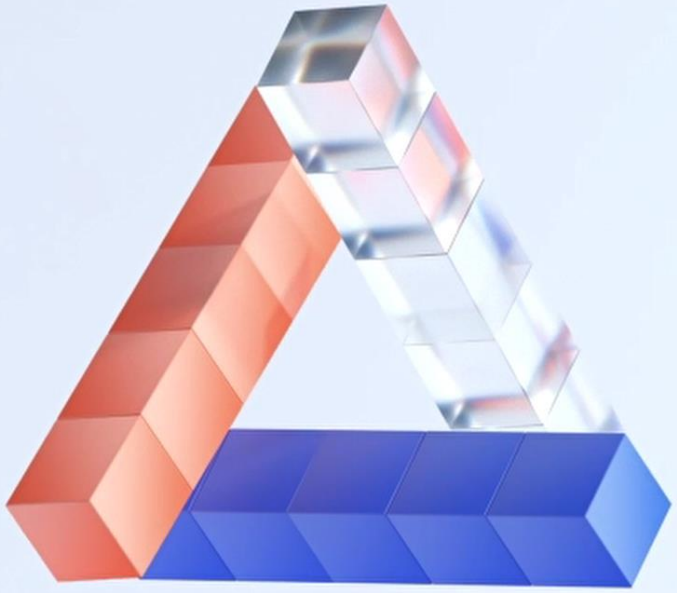
메타버스 세상의 표현 : Volumetric Capture 기술

@ Jump Studio

임국찬
SK telecom,
Metaverse Contents 개발,
Jump Studio, Stage Tech.

신동엽
SK telecom,
Metaverse Contents 개발,
Jump Studio, Stage Tech.

DEVOCEAN

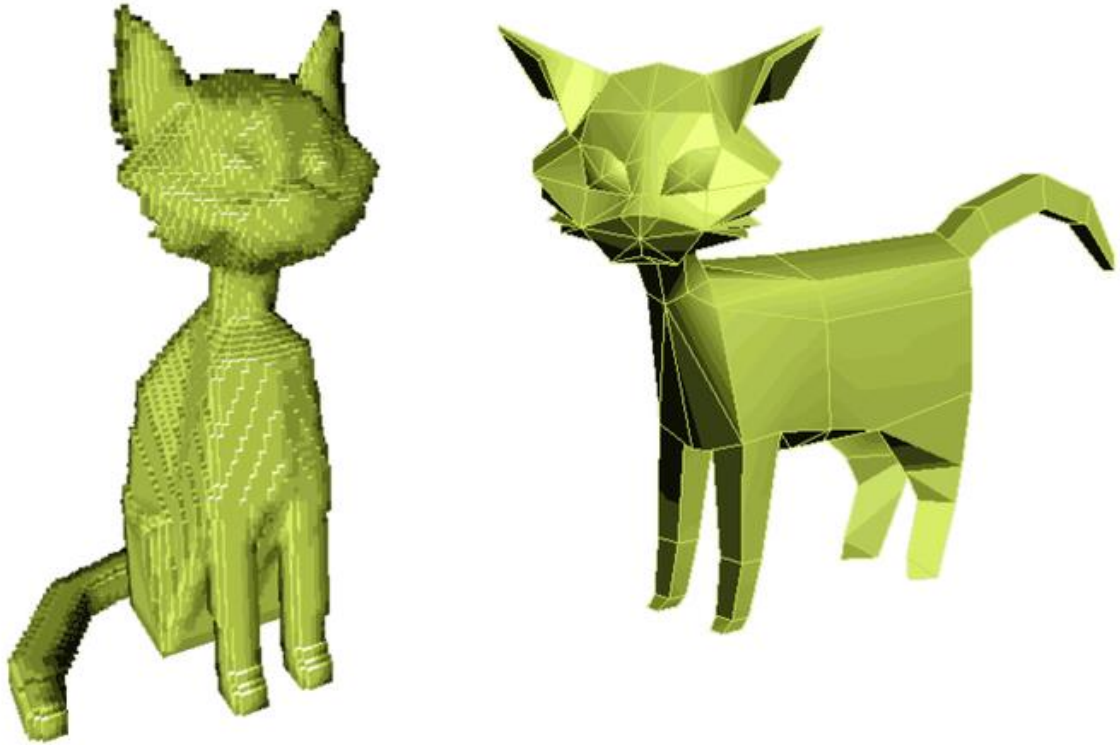


1. Volumetric Capture 기술 Overview

2. Workflow에 따른 기술 소개
- Capture (@Stage)
- Preprocessing
- Point generation
- Meshing and texuring
- Mesh tracking
- Compression and encoding

Volumetric Capture

Volumetric 솔루션에서 고려할 사양/방식

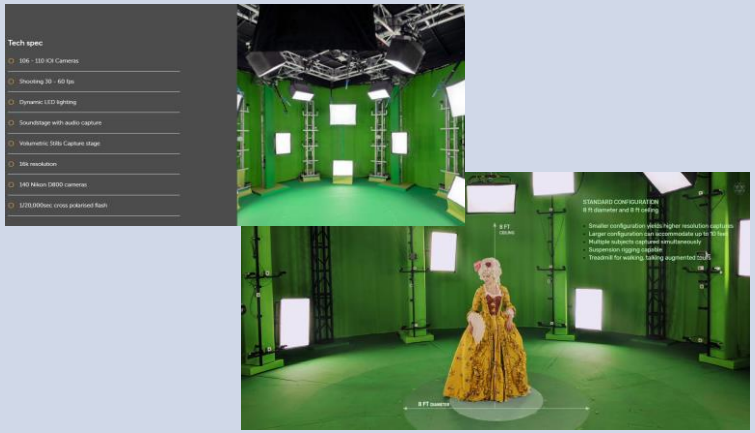


Voxel / Polygon

Capture Spec.

Volumetric Capture

솔루션 종류 및 비교

Microsoft	8i	4D View
		
System Space: D7.6m x H6.1m Capture Volume: D2.5m x H2.5m Camara: 106ea (53IR + 53RGB) 4k Capture Rate: 30~60 fps Recording Capacity: Customize Storage Capacity: Customize (Azure 가능) Processing Rate: Customize, (1day/25m by Azure) format: OBJ (10GB/30s), MP4(10MB/s) Plug-in: Unity, Unreal	System Space: D10m x H6.1m Capture Volume: D2.5m x H2.2m Camara: 60ea, 4k Capture Rate: 15, 30fps Recording Capacity: 5.5h (30fps) format : HVR(Voxel), Polygon(GLB)	System Space: D9m x H5m Capture Volume: D4m x H2.4m Capture Rate: up to 60fps Recording Capacity: 110m (30fps) Storage Capacity: 20~30h (30fps) Processing Rate: 3~10h/m format: 4DS, ABC - 300 frame, 200MB - 30k Polygon
SK Telecom, Jump Studio	K-실감스튜디오(NIPA)	

Volumetric Capture

Microsoft MRCS(Mixed Reality Capture Studio)의 특징

- ✓ RGB, 적외선(IR), 실루엣 정보를 활용하여 depth map 생성
- ✓ 이상치 제거를 통해 효과적인 PSR(Poisson Surface Reconstruction) 클리핑
- ✓ 인지적으로 중요한 주요 관심 영역(얼굴, 손가락 등)을 지정하여 고품질의 mesh 및 texture 생성
- ✓ 시간적으로 일관된 시퀀스 생성을 위한 Mesh 추적 기술 (효과적인 Key frame 지정)
- ✓ MPEG Format encoding

Processing Pipeline

Camera Capture

Point Cloud

Mesh Generation

Mesh Smoothing

Texturing

Final

System overview 전체 과정 요약

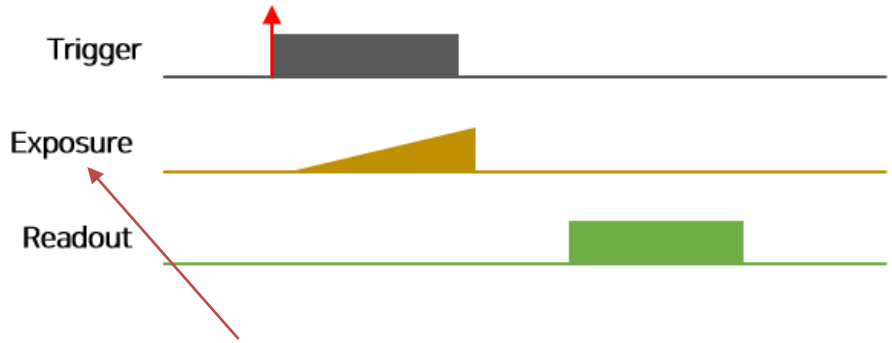
- ✓ **Capture and preprocessing**
- ✓ **Point Cloud Generation**
- ✓ **Mesh / Smoothing**
- ✓ **Texturing (Colorfill)**

Capture

카메라 영상 제어 신호

연속모드 (Free-Run mode)

Trigger Mode

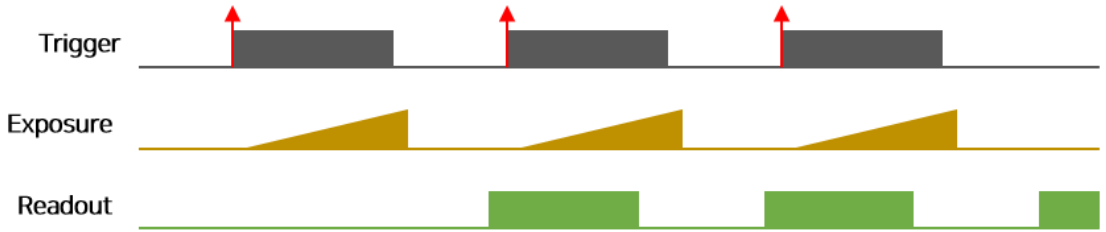


노출신호 :

고정시간 노출 / Trigger Pulse 노출

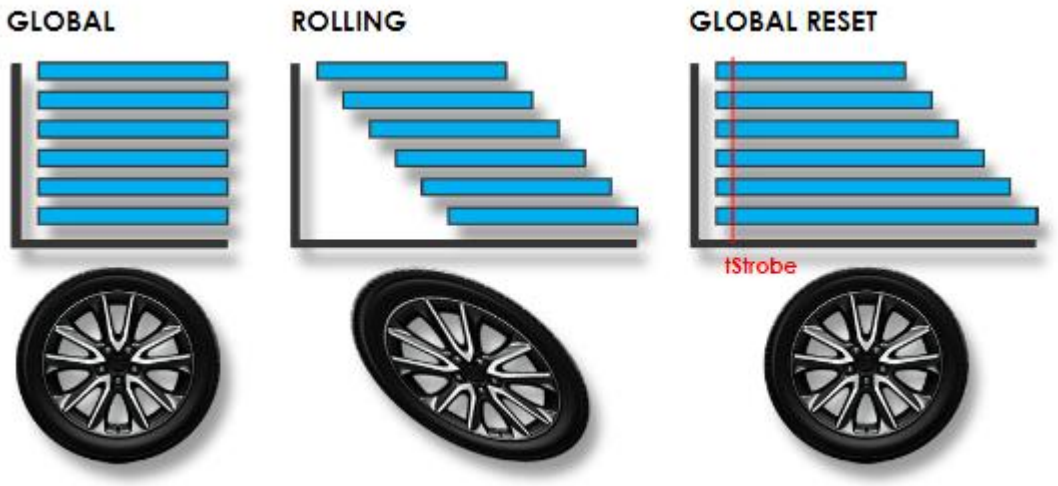
overlap off

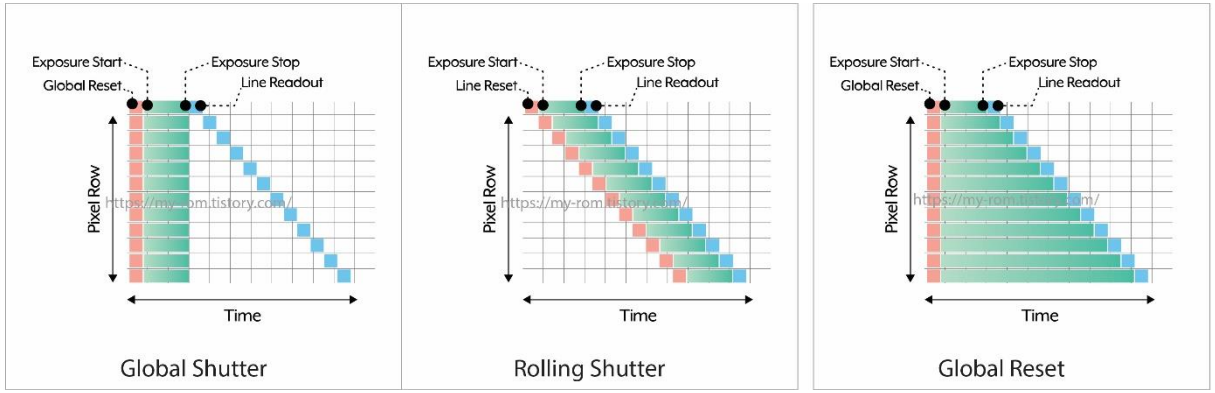
overlap on



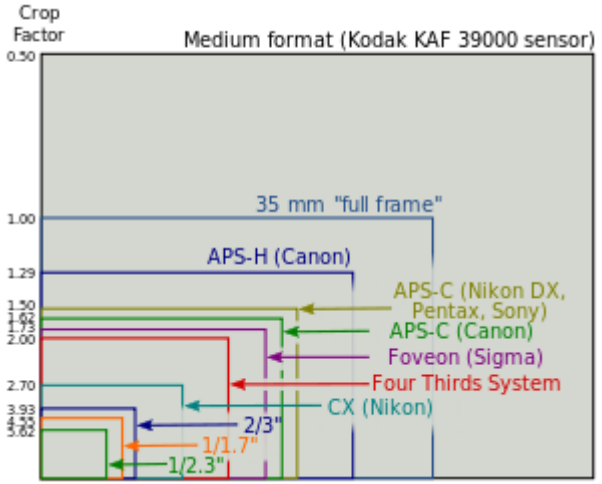
Capture

카메라 전자 셔터 Type



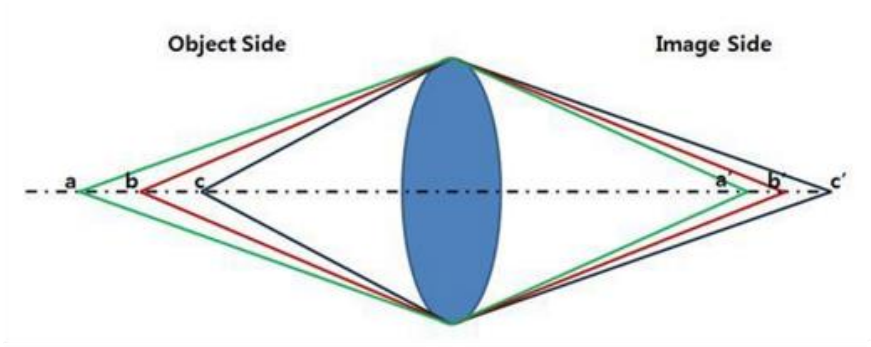


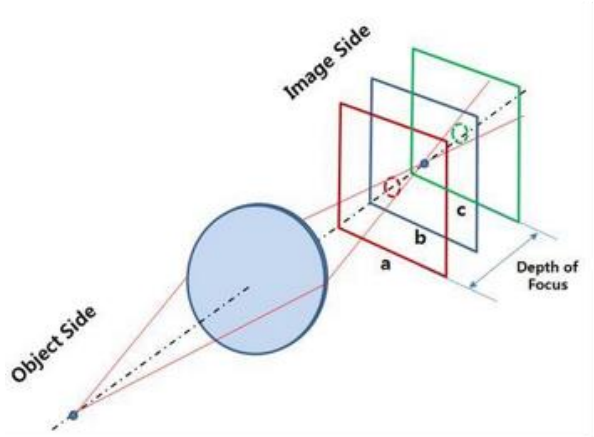
카메라 Mount

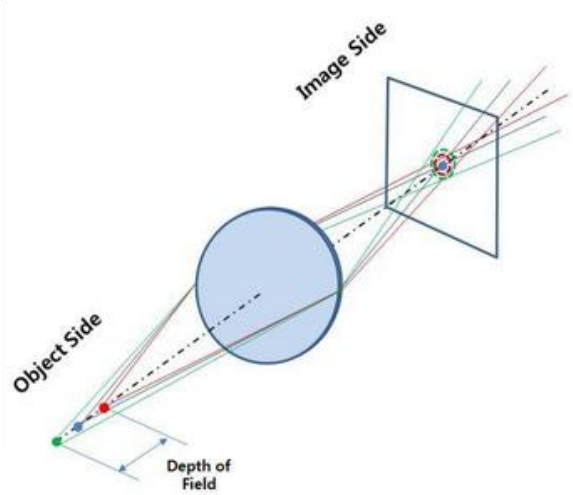


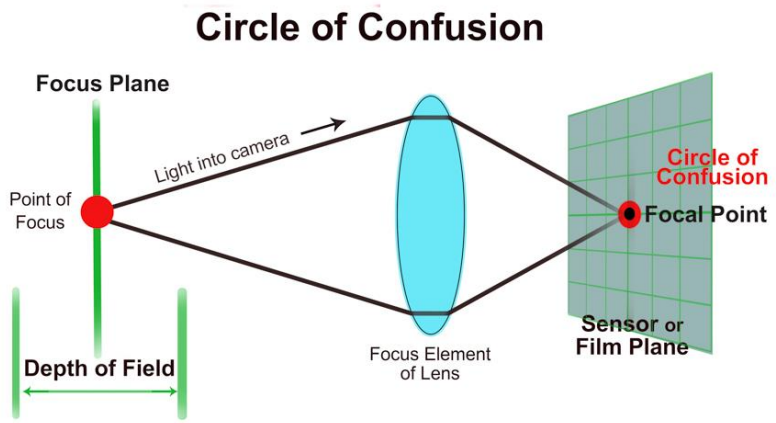
Capture

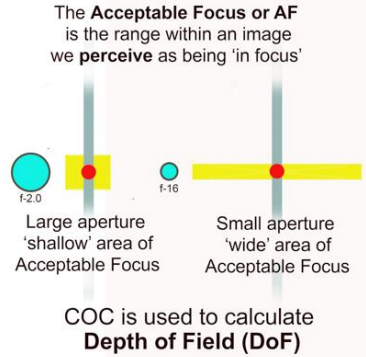
광학 사양 (EFL / FOV / DoF / CoC)

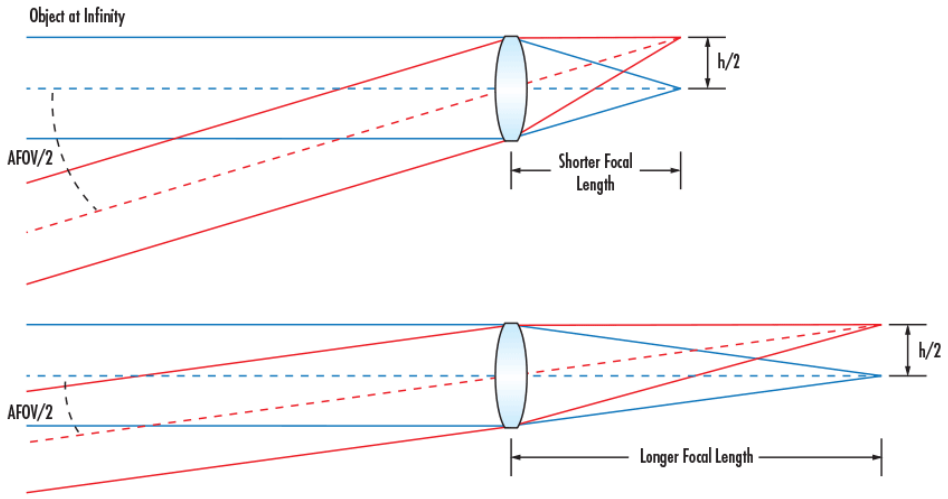








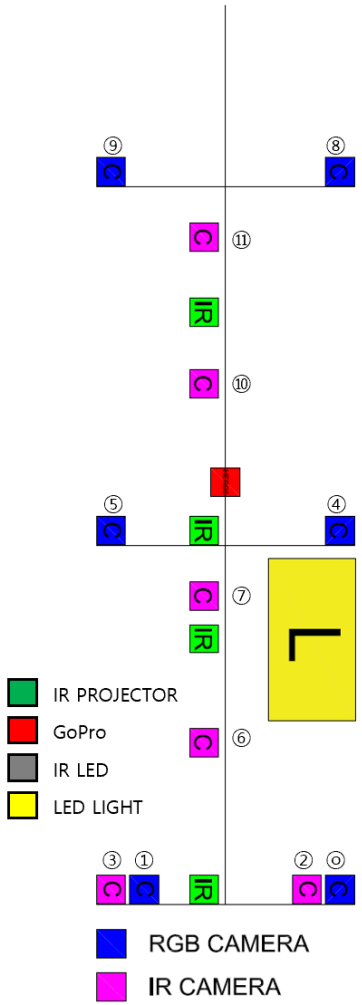


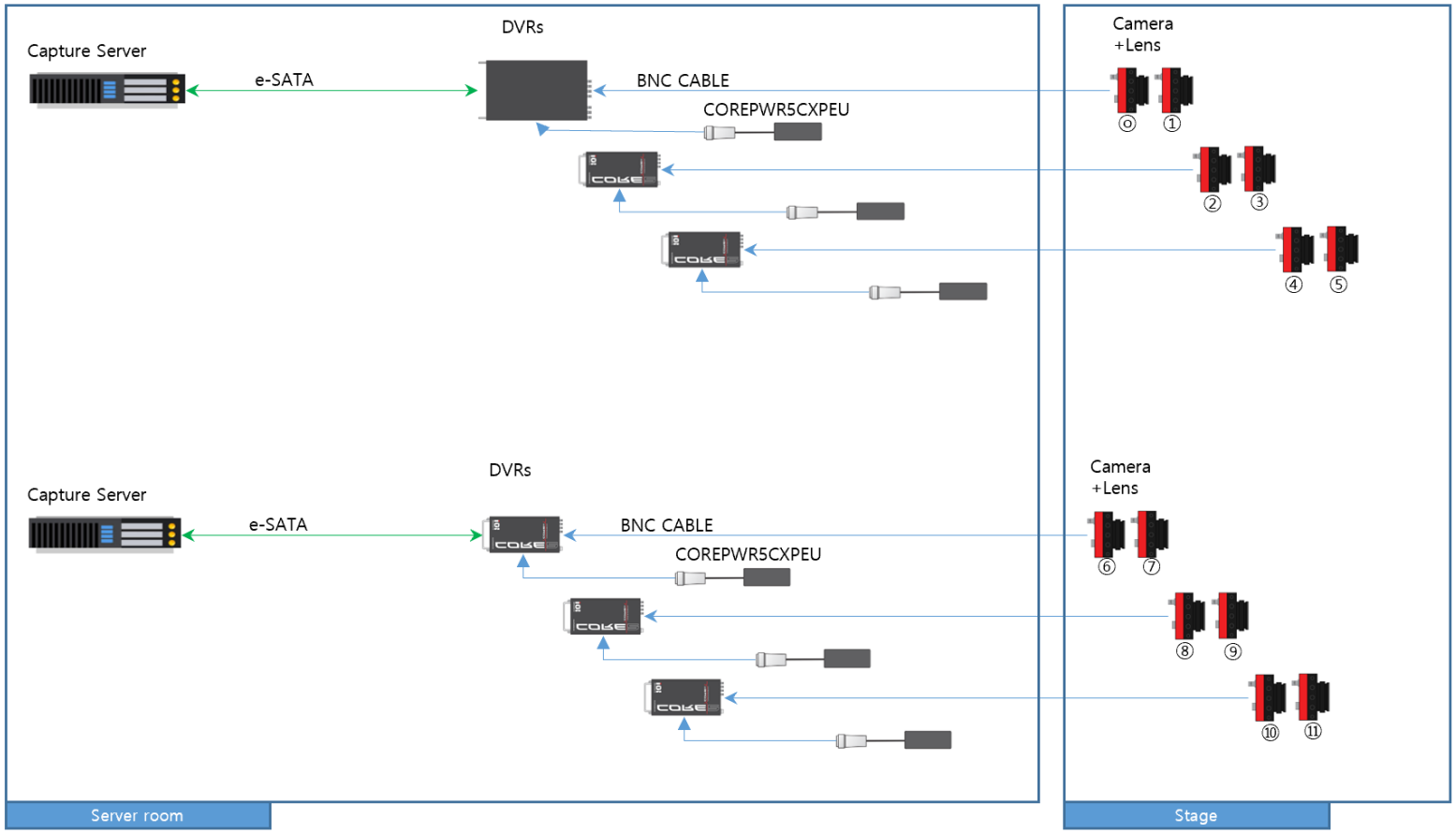


Capture

Capture System

Tower 배치





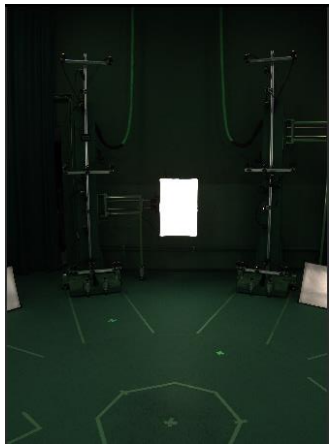
Capture

Calibration

- calibration chart 촬영으로 카메라의 공간 좌표 획득

Background

- hue 선택으로 chroma key 제거
- binary mask threshold 로 배경 제거



Light Calibration(W/B) with Sphere

- White Balance 셋업 (camera low level)

Point Generation

[Pod = 2 RGB + 2 IR] * 3 = Tower

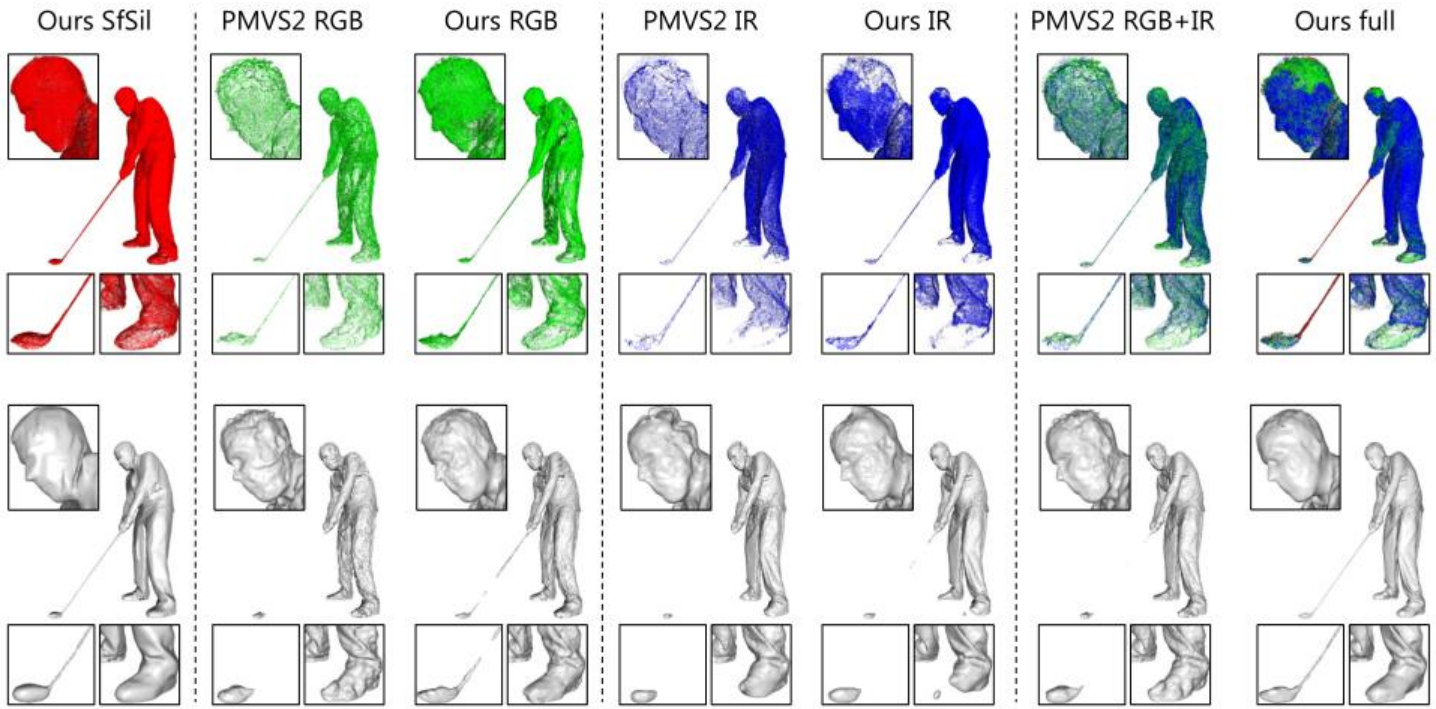
- Pod 단위로 Depth Map 생성 및 신뢰도 추정

  IR과 RGB에 대해 신뢰 가중 평균을 계산

  동일 물리 공간에 해당하는 기준 영상 선택하고 각각의 보기(IR, RGB 각)에 대해 집계
  이를 다면적으로 확장

Point Generation

RGB Stereo, IR Stereo, Shape from Silhuoette 의 최적 조합으로 3D 재구성



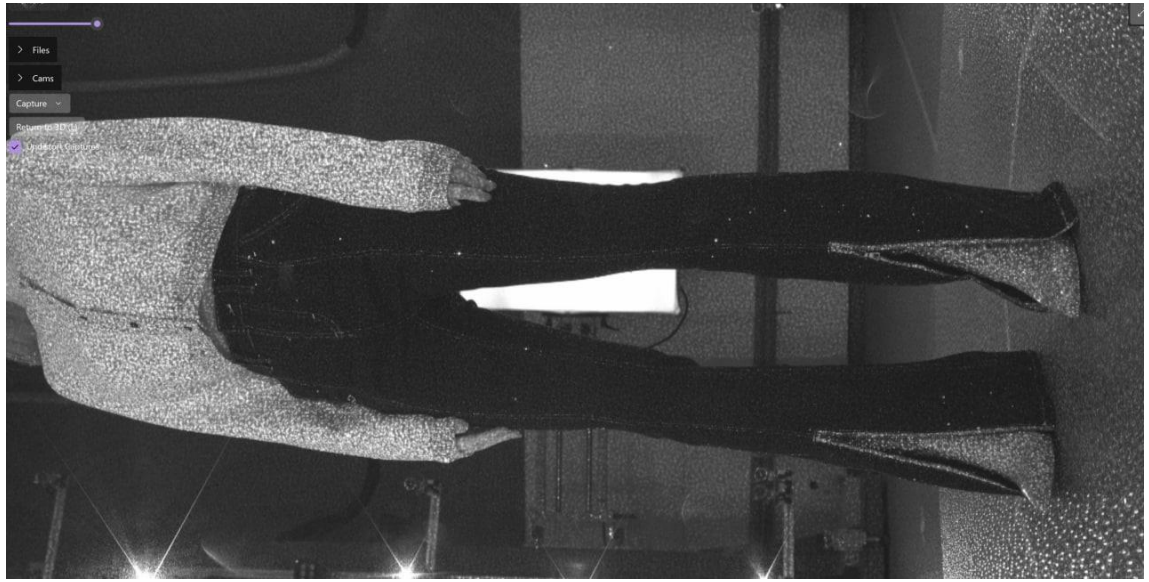
전체적인 체형 : IR 우세

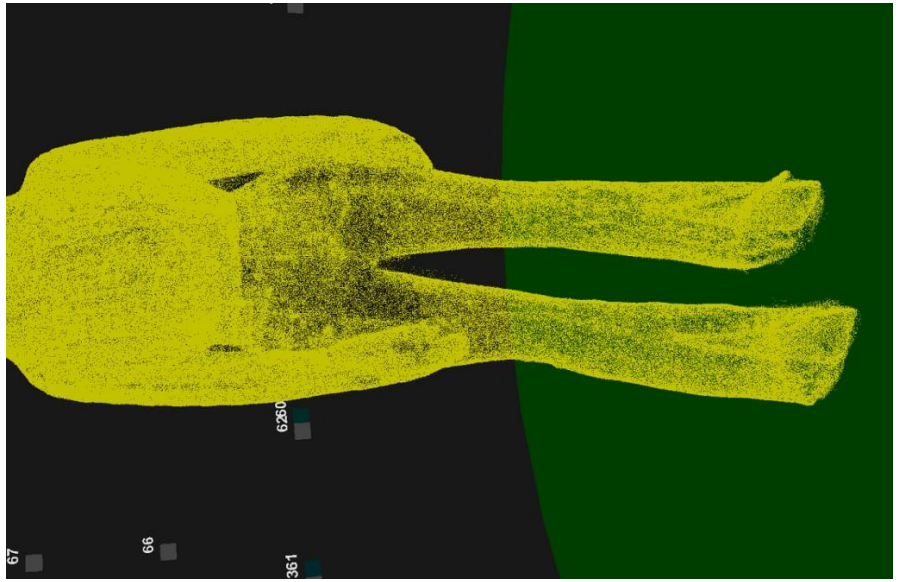
머리카락/신발 등 : RGB 우세 (IR이 흡수되는 영역)

골프채 : 실루엣 우세 (얇은 물체)

Point Generation

Case : IR을 흡수하는 형태의 표면 촬영시





(좌) 특정 angle에서의 IR 카메라 capture 이미지

바지 부분의 IR이 흡수되어 mesh 형성이 불완전해짐

(우) 실루엣 옵션을 적용할 경우, 바지 형상을 복원함

다른 부분의 noise가 다소 증가할 수 있음

Meshing and Texturing

- 포인트 클라우드로부터 Mesh 생성
- 잘 알려진 Screened PSR (Poisson surface reconstruction) 알고리즘
- 추가 constrain을 통해 보다 세밀한 Mesh 구성
- Mesh 의 Noise 제거
- 주요 관심 영역을 설정하여 해당 부분에 더 많은 Mesh 및 텍스쳐 할당

Meshing and Texturing

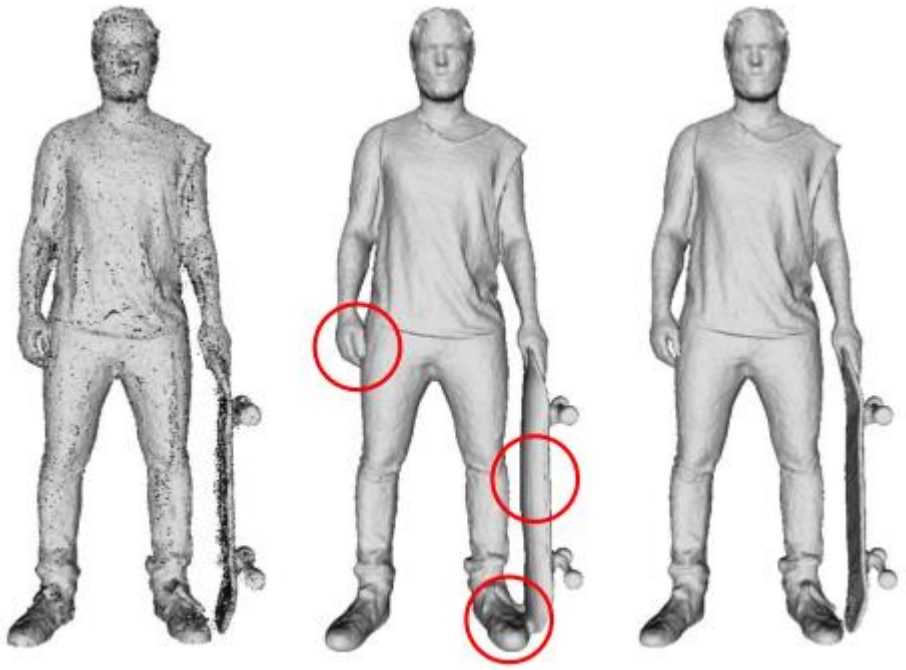
Screened Poisson surface reconstruction(PSR) 은 포인트 클라우드 매싱에 폭 넓게 사용되는 알고리즘

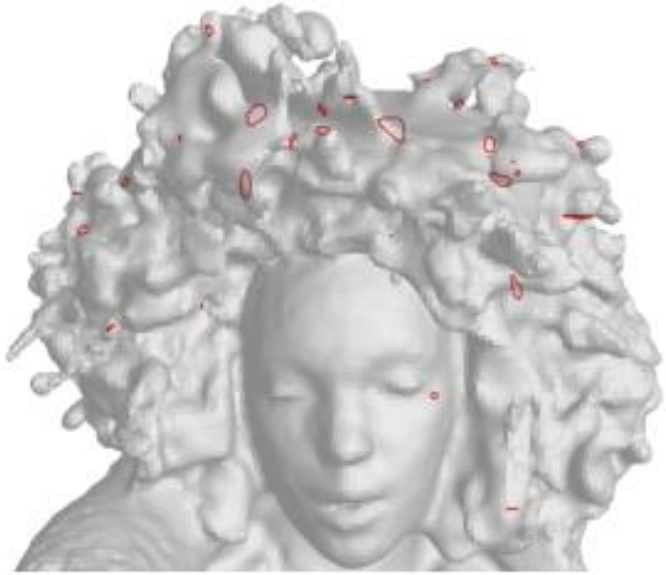
- 장점 : 빠르고, 불완전한 데이터에 대한 복원력이 유효함

Screened PSR에 필요에 따라 background mask 와 비교하여 객체의 외부에 있을 경우,

포아송 솔버의 스칼라 계수를 음수로 확정하는 작업을 통해 Hull-constrained PSR을 적용함

객체의 edge를 더 효과적으로 구현함



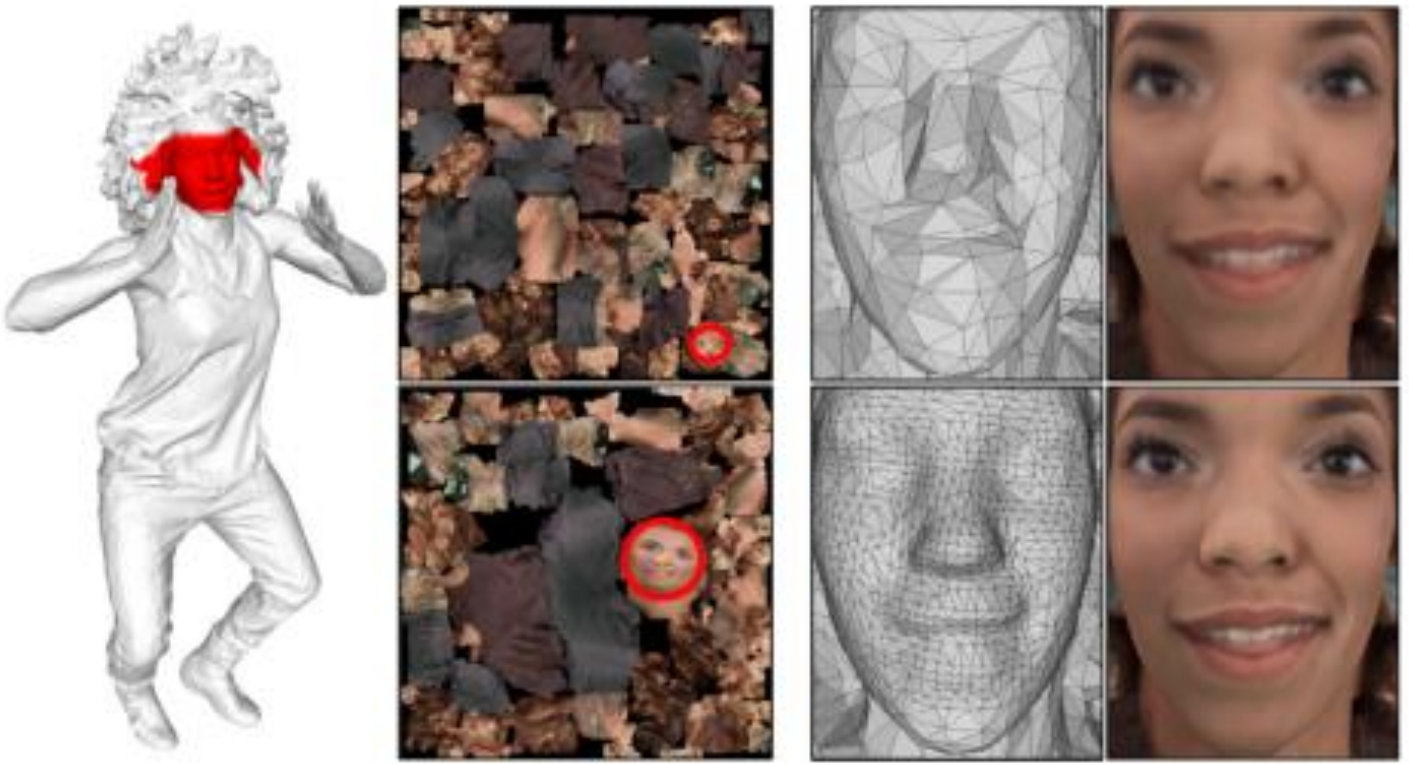


Topology Noise 제거

Meshing and Texturing

Importance 영역 지정

- 통상의 연출에서 인지적으로 중요한 요소(얼굴, 손가락 등)의 영역을 지정
- 이미지 객체 추적을 통해 영역을 자동 지정하거나, 선택하여 Mesh 및 Texture의 UV map 을 효과적으로 지정



Importance 영역으로 얼굴이 지정되었을 때,

1) Mesh의 density 증가
2) UV map 에서의 Texture 비율

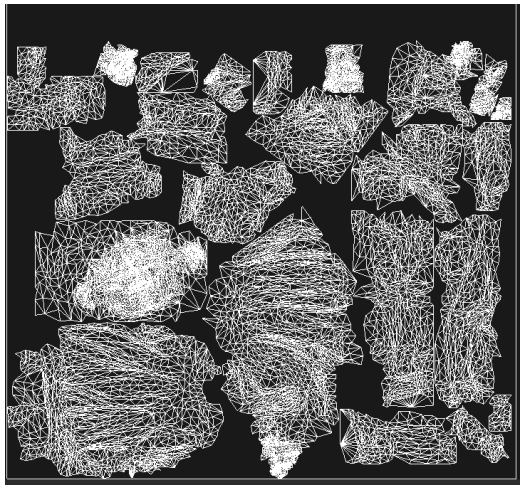
조정되어 최종 품질이 크게 향상됨

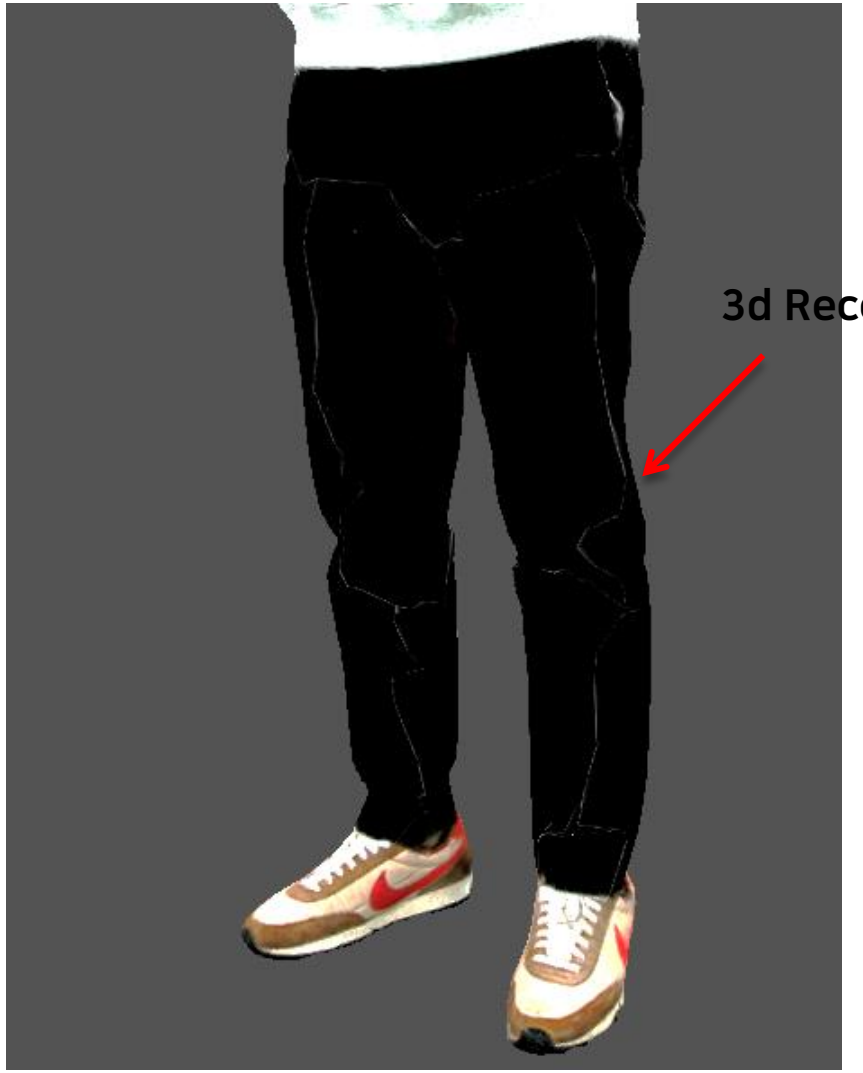
Mesh Tracking

3D 영상 구성

- Mesh / Texture / UV

- UV 는 Mesh와 Texture의 공간좌표 정보





Mesh Tracking

Algorithm 1 Coherent Tessellation via Mesh Tracking

procedure COHERENTTESS($\{\mathcal{M}\}$) ▷ Input meshes
  $\{\Psi\} = \emptyset$ ▷ Temporal mesh sequences
  $\{\mathcal{S}\} \leftarrow$ keyframe feasibility for all frames ▷ **Section 7.1**
  Sort $\{\mathcal{S}\}$ into priority queue Q ▷ Key: i, Value:$\mathcal{S}_i$
  while $Q \neq \emptyset$ **do** ▷ Until we cover all frames
    $i \leftarrow Q.RemoveHighest()$ ▷ Most feasible frame
    $K_i \leftarrow \mathcal{M}_i$ ▷ Set as keymesh
    $\{\widehat{\mathcal{M}}\}_f = \text{MeshTracking}(i,+1)$ ▷ Forward Tracking
    $\{\widehat{\mathcal{M}}\}_b = \text{MeshTracking}(i,-1)$ ▷ Backward Tracking
    $\{\Psi\} \leftarrow \{K_i, \{\widehat{\mathcal{M}}\}_f \cup \{\widehat{\mathcal{M}}\}_b\}$ ▷ Save subsequences
  return $\{\Psi\}$ ▷ Return temporal sequences

procedure MESHTRACKING(i,δ) ▷ i: keyframe, δ: offset
  $\{\widehat{\mathcal{M}}\} = \emptyset$ ▷ Initialize subsequence
  while $Q.HasEntry(i+\delta)$ **do** ▷ Proceed if uncovered
    $\widehat{\mathcal{M}}_i \leftarrow K_i$ ▷ Initialize transient mesh
    $\widehat{\mathcal{M}}_{i+\delta} \leftarrow \text{DeformInto}(\widehat{\mathcal{M}}_i, \mathcal{M}_{i+\delta})$ ▷ **Section 7.2**
    $e \leftarrow \text{FittingError}(\widehat{\mathcal{M}}_{i+\delta}, \mathcal{M}_{i+\delta})$ ▷ Compute error
    if $e < \epsilon$ **then** ▷ ϵ: error threshold
      $\{\widehat{\mathcal{M}}\} \leftarrow \widehat{\mathcal{M}}_{i+\delta}$ ▷ Save the tracked mesh
      $Q.Remove(i+\delta)$ ▷ This frame is covered
      $i \leftarrow i+\delta$ ▷ Move on to the next frame
    else ▷ Error above threshold
      return $\{\widehat{\mathcal{M}}\}$ ▷ Keymesh no longer usable
  return $\{\widehat{\mathcal{M}}\}$ ▷ Reach end of sequence

효율적인 압축을 위한 과정

앞선 과정에서 Mesh는 공간적으로는 일관성을 가지나

시간 축으로는 일관성이 없는 상태임

효과적인 압축을 위해서는 시간 축으로 일관성을 유지해야 함

특정 window 기간 안에서

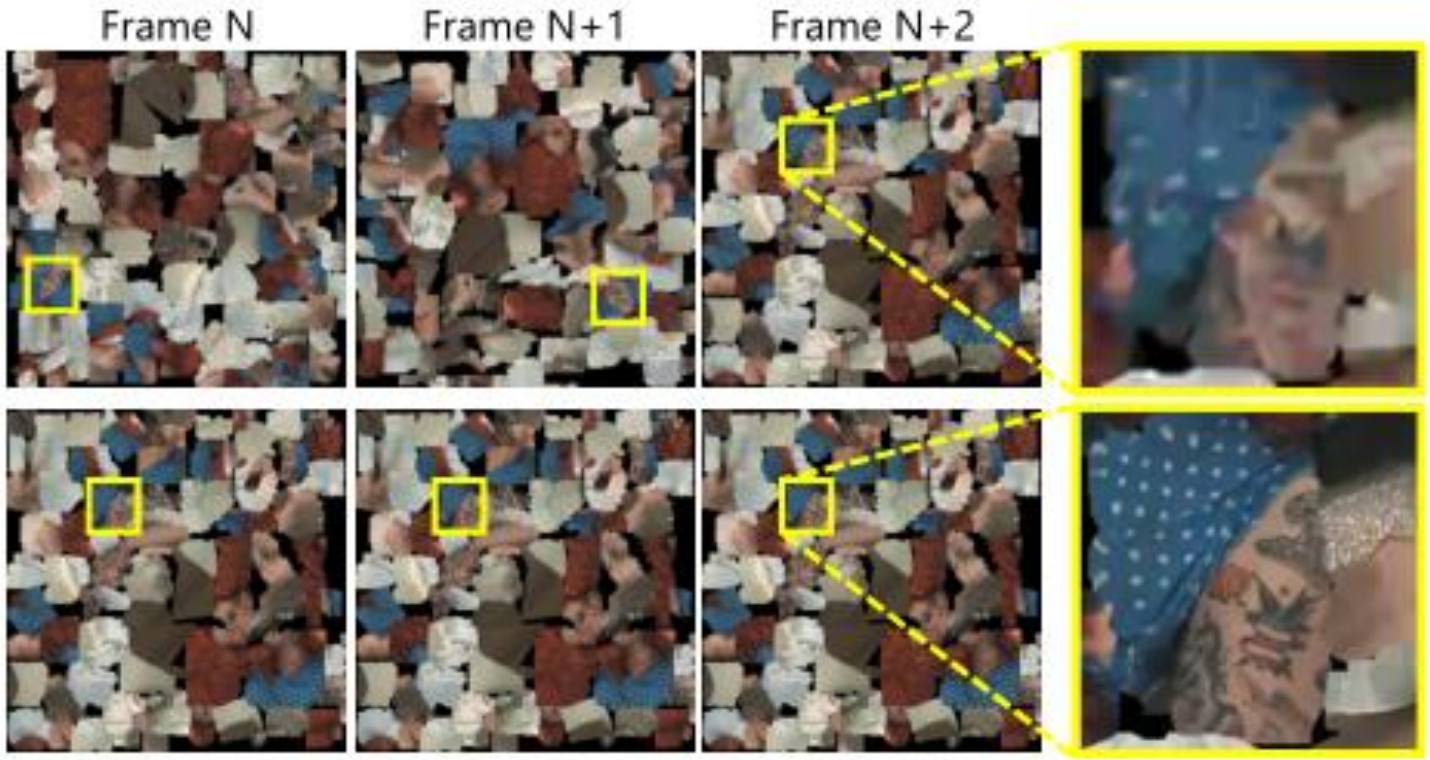
효과적으로 대표 frame(keyframe)을 설정하고

이를 바탕으로 UV map 이 생성된다면

영상 정보의 압축이 가능함.

Compression and encoding

- 스트리밍이 가능한 MPEG 포맷으로 출력
- NAL unit에 mesh를 embed 하는 형태 (MPEG 편집시 mesh 손실 없이 컨테이너 수정 가능)
- AAC / H.264 지원
- 비디오 압축의 핵심은 Mesh tracking으로 인한 시간 축으로 일시적 일관성을 갖는 UV 배치

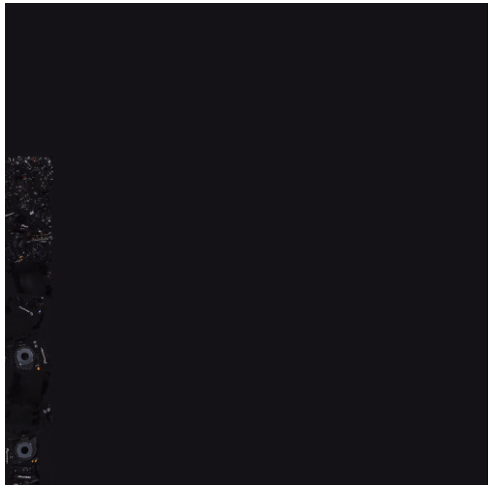


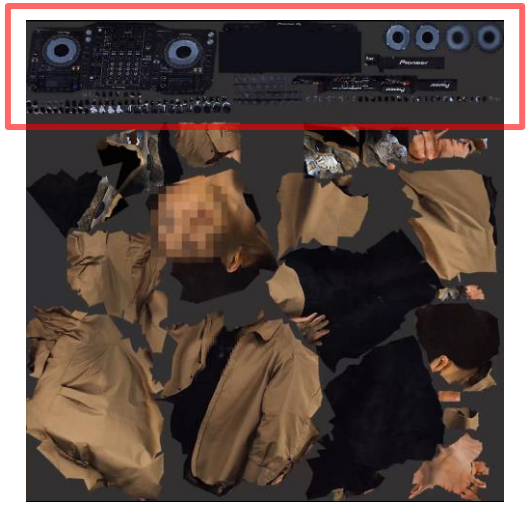
Mesh tracking에 따른 압축 후 품질 차이

Props / Clean-up

- Mesh / Texture의 수정을 통해 Props을 추가하거나 이미지 클린업을 수행
- Maya / Effter Effect 와 같은 상용 툴을 사용하여 수정

인물과 **props** 함께 **texture**





Props의 **texture** 정보를 분리

UV를 효과적으로 구성하기 위해 소품을 별도 처리해서 최종 output에 배치하면,
상대적으로 보다 중요한 영역에 mesh / texture를 할당하여 인지 품질을 향상할 수 있음.

Processing Pipeline

Camera Capture

Point Cloud

Mesh Generation

Mesh Smoothing

Texturing

Final

System overview 전체 과정 요약

- ✓ **Capture and preprocessing**
- ✓ **Point Cloud Generation**
- ✓ **Mesh / Smoothing**
- ✓ **Texturing (Colorfill)**

마치며

SK telecom

Thank you