3D Cinemagraphy

Presented By
추해수

Scenario



VS



Scenario



VS



흐르는 물

움직이는 화면

Scenario



VS



흐르는 물

움직이는 화면

Scenario_Animation

- 물을 흐르게 하는 기술 : Image to Animation



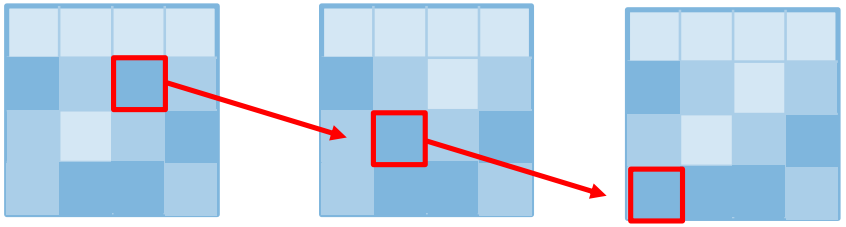
Animation = motion이 있다 = 픽셀의 이동이 있다

Scenario_Animation

- 물을 흐르게 하는 기술 : Image to Animation

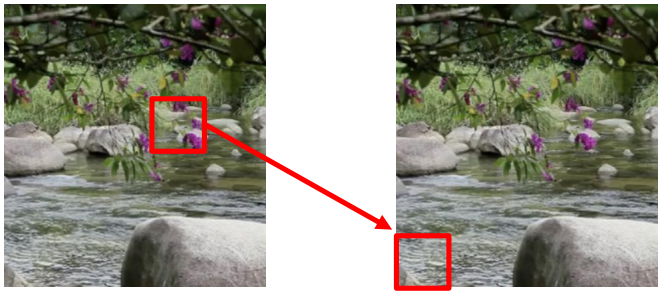


Animation = motion이 있다 = 픽셀의 이동이 있다

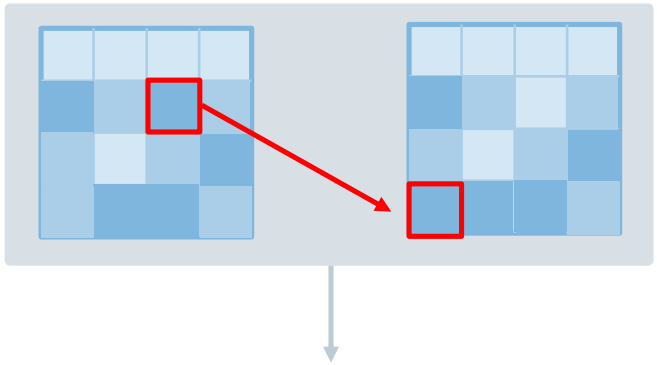


Scenario_Animation

여러 장의 still image



[image 1]　　[image 2]



픽셀의 이동 패턴

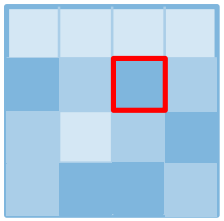
Scenario_Animation

여러 장의 still image

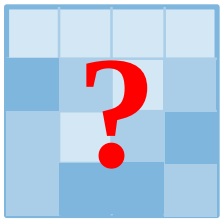
모든 픽셀에 대해 계산하여
Frame(image) 사이에서
각 픽셀의 움직임을 나타내는
Optical Flow

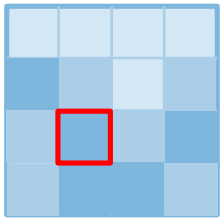


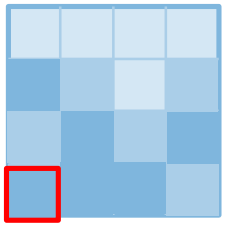
를 통해 만든 motion map



[image 1]







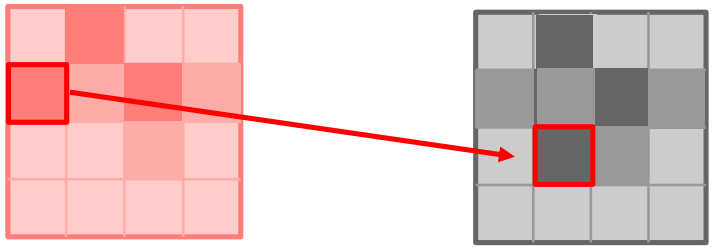
[image 2]

Scenario_Animation

한 장의 still image



[image 1]

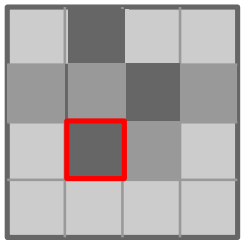


[image 1]

[image 2 가정]

Scenario_Animation

한 장의 still image



[image 2 가정]

Constant-velocity
(일정한 속도)

픽셀이 시간에 따라 이동할 때
일정한 속도 유지

Time-invariant
(시간 불변)

이미지의 각 픽셀에 대한
이동 벡터가 시간에 따라 일정하게 유지

등속 운동

: 물체가 운동을 할 때
속력과 방향이 변하지 않는
일정한 운동

Scenario_Animation

한 장의 still image

등속 운동

: 물체가 운동을 할 때 속력과 방향이 변하지 않는 일정한 운동

$$x(t) = x_0 + vt$$

- $x(t)$: 시간 t에서의 위치
- x_0: 초기 위치
- vt: 등속운동의 상수 속도

등속 운동의 위치 변화를 나타내는 수식

오일러 운동방정식

: 물체의 위치와 속도 간의 관계를 나타내는 방정식

$$x_{t+1} = x_t + v \cdot \Delta t$$

- x_{t+1}: 다음 시간 단계에서의 위치
- x_t: 현재 위치
- v: 등속운동의 상수 속도
- Δt: 시간 간격

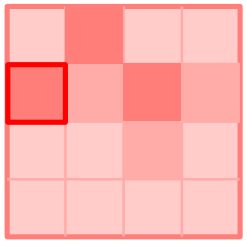
등속 운동의 위치 변화를 시간 간격 동안의 이동으로 나타내어 등속 운동의 위치를 오일러 운동방정식을 통해 모델링

Scenario_Animation

한 장의 still image



[image 1]



[image 1]

Motion Estimation



[motion map]

Scenario



VS



흐르는 물

움직이는 화면

Scenario_Novel View Synthesis

- 화면을 움직이게 하는 기술 : Novel View Synthesis





방향

각도

Novel View

: 새롭거나 이전에 관찰되지 않았던 시각적인 관점

***Parallax Effect**
(시차 효과)

: 멀리 있는 물체는 천천히 움직이고 가까이에 있는 물체는 빨리 움직임

Scenario_Novel View Synthesis



[RGB image]



[Depth map]

Depth (깊이) : 특정 지점이나 표면으로부터의 거리를 나타내는 값

Depth map : image의 각 픽셀에 해당하는 depth 정보를 표현한 map

Scenario_Novel View Synthesis

2D image를 3D 공간으로 복원 : 2D image는 단일 시점이기에



[Depth X]



[Depth O]

다양한 시점에서 장면을 볼 때 3D 공간에 표현된 깊이와 구조를 통해
카메라 각도와 방향에 따라 새로운 장면을 볼 수 있다

Scenario_Motivation



단일 image

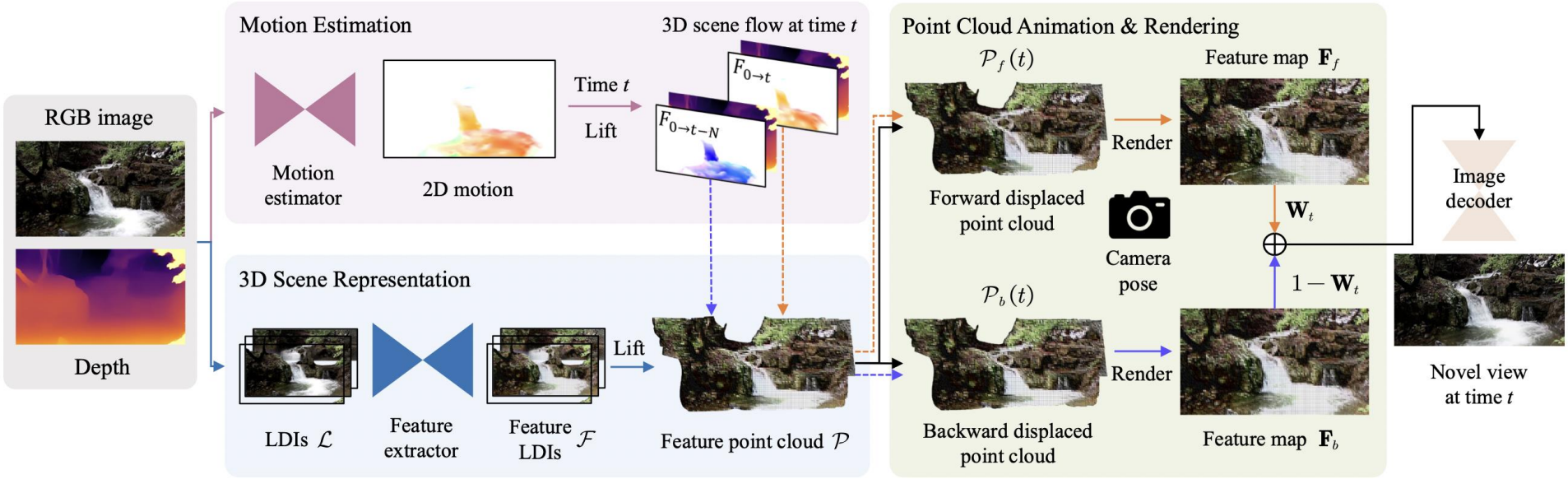
↓

animation + 카메라 motion = **cinemagraphy**

↓

!! in 3D space !!

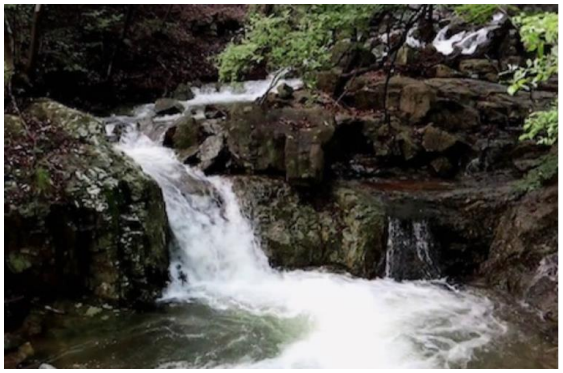
Method_Frame Work



3D Cinemagraphy from a Single Image (2023)

Method

input



[RGB image]

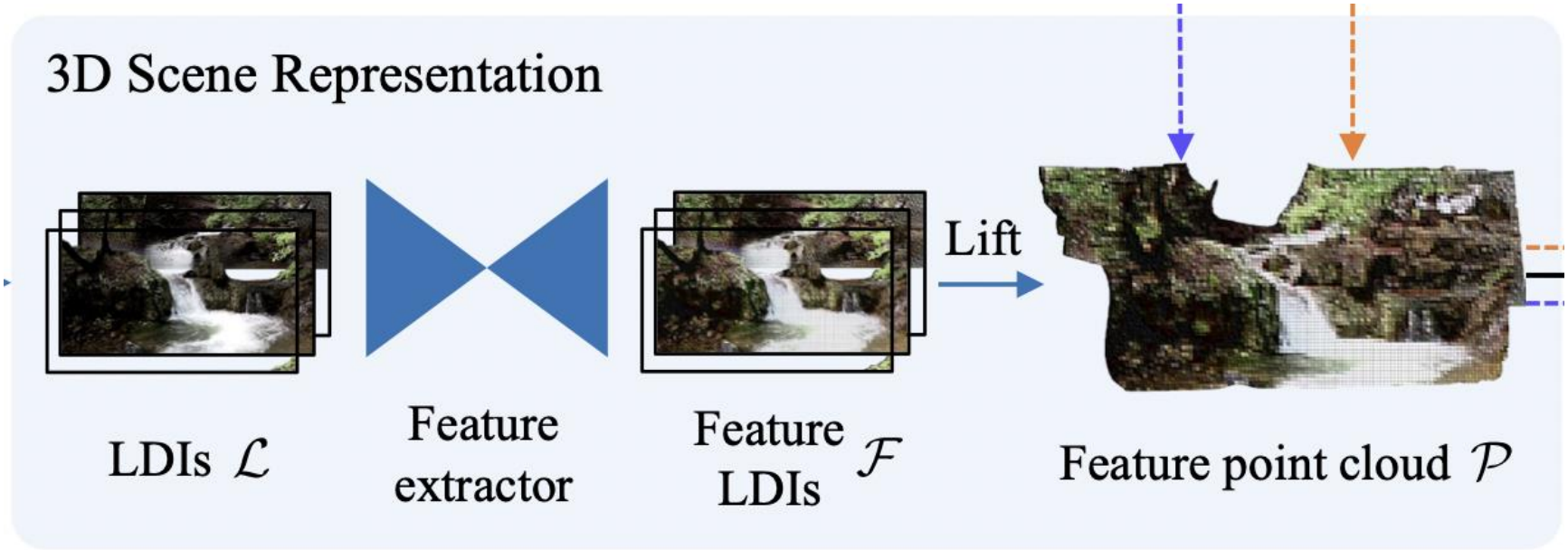
Depth Estimation Model



[Depth map]

Method

3D Scene Representation

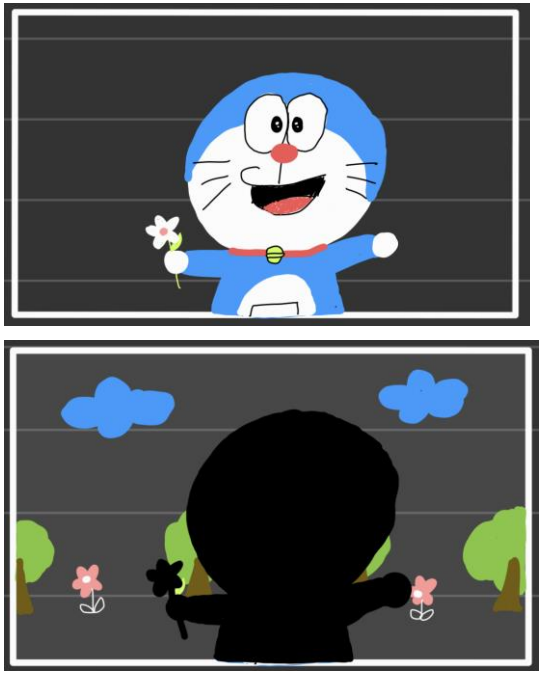


Method

3D Scene Representation



input



Depth 기반 layer 분리

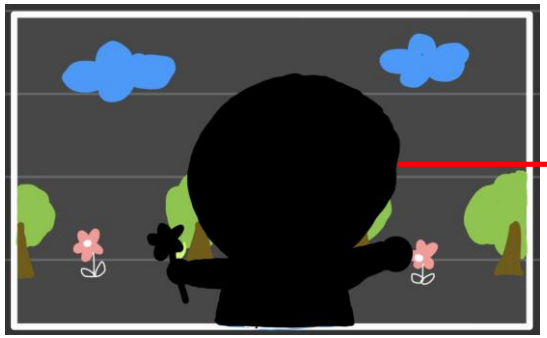
Agglomerative Clustering

: 비슷한 개체들을 하나로 합치면서 시작하여 군집간의 거리를 최소화해가는 군집화 알고리즘

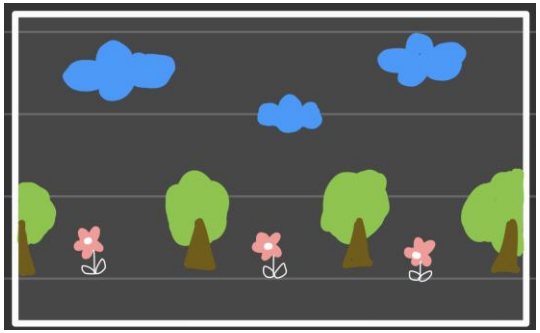
Source depth map의 값을 유사한 depth 값으로 묶어 레이어를 나눔

Method

3D Scene Representation



가려진 부분에 대한
Occluded regions



Inpainting

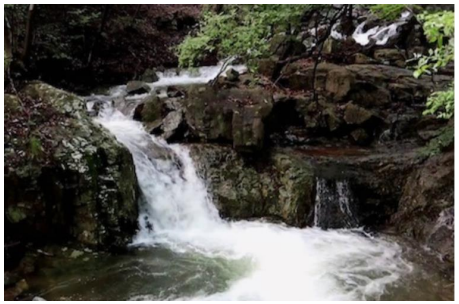




LDI(layered depth images) 생성

Method

3D Scene Representation



[RGB image]



[Depth map]



LDIs $\mathcal{L}$

해당 layer의 색상 정보

$$\mathcal{L} = \{\mathbf{C}_l, \mathbf{D}_l\}_{l=1}^{L}$$

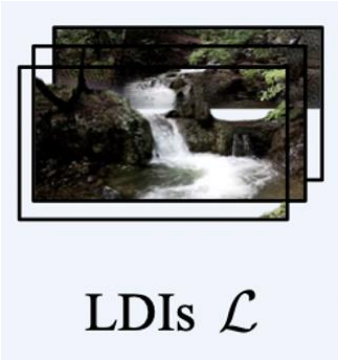
layer의 집합

해당 layer의 depth 정보

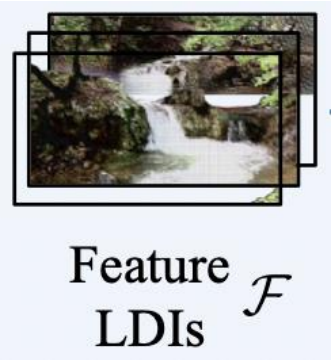
LDI(layered depth images) 생성

Method

3D Scene Representation



Feature Extractor



해당 layer의 색상 정보

$$\mathcal{L} = \{\mathbf{C}_l, \mathbf{D}_l\}_{l=1}^{L}$$

layer의 집합

해당 layer의 depth 정보

각 layer에 대한 색상 feature

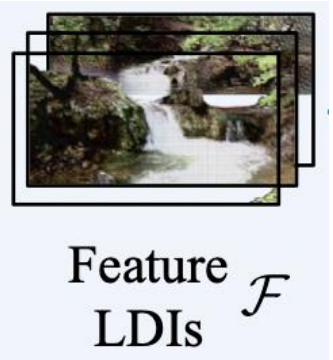
$$\mathcal{F} = \{\mathbf{F}_l, \mathbf{D}_l\}_{l=1}^{L}$$

여러 layer의 feature 정보를 포함하는 집합

각 layer에 대한 depth feature

Method

3D Scene Representation



Lift

*point cloud



: 3D 공간에서 point들로 구성된 데이터 구조
각 point는 공간 상의 위치를 나타내며
이 point들이 모여 하나의 cloud를 형성

각 layer에 대한
색상 feature

$$\mathcal{F} = \{\mathbf{F}_l, \mathbf{D}_l\}_{l=1}^{L}$$

여러 layer의 feature
정보를 포함하는 집합

각 layer에 대한
depth feature

feature point 좌표

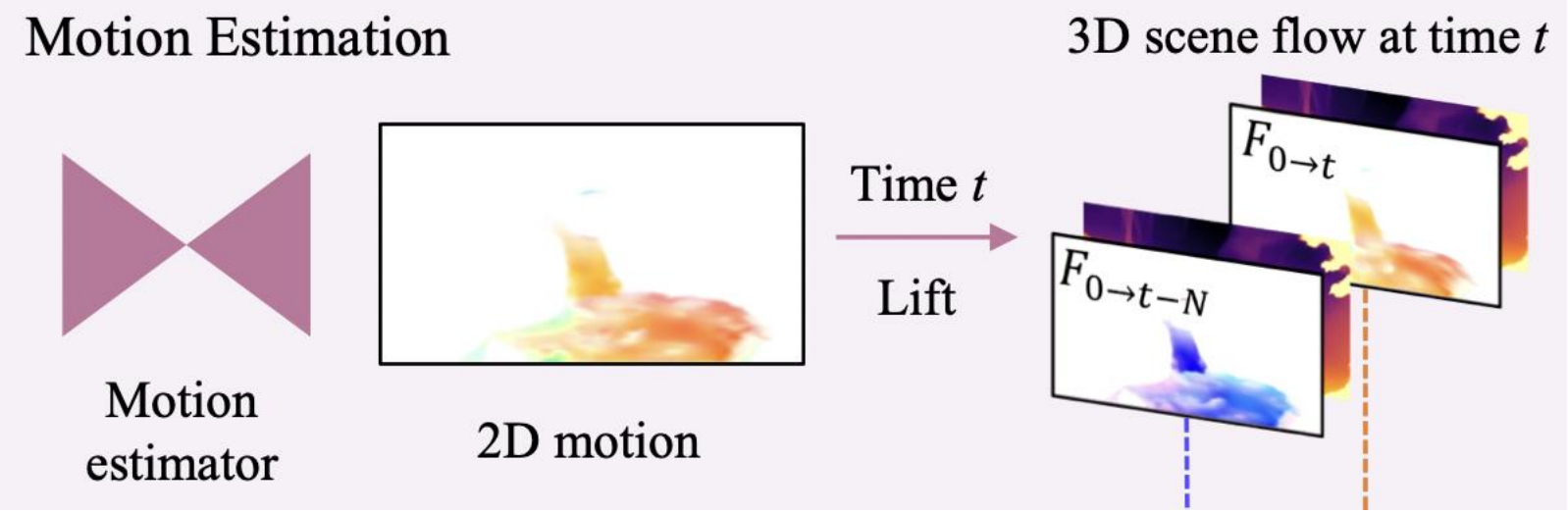
$$\mathcal{P} = \{(\mathbf{X}_i, \mathbf{f}_i)\}$$

feature
point cloud

feature
vector

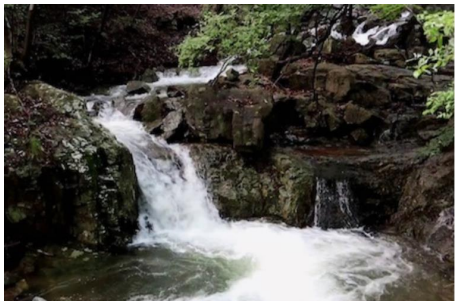
Method

Motion Estimation



Method

Motion Estimation



[RGB image]



[Depth map]

Motion Estimation



2D motion

Eulerian flow

$$F_{t \to t+1}(\cdot) = M(\cdot)$$

t frame에서
t+1 frame으로의
optical flow map

Method

Motion Estimation

Eulerian flow

t frame에서
t+1 frame으로의
optical flow map

$$F_{t \to t+1}(\cdot) = M(\cdot)$$

시간 t에서의 픽셀 좌표

다음 frame에서의
픽셀 좌표

$$\mathbf{x}_{t+1} = \mathbf{x}_t + M(\mathbf{x}_t)$$

$$F_{0 \to t}(\mathbf{x}_0) = F_{0 \to t-1}(\mathbf{x}_0) + M(\mathbf{x}_0 + F_{0 \to t-1}(\mathbf{x}_0))$$

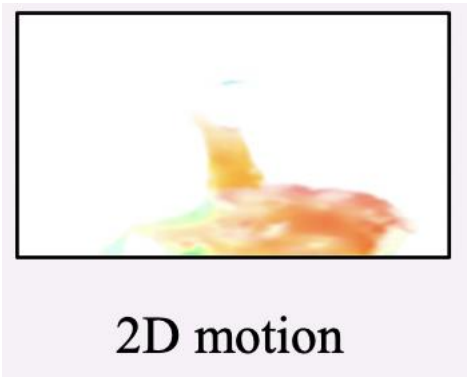
현재 픽셀에서의 Optical flow
(Eulerian Flow Field에 따라
이동한 위치에서의 모션을
나타냄)

이전 frame에서 현재 frame으로의
픽셀 이동을 나타냄
이전 frame의 Optical flow map

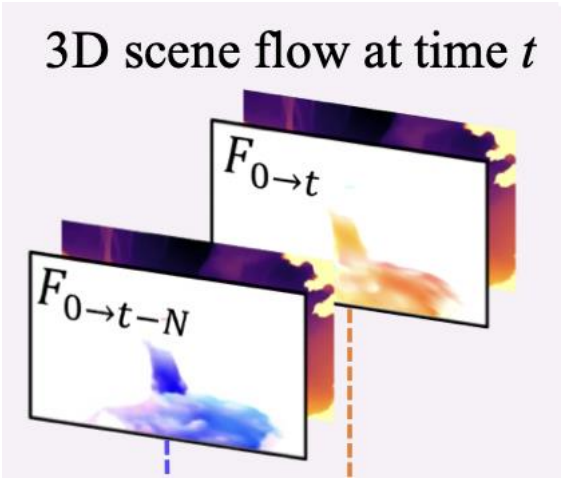
현재 픽셀 위치에 이전 frame으로의
픽셀 이동을 더함으로써
현재 픽셀이 이동할 위치를 얻음

Method

Motion Estimation



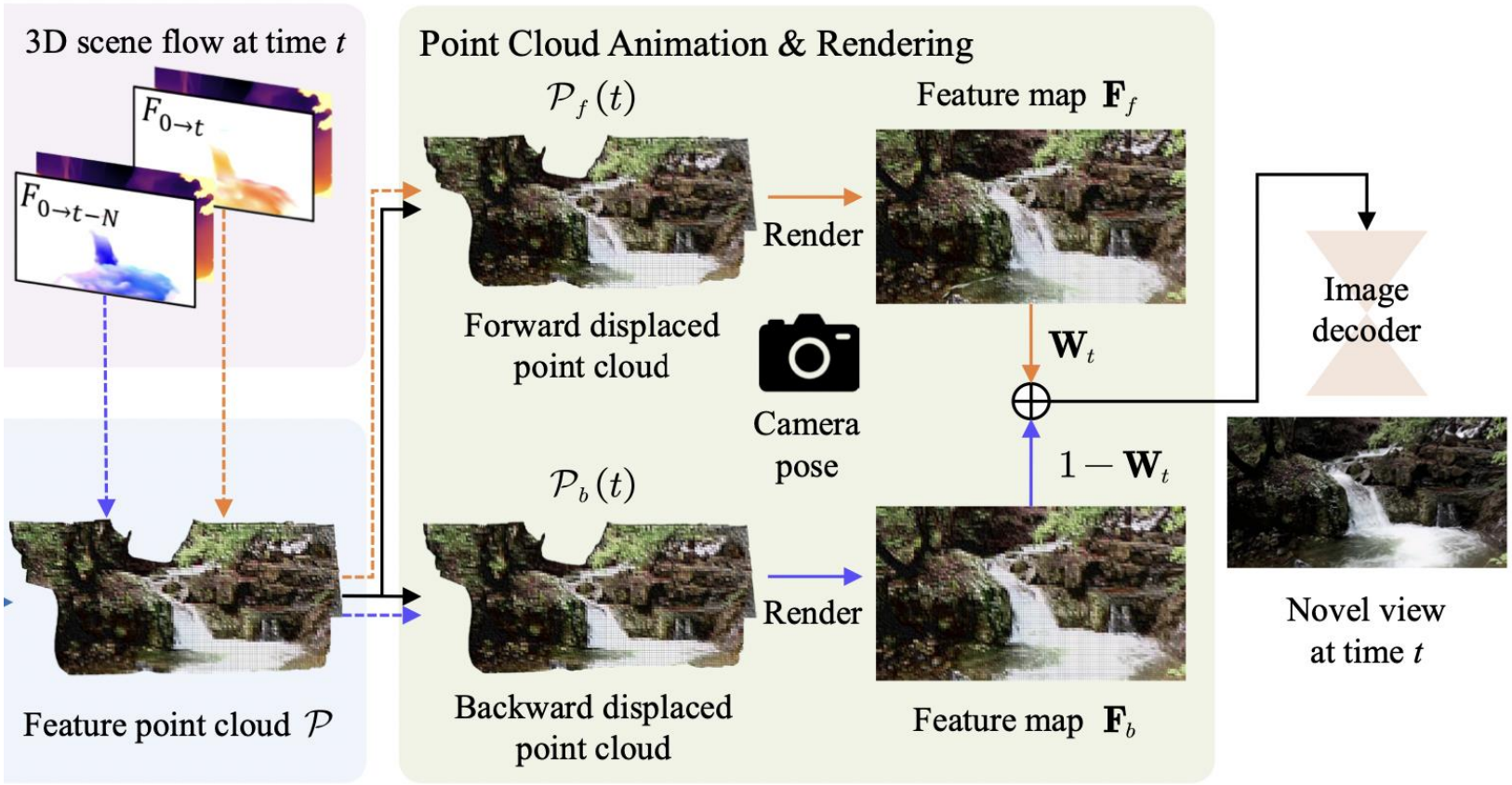
Lift



Image에서 얻은 displacement field를 해당하는 depth map과 결합하여 3D 공간으로 확장

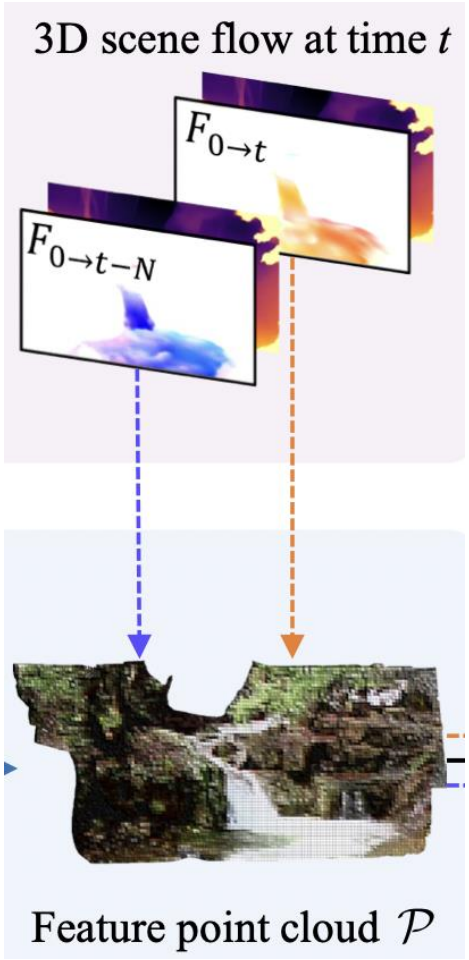
Method

Point Cloud Animation & Rendering



Method

Point Cloud Animation & Rendering



Image에서 얻은 displacement field를
해당하는 depth map과 결합하여 3D 공간으로 확장

↓

시간 t와 3D point의 좌표를 사용하여
해당 3D point를 이동할 수 있는

Feature point cloud ← $\mathcal{P}(t) = \{(\mathbf{X}_i(t), \mathbf{f}_i)\}$ → Feature vector

↓

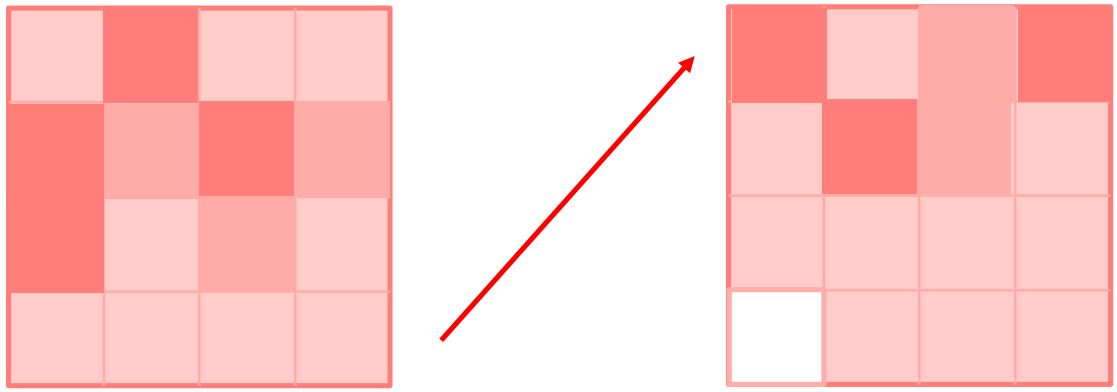
t에 대한 feature point

Point cloud를 한 시점에서 다른 시점으로 animate 가능

Method

Point Cloud Animation & Rendering

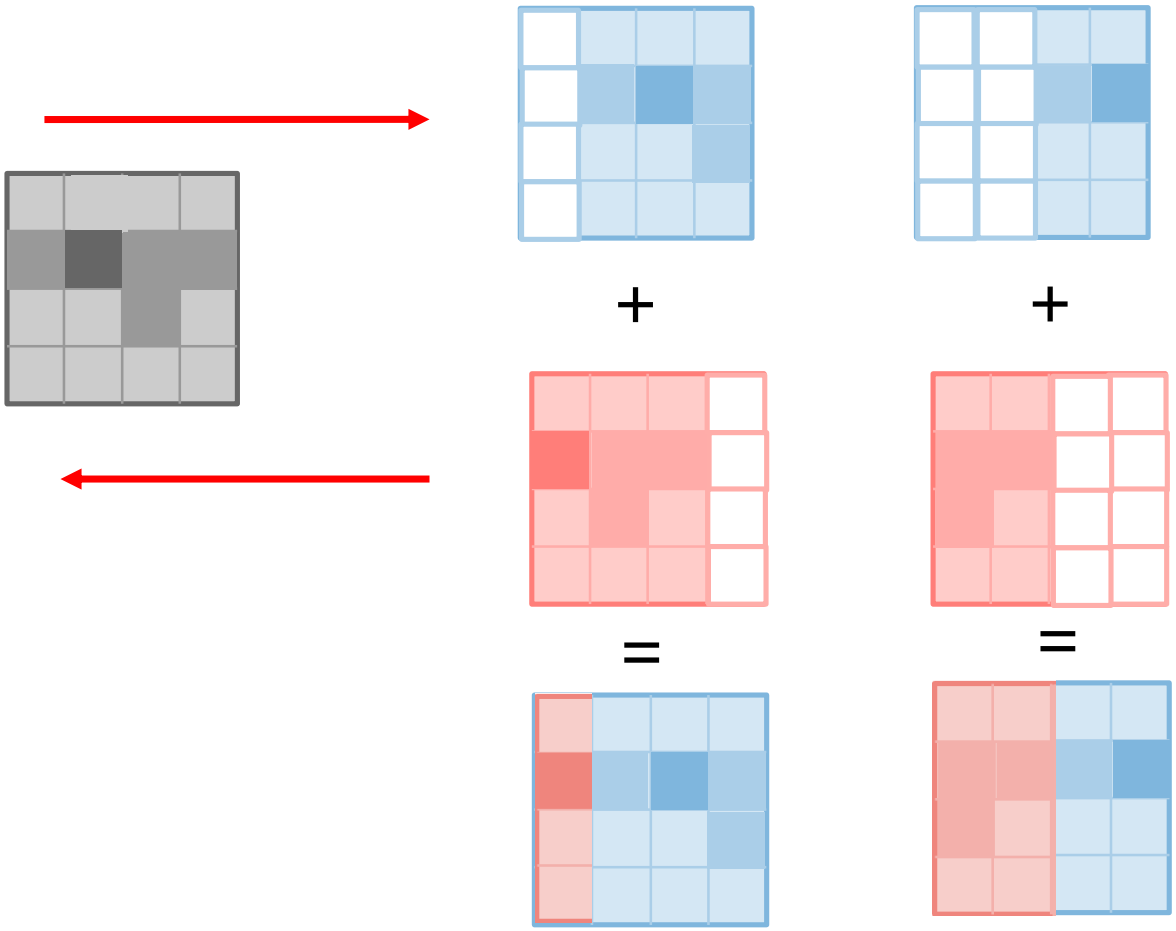
!!문제 발생!!



Point가 한 방향으로 이동함에 따라 hole(구멍)이 생김

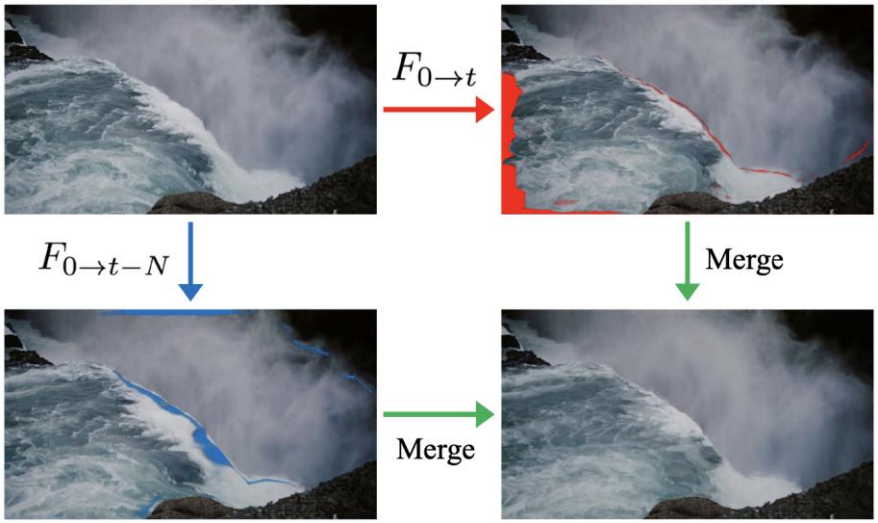
Method

Point Cloud Animation & Rendering



Method

Point Cloud Animation & Rendering



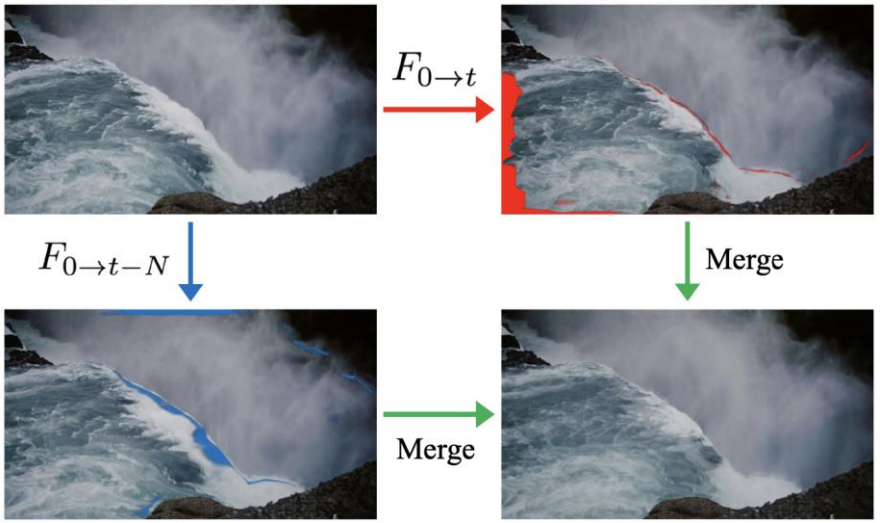
$$\mathcal{P}(t) = \{(\mathbf{X}_i(t), \mathbf{f}_i)\}$$

forward $\mathcal{P}_f(t) = \{(\mathbf{X}_i^f(t), \mathbf{f}_i)\}$

backward $\mathcal{P}_b(t) = \{(\mathbf{X}_i^b(t), \mathbf{f}_i)\}$

Method

Point Cloud Animation & Rendering



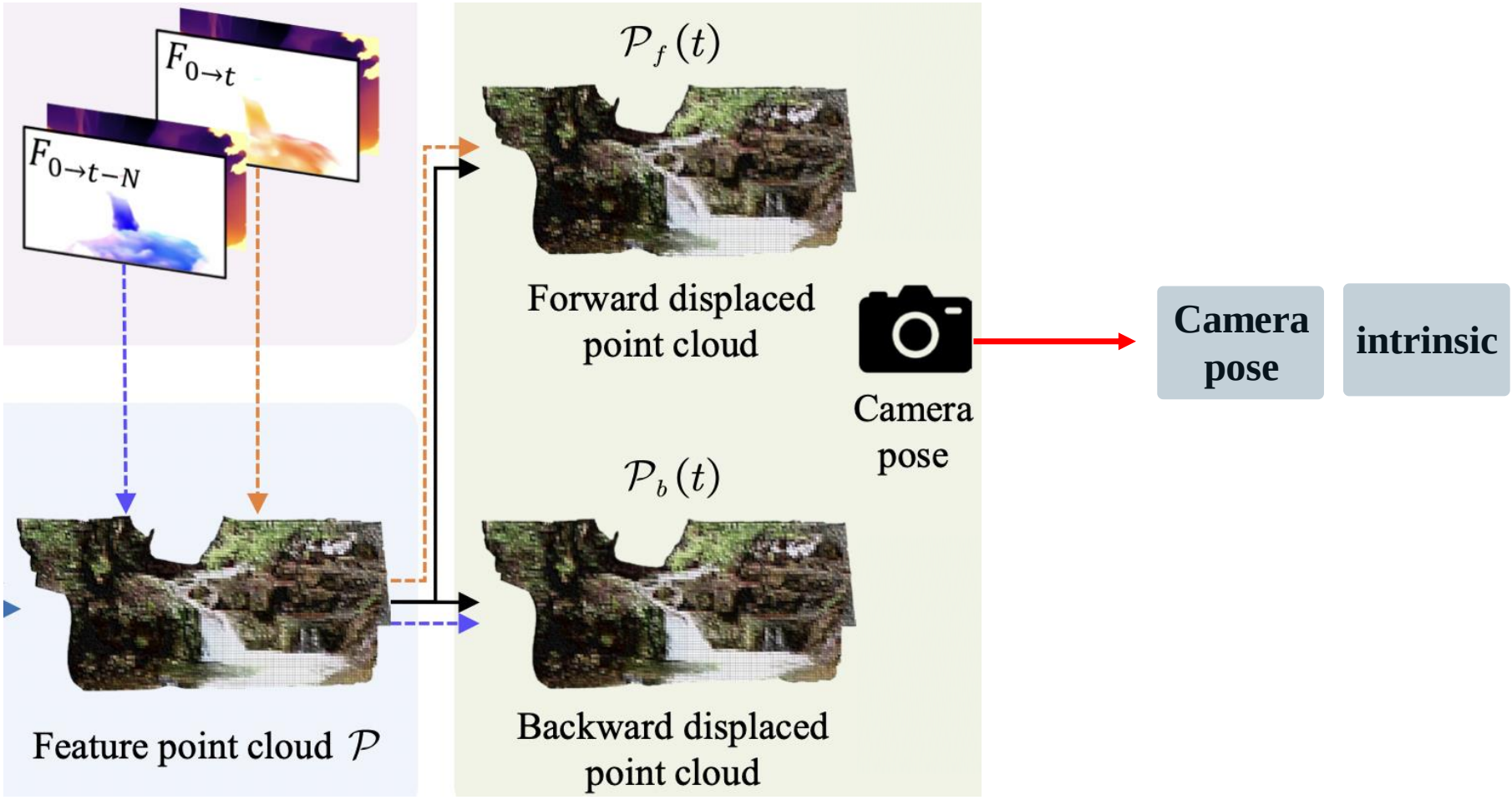
$$\mathcal{P}(t) = \{(\mathbf{X}_i(t), \mathbf{f}_i)\}$$

forward $\mathcal{P}_f(t) = \{(\mathbf{X}_i^f(t), \mathbf{f}_i)\}$

backward $\mathcal{P}_b(t) = \{(\mathbf{X}_i^b(t), \mathbf{f}_i)\}$

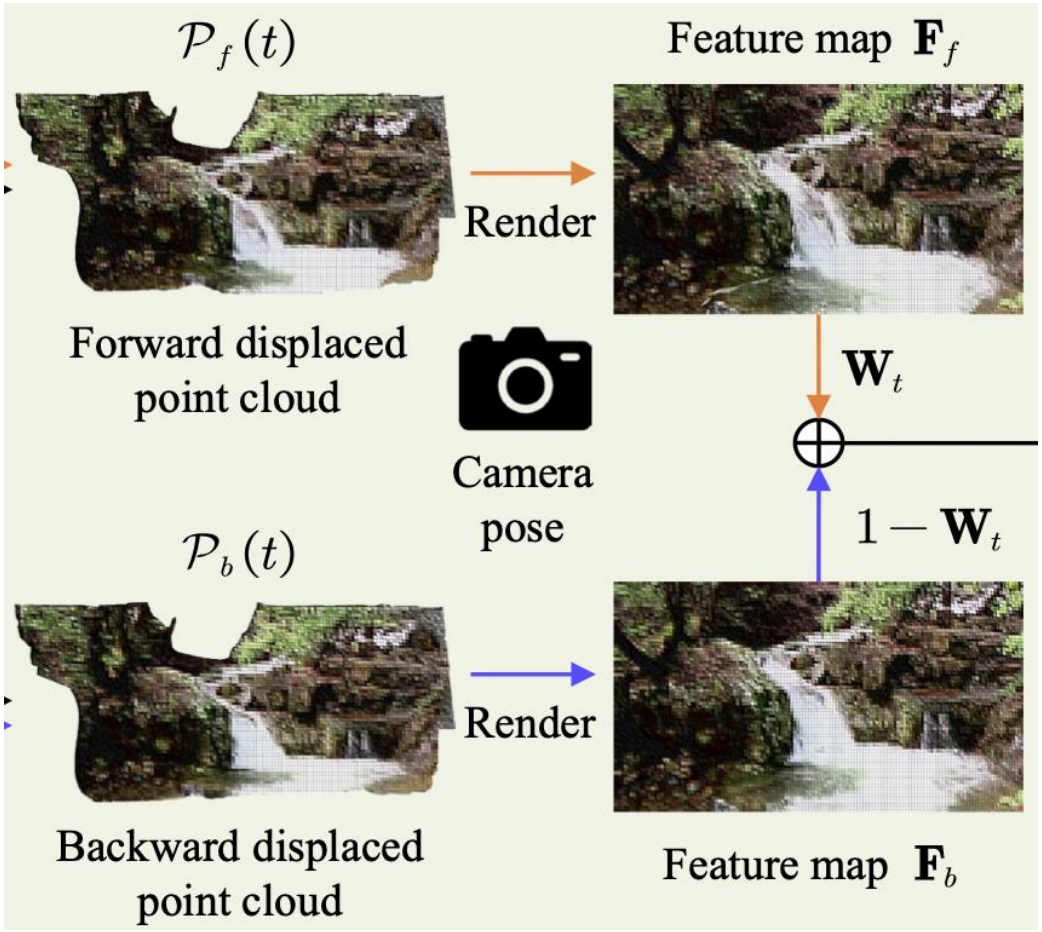
Method

Point Cloud Animation & Rendering



Method

Point Cloud Animation & Rendering



3D Rendering

: 3차원 공간에 존재하는 장면을 2D image로 변환하는 과정

$\mathbf{F}_f\,\mathbf{F}_b$	2D feature map
$\mathbf{D}_f\,\mathbf{D}_b$	Depth map
$\alpha_f\ \alpha_b$	투명도

Method

Point Cloud Animation & Rendering

Feature map $\mathbf{F}_f$



$\mathbf{W}_t$

+

$1 - \mathbf{W}_t$



Feature map $\mathbf{F}_b$

$$\mathbf{W}_t = \frac{(1 - \frac{t}{N}) \cdot \boldsymbol{\alpha}_f \cdot e^{-\mathbf{D}_f}}{(1 - \frac{t}{N}) \cdot \boldsymbol{\alpha}_f \cdot e^{-\mathbf{D}_f} + \frac{t}{N} \cdot \boldsymbol{\alpha}_b \cdot e^{-\mathbf{D}_b}}$$

$\mathbf{D}_f$ $\mathbf{D}_b$ Depth map

$\boldsymbol{\alpha}_f$ $\boldsymbol{\alpha}_b$ 투명도

1. Looping을 위해 첫 번째와 마지막 frame이 동일하도록 weight 할당
2. Depth 값이 작을 수록 가까이에 있는 물체이기에 작은 depth값을 선택하도록 weight 할당
3. Hole을 메울 수 있는 픽셀 위치에 weight 증가



Method

Point Cloud Animation & Rendering

통합 feature map

$$\mathbf{F}_t = \mathbf{W}_t \cdot \mathbf{F}_f + (1 - \mathbf{W}_t) \cdot \mathbf{F}_b$$

통합 depth map

$$\mathbf{D}_t = \mathbf{W}_t \cdot \mathbf{D}_f + (1 - \mathbf{W}_t) \cdot \mathbf{D}_b$$

$\mathbf{F}_f \, \mathbf{F}_b$ 2D feature map

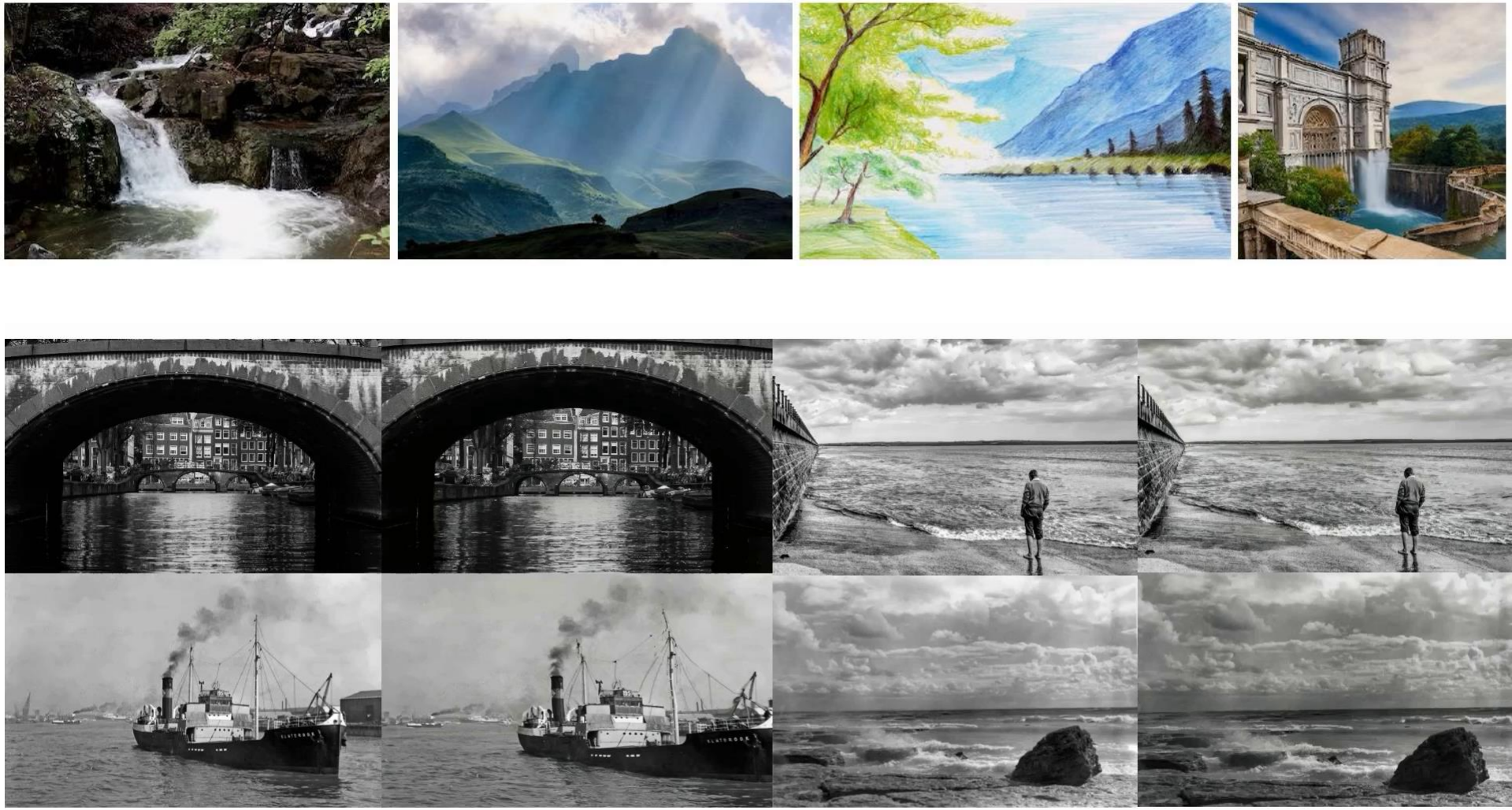
$\mathbf{D}_f \, \mathbf{D}_b$ Depth map

Image decoder



Novel view
at time t

Result



감사합니다.