Invisibility Cloak using Video Inpainting

정하빈

2021.11.30

Introduction

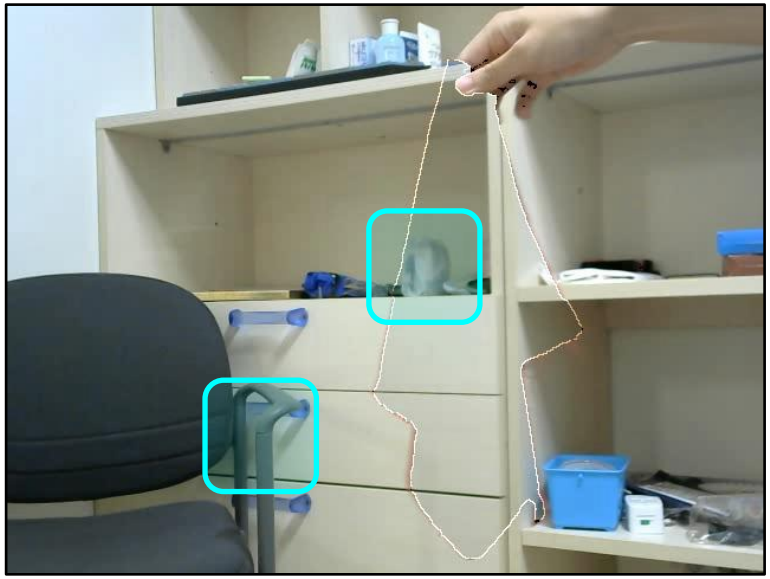
✓ Invisibility Cloak (투명 망토)

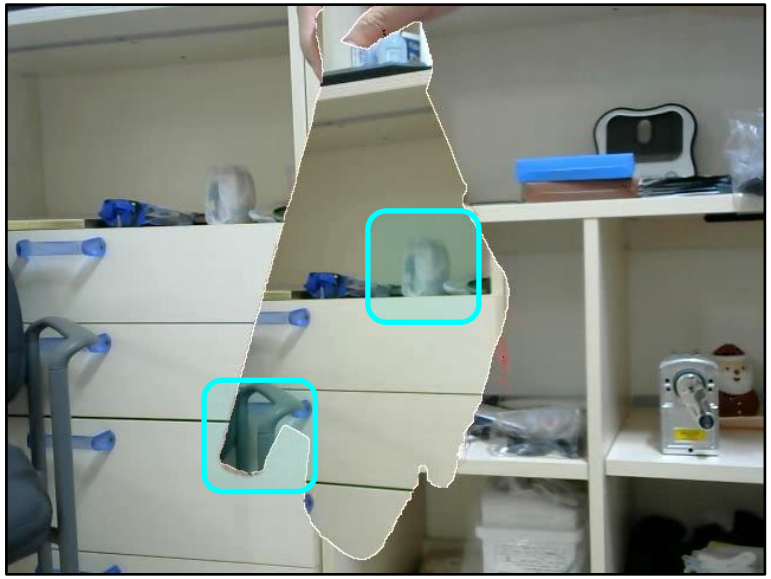
- 특정 색상을 검출하여 마스킹 처리를 하고, 새로운 내용으로 이미지 복원을 하여 투명 망토의 효과를 구현

Introduction
Previous Project Problem

✓ 고정된 배경

- 망토 없이 미리 캡쳐한 배경 이미지를 마스킹 영역에 적용하였기 때문에 카메라의 이동 불가능

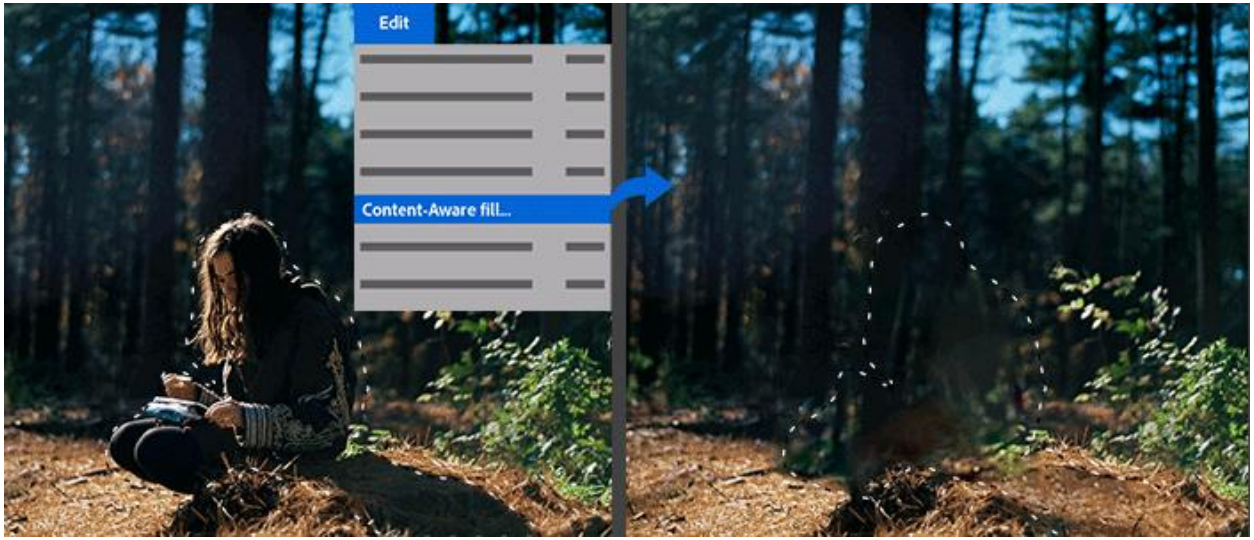




Introduction

✓ Image Inpainting (이미지 복원)

- 마스킹 영역의 주변에서 샘플링한 컨텐츠로 새로운 내용을 생성하여 마스킹 영역을 복원하는 작업

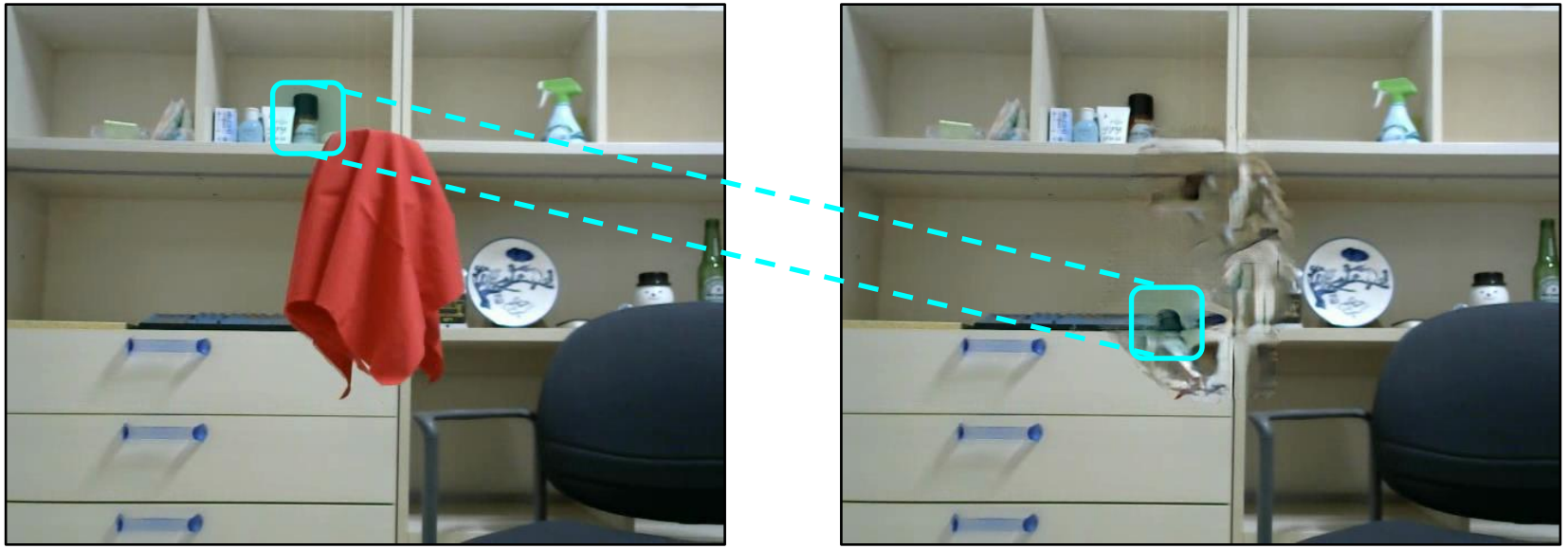


예시) Photoshop [내용 인식 채우기]

Introduction
Single Inpainting Problem

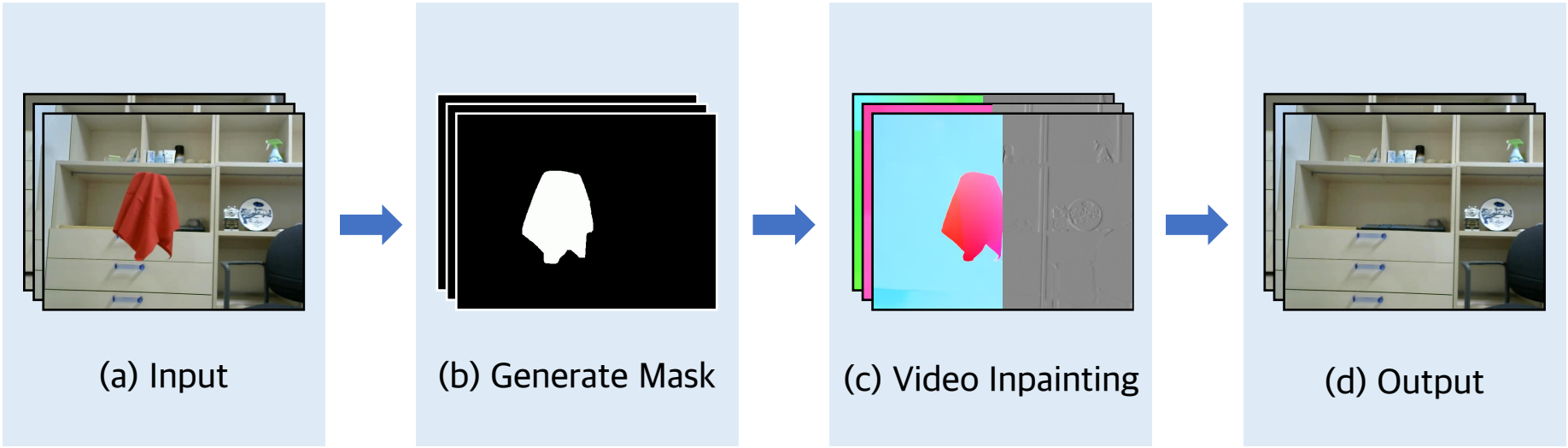
✓ 단일 이미지 복원 문제

- 기존의 이미지 복원 방법은 패치 기반 합성 기술 사용
- 이미지에 존재하는 패치 영역만 작업 가능



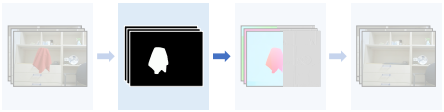
Process Overview

Video Inpainting



Method

(a) Input



Method

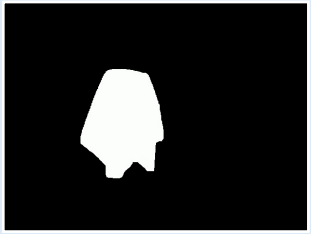
(b) Generate Mask

Input

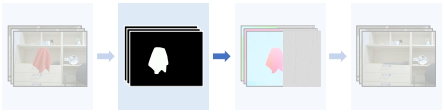
Convert HSV

Detect Color

Noise Filtering



Convert Mask



Method

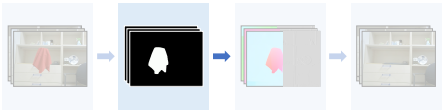
(b) Generate Mask

✓ Convert RGB to HSV

- 사람의 눈과 비슷한 HSV 채널 사용
- `cv2.cvtColor(image, cv2.COLOR_BGR2HSV)`

Input

Convert HSV

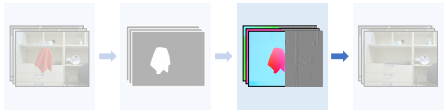


Method
(b) Generate Mask

✓ Detect Color & Noise Filtering

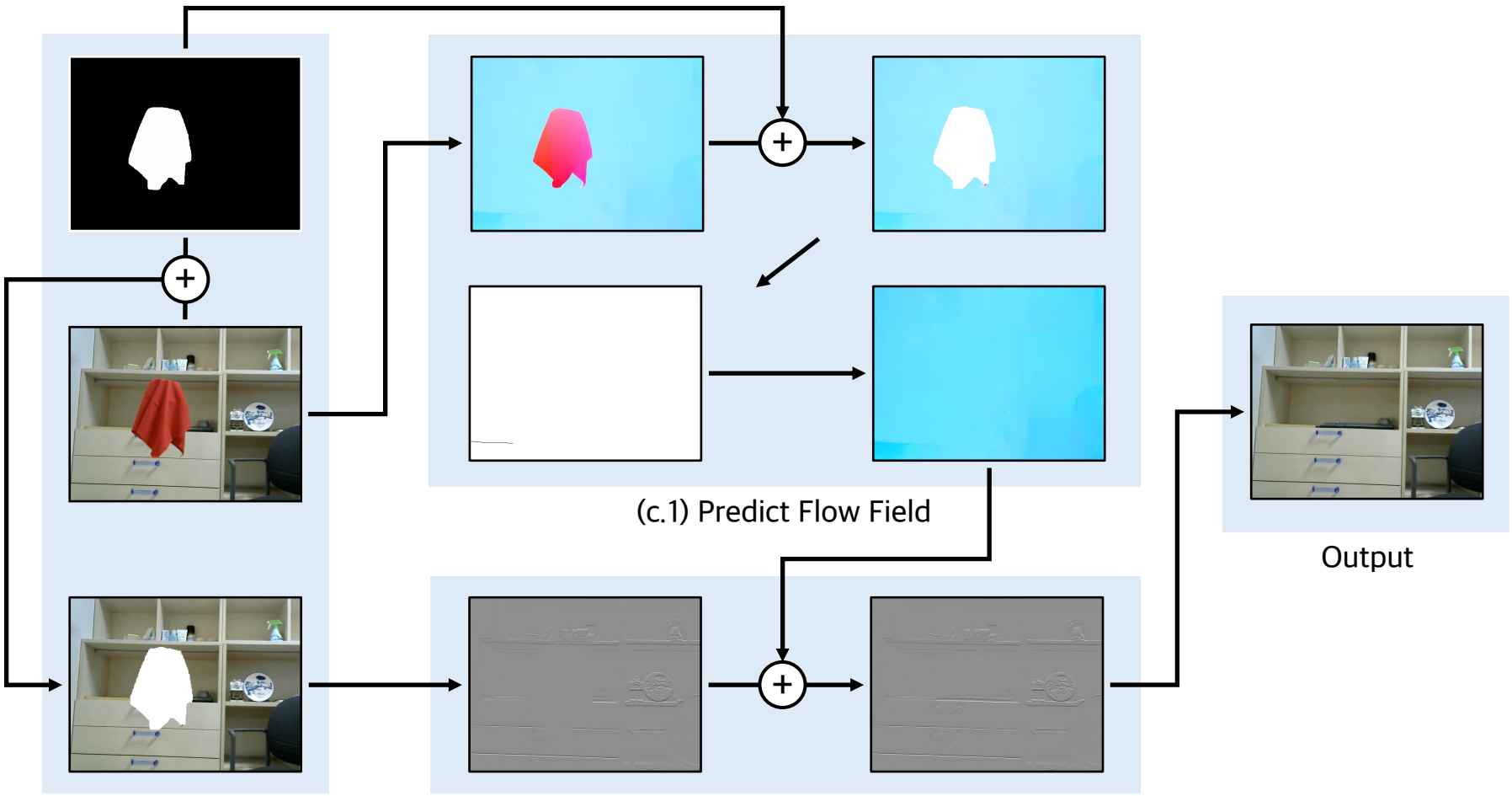
- 잡음 제거, 마스크 영역 확장을 위해 `cv2.morphologyEx(...)` 사용
- `cv2.MORPH_OPEN` → `cv2.MORPH_CLOSE` → `cv2.MORPH_DILATE`

Detect Color　　Noise Filtering　　Convert Mask

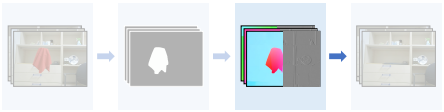


Method

(c) Video Inpainting with FGVC[1]



[1] Flow-edge Guided Video Completion



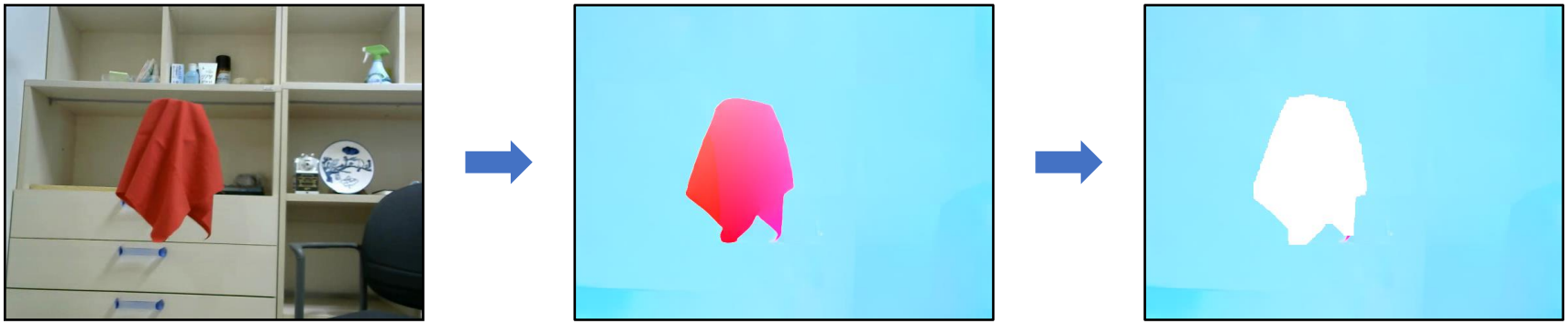
Method

(c.1) Predict Flow Field

✓ Flow Computation

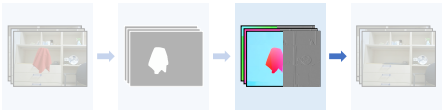
- RAFT[2]를 사용하여 인접 프레임 사이의 optical flow를 계산

$$\boldsymbol{F}_{i\to j} = \mathcal{F}\left(\boldsymbol{I}_i,\ \boldsymbol{I}_j\right), \qquad |i-j| = 1$$



Input　　Optical Flow　　Apply Mask

[2] RAFT: Recurrent All-Pairs Field Transforms for Optical Flow

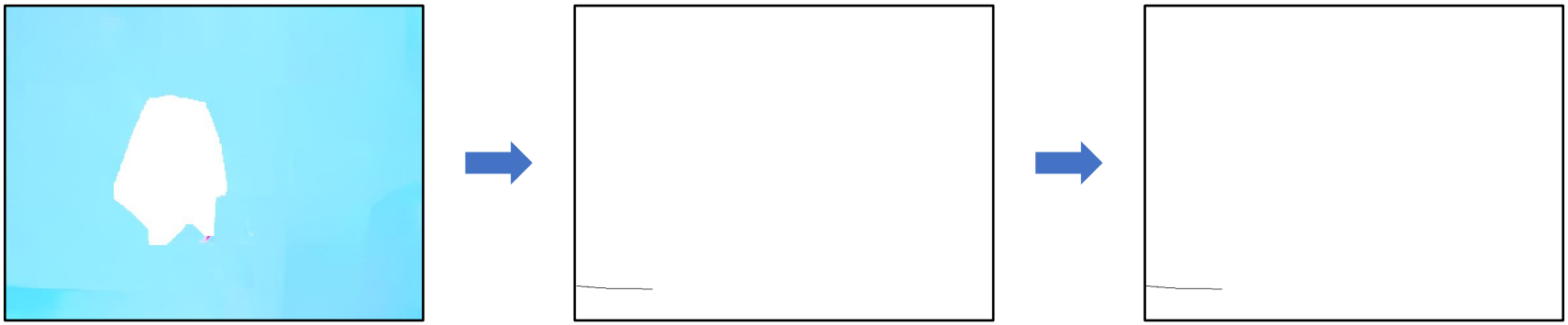


Method

(c.1) Predict Flow Field

✓ Edge Completion

- EdgeConnect[3]를 사용하여 edge 예측
- 경계를 구분하여 object 판별에 유리

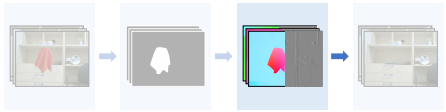


Input Optical Flow

Detect Canny Edge ($\boldsymbol{E}_{i\to j}$)

Connect Edge ($\widetilde{\boldsymbol{E}}_{i\to j}$)

[3] EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning

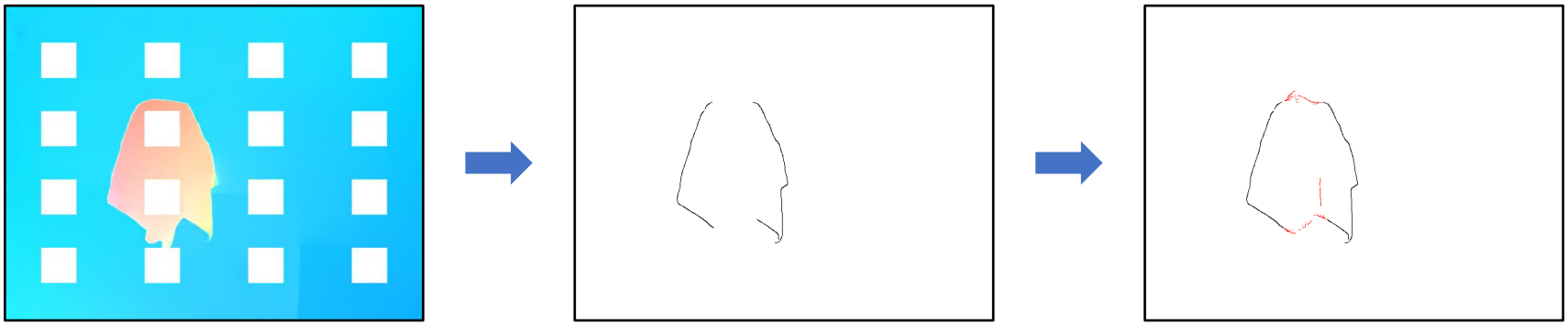


Method

(c.1) Predict Flow Field

✓ Edge Completion

- EdgeConnect[3]를 사용하여 edge 예측
- 경계를 구분하여 object 판별에 유리

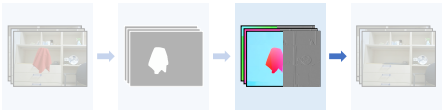


Input Optical Flow

Detect Canny Edge ($\boldsymbol{E}_{i\to j}$)

Connect Edge ($\widetilde{\boldsymbol{E}}_{i\to j}$)

[3] EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning



Method

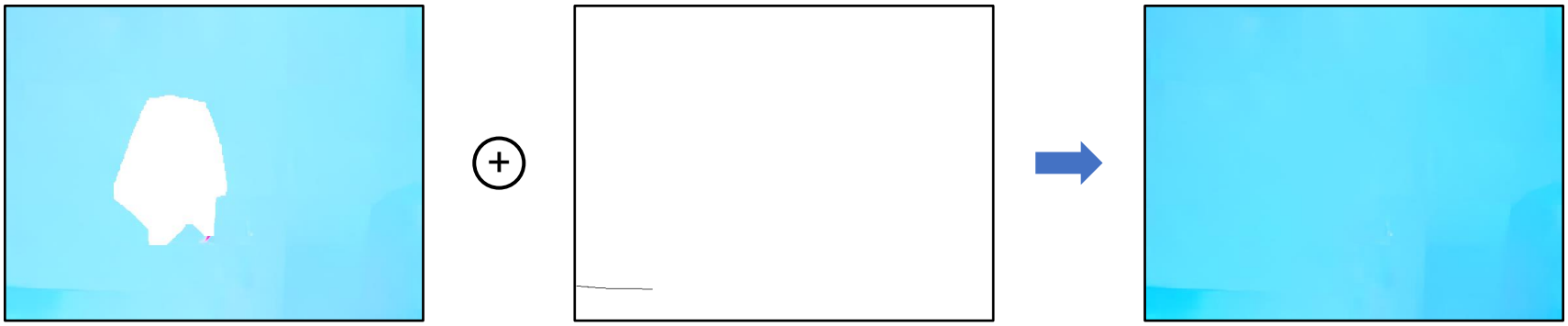
(c.1) Predict Flow Field

✓ Flow Completion

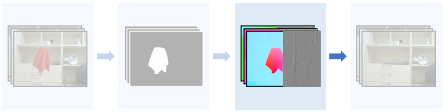
- Edge를 제외한 영역에 대하여 flow 완성

$$\underset{\widetilde{\boldsymbol{F}}}{\operatorname{argmin}} \sum_{p|\widetilde{\boldsymbol{E}}(p)=1} \left\|\Delta_x \widetilde{\boldsymbol{F}}(p)\right\|_2^2 + \left\|\Delta_y \widetilde{\boldsymbol{F}}(p)\right\|_2^2,$$

subject to $\widetilde{\boldsymbol{F}}(p) = \boldsymbol{F}(p) \mid \boldsymbol{M}(p) = 0$



Input Optical Flow　　Connect Edge　　Flow Completion

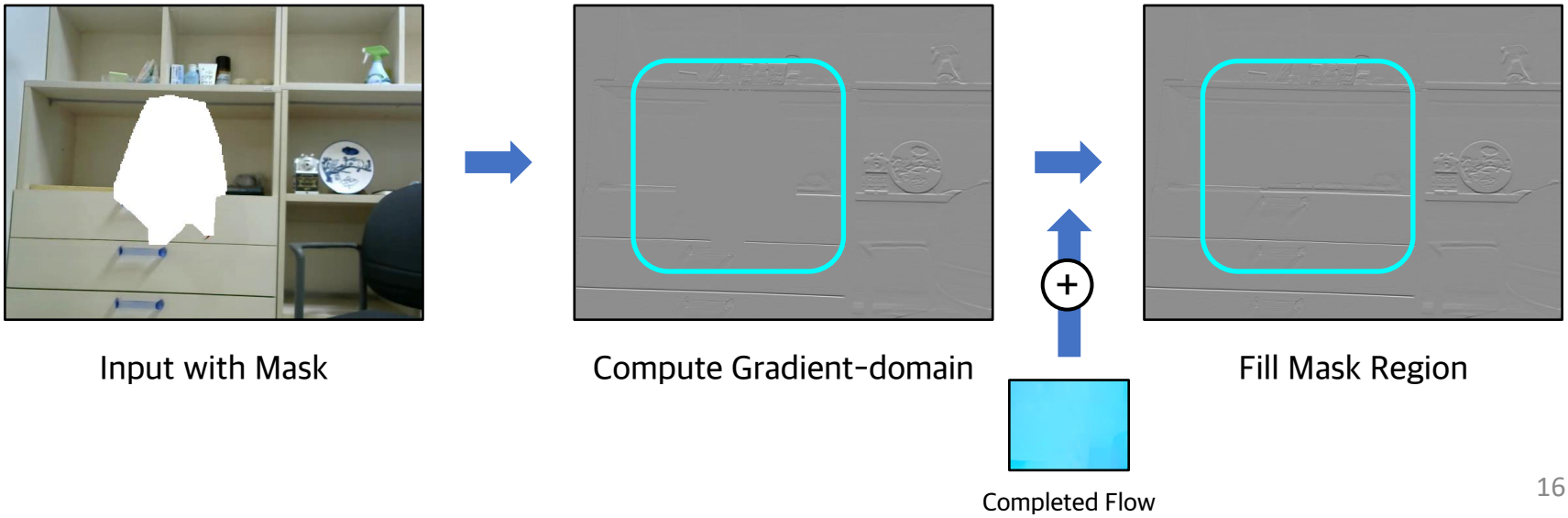


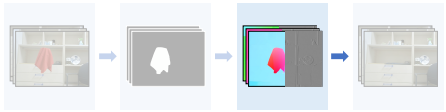
Method
(c.2) Fill Gradient

✓ Gradient-domain Processing

- Gradient-domain에서 각 픽셀 p에 대하여 Weighted Average 계산

$$\tilde{I}(p) = \frac{\sum_k w_k c_k}{\sum_k w_k}, \qquad \tilde{G}(p) = \frac{\sum_k w_k \Delta c_k}{\sum_k w_k}$$

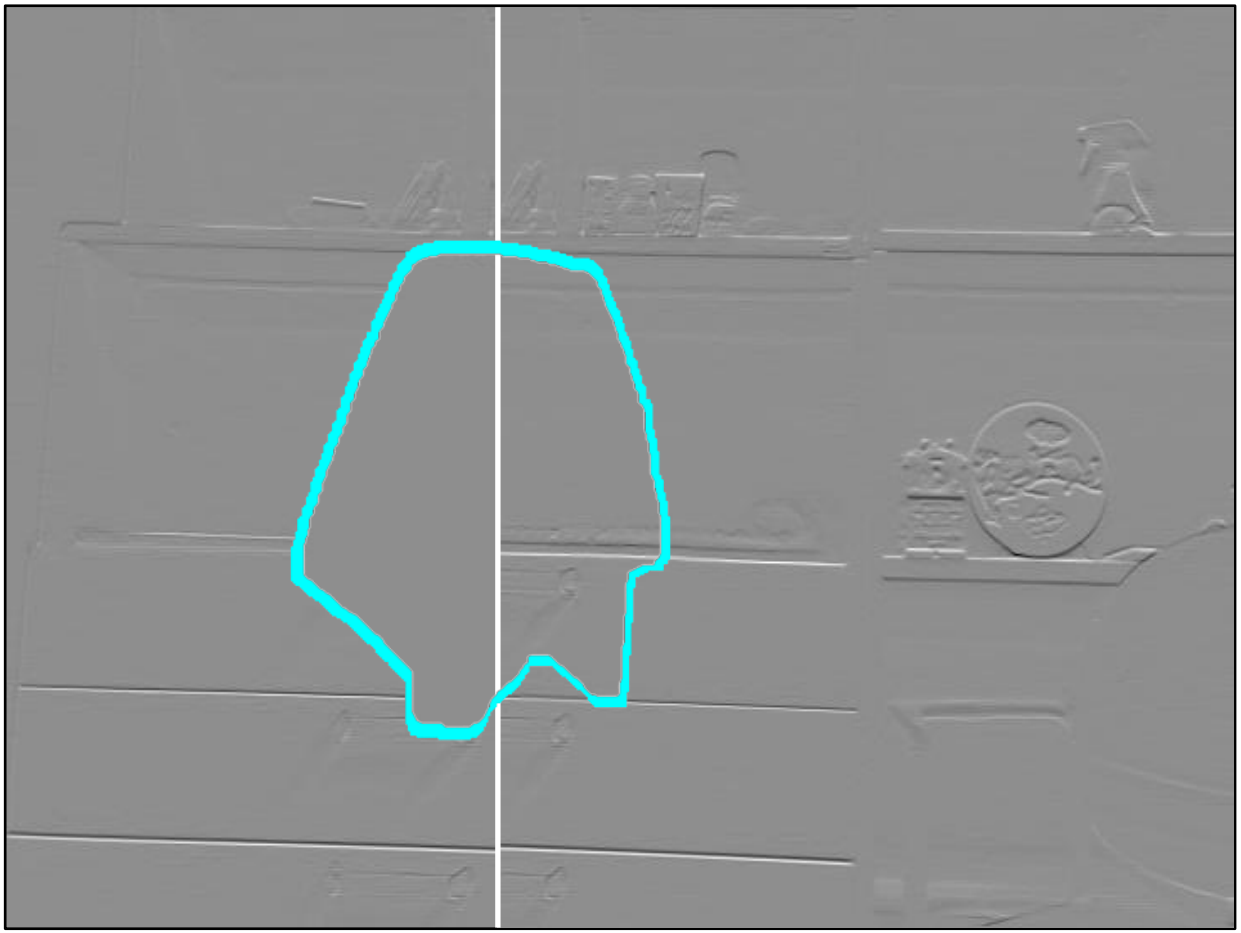


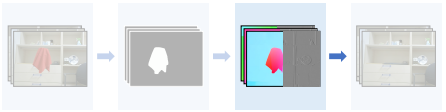


Method

(c.2) Fill Gradient

✓ Gradient-domain Processing





Method
(c.2) Fill Gradient

✓ Fusion Gradient-domain with input

- Poisson Blending [4]을 통한 합성

$$\underset{\tilde{I}}{\operatorname{argmin}} \quad \left\|\Delta_x \tilde{I} - \tilde{G}_x\right\|_2^2 + \left\|\Delta_y \tilde{I} - \tilde{G}_y\right\|_2^2,$$

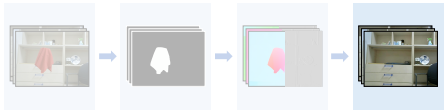
subject to $\tilde{I}(p) = I(p) \mid M(p) = 0$

Input　　Filled Gradient-domain　　Output

[4] Fast Poisson Blending using Multi-Splines

Method

(d) Output



Limitation
Failure Case

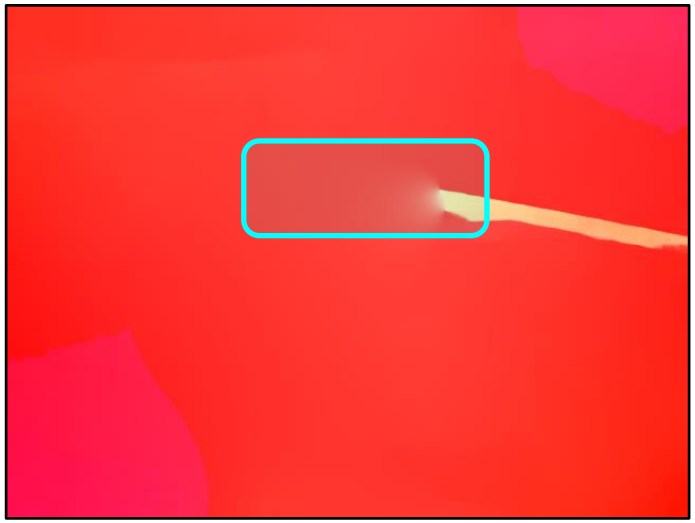
Limitation
Failure Case

Limitation
Failure Case

✓ 빠른 카메라 전환

Limitation
Failure Case

✓ 불필요한 대상 인식

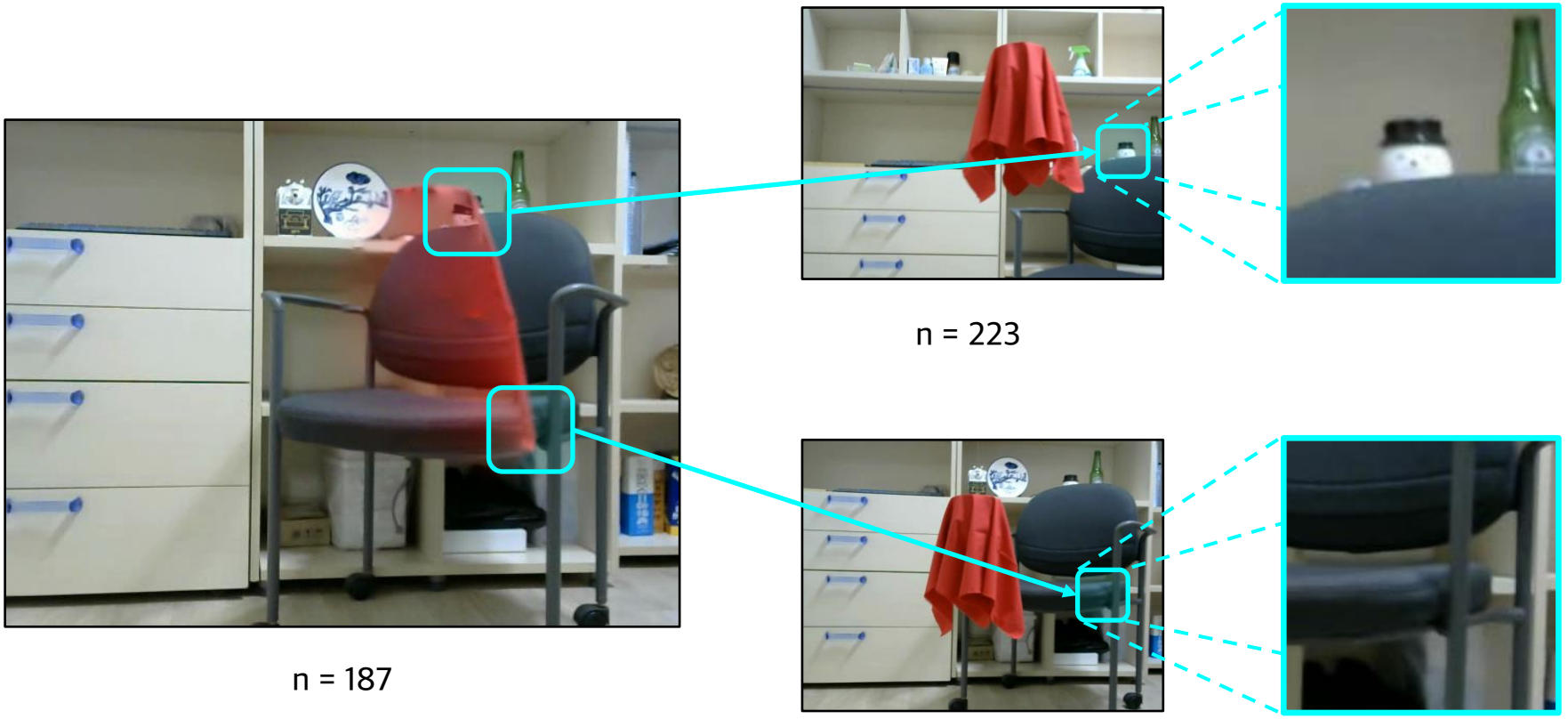


Optical Flow with Removed Mask

Output

Limitation
Low Quality Frame

✓ Non-local Frame을 통한 보완 필요



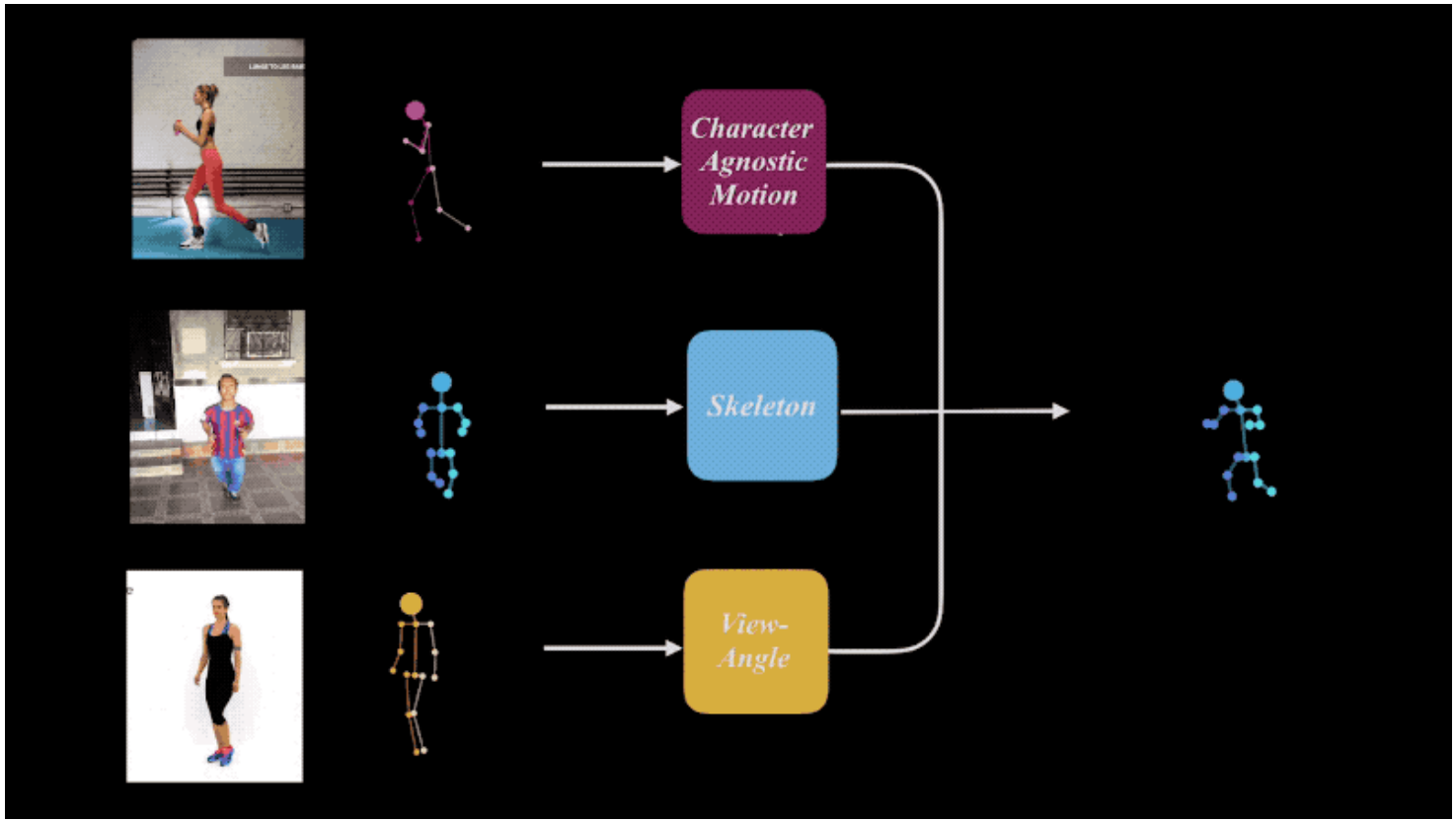
n = 187

n = 223

n = 176

Interested Topic

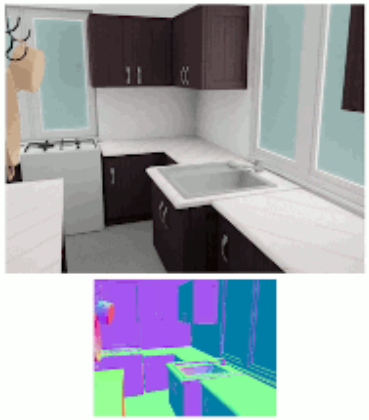
✓ 2D Motion Retargeting [5]



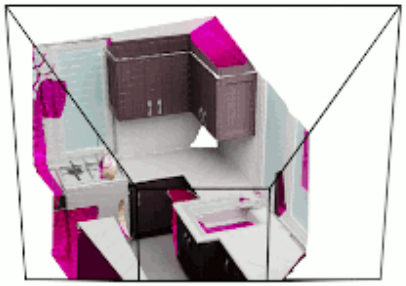
[5] Learning Character-Agnostic Motion for Motion Retargeting in 2D

Interested Topic

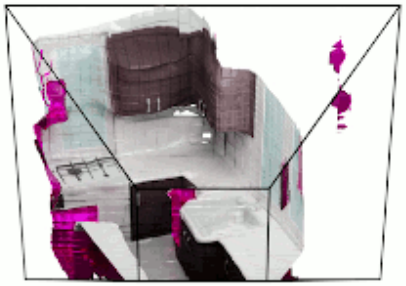
✓ 3D Scene Reconstruction [6]



Color + Normal Image



Ground Truth



Prediction

[6] 3D Scene Reconstruction from a Single Viewport

Conclusion

- ✓ Computer Vision Basic
- ✓ Paper Reading
- ✓ Topic Search
- ✓ Code Review

2개월 간의 북마크

감사합니다.