Track Anything Model

<TAM>

- meta AI <2023.04.28>

ComputerVision, Tracking

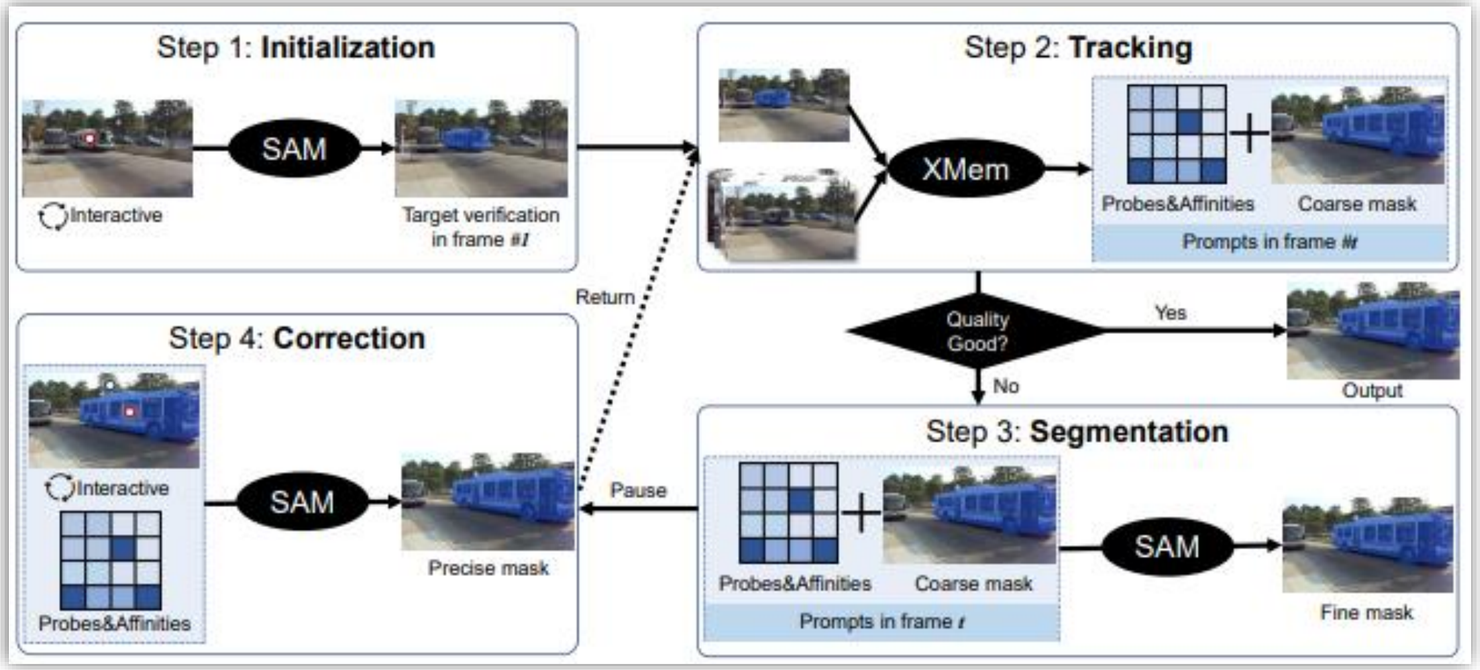
백 대 환

PaperReview

Contents

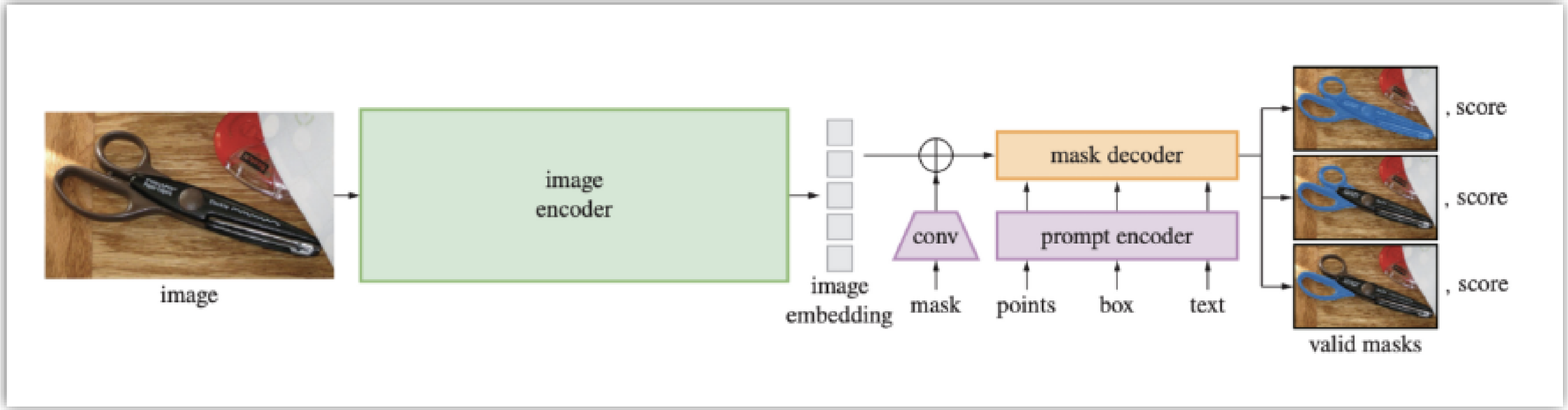
Introduction

- **VOT는 CV의 기본 작업**이고, **VOS를 통해 배경에서 객체 분리, 세분화된 객체 추적**
- VOT를 위해서는 많은 학습이 필요하고, 이를 위한 많은 인력과 시간이 필요
- VOS 중 SAM은 **유연한 prompt 지원, 여러 mask를 계산하여 상호작용 가능, zero-shot에 강점**
- **1. SAM과 2. XMEM**을 결합하여 VOT 모델인 **TAM**을 만듦
- XMEM : https://arxiv.org/pdf/2207.07115.pdf, SAM: https://arxiv.org/pdf/2304.02643.pdf



Segment Anything Model & XMem

- SAM (2023. 04)
- 1,100만 개의 이미지와 11억 개의 마스크에 대해 학습되어 고품질 마스크를 생성
- 일반 시나리오에서 **zero-shot segmentation** 수행 가능
- 다양한 종류의 prompt와의 높은 상호작용성. (만족스러운 segmentation mask 제공 가능)
- But, 시간적인 correspondence가 부족하여 인상적인 성능 제공 못함.
- **상호 작용 방식을 이용한 동영상에서 고성능 tracking과 segmentation을 할 수 있을까?**



Segment Anything Model & XMem

- SAM (2023. 04)
- 1) 대화형 비디오 객체 추적을 위해 SAM을 비디오에 적용, 프레임당 SAM이 아닌 **시간적 대응 구축 프로세스**를 통합
- 2) 효율적인 initialization을 위한 간편한 인터페이스
- 3) 우수한 성능과 높은 사용성. 복잡한 장면에서 우수한 성능과 높은 사용성
- -> + XMem

Our contributions can be concluded as follows:

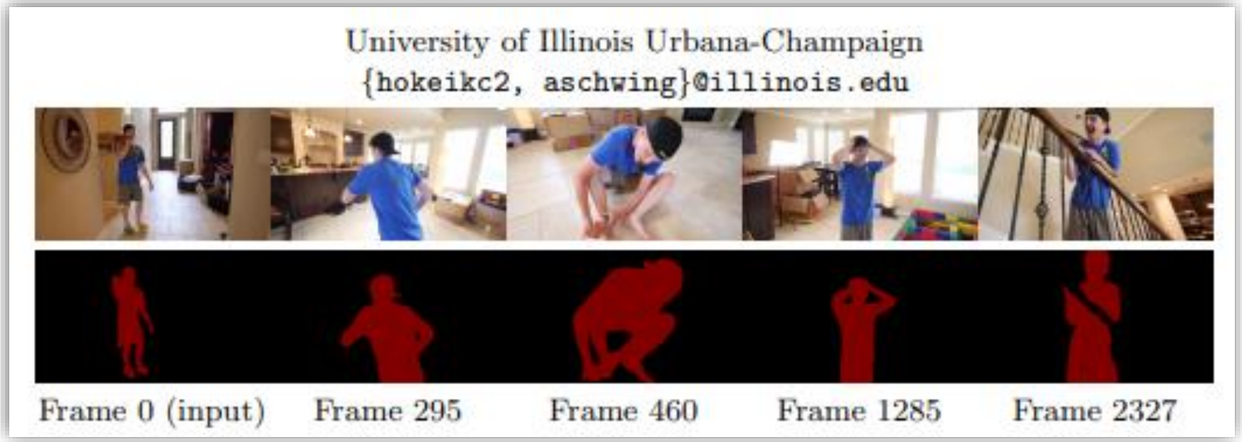
1) We promote the SAM applications to the video level to achieve interactive video object tracking and segmentation. Rather than separately using SAM per frame, we integrate SAM into the process of temporal correspondence construction.

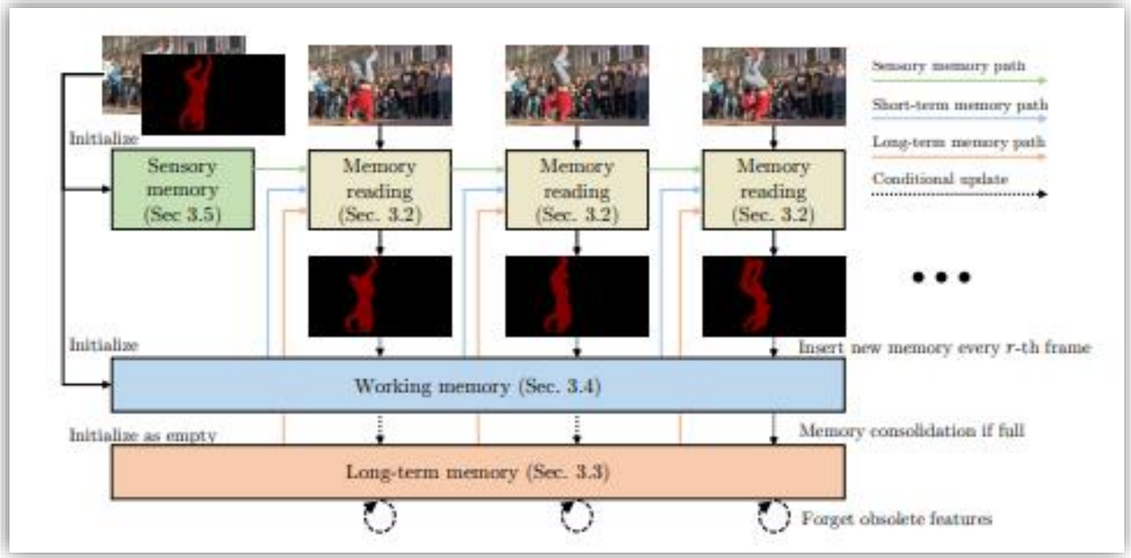
2) We propose one-pass interactive tracking and segmentation for efficient annotation and a user-friendly tracking interface, which uses very small amounts of human participation to solve extreme difficulties in video object perception.

3) Our proposed method shows superior performance and high usability in complex scenes and has many potential applications.

Segment Anything Model & XMem

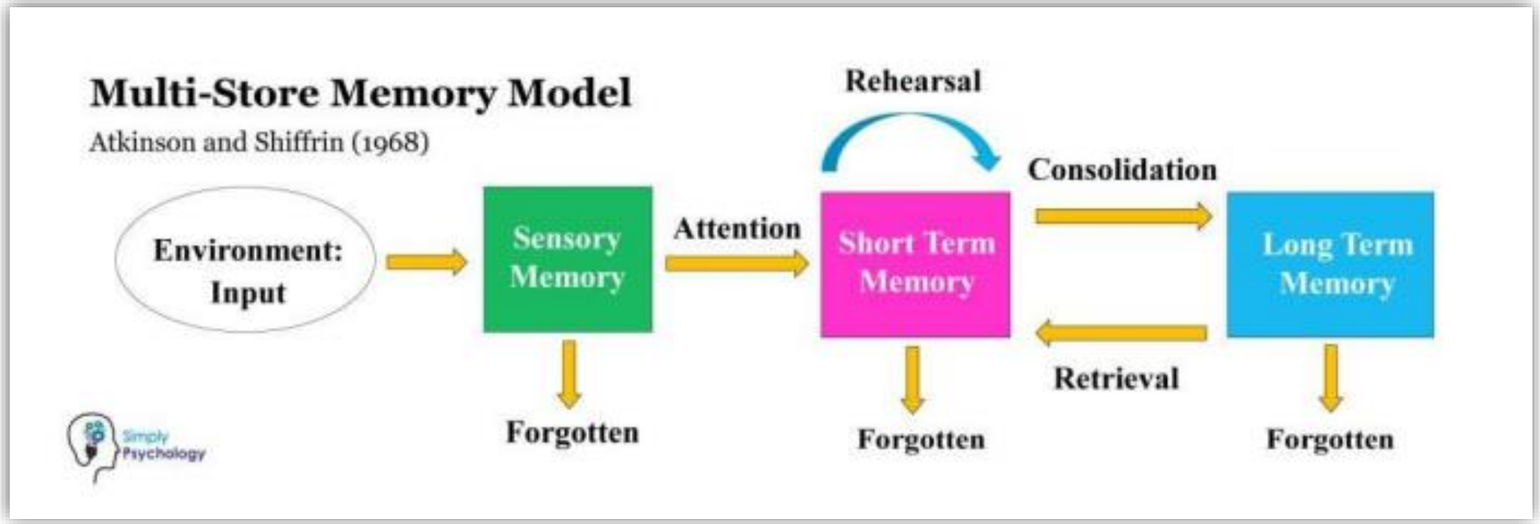
- XMEM (2022. 07)
- VOS의 연구들은 하나의 feature memory만 사용. 그러므로 **1분 이상의 video 처리에 한계가 있음**
- Past frame representation을 memory에 저장, attention을 통해 query 해옴
- 이를 다룰 때 영상의 feature를 압축해서 저장하는 방식. 일부 정보손실을 감수





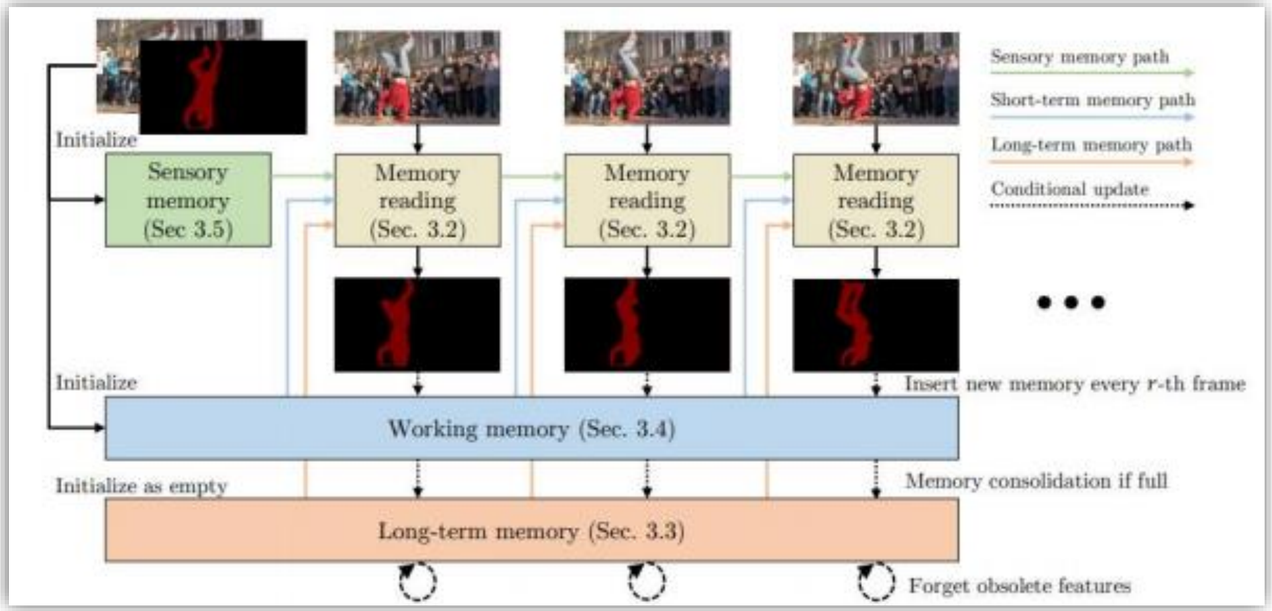
Segment Anything Model & XMem

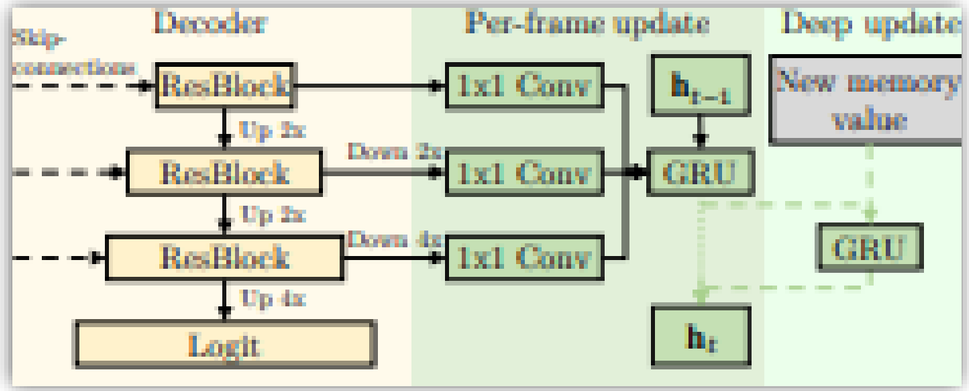
- XMEM (2022. 07)
- A sensory memory
- Fastest reacting, locality sensitive, Implemented with a Conv-GRU
- A working memory
- High-resolution features for query-key-value matching
- A long-term memory
- Compact features consolidated from the working memory, Lasts for 10,000+ frames



Segment Anything Model & XMem

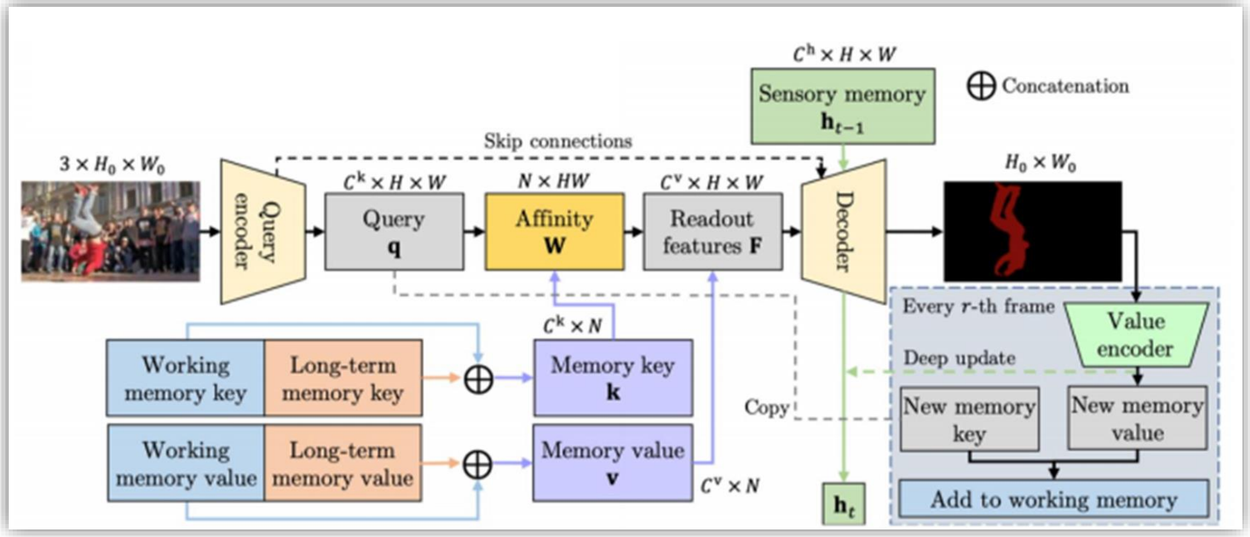
- XMEM (2022. 07) Method
- **Memory reading**은 모든 memory store로부터 얻은 context info를 이용해 매 frame 마다 mask를 출력하는 역할을 맡음 **sensory memory**는 매 frame 마다 업데이트 되고, **working memory**도 매 frame마다 업데이트 됨
- **working memory**가 꽉 차면 압축된 형태로 **long-term memory**로 이동하고, **long-term memory**도 꽉 차면 (thousands of frames 처리 후에 발생) 오래된 feature를 삭제함
- Deep update를 통해 working memory에 저장된 정보가 sensory memory에 중복으로 저장될 필요가 없게 해 줌





Segment Anything Model & XMem

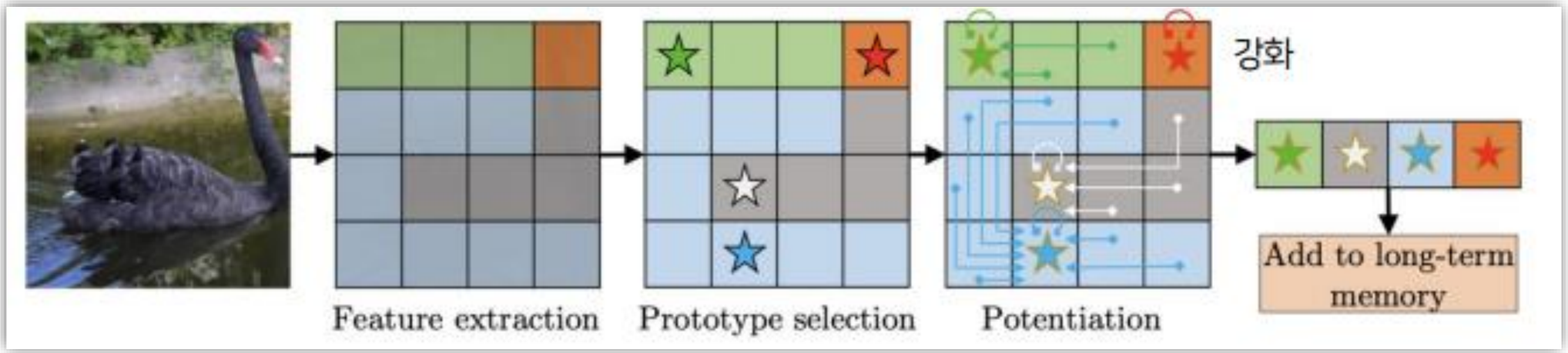
- XMEM (2022. 07) Method
- 현재 프레임과 **feature memory stores**를 입력으로 mask를 출력하는 역할
- **query-key-value attention이** 사용되는데, 현재 프레임이 query로, working memory와 long-term memory가 key-value로 사용
- 현재 프레임을 기반으로 단기/장기 기억 속에서 어떤 정보를 꺼내올지 **feature를 추출**하여 판단
- Affinity matrix는 query와 key의 유사도를 기반으로 계산되는데, **어떤 similarity function을 쓸지도 중요**
- **L2 distance가 더 안정적**이지만, memory element의 중요도를 설정할 수 없는 등 표현력이 부족한 단점이 있음



- Memory Reading : F = vW(k, q)
- 유사도: $\mathbf{S}(\mathbf{k},\mathbf{q})_{ij} = -\mathbf{s}_i \sum_{c}^{C^k} \mathbf{e}_{cj} (\mathbf{k}_{ci} - \mathbf{q}_{cj})^2$

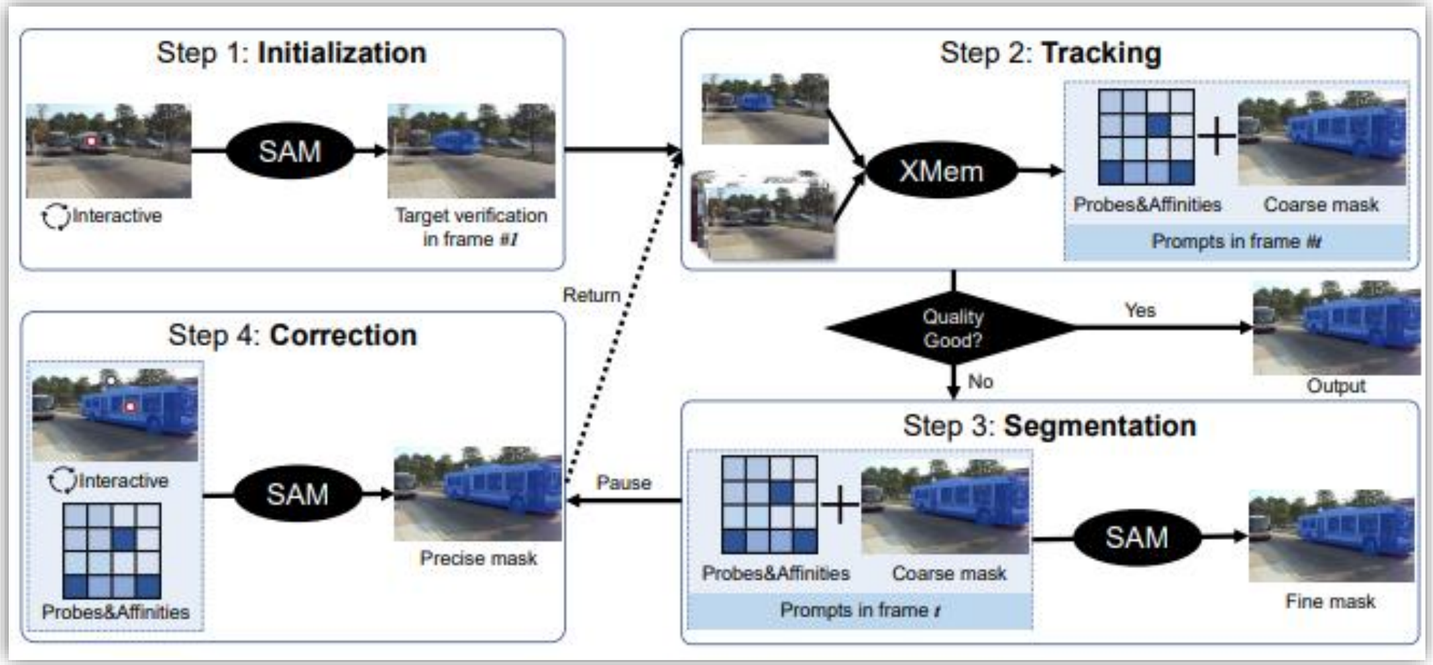
Segment Anything Model & XMem

- XMEM (2022. 07) Method
- Long term memory
- 중요한 것만 넣어야 함
- 사람의 뇌를 모방한 **prototype selection**: Affinity matrix 기준, 가장 자주 사용된 top-P memory element를 고름.
- 6GB GPU memory 기준, 34000 프레임 정도 처리하면 메모리가 가득 참
- 마찬가지로 affinity matrix 기준, **least-frequently-used (LFU) memory element**를 제거
- 실제로는 long-term memory의 크기를 10000 프레임으로 제한해 1.4GB 이상 차지 않게 함



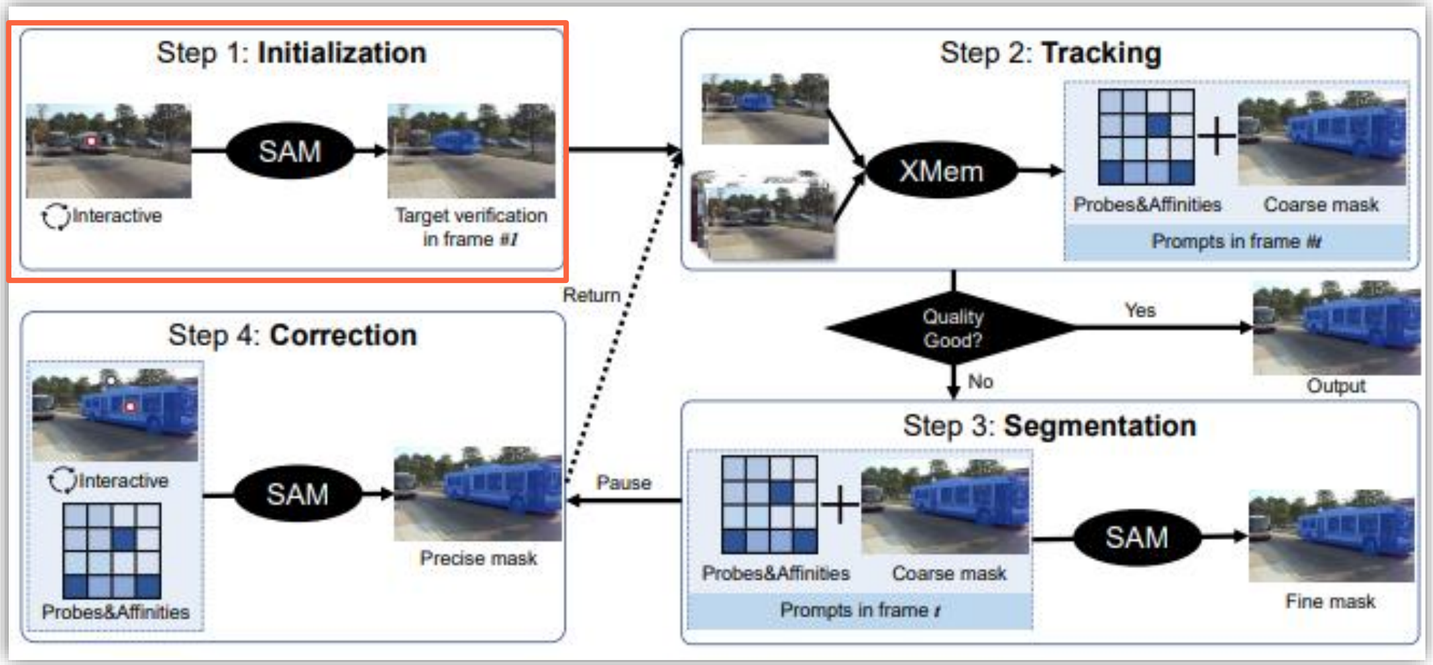
Track Anything Task

- Methodology
- Step 1: Initialization with SAM
- Step 2: Tracking with XMem
- Step 3: Refinement with SAM Permalink
- Step 4: Correction with human participation



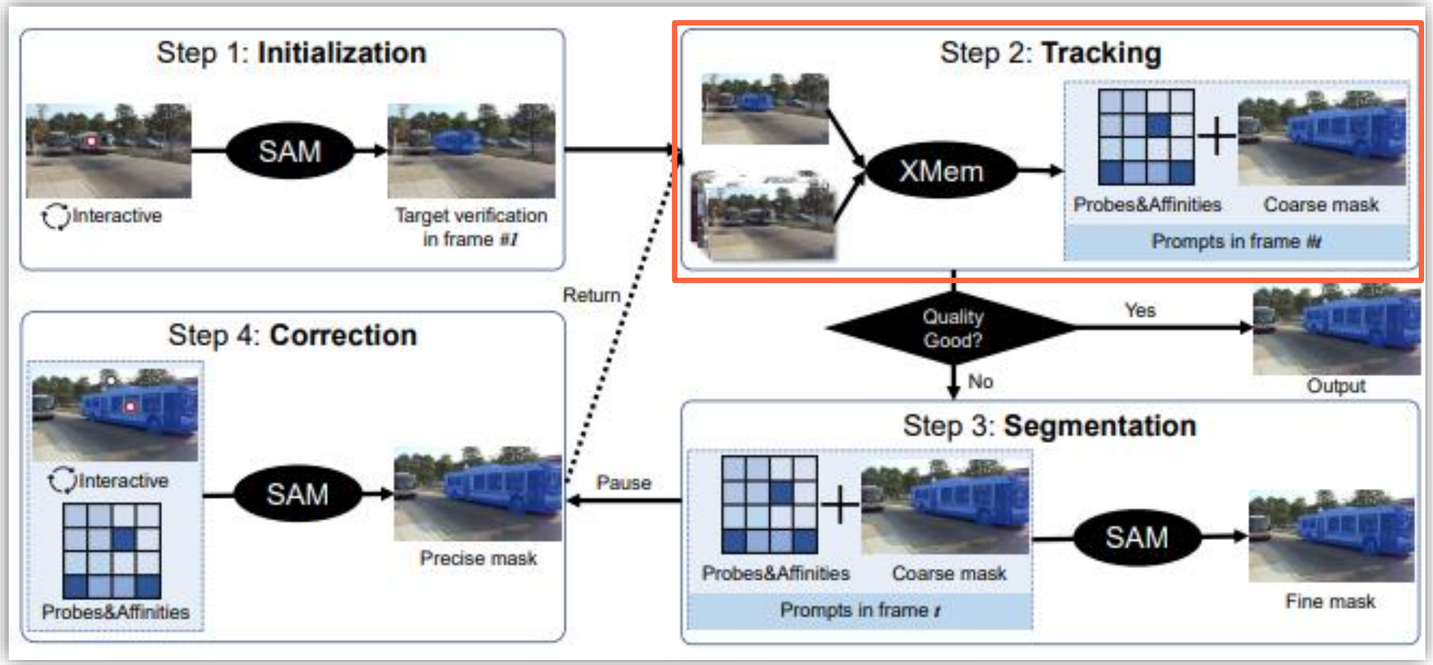
Methodology

- **Step 1: Initialization with SAM**
- 첫번째 프레임에서 SAM을 통해 관심 객체의 **초기 마스크를 얻음**
- SAM에 이어 사용자는 클릭 한 번으로 관심 객체에 대한 마스크 설명을 얻거나 몇 번의 클릭으로 객체 마스크를 수정하여 **만족스러운 initialization를 달성**



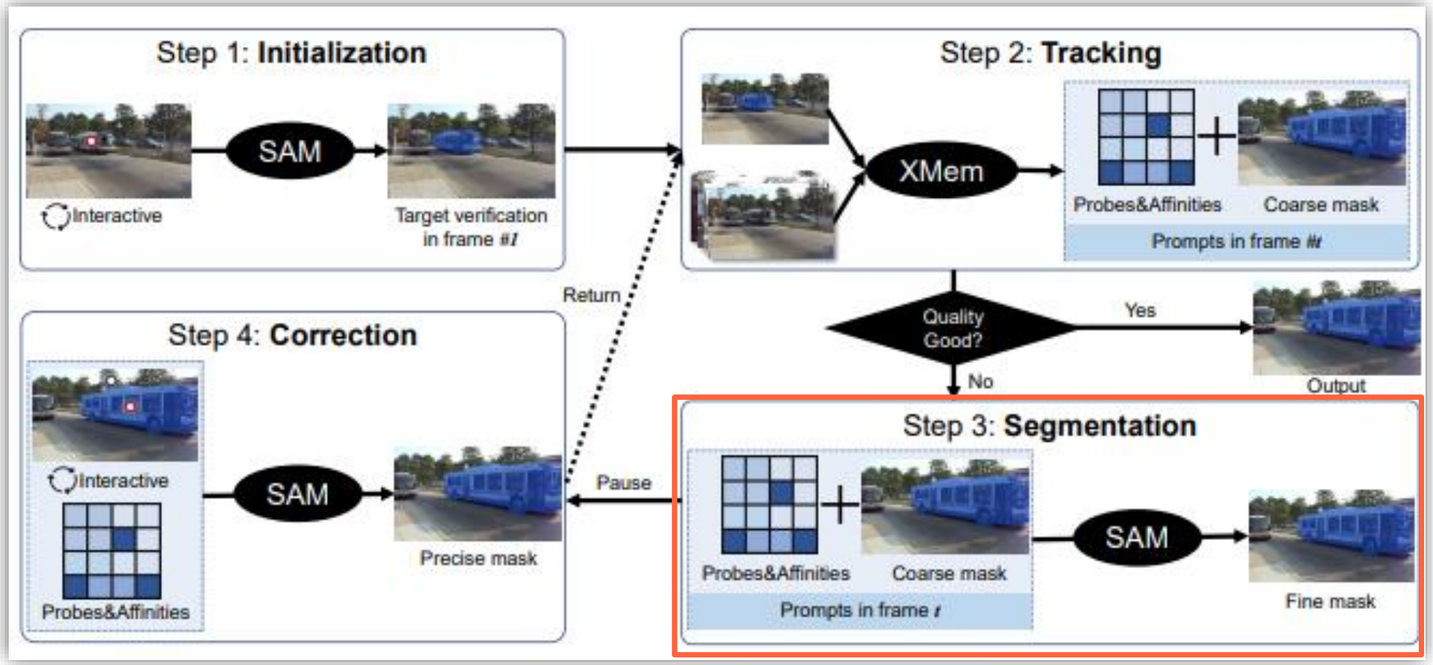
Methodology

- **Step 2: Tracking with XMem**
- XMem을 이용해서 이후 프레임들의 마스크를 얻게 됨
- XMem은 간단한 시나리오에서 만족스러운 결과를 생성할 수 있는 고급 VOS 방법이므로 대부분의 경우 XMem에서 예측된 마스크를 출력
- 마스크의 품질이 좋지 않은 경우 **XMem 예측과 해당 중간 parameter(ex> probe, affinity)를 저장**, 3단계로 건너뜀



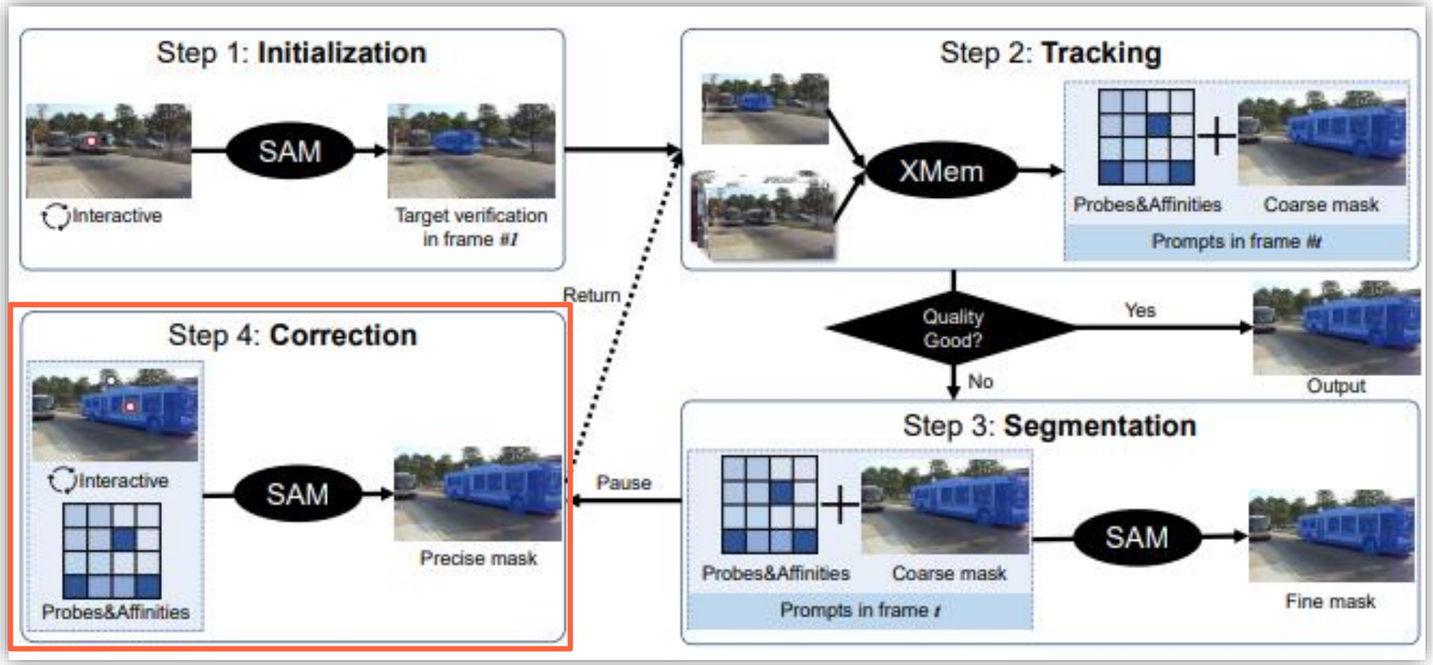
Methodology

- Step 3: Refinement with SAM Permalink
- 데이터의 품질 평가가 만족스럽지 않은 경우 SAM을 사용하여 **XMem이 예측한 마스크를 개선**
- probe와 affinity를 SAM의 포인트 프롬프트로 사용하고 2단계에서 예측된 마스크를 SAM의 마스크 프롬프트 사용
- SAM은 이러한 프롬프트를 사용하여 세분화된 세분화 마스크를 생성
- **세분화된 마스크는 XMem의 시간적 대응에 추가**되어 이후의 모든 객체 식별을 세분화

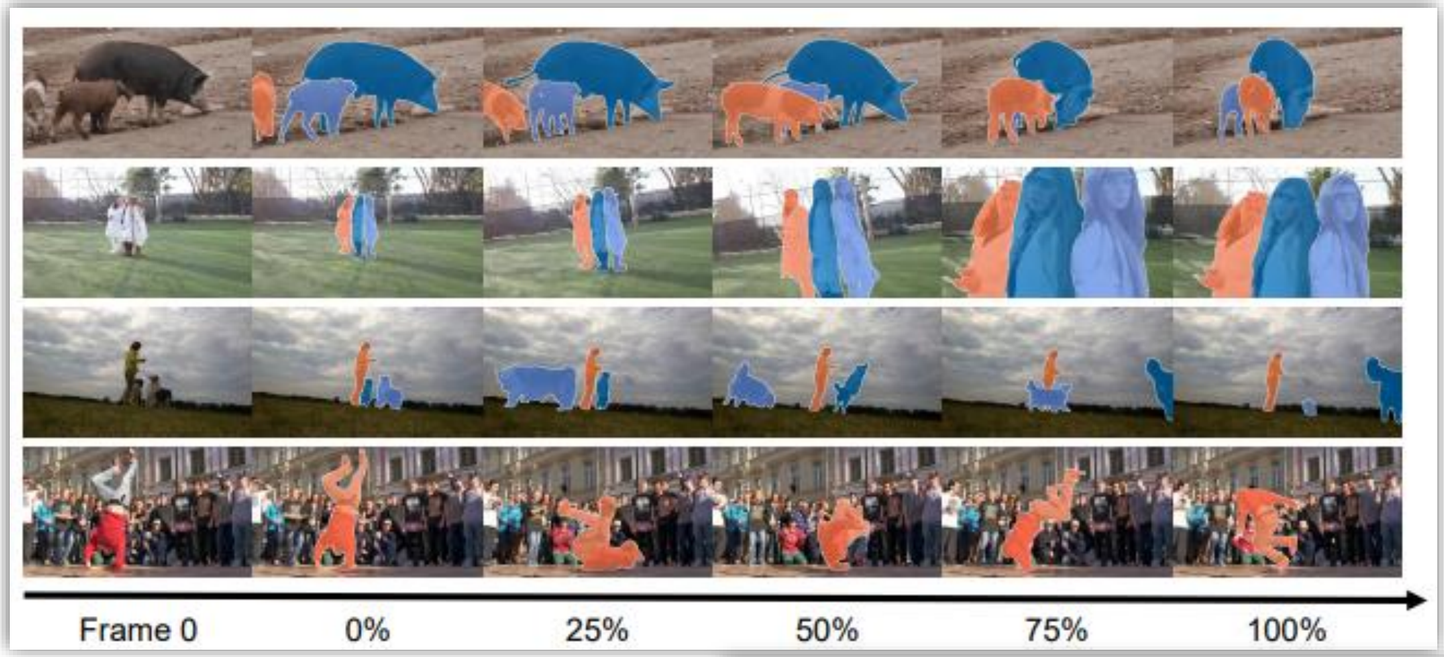


Methodology

- Step 4: Correction with human participation
- 위의 세 단계를 거친 후, TAM은 이제 몇 가지 일반적인 문제를 성공적으로 해결하고 세그먼트 마스크를 예측
- 특히 긴 동영상을 처리할 때와 같이 **매우 어려운 시나리오에서는 여전히 객체를 정확하게 구분하기 어렵다는 점**
- 따라서 추론 중에 사람의 보정을 추가, 이를 통해 사람의 노력을 거의 들이지 않고도 **성능의 질적 향상**
- 특히, 사용자는 **TAM 프로세스를 강제로 중지하고 포지티브 및 네거티브 클릭으로 현재 프레임의 마스크를 수정 가능**



Experiments



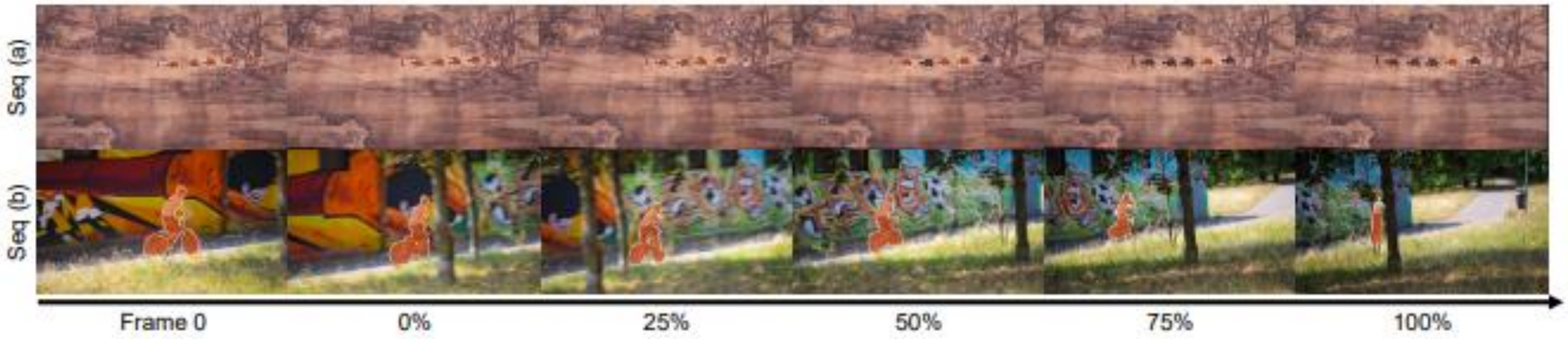
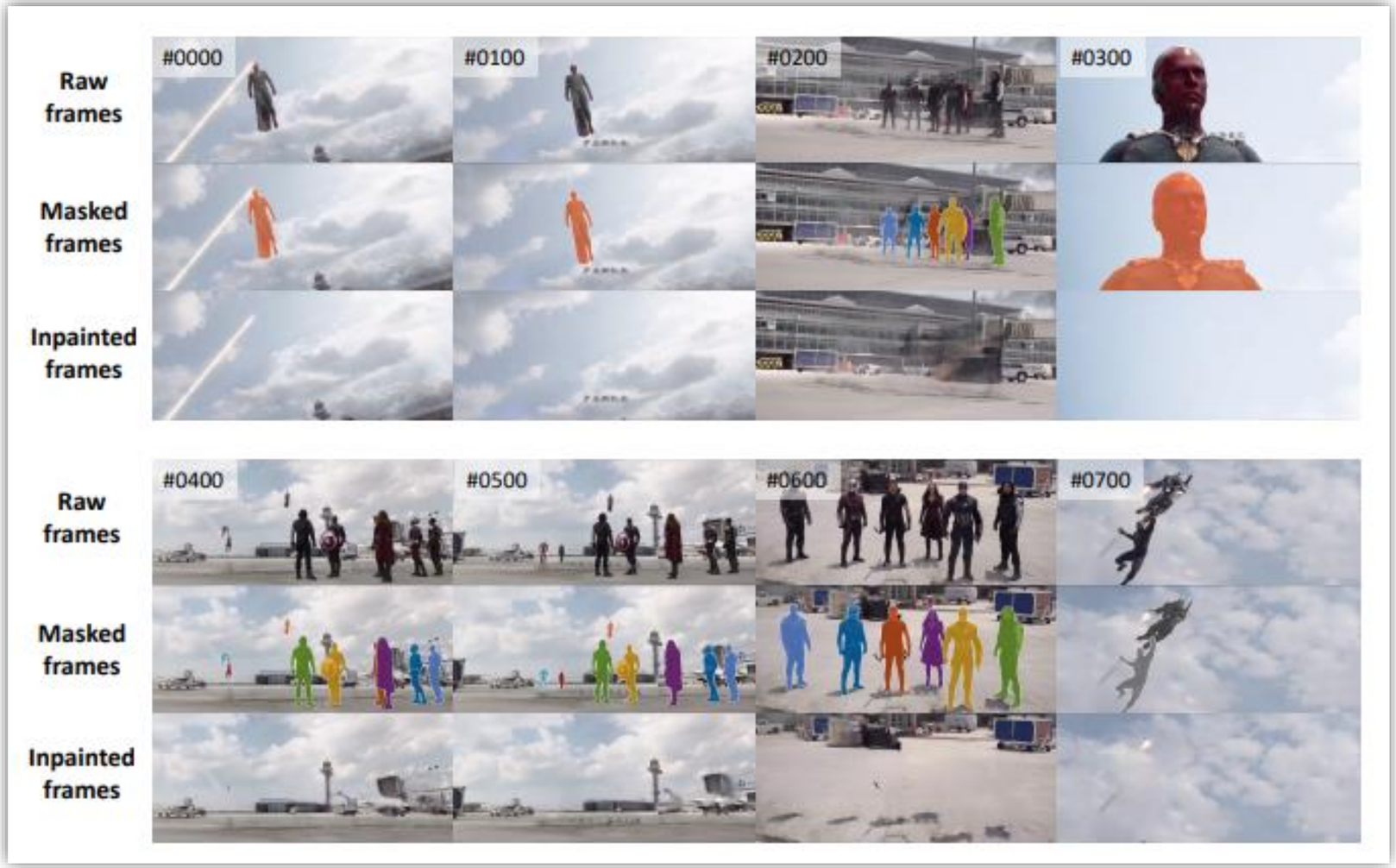


Figure 3: Failed cases.

Experiments



Fail Cases

- 1) 현재 VOS 모델은 대부분 짧은 동영상용으로 설계되어 장기 기억보다 단기 기억 유지에 더 중점. 이는 긴 동영상에서 마스크 축소 또는 개선 부족으로 이어짐. **본질적으로 SAM의 개선 능력으로 Step 3에서 이를 해결하는 것을 목표**로 하지만 실제 적용에서는 그 효과가 예상보다 낮음 인간의 참여/상호작용은 이러한 어려움을 해결하는 접근이 될 수 있지만 **너무 많은 상호작용은 효율성을 떨어뜨리는 결과를 낳음**

- 2) **오브젝트 구조가 복잡한 경우, 클릭을 전파하여 세밀한 초기화 마스크를 얻는 것이 매우 어려움.** 따라서 대략적으로 초기화된 마스크는 후속 프레임에 부작용을 일으켜 잘못된 예측으로 이어질 수 있음. 이것은 또한 SAM이 여전히 복잡하고 정밀한 구조와 씨름하고 있음을 보여줌