

# 芭蕾呪法

## 프레임 뷰어 구현 리포트

2026-02-22 | Streamlit + SMPL 인터랙티브 뷰어

### 1 개요

사용자가 구간을 직접 설정하고 슬라이더로 프레임을 탐색하며 SMPL 메시 및 스켈레톤 비교를 실시간으로 확인할 수 있는 인터랙티브 웹 뷰어를 구현하였다.

**실행 명령:**

```
streamlit run src/viewer_app.py
```

**접속 URL:** <http://localhost:8510>

### 2 구현 기능

기능	상태	비고
사이드바 구간 설정 (시작/종료 프레임)		number_input + Apply 버튼
프레임 슬라이더 (설정 구간 내 탐색)		st.slider, 실시간 업데이트
현재 프레임 정보 표시		Loss, Ref/Comp 원본 프레임 번호
SMPL 메시 비교 이미지		캐시된 PNG 자동 감지 후 표시
스켈레톤 폴백 렌더링		SMPL 없으면 matplotlib 3D 스켈레톤
손실 타임라인 (Plotly)		다크 테마, 현재 프레임 빨간 점선
관절별 손실 TOP 5 표시		우측 컬럼, 실시간 업데이트
JSON 데이터 캐싱		st.cache_data, 재로드 불필요
가장 가까운 SMPL 힌트		“Frame 519에서 캐시 ±63프레임” 등

### 3 화면 구성

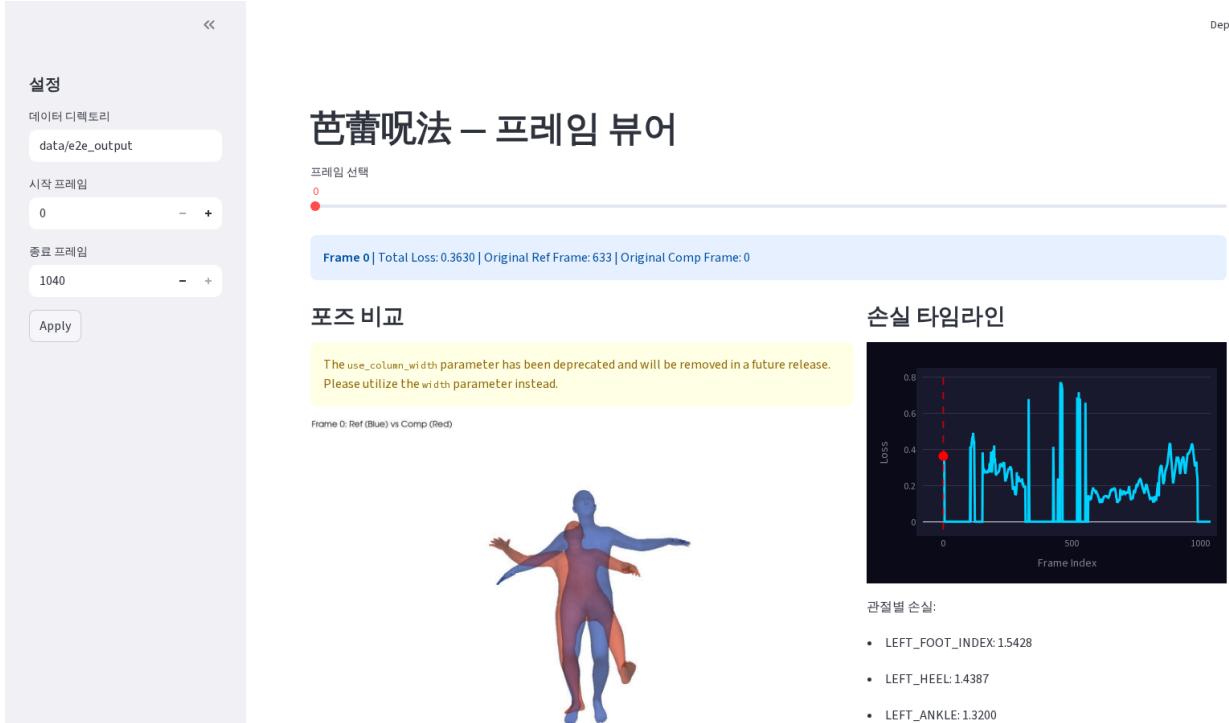


Figure 1: Frame 0 — SMPL 메시 비교 모드. 좌: 포즈 비교(SMPL 메시), 우: 손실 타임라인

## 4 스켈레톤 폴백 모드 (SMPL 미캐시 프레임)

SMPL 이미지가 없는 프레임에서는 world\_landmarks 3D 좌표 기반으로 스켈레톤을 실시간 렌더링한다. 관절명 기반 연결선(어깨-팔꿈치-손목, 힙-무릎-발목 등 16개 연결)으로 자세를 시각화한다.

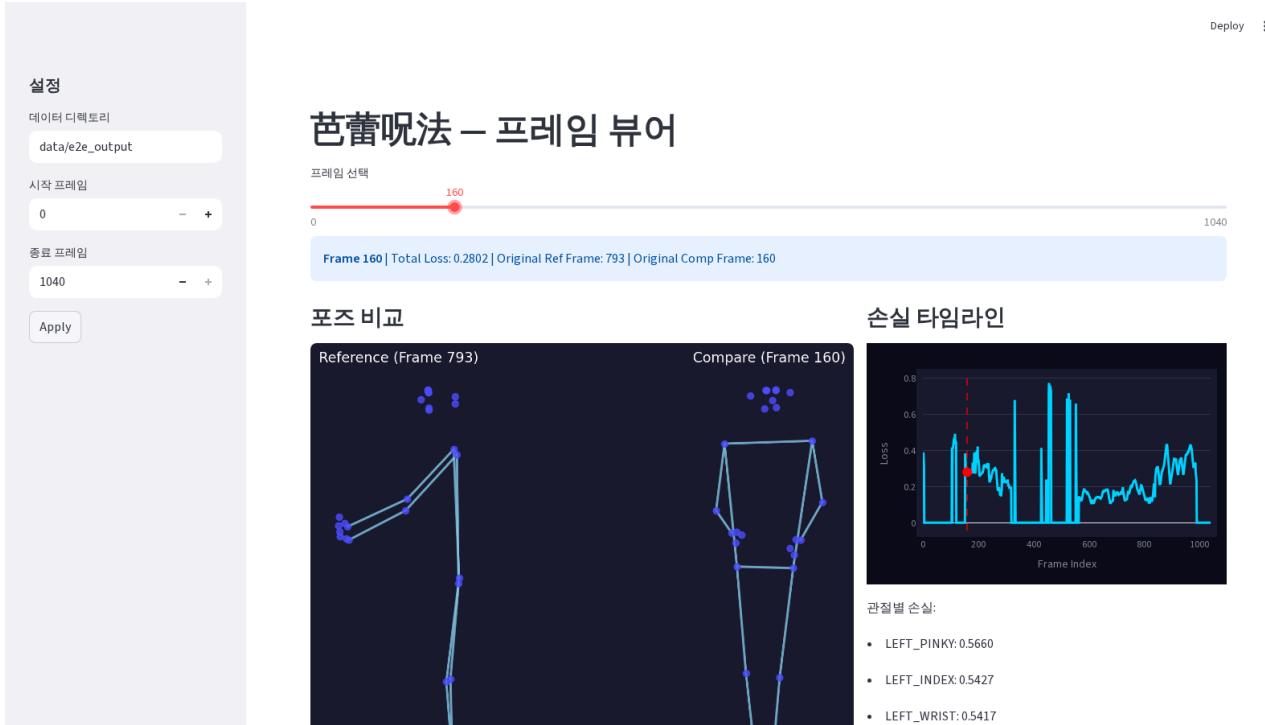


Figure 2: Frame 160 – 스켈레톤 폴백 모드. Reference (Frame 793) vs Compare (Frame 160)

**관찰:** Reference는 한쪽 팔을 측면으로 올린 자세, Compare는 정면을 향한 기본 서기 자세. 두 수행자의 팔 위치 차이가 명확히 시각화되어 있다. Loss: 0.2802.

## 5 최대 손실 프레임 시각화

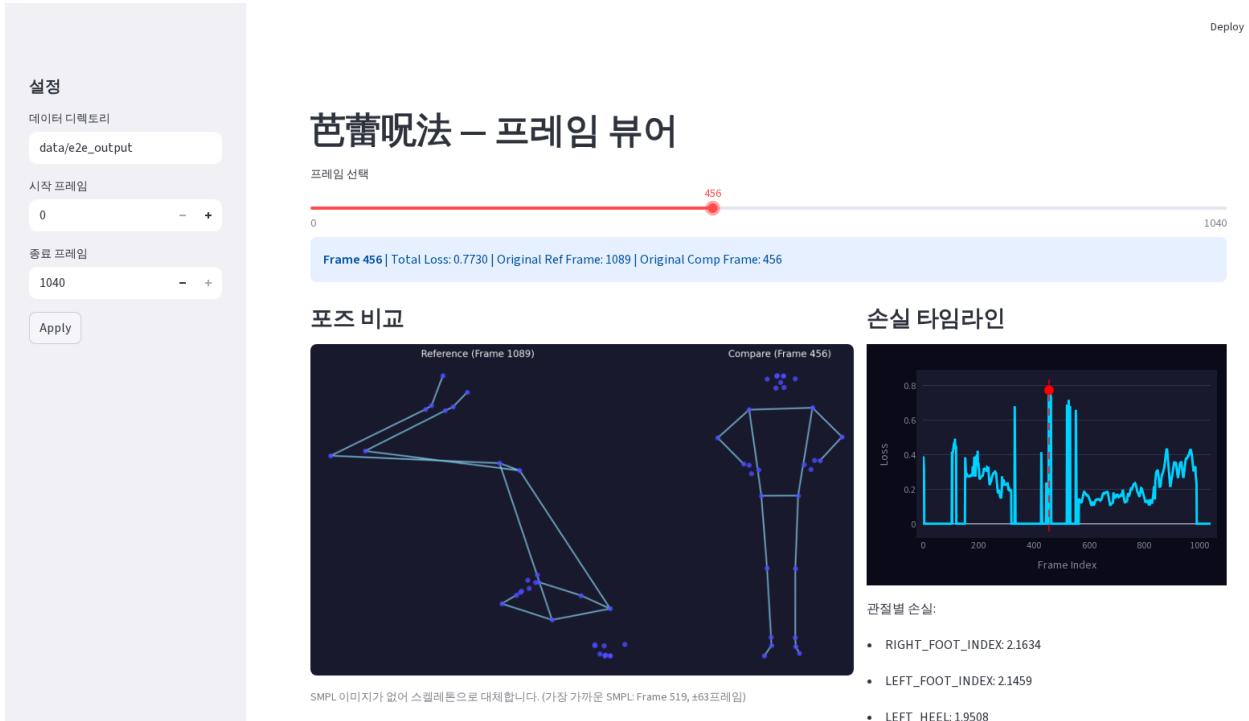


Figure 3: Frame 456 — 최대 손실 프레임 (Loss: 0.7730). 손실 타임라인에서 빨간 점이 피크에 위치

**관찰:** Frame 456은 전체 시퀀스에서 가장 큰 포즈 차이를 보이는 프레임이다. Reference(Frame 1089)는 몸통이 앞으로 크게 기울어진 자세인 반면, Compare(Frame 456)는 직립 서기 자세다. 발끝 관절(FOOT\_INDEX) 오차가 2.16으로 최고치. 손실 타임라인의 빨간 수직선이 피크 위치를 명확히 가리키고 있다.

## 6 후반부 프레임

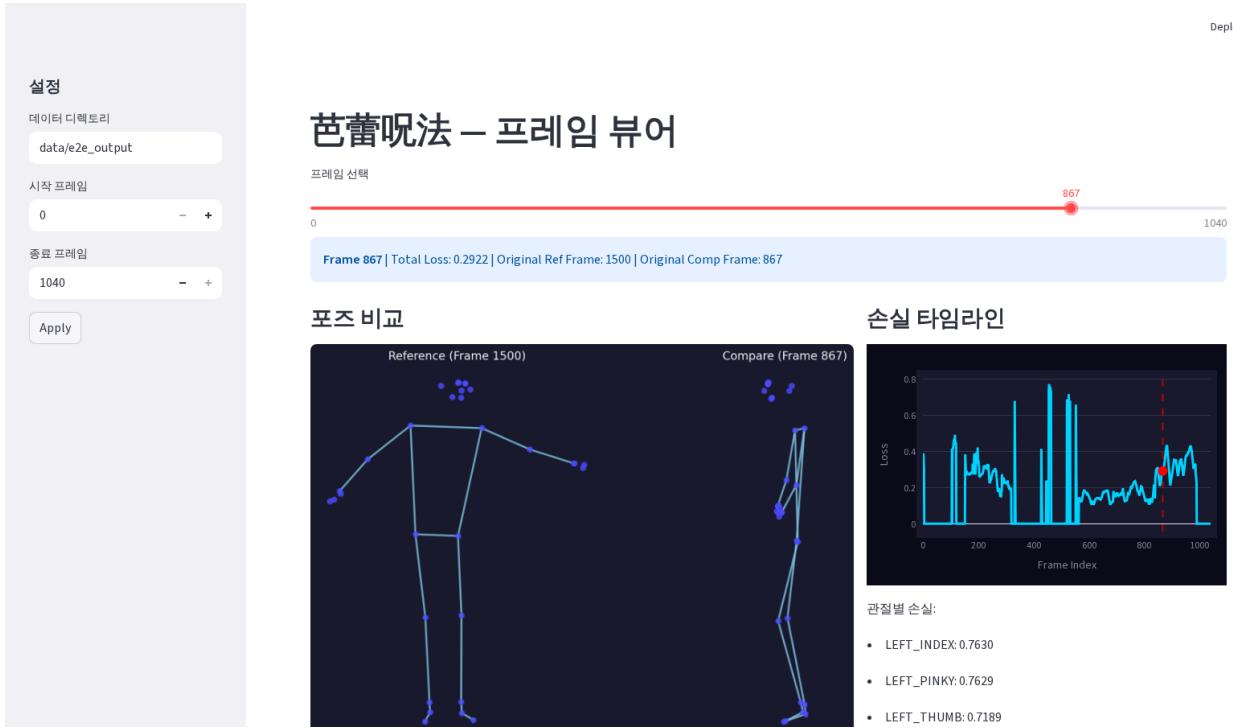


Figure 4: Frame 860 — 후반부. Reference vs Compare 자세 비교 및 손실 분포

## 7 구현 상세

### 7.1 파일 구조

```
src/viewer_app.py          # 메인 Streamlit 앱 (약 280줄)
data/e2e_output/
    comparison_result.json # per_frame_loss, metadata
    reference_poses.json   # world_landmarks (dict 형태)
    compare_poses.json     # world_landmarks (dict 형태)
smpl_seq/
    smpl_seq_frame_XXXX.png # 캐시된 SMPL 프레임들
```

### 7.2 핵심 구현 포인트

**1. 데이터 형식 처리:** `world_landmarks`는 `{"NOSE": {"x": ..., "y": ..., "z": ...}, ...}` 딕셔너리 형태로 저장되어 있어, 리스트로 가정한 초기 코드에서 버그가 발생하였다. 관절명 기반 처리로 수정하여 해결.

#### 2. 스켈레톤 연결 좌표계:

- `world_landmarks` 사용 시:  $y' = -y$  (MediaPipe  $Y \downarrow \rightarrow$  화면  $Y \uparrow$  반전)
- `landmarks` (2D 정규화) 사용 시: `ax.invert_yaxis()` 적용

**3. SMPL 캐시 매팅:** `smpl_seq_frame_{frame_idx:04d}.png` 파일을 스캔하여 현재 프레임과 가장 가까운 캐시 자동 선택. 정확히 일치하는 프레임이 있으면 SMPL 메시 표시, 없으면 스켈레톤 풀백.

#### 4. 성능:

- `@st.cache_data`로 3개 JSON 파일 캐싱 (최초 로드 약 3초, 이후 즉시)
- 스켈레톤 렌더링: `plt.close(fig)` 명시적 호출로 메모리 누수 방지

### 7.3 수정 이력

시도	문제	수정
1	plotly 미설치로 ModuleNotFoundError	<code>pip install plotly</code>
2	<code>world_landmarks</code> 가 dict인데 리스트로 접근하여 IndexError	관절명 기반 dict 처리로 전환

시도	문제	수정
3	비어있는 landmarks 배열에서 ndim 체크 실패	None 필터링 + 관절수 최소값 검증 추가

## 8 실행 방법

# 프로젝트 루트에서 실행

```
streamlit run src/viewer_app.py
```

# 브라우저에서 접속

```
# http://localhost:8501
```

# 사용법:

# 1. 사이드바에서 데이터 디렉토리 입력 (기본: data/e2e\_output)

# 2. 시작/종료 프레임 입력 후 Apply 클릭

# 3. 메인 슬라이더로 프레임 탐색

# 4. 좌측: SMPL 메시 또는 스켈레톤 비교

# 5. 우측: 손실 타임라인 + 관절별 오차

## 9 결론

인터랙티브 프레임 뷰어가 정상 동작하며, 사용자는 발레 동작의 어느 구간에서 포즈 차이가 크고 작은지를 시각적으로 확인할 수 있다. SMPL 메시가 캐시된 프레임(6개)에서는 3D 메시 비교가 제공되고, 나머지 프레임(1,035개)에서는 실시간 스켈레톤 렌더링으로 풀백된다.

향후 SMPL 생성을 더 많은 프레임에 대해 사전 계산하거나, 뷰어 내에서 “이 프레임 SMPL 생성” 버튼을 추가하면 완성도를 높일 수 있다.