

芭蕾呪法

Phase 1: 실제 영상 포즈 추출 + 두 영상 L2 Loss 비교

테스트 리포트

2026-02-14 | /sc:duo 자동 생성

1 작업 요약

유튜브 그랑 바뜨망(Grand Battement) 영상 2개를 다운로드하여 MediaPipe로 포즈를 추출하고, 프레임별 L2 Loss를 계산하는 비교 엔진과 시각화 모듈을 구현하였다.

- **구현 방식:** Claude (오케스트레이터) + Gemini CLI (구현자) 분업
- **Gemini 재시도:** pose_extractor.py MediaPipe API 변경으로 1회 재시도
- **Claude 직접 수정:** visualizer.py import 오류 및 데이터 구조 불일치 4건 수정

1.1.1 구현된 모듈

파일	역할	구현자
src/pose_extractor.py	영상 → MediaPipe Tasks API → 24fps 포즈 JSON 추출 (모델 자동 다운로드)	Gemini + Claude
src/comparator.py	두 포즈 JSON의 프레임별 L2 Loss 계산. 관절별 가중치(하체 2x) 지원.	Gemini + Claude
src/visualizer.py	Loss 그래프, 스켈레톤 오버레이, 관절별 히트맵 PNG 생성.	Gemini + Claude

1.1.2 테스트 데이터

영상	출처	해상도	길이
Reference	YouTube “How to Do a Grand Battement”	854x480	10초
Comparison	YouTube “Quick Ballet Tips - Grand Battement”	640x360	10초

2 테스트 결과

#	테스트 항목	결과	상세
1	포즈 추출 - reference 영상	PASS	297/301 프레임 (98.7%)
2	포즈 추출 - compare 영상	PASS	301/301 프레임 (100%)
3	L2 Loss 비교 엔진	PASS	mean=4.6161, 301프레임
4	시각화 PNG 4종 생성	PASS	4/4 파일

전체 결과: 4/4 PASS

3 비교 분석 결과

3.1 Loss 통계

지표	값
평균 Loss (mean)	4.6161
최대 Loss (max)	11.4594 (프레임 49)
최소 Loss (min)	0.0000 (프레임 12)
표준편차 (std)	0.9537
비교 프레임 수	301

3.2 프레임별 Loss 그래프

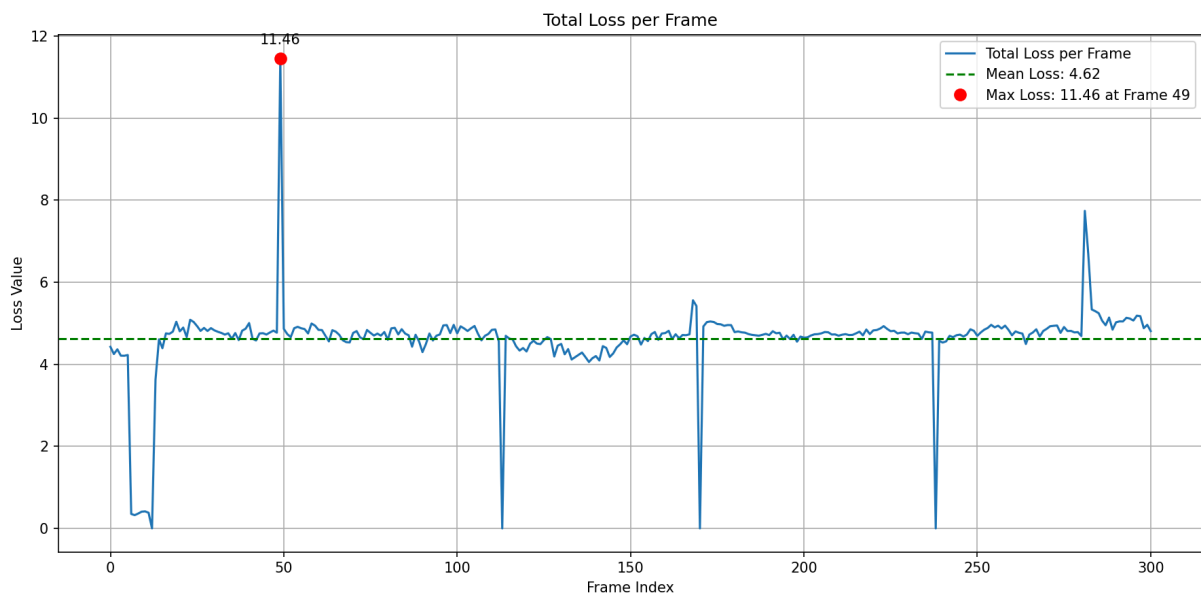


Figure 1: 프레임별 L2 Loss 추이. 주황 점선은 평균, 빨간 점은 최대 Loss 지점(프레임 49).

3.3 스켈레톤 비교: 최대 Loss 프레임 (49)

Skeleton Comparison at Frame 49

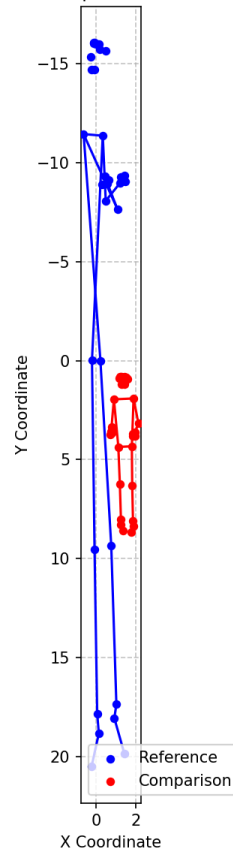


Figure 2: 프레임 49에서의 스켈레톤 비교. 파란색=Reference, 빨간색=Comparison. 두 동작이 가장 큰 차이를 보이는 구간.

3.4 스켈레톤 비교: 최소 Loss 프레임 (12)

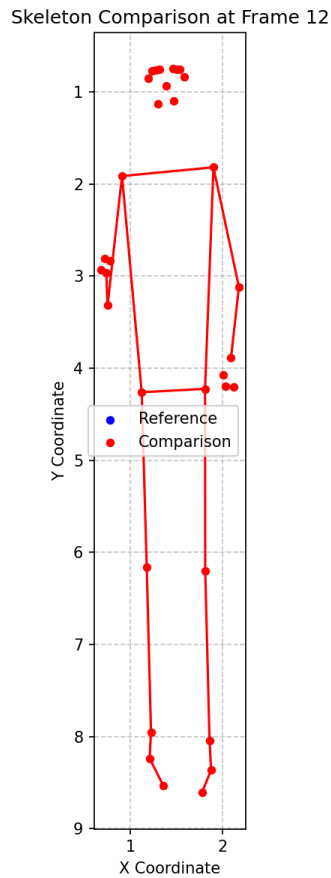


Figure 3: 프레임 12에서의 스켈레톤 비교. 두 동작이 가장 유사한 구간.

3.5 관절별 평균 Loss (상위 10개)

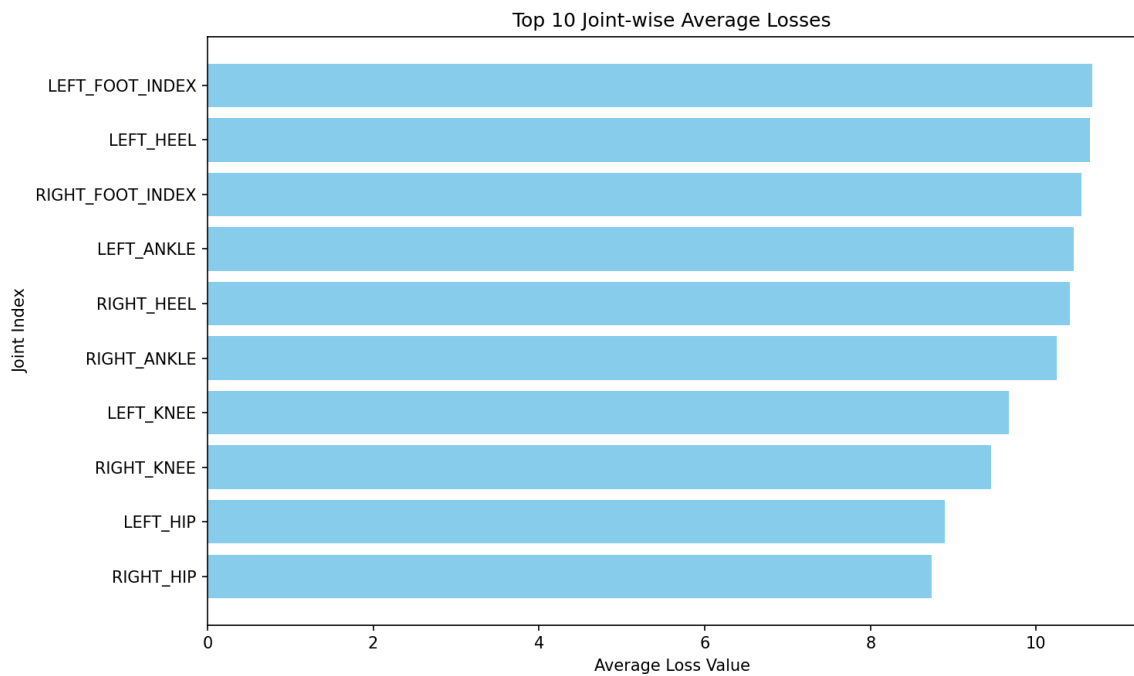


Figure 4: 관절별 평균 Loss. 하체 관절에 2x 가중치가 적용되었으므로 하체 관절이 상위에 위치.

4 Gemini CLI 실행 로그

#	작업	Gemini 성공	Claude 수정
1	pose_extractor.py - MediaPipe Tasks API 마이그레이션	재시도 1회 후 성공	없음
2	comparator.py - L2 Loss 비교 엔진	1회 성공	DEFAULT_WEIGHTS 중복 선언 수정
3	visualizer.py - 시각화 모듈	1회 성공	import 오류 + 데이터 구조 불일치 4건 수정

5 결론 및 다음 단계

5.1.1 완료된 작업

- 유튜브 그랑 바뜨망 영상 2개 다운로드 및 포즈 추출 (MediaPipe Tasks API)
- 좌표 정규화 후 프레임별 L2 Loss 비교 엔진 구현
- Loss 그래프, 스켈레톤 오버레이, 관절별 히트맵 시각화

5.1.2 발견 사항

- MediaPipe 0.10.32에서 기존 `mp.solutions` API가 제거됨 → `mp.tasks.vision` 사용 필요
- 두 영상의 카메라 앵글/거리 차이로 인해 정규화 후에도 Loss가 높음 (mean 4.6)
- 이는 “동작의 차이”보다 “촬영 환경의 차이”가 더 큰 영향을 줌 → 향후 3D 포즈 필요

5.1.3 다음 단계 (Phase 2)

1. 음악 BPM 동기화 (librosa로 비트 추출 → 프레임 매핑)
2. 비트 기준 Loss 집계 (프레임별 → 박자별)
3. 실시간 카메라 입력 지원 (배치 → 스트리밍)