

동영상 내 중복 프레임 제거를 통한 효율적 이미지 추출 기법

김병현¹, 김주하², 남현우³
^{1,2} 인덕대학교 컴퓨터 소프트웨어학과 학부생
³ 인덕대학교 컴퓨터 소프트웨어학과 교수

Kimbh0013@gmail.com, khabh@naver.com, namhw@induk.ac.kr

An Efficient Method for Image Extraction from Videos by Removing Duplicate Frames

ByeongHyeon Kim¹, Juha Kim², HyunWoo Nam³
^{1,2} Dept. of Computer Science, Induk University
³ Dept. of Computer Science, Induk University

요 약

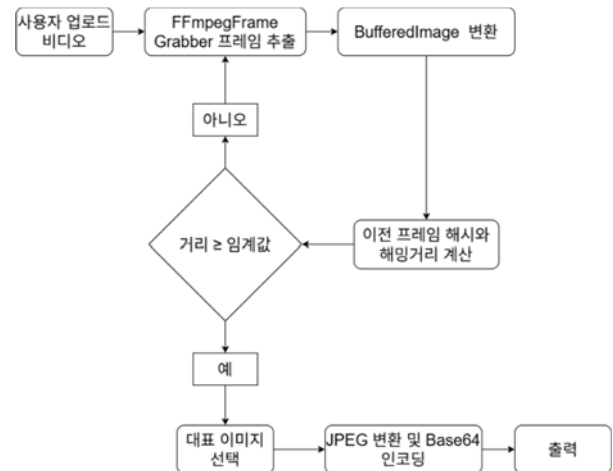
본 논문은 동영상의 중복 프레임을 제거하고 대표 장면을 효율적으로 추출하는 기법을 제안한다. 프레임 간의 시각적 유사도를 평균 해시(Average Hash)와 해밍 거리(Hamming Distance)로 측정하고, 임계값 기반 필터링을 통해 유사도가 낮은 프레임만을 선별해서 추출한다. 이 방식은 저장 비용을 크게 절감하며, 직관적인 제어를 통해 효과적인 시각적 요약 생성을 가능하게 한다.

1. 서론

디지털 미디어 기술의 발전으로 동영상 콘텐츠의 양이 폭발적으로 증가하고 있다. 하지만 동영상은 연속된 프레임의 집합이므로, 사용자가 원하는 특정 장면이나 제품의 세부 모습을 확인하기 위해서는 타임라인을 직접 조작해야 하는 비효율적인 탐색 과정을 거쳐야 한다. 특히 360 도 제품 영상과 같이 유사한 프레임이 반복되는 콘텐츠의 경우 이러한 문제는 더욱 심화된다. 이러한 문제를 해결하기 위해, 본 논문에서는 동영상 내의 시각적 중복성을 효과적으로 제거하고 내용의 변화를 잘 나타내는 대표 프레임만을 자동으로 선별하는 기법을 제안한다. 제안 기법은 인접 프레임 간의 유사도를 평균 해시(Average Hash)와 해밍 거리(Hamming Distance)로 측정하고, 사전에 정의된 임계값을 기반으로 내용 변화가 감지된 프레임만을 추출한다. 이를 통해 별도의 특수 장비 없이도 동영상을 직관적인 이미지들의 집합으로 요약하여 제공함으로써, 데이터 저장 비용을 절감하고 사용자의 콘텐츠 탐색 효율을 극대화하는 것을 목표로 한다. 본 논문은 다음과 같이 구성된다. 2 장에서는 제안하는 프레임 추출 기법의 상세한 과정을 기술하고, 3 장에서는 실험을 통해 기법의 성능을 분석한다. 마지막으로 4 장에서 결론을 맺는다.

2. 제안하는 동영상 이미지 추출 기법

본 장에서는 동영상으로부터 대표 이미지를 효율적으로 추출하기 위해 제안하는 알고리즘의 전체적인 구조와 각 단계별 세부 사항을 설명한다.



(그림 1) 이미지 추출 플로우 차트

2.1. 프레임 추출

사용자가 업로드한 비디오를 입력으로 받아 FFmpeg[1] 라이브러리를 활용한 FFmpegFrameGrabber[2]로 프레임을 순차적으로 획득한다. 각 프레임은 Java2DFrameConverter[2]를 통해 Java에서 처리 가능한 BufferedImage 객체로 변환한다. 이 단계는 원본 영상 데이터를 후속 처리 단계에서 분석 가능한 이미지 형식으로 준비하는 과정이다.

2.2. 평균 해시 기반 특성화

각 프레임 이미지에 대해 ImageHashUtil.averageHash를 적용하여 고정 길이의 평균 해시(Average Hash)를 생성한다. 평균 해시(Average Hash)는 이미지의 전역적

명암·구조 정보를 64 비트의 해시값으로 요약하므로, 조명 변화나 미세한 노이즈에 비교적 강인하게 유사도를 판별할 수 있는 특징을 가진다.

2.3. 해밍 거리 기반 유사도 산출

연속되는 프레임의 이미지 해시(hash)값에 대해 해밍 거리(Hamming Distance)를 계산하여 유사도를 판별한다. 해밍 거리는 두 해시 문자열의 각 자리를 비교하여 서로 다른 문자의 총 개수를 세는 방식으로, 이 값이 작을수록 시각적으로 유사한 프레임임을 의미한다. 이렇게 산출된 거리 값이 사전에 정의된 임계값을 넘을 경우에만, 해당 프레임을 의미 있는 장면 전환으로 판단하여 이미지로 추출한다. 이를 통해 유사한 장면은 건너뛰고 영상의 주요 장면만을 효율적으로 선택할 수 있다.

2.4. 임계값 기반 선택

해밍 거리가 사전에 정의된 임계값 이상일 때만 해당 프레임을 대표 이미지로 채택한다. 임계값은 열거형 SnapshotDetailLevel 을 통해 조절되며, 임계값이 크면 클수록 변화가 큰 사진들만 대표 이미지로 채택이 된다.

```
public enum SnapshotDetailLevel {
    HIGH( hammingThreshold: 7), // 세밀하게 - 비슷한 것도 따로 추출
    MEDIUM( hammingThreshold: 15), // 기본값 - 적당히 유사한 것만 제거
    LOW( hammingThreshold: 20); // 굵직하게 - 많이 유사해야 중복으로
```

(그림 2) 임계값 선택을 위한 임계값 정의 및 값 표시

2.5. 출력 형식

선택된 프레임은 JPEG 로 인코딩한 후 Base64 문자열로 변환하여 data:image/jpeg;base64 형태의 데이터 URI 스킴(Data URI Scheme)으로 반환된다.

이 방식은 이미지를 별도의 파일로 저장하고 URL 을 통해 접근하는 방식과 달리, 이미지 데이터를 텍스트 형태로 API 응답(e.g. JSON)에 직접 포함시킬 수 있다. 그 결과, 클라이언트는 추가적인 HTTP 요청 없이 즉시 이미지를 렌더링할 수 있으며, 데이터베이스에도 별도의 파일 처리 로직 없이 텍스트 필드에 저장 가능해 전체 시스템 아키텍처를 단순화하는 장점을 가진다.

3. 실험 결과 및 기대효과

본 장에서는 2 장에서 제안한 동영상 대표 이미지 추출 기법의 성능과 효율성을 검증하기 위해 수행한 실험 내용과 그 결과를 기술한다. 실험은 제안된 기법이 다양한 임계값 설정에 따라 얼마나 효과적으로 중복 프레임을 제거하고 의미 있는 장면을 추출하는지 정량적, 정성적으로 평가하는 것에 목적을 둔다.

3.1. 실험 방법

제안하는 기법의 성능 평가를 위해, ‘디지털 전시

솔루션’ 시나리오에 맞춰 360 도 제품 동영상(5 초, 30fps)을 실험 데이터로 사용하였다.

3.2. 실험 결과



(그림 3) 임계값 설정에 따른 추출 결과 비교
좌: 세밀한 임계값, 우: 굵직한 임계값

3.3. 타 기법과의 성능 비교

알고리즘 (Algorithm)	처리 시간 (ms)	추출된 프레임 수
Average Hash (제안 기법)	180	4
Perceptual Hash (dHash)	250	4
Cosine Similarity	410	6

(표 3) 각 유사도 평가 기법의 성능 비교표
실험 결과, 제안 기법인 Average Hash 는 Perceptual Hash 대비 약 28%, Cosine Similarity 대비 약 56% 빠른 처리 속도를 보였다. 이는 실시간 처리가 요구되거나 대용량 비디오를 다루는 환경에서 제안 기법이 매우 실용적임을 시사한다.

3.4. 기대 효과

논문에서 제안하는 기법은 다음과 같은 기대효과를 가진다. (i) 비용 절감: 중복 프레임 저장을 원천적으로 방지하여 저장 공간 및 데이터 전송 비용을 크게 절감한다. (ii) 품질 향상: 장면 전환이 일어나는 주요 지점의 프레임을 대표 이미지로 선별함으로써 동영상 요약의 품질을 향상시킨다. (iii) 제어 용이성: 단일 임계값 파라미터를 통해 추출 정밀도와 결과물의 압축률 사이의 트레이드 오프(trade-off)를 직관적으로 제어할 수 있다.

4. 마무리

본 논문은 평균 해시 기반의 효율적인 동영상 프레임 추출 기법을 제안했다. 정량적 비교를 통해 타 기법 대비 높은 계산 효율성을 입증했으며, 이는 비용 절감과 사용자 경험 개선에 기여하여 실용적 가치가 높다.

참고문헌

- [1] [JAVA] FFmpeg 로 Thumbnail 추출하기 - 2025.09.16. <https://byul91oh.tistory.com/417>
- [2] The Bytedeco Team, “JavaCV”, Bytedeco, - 2025.09.16. <https://github.com/bytedeco/javacv>.