

감시 시스템을 위한 동영상 데이터의 다단계 관리 및 시공간 검색 기법 연구

이회정(李姬貞), 이원석(李元碩)
연세대학교 컴퓨터과학과 데이터베이스 연구실
(rosa,leewo)@amadeus.yonsei.ac.kr

A Study on Multi-stage Management and Spatio-Temporal Search of Video Features for a Surveillance System

Hui-Jeong Lee, Won-Suk Lee
Database Lab., Dept. of CS, Yonsei University
(rosa,leewo)@amadeus.yonsei.ac.kr

요약

오늘날 멀티미디어 및 인터넷 서비스가 눈에 띄게 증가하면서 다양한 응용분야에서의 동영상 데이터 활용이 급증하였고 이에 사용자가 원하는 동영상 데이터를 빠르고 정확하게 검색하기 위한 내용기반 검색기법이 필수적이다. 본 논문은 high-level features와 더불어 동영상의 고유 내용 속성에 속하는 low-level features를 기반으로 Video Feature Classification을 정의하는 한편 효율적인 시공간 내용기반 검색을 지원한다. Video Feature Classification은 사용자가 사용하기 불편한 low-level features를 자동 일반화(generalization)하여 다단계 관리하고 features에 대한 가중치 적용질의를 제공함으로써 기존 내용기반 검색 연구와는 뚜렷한 차별성을 갖는다. 또한 low-level features와 high-level features간의 자동 변환(translation)을 가능하게 함으로써 동영상 데이터베이스의 사용자 접근 효율을 한단계 높이고 보다 의미구조화된 동영상 관리 및 내용기반 검색을 지원한다.

1. 서론

오늘날 멀티미디어 및 인터넷 서비스가 증가되고 동영상 데이터의 활용 및 정보 가치가 눈에 띄게 향상된 가운데 동영상 처리 기술에 대한 질 높은 연구가 국내외에서 활발히 수행되고 있다. 동영상 데이터는 감시 영상, 의학 영상, 위성 사진, 과학 실험 등 다양한 분야에서 활용되고 있으며 동영상 처리기술 연구와 더불어 방대한 동영상 데이터에 대해 원하는 데이터를 빠르고 정확하게 검색하는 내용 기반 연구가 필수적이다.

기존의 동영상 내용기반 검색 방법은 keyword, subject와 같이 텍스트위주의 high-level features를 이용한 검색 방법 [1,2]과 color, texture, shape와 같이 단순한 Image features위주의 low-level features를 이용한 검색 방법 [3,4]이 주를 이루어 왔다. [1,2]의 경우 방대하고 복잡한 의미를 갖는 동영상 데이터를 색인하고 검색하기에는 많은 시간과 노력이 요구되므로 시스템 관리상 부적합하며 [3,4]의 경우 동영상은 Image와 달리 시공간적인 특징을 가지고 있기 때문에 단순한 color, texture, shape 등의 Image features만 가지고 검색하기에는 미흡한 점이 많다. 따라서 high-level features와 동영상이 실제로 가진 고유의 내용적 특징 즉, low-level features를 함께 사용하는 방법이 고려되어야 한다 [5]. 본 논문은 표1에서 정의한 low-level features와 high-level features를 결합한 시공간 내용기반 검색 기법을 제시함과 동시에 Video Features Classification을 이용해 다단계 low-level features를 자동 관리한다. 사용자가 사용하기 불편한 low-level features를 손쉽게 질의에 반영하고 low-level features 가중치 적용 기법을 제공함으로써 기존 내용기반 검색 기법의 단점이던 high-level

features와 low-level features간의 변환(translation) 문제를 해결한다.

표1. 동영상 features

Low-level Features	Visual Features Motion Features	Color, Texture, Shape, Object# Motion, Time, MotionActivity
High-level Features	Semantic Features	이름, 행위속성, 내용속성

컴퓨터 동영상 처리 기술에는 아직 한계가 있기 때문에 Image 검색에 비해 동영상 검색 기술은 상당한 제약을 받고 있다. 이에 응용사례 또한 제한된 범위에서만 가능하다. 본 논문은 감시 시스템이라는 특정 도메인 안에서 보다 정확한 내용기반 검색 실험 결과를 얻고자 하며 동시에 일반적인 동영상 features에 대한 데이터베이스 자동화 관리기법을 연구한다.

2. Video Feature Classification

Video Feature Classification이란 일반적으로 color, texture, shape, motion등과 같은 동영상 low-level features를 데이터베이스 내에서 자동 일반화하여 관리하는 기법이다. 데이터베이스 내 소스 예제들의 공통 feature를 루트 노드까지 일반화 (Generalization)함으로써 해당 노드에 공통 features값이 부분적으로 유지된다. 따라서 사용자는 원하는 동영상에 대한 low-level feature를 선택하여 질의 가능하다. Feature Generalization 알고리즘은 Video Features Classification의 핵심 부분이며 low-level features의 가중치 적용 질의 및 high-level features와 low-level features 간의 자동 변환 (translation)을 지원한다. high-level features를 포함한 자동 low-level features 질의와 가중치 적용 질의, 사용자 또는 시

스텝 정의 브라우징 검색이 주요 특징이다.

2.1. 동영상 Features

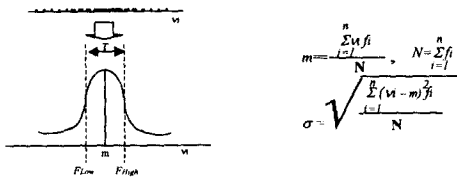
동영상 features는 처리과정에서 수학적 표현되며 표2는 유사도 측정에 앞서 각 feature값을 정의한 것이다. 비교 안의 응용 기반으로 수학적 표현되어 데이터베이스에 저장된다.

표2. 동영상 features

Color	(R, C, G, C2, B - C)	$C = \frac{\sum_{i=1}^n f_i}{N}$	각 Feature는 0 ~ 255의 범위를 가진다. Color는 0 ~ 255의 범위를 가진다. Color는 0 ~ 255의 범위를 가진다.
Texture	(R, T1, G, T2, B, T3)	$T = \frac{\sum_{i=1}^n f_i}{N}$	각 Feature는 0 ~ 255의 범위를 가진다. Texture는 0 ~ 255의 범위를 가진다. Texture는 0 ~ 255의 범위를 가진다.
Shape	S		각 Feature는 0 ~ 255의 범위를 가진다. Shape는 0 ~ 255의 범위를 가진다. Shape는 0 ~ 255의 범위를 가진다.
Motion	M		각 Feature는 0 ~ 255의 범위를 가진다. Motion는 0 ~ 255의 범위를 가진다. Motion는 0 ~ 255의 범위를 가진다.
Duration (=Time)	D		각 Feature는 0 ~ 255의 범위를 가진다. Duration는 0 ~ 255의 범위를 가진다. Duration는 0 ~ 255의 범위를 가진다.
Object#	O		각 Feature는 0 ~ 255의 범위를 가진다. Object#는 0 ~ 255의 범위를 가진다. Object#는 0 ~ 255의 범위를 가진다.
MotionActivity	MA		각 Feature는 0 ~ 255의 범위를 가진다. MotionActivity는 0 ~ 255의 범위를 가진다. MotionActivity는 0 ~ 255의 범위를 가진다.
이동, 움직임, 이동속성, 이동속성	이동, 움직임, 이동속성, 이동속성	(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z)	각 Feature는 0 ~ 255의 범위를 가진다. 이동, 움직임, 이동속성, 이동속성은 0 ~ 255의 범위를 가진다. 이동, 움직임, 이동속성, 이동속성은 0 ~ 255의 범위를 가진다.

2.2. Probability Density Graph

Video Feature Classification에 포함된 소스 예제의 유사도를 측정하기 위해 그룹내 각 feature에 대해 밀도 그래프를 생성한다. PDG(Probability Density Graph)는 분포값 중 가장 밀집되어 있는 값들을 뽑아내며 평균(m)과 표준편차(o)로 구성된 정규분포 그래프 형태를 갖는다. PDG는 실제 분포값에서 [그림1]과 같이 생성되고 [그림2]의 RC(Range Components)로 데이터베이스에 저장된다.

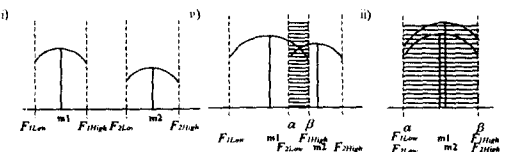


[그림1] Probability Density Graph 생성

(m, Flow, Fhigh, C, R(n)) F ∈ (Features)
 Features = {color, texture, shape, motion, duration}
 n = Fhigh - Flow
 C = Flow / Fhigh
 R(n) = Flow + Fhigh * (n - 1) / n

[그림2] Range Components 생성

동영상 features를 일반화하는 과정에서 PDG는 결합을 한다. [그림3]은 각 RC에 대해 교집합이 있는 경우와 없는 경우이다.



[그림3] Probability Density Graph 결합

[그림3]에서 i)는 교집합이 없는 경우로 RC가 생성되지 않고 F의 RC플래그는 false로 설정되는 반면 ii),iii)는 교집합이 있는 경우로 F의 RC플래그는 true로 설정된다. 이 경우에는 PDG간 여러 개의 RC가 생길 수 있으며 RC값 계산은 [그림4]와 같다.

$$R(n) = \frac{\sum_{i=1}^n f_i}{N}$$

$$C = \frac{Flow}{Fhigh}$$

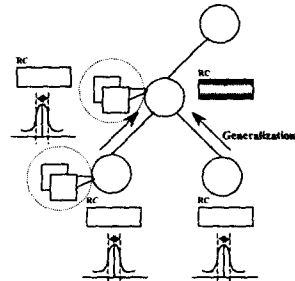
$$M = \frac{\sum_{i=1}^n f_i}{N}$$

$$R(n) = n * C + Flow * (1 - C)$$

[그림4] Range Components 계산

2.3. Feature Generalization 알고리즘

부모 또는 자식 노드로 Video Feature Classification의 클래스 구조를 생성하고 각 클래스에 semantic feature를 할당한다. 리프 노드에 소스예제를 그룹핑하여 추가시킨 다음 FG(Feature Generalization) 알고리즘을 실행하면 [그림5]와 같은 하나의 Video Feature Classification이 생성된다.



[그림5] Feature Generalization

[그림6]은 Feature Generalization 알고리즘이다.

```

1: Video Feature Classification에 추가된 소스 예제를 가장노드로 생성한다.
2: If 가장노드
3: 소스 예제의 각 Feature 명의를 생성한다.
   While F ∈ (Features)
4: F에 대한 소스 예제의 Probability Density Graph 생성하기
5: 임계치 T를 이용한 Range Components 구하기
6: (Features) = (Features) - F
7: Do
8: Do
9: End If
10: 각 노드를 후위 순회(postorder traversal)로 탐색(traverse)하면서 Features를 일반화
    (Generalization)한다.
11: 자식 노드의 각 F에 대한 Range Components를 구한다.
12: While n ∈ (Nodes)
13: While F ∈ (Features)
14: F의 Range Intersection을 구한다. 이때 자식 노드가 n개라고 가정
    한다.
15: If (F.child 1 ∈ F.child 2, ..., F.child n)
16: F.enabled = false
17: else
18: F.enabled = true
19: F에 대한 Probability Density Graph 결합하기.
20: 해당 노드의 Range Components를 구하기.
21: Do
22: Do
23: 루트 노드에서 각 Feature의 가중치를 구한다.
24: n
25: Features를 Fw에 따라 오름차순으로 정렬하기
    Fw = (sum of C) / n
    [0,1]의 값 n-각 Feature에 대한 C(Confidence Measure)의 개수

```

[그림6] Feature Generalization 알고리즘

2.4. Features Weight

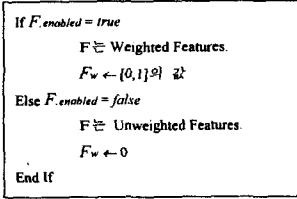
[그림6]의 FG 알고리즘에 의해 각 Feature Weight(Fw)이 계산된다. 루트노드의 C(Confidence Measure)가 Fw 측정 기준이 되며 C값이 높을수록 Fw은 커지고 C값이 낮을수록 Fw은 작아진다. Fw에 따라 동영상 features를 오름차순으로 정렬하면 [그림7]과 같다.

3. 동영상 시공간 검색

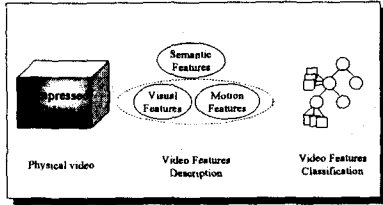
3.1. 동영상 검색구조 및 형태

동영상화일 분할과정에서 자동 추출된 color, texture, shape, motion 등의 low-level features는 [그림8]의 동영상 데이터베이스

이스에 저장된다.



Weighted Features > Unweighted Features
 $Fw1 > Fw2 > \dots > Fwi$
 [그림7] Feature Weight



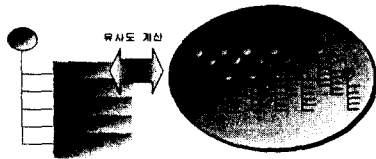
[그림8] 동영상 데이터베이스 구성요소

동영상 데이터베이스는 텍스트뿐만 아니라 features를 사용한 스케치, 동영상 파일등의 다양한 질의로 feature간 거리를 이용한 유사도 측정방식으로 접근된다. 검색은 Scene과 Object 단위로 이루어진다. Scene은 동영상처리과정에서 Scene Change Detection 알고리즘에 의해 자동 분할된 일련의 동영상 프레임 시퀀스이며 Object는 하나의 유사한 움직임 벡터 (motion vector)값을 갖는 영역의 집합이다.

3.2. 동영상 유사도 측정

3.2.1. Object 질의

1-object 질의를 일컫는다. Query는 각 feature에 대해 데이터베이스내 Target object와 각 유사도를 계산한 다음 측정된 유사도 총합으로 최종 검색 결과를 생성한다. [그림9]는 데이터베이스와의 object 질의 형태이다.



[그림9] Object 질의 형태

Q(Query)와 T(Target Object)의 유사도는 Euclidean Distance로 계산하며 이 값을 O_d 라고 정의할때 [식1]처럼 표현된다.

$$O_d = \sum OF_d \quad OF \in \{Features\} // OF = object\ features \quad \dots [식1]$$

Features = {color, texture, shape, motion, time}

O_d 의 크기가 작을수록 Q와 T는 유사하다. Q와 T와의 거리 $D(Q,T)$ 측정은 [식2]와 같다.

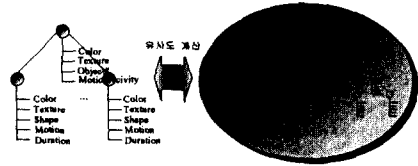
$$D(Q,T) = S_d + \sum_{i=1}^n O_{d_i} \quad S_d > 0, n \geq 1, n \in \text{Template상에 기술된 object수} \quad \dots [식2]$$

그러나, $S_d = 0, n=1$ 이기 때문에 Q와 T와의 거리는 단순히 O_d 로 계산된다.

3.3.2. Scene 질의

n-objects 질의를 일컫는다. 유사도 측정 방법은 3.3.1의 object질의와 비슷하고 Scene내 Object 계산방법이 별도로 계산된다. [그림10]은 데이터베이스와의 Scene 질의 형태이다.

Q(Query)와 T(Target Scene)와의 거리를 S_d 라고 정의할 때, [식3]처럼 표현된다.



[그림10] Scene 질의 형태

$$S_d = SF_d + \sum_{i=1}^n OF_{d_i} \quad SF \in \{Features\} // SF = scene\ features \quad \dots [식3]$$

Features = {color, texture, object#, motionactivity}

n 은 Template상에 기술된 object수

그러나, Q와 T는 object수가 일치하지 않기 때문에 [그림11]과 같은 유사도 측정 알고리즘을 사용한다. Q의 Object수를 n, T의 object수를 m이라 가정한다.

```

    i) n > m
    no matching
    ii) n = m, n < m
    for i=1 to n
        for j=1 to m
            [Tj]d ← D(Qi, Tj)
        Loop
    Old ← min [Tj]d . . . (j=1,2,...m)
    Loop
    
```

[그림11] Scene내 Object 유사도 측정 알고리즘

S_d 의 크기가 작을수록 Q와 T는 유사하다. Q와 T와의 거리 $D(Q,T)$ 측정은 [식2]와 같다.

5. 결론

본 연구는 보다 정확한 동영상 내용기반 검색을 하기 위해 기존의 검색 방법의 단점을 보완한 low-level features와 high-level features 결합방식을 사용하였다. 또한 low-level features에 기반한 Video Feature Classification을 정의하고 FG 알고리즘을 제안함으로써 데이터베이스내 low-level features를 다단계 자동 관리할 수 있다. 자동 low-level features 질의, low-level features 가중치 적용 질의, low-level features와 high-level features간의 자동 변환(translation) 및 브라우징 검색 등이 가능하며 여러 실세계 감시 구역에서 일어나는 모든 동영상 움직임을 분석하여 특정 행위 와 현상에 대한 감시 정보 검색 및 행위 패턴 처리 기법 개발로 응용가능하다.

참고 문헌

- [1] 정미영, 이원석, 효율적인 의미 검색을 위한 동영상 데이터 모델링, 한국정보처리학회논문지, 4권4호, 1997.4.
- [2] 성인용, 이원석, 비디오 데이터의 내용기반 검색을 위한 유동 속성 트리 및 가중치와 히스토리의 이용방법 연구, 석사학위논문, 연세대학교, 1999
- [3] M.Flickner et al., "Query by Image and Video Content : QBIC System", IEEE Computer, Vol.28, No.9, pp23-32, 1995
- [4] J.R.Smith and S.-F.Chang, "VisualSEEK : A Fully Automated Content-Based Image Query System," ACM Multimedia Conference, Boston, MA, Nov. 1996. (<http://www.ctr.columbia.edu/VisualSEEK>)
- [5] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong, "VideoQ- An Automatic Content-Based Video Search System Using Visual Cues," ACM Multimedia Conference, Nov. 1997, Seattle, WA. (<http://www.ctr.columbia.edu/VideoQ>)
- [6] A. Gupta, R. Jain, Visual Information Retrieval , Communications of ACM, Vol. 40, No. 5, 70-79, May 1997.
- [7] S.-F. Chang, "Content-Based Indexing and Retrieval of Visual Information" IEEE Signal Processing Magazine, Vol. 14 No.4, pp.45-48, July 1997.