객체의 시공간적 움직임 정보를 이용한 내용 기반 비디오 검색 알고리즘

(Content-Based Video Retrieval Algorithms using Spatio-Temporal Information about Moving Objects)

정 종 면 † 문 영 식 ††
(Jong Myeon Jeong) (Young Shik Moon)

요 약 본 논문에서는 객체의 움직임 정보를 이용한 효율적인 내용 기반 비디오 검색 알고리즘을 제안하는데, 움직임 속도에 무관(temporal scale-invariant)한 검색과 움직임 속도를 고려(temporal scale-absolute)한 검색을 포함한다.

움직임 속도에 무관한 검색에서는 객체의 이동 흔적(trail)을 거리 변환(distance transform)하여 거리 영상(distance image)을 얻은 다음, 거리 영상을 데이타베이스에 저장한다. 질의로 주어진 흔적의 좌표에 대응하는 데이타베이스의 거리 영상의 밝기는 질의 흔적을 이루는 한 점과 데이타베이스의 흔적 사이의 거리를 의미하기 때문에, 두 흔적 사이의 정합은 질의 흔적을 이루는 점들의 좌표에 대응하는 거리 영상의 평균 밝기를 계산함으로써 수행된다. 또한 움직임 속도를 고려한 검색을 수행하기 위해 움직임 검색 코드(Motion Retrieval Code)를 제안한다. 움직임 검색 코드는 작은 속도 변화 제약을 반영하여 설계되어 인간 시각 특성을 반영할 뿐 아니라, 효율적인 정합을 고려하여 비트 연산이 가능하도록 설계되어 정확하면서도 빠른 비디오 검색을 수행할 수 있다. 제안하는 검색 기법을 이용한 실험에서 제안하는 알고리즘의 타당성을 확인한다.

키워드 : 비디오 검색, 움직임 정보, 흔적, 궤적, 거리 변환, 움직임 검색 코드

Abstract In this paper, efficient algorithms for content-based video retrieval using motion information are proposed, including *temporal scale-invariant* retrieval and *temporal scale-absolute* retrieval.

In *temporal scale-invariant* video retrieval, the distance transformation is performed on each trail image in database. Then, from a given query trail the pixel values along the query trail are added in each distance image to compute the average distance between the trails of query image and database image, since the intensity of each pixel in distance image represents the distance from that pixel to the nearest edge pixel.

For *temporal scale-absolute* retrieval, a new coding scheme referred to as Motion Retrieval Code is proposed. This code is designed to represent object motions in the human visual sense so that the retrieval performance can be improved. The proposed coding scheme can also achieve a fast matching, since the similarity between two motion vectors can be computed by simple bit operations. The efficiencies of the proposed methods are shown by experimental results.

Key words : video retrieval, motion information, trail, trajectory, distance transformation, Motion Retrieval Code

· 이 논문은 2000년 한양대학교 교내연구비 지원으로 연구되었음.
† 정 회 원 : 한국전자통신연구원 방송미디어연구부 선임연구원
jmjcong@etri.re.kr
†† 종신회원 : 한양대학교 컴퓨터공학과 교수
ysmoon@cse.hanyang.ac.kr
논문접수 : 2001년 11월 16일
심사완료 : 2002년 7월 12일

1. 서 론

1.1 연구 배경

멀티미디어 데이타에 대한 수요와 관심이 증가함에 따라 비디오 데이타베이스의 구축 및 검색에 관한 연구가 활발히 진행되고 있다. 그 동안 개발된 내용 기반 영

상 검색 시스템(content-based image/video retrieval system)은 대부분 정지 영상에 대한 것으로써, 영상의 전체적 색상(color) 정보 혹은 분할된 객체(segmented object)의 색상 정보를 비교, 분석하고, 영상의 전체적 에지(edge) 정보 혹은 분할된 객체의 에지 정보를 분석하였으며, 영상의 질감(texture)에 관한 정보를 비교, 분석함으로써 데이타베이스 검색을 수행하였다[1]-[3]. 비디오에 대한 내용 기반 영상 검색은 정지 영상에 대한 검색 시스템에서 사용되었던 여러 가지 특징들과 함께 비디오의 움직임(motion) 정보를 사용하는데, 비디오의 움직임 정보를 추출하기 위해서는 동영상 분석(motion analysis) 관련 기술이 필수적이다[4]-[6]. 한편, 의미 정보(semantic information)를 이용하여 정지 영상과 비디오에 대한 검색을 수행하는 연구가 활발히 진행되고 있다[7]-[8]. 이 방법들은 영상을 구성하는 영역의 기하학적인 관계(topological relationship)나 색상, 움직임의 의미 정보를 이용하여 상위 단계에서의 검색을 수행할 수 있도록 하는데, 비디오가 표현하는 실세계에 대한 의미 관계의 정의가 선행되어야 한다.

그 동안 국내외에서 내용 기반 영상 데이타베이스의 구축에 관련된 연구가 있었으나 대부분은 정지 영상에 관련된 것이었고 비디오 검색에 관련된 연구도 움직임 정보를 검색의 주요 특징으로 사용한 예는 많지 않았다. 비디오 데이타에서 가장 중요하고 핵심적인 정보이면서도 움직임 정보가 많이 사용되지 않은 이유는 움직임 정보는 영상의 다른 특징에 비해 추출 및 정합이 대단히 어렵고, 검색을 위한 정합 척도(matching measure)가 사용자의 일반적인 요구 조건을 만족시키는 것이 힘들기 때문이라고 할 수 있다.

1.2 기존의 연구

QBIC[4], VisualSEEk[9], Virage[10], Informedia [11] 그리고 PhotoBook[12] 등은 1990년대 중반부터 보고되기 시작한 내용 기반 비디오 검색 시스템의 예로서, 내용에 기반한 영상 검색을 수행하였다. 이들 방법에서의 비디오의 묘사는 색상, 형태(shape), 질감과 같은 저수준(low-level)의 특징과 주파수 영역(frequency domain)의 특징 그리고 캡션(caption) 등을 이용하여 이루어 졌는데, 움직임 정보를 검색에 이용하지는 않았다. 이동 객체의 움직임 정보를 비디오 검색에 적용한 기존의 연구 결과물 중 주목할 만한 것은 [13]-[18] 등으로서 이들을 간략히 설명하면 다음과 같다.

Yoshitaka 등[15]은 객체의 움직임에 대한 체인 코드(chain code)를 이용한 비디오 검색 방법을 제안하였는데 검색의 정확도에 문제가 있으며, 샘플링 간격을 고려하기 위한 계산량이 많을 뿐 아니라 시공간(spatio-temporal)에 대한 고려가 부족하여 실제 비디오 검색에 사용하기에는 어려움이 많다. Chang 등[13]은 비디오 데이타베이스의 구성과 검색을 위해 색상, 질감, 형태, 크기(size), 움직임 정보 등을 사용하였고, 이들 특징에 적절하게 가중치를 줌으로써 사용자가 원하는 비디오를 검색하도록 하였다. 그러나 움직임 정보에 관한 사용자의 요구에 적절히 동작하기에는 시간축에 대한 고려가 불충분하고, 객체 움직임에 관한 사용자의 다양한 요구를 반영하기 어렵다. Dagtas 등[14]은 궤적 모델(trajectory model)을 이용하여 움직임 속도를 고려한(temporal scale-absolute) 검색을 수행하였고, 흔적 모델(trail model)을 이용하여 움직임 속도에 무관(temporal scale invariance)한 검색을 수행하였다. 그들의 방법은 영상 검색에서 시공간 정보에 대한 각각의 경우를 모두 고려하였기 때문에 기존의 방법보다 한단계 진보한 비디오 검색 시스템이라고 할 수 있으나, 계산량과 검색의 정확도에 개선할 점이 많다. Lee 등[16]은 움직임 궤적 (motion trajectory)을 이용한 비디오 검색 기법을 제안하였는데, 사용자의 다양한 요구를 반영할 수 있고 고속의 검색을 수행할 수 있으나 검색의 정확성에 문제가 있다. Chen 등[17]은 웨이블릿 변환(wavelet transform)을 이용한 비디오 검색 시스템을 제안하였는데, 주어진 궤적을 부분 궤적(sub-trajectory)으로 나누는 과정에 오류가 있을 경우 잘못된 결과를 출력하기 쉽다. 한편, Aghbari 등[18]은 비디오 샷을 계층적으로 묘사하여 비디오 검색에 이용하는 방법을 제안하였는데, 다중 객체의 검색에 장점이 있으나 검색의 정확도와 계산량에서는 단점이 있다. 그밖에 움직임 정보를 이용하여 비디오를 검색하기 위한 방법으로 AbouGhazaleh 등의 방법[19], Panchanathan 등의 방법[20], Singla 등의 방법[21] 등이 있으나 비디오 검색에 적용하기에는 그 정확도에 문제가 있다.

본 연구와 관련된 국내외 연구 동향을 종합적으로 살펴보면, 기존 내용 기반 비디오 검색 기법들은 키 프레임(key frame)이나 프레임 전체의 색상, 질감을 주요 특징으로 사용하였을 뿐, 비디오 검색에 가장 적합한 객체의 움직임 정보는 이용하지 않은 경우가 대부분이고[4],[9]-[11], 움직임 정보를 이용한 경우에도 검색을 위한 계산량이 많아 효율적인 검색이 어렵거나[14], 검색 결과가 정확하지 않으며[16],[19]-[21] 비디오의 시공간적인 특징에 대한 고려가 부족하여 유연한 검색이 어려웠다[13].

본 논문에서는 객체의 움직임 궤적을 이용한 효율적

인 비디오 검색 방법을 제안한다. 제안하는 방법은 비디오 데이타베이스 시스템에 대한 사용자의 요구를 적절히 반영하기 위해, 객체의 움직임 속도에 무관(temporal scale invariance)한 검색과 움직임 속도를 고려한 검색(temporal scale-absolute)으로 나누어 처리한다. 본 논문의 이후 구성은 다음과 같다. 먼저 1.3절에서 비디오 데이타베이스 검색 시스템이 갖춰야 할 특징에 관하여 설명한 다음 2 장과 3 장에서 객체의 움직임 속도에 무관한 비디오 검색 기법과 움직임 속도를 고려한 비디오 검색 기법을 제안하고 4 장에서 제안하는 방법들의 성능을 평가한 후 5 장에서 결론을 맺는다.

1.3 비디오 데이타베이스 검색 시스템의 요구 특성

비디오 데이타의 검색은 그 데이타의 방대함으로 인해서 원시 데이타(raw data)에 대한 직접적인 처리가 사실상 불가능하기 때문에 비디오 데이타베이스 시스템은 그 구성 단계에서부터 데이타베이스 저장 시스템과 질의, 검색 시스템으로 나누어 설계하는 것이 바람직하며 사용자 질의, 데이타베이스 검색, 결과 출력 과정에서 서로 연동할 수 있어야 한다. 또한 비디오 데이타는 동일한 움직임, 색상, 질감, 형태를 갖는 비디오에 대해서도 보는 사람의 주관과 감정에 따라 그 내용을 다르게 해석하기 때문에 기존의 문자 데이타와는 달리 애매성(*ambiguity*)을 갖는다. 사용자의 질의 방식도 전통적인 문자 데이타베이스와 비디오 데이타베이스는 서로 다르다. 즉, 문자 데이타베이스에 비해서 비디오 검색을 위한 사용자의 질의는 대단히 추상적이며 그 의미가 애매한 경우가 많고, 따라서 질의에 대한 검색 결과가 사용자가 기대했던 결과와 전혀 다른 내용일 수 있다. 따라서 비디오 데이타베이스 시스템을 구축할 때는 비디오 데이타의 대용량성과 애매성, 그리고 사용자 질의의 추상성을 고려해야 한다. 사용자의 요구사항에 따라 적절하게 응답할 수 있는 비디오 데이타베이스 시스템이 갖추어야 할 특징을 요약하면 다음과 같다.

• 내용 기반 검색(content-based retrieval) : 비디오 데이타베이스 시스템에서 비디오 데이타의 대용량성과 애매성, 사용자 질의의 추상성을 극복하기 위하여 내용 기반 검색이 필수적이다는 사실은 이미 일반적이다[4], [9]-[11],[13].

• 퍼지성(*fuzziness*) : 비디오 검색에서는 사용자의 추상적인 요구에 적절하게 동작하기 위하여 문자 데이타베이스처럼 질의와 정확하게 일치하는 것만 검색해서는 안된다. 즉, 비디오 데이타베이스 시스템은 퍼지성을 갖추어야 하는데, 이는 질의와 검색 과정에서 애매성에 대한 고려를 포함하여 데이타베이스의 비디오가 질의와 정확하게 일치하지 않더라도 경우에 따라서는 검색 결과에 포함시켜야 함을 의미한다.

• 효율성(efficiency) : 비디오 데이타베이스는 그 데이타가 대단히 방대하므로 검색을 위한 검색 알고리즘과 정합 척도는 계산 부담이 적은 것을 사용하는 것이 중요하다. 즉, 검색을 위한 정합이 효율적이어야 한다.

• 유연성(flexibility) : 비디오 검색을 위한 사용자의 질의는 문자 데이타베이스 검색을 위한 질의에 비해 대단히 추상적이기 때문에 사용자의 요구를 가능한 정확히 전달하기 위하여, 비디오 검색을 위한 질의와 더불어 사용자가 몇 가지 선택적인 검색 요구사항을 부가하여 결과를 요구할 수 있도록 하는 것이 바람직하다. 즉, 데이타베이스 검색 시스템은 사용자의 다양한 요구에 유연하게 동작하는 것이 바람직하다.

• 정확성(accuracy) : 사용자의 질의로부터 사용자가 필요로 하는 비디오 데이타를 정확하게 출력해야 한다. 즉, 사용자의 직관과 가장 유사한 비디오를 찾을 수 있어야 한다.

2. 움직임 속도에 무관한 비디오 검색(Temporal scale-invariant video retrieval)

2.1 필요성

움직임 속도에 무관한 검색이란 객체 움직임의 속도는 고려하지 않고 이동 경로만을 고려 대상으로 삼아 비디오 검색을 수행하는 것을 의미한다. 본 논문에서는 움직임 속도에 무관한 검색을 수행하기 위하여 객체의 이동 흔적(trail)을 이용한다. 객체의 이동 흔적이란 객체가 이동해 나가는 경로를 하나의 평면에 점(point)이나 선(line), 또는 사각형(bounding box)으로 나타낸 것으로서 움직임 속도에 관한 정보는 자연히 사라지게 된다.

그림 1은 움직임 속도에 무관한 검색의 의미를 설명하고 있는데, 그림 1에서 화살표는 객체의 이동을 나타내고 화살표의 길이는 공간상의 거리를 의미하며 검정색의 점(dot)은 시간 t 에 따라 이동해 나가는 객체를 나타낸다. 그림 1(a)에서 점의 움직임은 그림 1(b)에서의

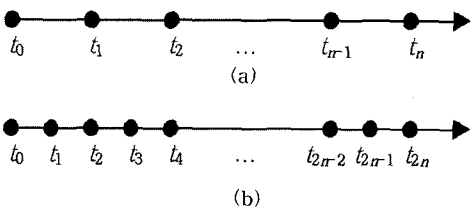


그림 1 움직임 속도에 무관한 검색

그것보다 2배 빠른 움직임 속도를 보이고 있기 때문에 두 움직임은 서로 다르다고 할 수 있는데, 움직임 속도에 무관한 검색에서는 그림 1(a)의 움직임과 그림 1(b)의 움직임을 서로 같은 것으로 간주한다. 즉, 움직임 속도에 무관한 검색에서는 객체의 움직임의 속도는 고려하지 않고, 이동 경로만을 고려 대상으로 삼아 비디오 검색을 수행한다.

흔적을 이용한 움직임 속도에 무관한 검색이 유용하게 사용되는 예로는, 스포츠 비디오에서 선수가 골인을 넣는 장면을 필요로 할 때 원래 속도에 의한 비디오(normal motion)와 느린 비디오(slow motion) 모두 찾아서 응답할 수 있으며, 특히 교통 감시 시스템(traffic surveillance system)에서 불법 횡단, 차선 위반 등을 찾는데 유용하다.

2.2 공간 위치를 고려한 검색(Temporal scale-invariant and spatial translation-absolute retrieval)

이동 흔적을 이용한 공간 위치를 고려한 비디오 검색은 그림 2에서 보이는 바와 같이 질의와 동일한 공간적 위치에 있는 움직임을 찾는 것을 목적으로 하는데, 객체 움직임의 속도에는 관계없이 사용자가 원하는 위치에, 원하는 형태의 이동 경로를 갖는 비디오 샷을 찾기 위하여 사용된다. 이를 위하여 본 논문에서는 객체 추적 결과 얻어진 객체의 이동 경로를 하나의 평면에 계속 그려나감으로써 흔적 영상을 얻고, 이 흔적을 다항식 근사화(polynomial fitting)하여 객체 추적 과정에 존재할 수 있는 잡음과 오류의 영향을 줄인 다음 거리 변환을 수행하여 거리 영상(Distance Image)을 얻어 저장한다.

2.2.1 다항식 근사화(Polynomial fitting)

주어진 비디오에 대해 객체 추출과 추적 기법을 적용하면 객체의 이동 경로를 표현하는 바운딩 박스의 시퀀스를 얻을 수 있다. 그러나 대부분의 경우 객체의 이동 경로는 잡음과 객체 추출, 추적 알고리즘의 문제로 인하여 잡음과 오류를 포함하고 있다. 잡음과 오류의 영향을 줄이기 위하여 식 1, 식 2와 같이 바운딩 박스의 중심 좌표로 표현된 객체의 이동 흔적을 x, y 좌표로 나누어, 각각에 대하여 다항식 근사화를 수행한다.

$$y(t) = b_0 + b_1 t + b_2 t^2 + \cdots + b_{n-1} t^{n-1} + b_n t^n \quad (1)$$

$$x(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_{n-1} t^{n-1} + a_n t^n \quad (2)$$

여기에서 $x(t)$, $y(t)$는 t 번째 프레임에 있는 바운딩 박스의 중심 좌표를 이루는 점의 x, y 좌표를 각각 의미하며, n 은 다항식 근사화의 차수이고, a_i와 b_i는 다항식의 계수인데 n 차 다항식일 경우 $n+1$ 개의 계수가 각각 존재한다.

k 개의 좌표가 주어진 흔적에서 x 좌표를 식 1과 같이 n 차 다항식 근사화 하면, x 좌표에 대한 $n+1$ 개의 다항식 계수는 k 개의 중심 좌표의 x 좌표로부터의 거리 제곱의 합(sum of the squares of the deviations)을 최소화하는 근사화된 x 좌표의 흔적을 표현할 수 있다. 한편 y 좌표에 대해서도 x 좌표에서와 동일한 과정의 다항식 근사화를 수행한다. 즉, 바운딩 박스의 중심 좌표로 표현된 객체의 이동 흔적에 대하여 x, y 좌표로 나누어 각각에 대하여 다항식 근사화를 수행하여 잡음과 오류의 영향을 제거한 다음, 다시 x, y 좌표를 하나의 좌표로 구성하여 원래의 x, y 좌표에 근사하는 근사화된 흔적을 얻은 다음 거리 변환을 수행하여 거리 영상을 얻는다.

2.2.2 거리 변환(Distance transform)

Borgerfors가 제안한 거리 변환은 영상의 이진화된 에지 영상을 입력으로 받아 영상의 각 픽셀에서 가장 가까운 경계선까지의 거리를 그 픽셀의 값으로 바꾸어 출력하는 변환을 말한다[22]. 거리 변환은 유클리디안 거리(Euclidean distance)를 정수로 근사화한 Chamfer

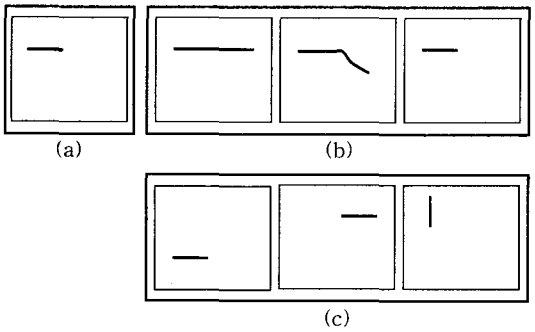


그림 2 움직임 속도에 무관하고 공간 위치를 고려한 비디오 검색의 예 : (a) 질의 흔적, (b) 비슷한 흔적, (c) 다른 흔적

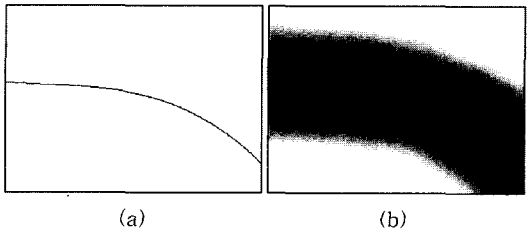


그림 3 거리 변환의 예 : (a) 이진화된 에지 영상, (b) (a)에 대한 거리 영상

3/4 거리를 이용하여 경계선으로부터 모든 픽셀의 거리를 단 두 번의 처리를 통해 얻을 수 있으므로 매우 효율적이다. 거리 변환을 거친 원래의 에지 영상은, 에지 픽셀은 0값을 갖고, 나머지 픽셀은 자신으로부터 가장 가까운 에지까지의 Chamfer 3/4 거리를 갖게 된다. 따라서 그림 3(b)에서 각 픽셀의 밝기 값(intensity)은 그림 3(a)에서 가장 가까운 에지까지의 거리를 의미하며 그림 3(b)에서 어둡게 나타나는 부분은 에지로부터 거리가 가깝다는 것을 의미하고 밝게 나타나는 부분은 에지로부터 거리가 멀다는 것을 의미한다.

2.2.3 정합 과정

주어진 원시 비디오 데이타(raw video data)에 대한 객체 추출, 추적 결과 얻은 궤적의 중심 좌표를 다항식 근사화 한 후 혼적 영상을 생성하고, 혼적 영상을 거리 변환하여 얻은 거리 영상을 데이타베이스에 저장하여 검색에 이용한다. 스케치 혹은 예제를 통해 입력받은 질의를 Q라고 했을 때 Q는 식 3과 같이 표현될 수 있다.

$$Q=[q_x(i),q_y(i)]^t,\quad i=1,2,\cdots,n \tag{3}$$

여기에서 $q_x(i)$와 $q_y(i)$는 질의를 이루는 i 번째 점의 x, y 좌표를 각각 의미한다. 본 논문에서는 Q를 다항식 근사화하여 오류와 잡음의 영향을 줄이는데, 다항식 근사화 과정을 거친 질의 혼적을 Q^*라고 하면 Q^*는 식 4와 같이 표현된다.

$$Q^*=[q_x^*(i),q_y^*(i)]^t,\quad i=1,2,\cdots,n \tag{4}$$

한편, 다항식 근사화를 거친 데이타베이스의 j 번째 혼적 D^{j*}에 대한 혼적 영상을 거리 변환하여 얻은 거리 영상을 $I^{D^{j*}}$라고 하면, 질의 Q^*을 이루는 한 점의 좌표 $(q_x^*(i),q_y^*(i))$에 대응하는 D^{j*}의 거리 영상 $I^{D^{j*}}(q_x^*(i),q_y^*(i))$의 밝기 값은 $(q_x^*(i),q_y^*(i))$로 부터 가장 가까운 D^{j*} 혼적까지의 최단 거리를 의미한다. 따라서 질의 혼적 Q^*와 데이타베이스의 j 번째 혼적 D^{j*} 사이의 거리는 식 5와 같이 계산될 수 있다.

$$Dist(Q^*,D^{j*})=\frac{\sum_{i=1}^{n} I^{D^{j*}}(q_x^*(i),q_y^*(i))}{n} \tag{5}$$

여기에서 $Dist(Q^*, D^{j*})$은 질의 혼적 Q^*와 데이타베이스의 j 번째 혼적 D^{j*} 사이의 거리를 의미하고, $(q_x^*(i),q_y^*(i))$는 질의 혼적 Q^*를 이루는 바운딩 박스의 중심 좌표들을 의미하며, n은 질의 혼적 Q^*를 이루는 바운딩 박스의 중심 좌표의 개수를 의미한다. 그림 4는 제안하는 방법을 블록도로 나타낸 것이다.

제안하는 방법은 데이타베이스에 저장되어 있는 거리 영상에서 질의 궤적이 지나가는 각 픽셀들의 밝기 값을

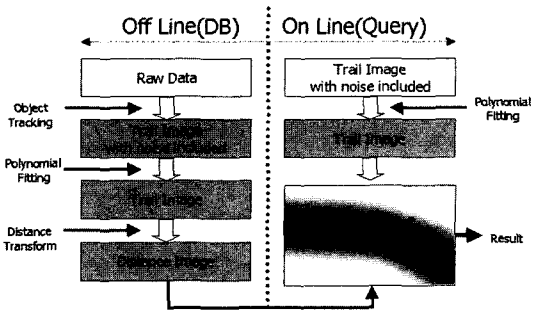


그림 4 혼적을 이용한 공간 위치를 고려한 비디오 검색 기법의 블록도

더한 다음 한번의 나눗셈만 하면 이루어진다. 그러므로 제안하는 방법을 이용한 검색을 위해서는 질의 혼적이 n 개의 점으로 이루어졌을 경우 $O(n)$의 계산량이 필요하다.

2.3 공간 위치에 무관한 검색(Temporal scale-invariant and spatial translation-invariant retrieval)

공간 위치에 무관한 비디오 검색이란 그림 5와 같이 질의로 주어진 움직임의 공간적인 위치에 관계없이 비슷한 이동 형태를 갖는 움직임을 찾는 것을 말한다. 공간 위치에 무관한 비디오 검색을 수행하기 위해서는 데이타베이스의 혼적에 대하여 질의 혼적을 상하 좌우로 이동(translation)시켜 가면서 혼적의 이동 가능성을 모두 고려해야 하므로 많은 계산이 필요하다. 본 논문에서는 공간 위치에 무관한 비디오 검색을 수행하기 위하여 2.2절에서 설명한 공간 위치를 고려한 검색 기법을 이용하는데, 계산량을 줄이기 위하여 혼적의 모든 이동 가능성은 고려하지 않고, 질의 혼적의 첫번째 점의 좌표를 데이타베이스에 있는 혼적의 좌표에 보상(compensation)

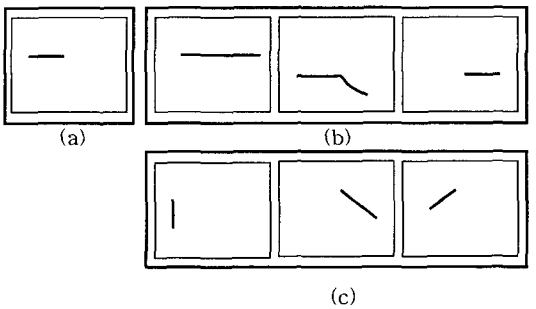


그림 5 움직임 속도에 무관하고 공간 위치에 무관한 비디오 검색의 예 : (a) 질의 혼적, (b) 비슷한 혼적, (c) 다른 혼적

해 줌으로써 공간 위치에 무관한 비디오 검색을 수행한다.

알고리즘의 계산량은 질의와 데이타베이스의 흔적을 이루는 점들의 수가 각각 m, n 개 일 경우 $O(mn)$으로 주어진다.

3. 움직임 속도를 고려한 비디오 검색(Temporal scale-absolute video retrieval)

3.1 개요

본 논문에서는 움직임 속도를 고려한 검색을 수행하기 위해 객체의 이동 궤적(motion trajectory)을 사용한다. 객체의 궤적이란 그림 6에서 보이는 바와 같이 비디오를 이루는 각 프레임 상에서 객체가 이동해 나가는 경로를 의미하는데, 이동 객체의 시공간적인 정보를 모두 나타낸다. 따라서 궤적을 이용한 비디오 검색은 사용자가 객체의 움직임 속도를 고려한 검색 결과를 얻고자 할 때 유용하게 사용될 수 있다.

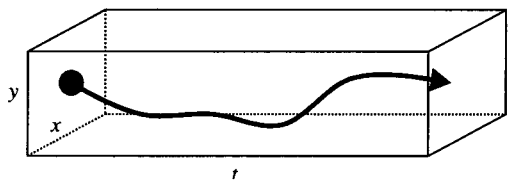


그림 6 객체 이동 궤적의 예

흔적을 이용한 검색에서와 마찬가지로 궤적을 이용한 검색도 공간 위치를 고려한 검색과 공간 위치에 무관한 검색으로 나눌 수 있다. 궤적을 이용한 공간 위치를 고려한 검색 기법은 객체의 움직임 속도와 움직임 방향, 궤적의 모양 그리고 움직임의 공간적 위치를 모두 고려하여 검색을 수행하는 것을 말하는데, 이를 위하여 Dagtas 등의 방법[14]이 유용하게 사용될 수 있다.

본 논문에서는 객체의 움직임 궤적을 이용한 공간 위치에 무관한 비디오 검색 기법을 제안한다. 제안하는 방법은 움직임 궤적을 비디오 검색에 적합하도록 표현하는 기법 즉, 움직임 궤적의 새로운 표현 방법과, 제안하는 방법으로 표현된 궤적의 정합 기법을 포함한다. 제안하는 움직임 표현 기법은 객체의 움직임 정보를 움직임의 크기와 방향에 따라 표현함으로써 비디오 검색에 효과적으로 사용될 수 있는데, 이와 관련된 기존의 연구 결과 중 주목할 만한 연구 결과는 객체의 공간 좌표를 직접 검색에 이용하거나[13],[14], 체인 코드에 기반한 방법[20],[23], 다항식 계수를 이용한 방법[16], 웨이블릿 변환을 이용한 방법[17] 등이 있다.

3.2 공간 위치에 무관한 검색(Temporal scale-absolute and spatial translation-invariant retrieval)

움직임 속도를 고려하고 공간 위치에 무관한 검색은 객체의 움직임 속도, 움직임 방향 그리고 궤적의 모양을 이용하여 검색을 수행하는 것을 말하는데, 사용자가 원하는 움직임이 공간적으로 어디에 존재하든 모두 찾아내는 것을 목적으로 한다. 그림 7은 움직임 속도를 고려하고 공간 위치에 무관한 비디오 검색의 예를 보이고 있다. 그림 7에서 객체의 중심 좌표를 의미하는 검정색 원은 각 프레임에 존재하는 객체의 중심 좌표가 시간에 따라 화살표 방향으로 이동하고 있음을 나타내는데, 화살표의 길이는 공간상의 거리를 의미한다. 각 프레임의 시간 간격은 일정하기 때문에 원의 간격이 좁다는 것은 움직임 속도가 느리다는 것을 의미한다. 궤적을 이용한 공간 위치를 고려한 비디오 검색에서는 그림 7에서 보이는 바와 같이 객체의 움직임 속도가 질의 궤적과 비슷해야 하고, 움직임 방향도 비슷해야 하지만, 움직임이 발생하는 공간 위치는 고려하지 않고 검색을 수행한다.

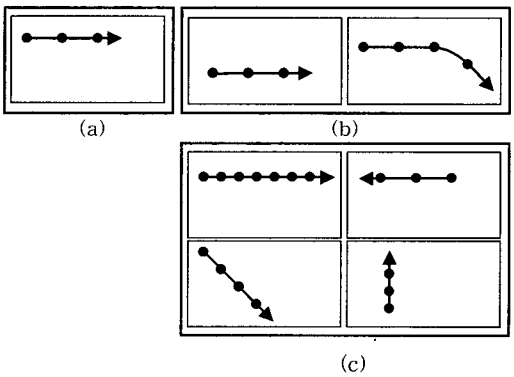


그림 7 움직임 속도를 고려하고 공간 위치에 무관한 비디오 검색의 예 : (a) 질의 궤적, (b) 비슷한 궤적, (c) 다른 궤적

움직임 속도를 고려하고 공간 위치에 무관한 비디오 검색을 위하여 Chang 등[13]과 Dagtas 등[14]은 두 궤적사이의 거리를 계산하기 위한 정합의 척도로써 두 점 사이의 유클리디언 거리를 사용하였고, Panchanathan 등[20]과 Dimitrova 등[23]은 두 체인 코드 사이의 유클리디언 거리를 사용하였다. 그러나 유클리디언 거리는 움직임 특성을 반영한 척도가 아니기 때문에 인간의 시각 특성과는 다른 결과를 출력할 수 있다. 그림 8은 유클리디언 거리를 움직임 검색을 위한 정합 척도로 사용

(a) (b)

그림 8 유클리디언 거리의 문제점

했을 때의 문제점을 보이고 있는데, 인간의 시각 특성에 의하면 그림 8(a)에 있는 두개의 움직임 벡터는 그림 8(b)의 움직임 벡터들보다 더 유사한 움직임을 보이고 있지만 유클리디언 거리를 정합의 척도로 사용했을 경우에는 그림 8(b)의 벡터를 더 유사한 것으로 판정한다. 이런 문제를 해결하기 위하여 본 논문에서는 인간 시각 특성을 반영한 움직임 검색 코드(Motion Retrieval Code)와 이를 이용한 비디오 검색 기법을 제안한다.

3.2.1 움직임 맵(Motion map)

김소연 등이 제안한 움직임 맵[24]은 그림 9와 같이 8 방향 체인 코드에 의한 객체의 움직임 방향과 함께 객체의 움직임 속도를 4단계로 나누어 표현함으로써, 움직임의 방향과 속도를 고려한 총 32개의 숫자를 이용하여 객체의 움직임을 표현한다.

그러나 움직임 맵을 검색에 사용하기에는 32개의 숫자로 표현된 움직임 맵의 서로 인접해 있는 숫자가 서로 비슷한 움직임을 표현하지 못하기 때문에, 움직임 맵으로 표현된 객체의 움직임의 차이를 표현할 수 있는 적절한 정합 척도를 정의하기 힘들다. 즉, 그림 9의 움직임 맵에서 움직임 "3"과 "4"는 서로 인접해 있는 숫자이지만 전혀 다른 움직임을 표현하고 있으며 "31"로 표현된 움직임은 "3"과 서로 비슷한 움직임이지만 그 숫자 차이는 매우 크다. 따라서 본 논문에서는 움직임 맵을 비디오의 인덱싱과 검색에 적합하도록 표현한 움직임 검색 코드를 제안한다.

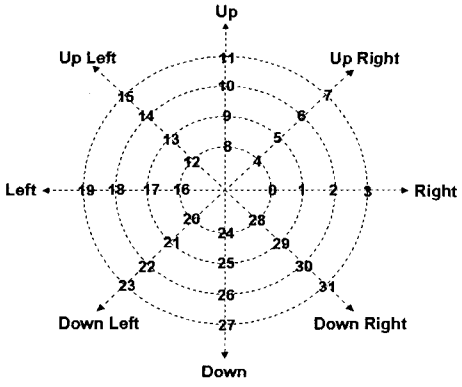


그림 9 움직임 맵

3.2.2 움직임 검색 코드(Motion Retrieval Code)

움직임 검색 코드는 움직임 맵에서 8단계의 움직임 방향 정보를 표현하기 위하여 4개의 비트를 사용하고 4단계의 움직임 크기를 표현하기 위하여 3개의 비트를 사용함으로써, 총 7개의 비트로 32개의 움직임을 표현한다. 이때 서로 비슷한 움직임은 서로 비슷한 비트 구조를 가지도록 하고 서로 다른 움직임은 비트 구조의 차이가 많도록 함으로써, 두 움직임을 비교할 때에는 서로 차이가 나는 비트의 수만 계산(count)함으로써 정합이 이루어지도록 한다. 표 1은 제안하는 움직임 검색 코드의 움직임 방향에 대한 비트 할당을 나타내고 있는데, ID 는 8방향 체인 코드의 움직임 인덱스를 나타내며, 이에 대한 움직임 검색 코드는 서로 인접한 움직임 방향에 대해서 항상 1비트씩 다르도록 하였다.

표 1 움직임 방향에 대한 비트 할당

움직임 방향 ID	움직임 검색 코드
0	0000
1	0001
2	0011
3	0111
4	1111
5	1110
6	1100
7	1000

한편 표 2는 움직임 검색 코드의 움직임 크기에 대한 비트 할당을 나타내고 있는데, ID는 4단계의 움직임 크기의 인덱스를 나타내고, 이에 대한 움직임 검색 코드는 움직임 방향과 마찬가지로 서로 인접한 움직임 크기에 대해서 항상 1비트씩 다르도록 하였다.

표 2 움직임 크기에 대한 비트 할당

움직임 크기 ID	움직임 검색 코드
0	000
1	001
2	011
3	111

움직임 방향과 움직임 크기에 대한 비트 할당을 조합하면 표 3과 같은 움직임 검색 코드가 된다.

본 논문에서 제안하는 움직임 검색 코드는 32개의 움직임에 대하여 움직임의 방향이나 크기가 서로 인접하여 서로 비슷한 움직임을 나타내고 있는 경우에는 항상

표 3 움직임 검색 코드를 위한 비트 할당

움직임 ID	방향	크기	움직임 ID	방향	크기
0	0000	000	16	1111	000
1	0000	001	17	1111	001
2	0000	011	18	1111	011
3	0000	111	19	1111	111
4	0001	000	20	1110	000
5	0001	001	21	1110	001
6	0001	011	22	1110	011
7	0001	111	23	1110	111
8	0011	000	24	1100	000
9	0011	001	25	1100	001
10	0011	011	26	1100	011
11	0011	111	27	1100	111
12	0111	000	28	1000	000
13	0111	001	29	1000	001
14	0111	011	30	1000	011
15	0111	111	31	1000	111

1 비트만 다르고, 서로 반대가 되는 움직임에 대해서는 7 비트 모두 차이가 나도록 설계되었다. 즉 움직임 맵에서 "3"으로 표현된 움직임의 경우 움직임 검색 코드에서는 "0000111"이고 "3"과 비슷한 움직임을 갖는 "2", "7", "31"에 대한 움직임 검색 코드는 각각 "0000011", "0001111", "1000111"로 주어져서 모두 움직임 "3"과 1 비트만 차이가 나도록 되어있다. 또, 움직임 "3"과 정반대의 움직임인 "16"은 "1111000"으로 주어져서 7비트 차이가 남을 볼 수 있다. 움직임 검색 코드는 객체 움직임의 모든 표현이 비트 연산을 고려하여 설계되었기 때문에 실제 비디오 검색을 위한 정합을 수행할 때 비트 연산에 의한 검색이 가능하고, 따라서 기존의 픽셀의 유클리디언 거리를 직접 계산하는 방법[13], [19], 퓨리에 변환을 이용한 방법[14]에 비해 계산량을 크게 줄일 수 있다는 장점이 있다.

한편, 표 3의 움직임 검색 코드는 32개의 움직임만을 표현하기 위한 것이며, 필요에 따라서 움직임의 종류를 줄이거나 늘릴 수도 있다. 만약 움직임의 방향에 대한 더 자세한 정보가 필요하다면 움직임을 표현하기 위한 비트의 수를 더 늘리면 된다. 예를 들어, 표현하고자 하는 움직임의 방향을 8개에서 10개로 늘리고 싶다면 표 4와 같이 비트를 할당하면 된다. 즉, 1개의 비트를 추가로 할당함으로써, 2개의 움직임 방향에 대한 정보를 더 표현할 수 있다. 또 움직임 크기를 5단계로 표현하고 싶다면 표 5 처럼 1개의 비트를 더 할당하면 된다. 즉, 1개의 비트를 더 할당함으로써, 1개의 움직임 크기에 대한 정보를 추가로 표현할 수 있게 된다.

표 4 10 방향에 대한 비트 할당

움직임 방향 ID	움직임 검색 코드
0	00000
1	00001
2	00011
3	00111
4	01111
5	11111
6	11110
7	11100
8	11000
9	10000

표 5 5개 움직임 크기에 대한 비트 할당

움직임 크기 ID	움직임 검색 코드
0	0000
1	0001
2	0011
3	0111
4	1111

3.2.3 움직임 검색 코드의 생성

주어진 영상열에 대한 객체 추적 결과 얻을 수 있는 이동 객체의 중심 좌표로부터 객체의 움직임 방향(direction : D)과 움직임 크기(magnitude : M)를 다음과 같이 계산할 수 있다.

$$D=\tan^{-1}\frac{dy}{dx}$$
$$M=\sqrt{dx^2+dy^2} \quad (6)$$

여기에서 dx, dy는 서로 인접해 있는 두 점 사이의 x 방향, y 방향 차이를 의미한다.

식 6에 의해 계산된 객체 움직임의 방향과 크기는 양자화 과정을 거쳐 32개의 움직임 맵으로 표현된다. 이를 위하여 김소연 등은[24] 객체 움직임 방향을 8단계로, 움직임 크기를 4단계로 각각 균일하게 양자화 하였다. 본 논문에서는 움직임 방향에 대한 양자화는 김소연 등의 방법을 이용한다. 그러나 객체의 움직임 크기, 즉 속도는 비디오의 종류에 따라 차이는 있으나 대부분의 경우에는 그 움직임이 $32\sqrt{2}$처럼 크게 나타나는 경우가 많지 않다. 즉, MPEG-1, 2는 최대 탐색 영역을 32 픽셀로 하고 있으나 실제로 대부분의 움직임 벡터는 $32\sqrt{2}$보다 훨씬 적은 움직임 크기를 갖는다. 그림 10은 MPEG-7의 테스트 시퀀스로 공인된 비디오 중 객체의 움직임 궤적에 의한 비디오 검색에 적합하여 본 논문에서 실험에 사용하고 있는 비디오들의 움직임 크기의 분포를 히스토그램으로 나타낸 것으로서 움직임 크기의

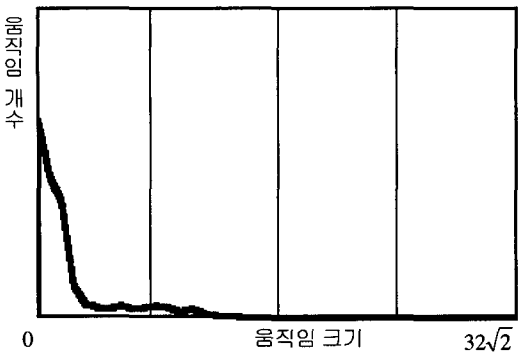


그림 10 MPEG 비디오의 움직임 크기 분포 히스토그램

대부분은 작은 움직임을 보이고 있는 것을 알 수 있다. 따라서, 정해진 비트 수로 움직임 크기를 효과적으로 표현하기 위해서는 MPEG 비디오의 움직임 특징에 따라 움직임 크기를 양자화 하는 것이 바람직하다. 본 논문에서는 데이타베이스로 주어진 비디오 영상들에 존재하는 모든 객체의 움직임 크기를 누적 히스토그램으로 나타낸 다음, 누적 히스토그램의 빈(bin)의 값이 전체 움직임 벡터 개수의 1/4에 해당 할 때를 기준으로 객체의 움직임 크기를 양자화 함으로써 움직임 크기에 대한 표현을 효과적으로 한다.

즉, 본 논문에서는 움직임 검색 코드를 생성하기 위하여 인접한 두 프레임의 이동 객체 중심 좌표들로부터 움직임 방향과 움직임 크기를 각각 얻은 다음, 움직임 방향은 균등하게 8단계로 양자화하고, 움직임 크기는 데이타베이스의 움직임 특성을 고려하여 4단계로 양자화 함으로써 주어진 이동 객체의 움직임을 32개로 양자화 한다. 그런 다음 표 3을 이용하여 움직임 검색 코드를 생성한다. 서로 인접해 있는 프레임의 이동 객체에 대해 움직임 검색 코드를 반복적으로 구하면 비디오 샷에 대한 움직임 검색 코드의 시퀀스를 얻을 수 있는데, 실제 비디오에 대한 검색은 움직임 궤적에 대한 움직임 검색 코드의 시퀀스를 이용하여 수행된다.

3.2.4 움직임 검색 코드의 타당성

컴퓨터 비전 분야에서 작은 속도 변화 제약(small velocity change constraint)은 동영상 분석을 위하여 널리 사용되어 왔다. 작은 속도 변화 제약이란 서로 인접한 프레임에서 객체나 픽셀의 움직임의 속도 차이는 없거나 거의 없다는 점을 이용하여 광류(optical flow)의 계산이나 대응점 탐색(correspondence search)을 수행할 때 서로 속도의 차이가 작은 점을 대응점으로 결정하는 제약을 말한다[25]. 식 7은 대응점 탐색에서 많이 쓰이는 정합의 척도로서 작은 속도 변화 제약을 반영하고 있다[26]-[27].

$$E = w_1 \frac{dS}{dt} + w_2 \frac{dD}{dt} \tag{7}$$

여기에서 w_1, w_2는 움직임 속도와 움직임 방향 성분에 대한 가중치를 의미하며, dS/dt, dD/dt는 움직임 속도와 움직임 방향의 차이를 각각 의미하는데, 작은 속도 변화 제약에 의하여 현재 프레임의 한 픽셀 또는 객체와, 움직임 속도의 차이와 움직임 방향의 차이의 합이 가장 적은 다음 프레임의 픽셀 또는 객체를 서로 대응되는 것으로 결정함으로써 광류를 계산하거나 객체 추적을 수행한다. 이 개념은 현재 프레임의 픽셀 또는 객체에 대한 움직임 벡터가 다음 프레임에서도 똑같이 나타날 것이라고 추정한 다음, 다음 프레임에서 움직임 벡터의 차이가 최소가 되는 픽셀 또는 객체를 대응점으로 결정하는 것으로써, 비디오 검색에서 질의로 주어진 움직임 벡터가 데이타베이스에 있을 것이라고 추정한 후, 데이타베이스에서 움직임 벡터의 차이가 적은 것을 찾는 개념과 같다. 따라서 식 7을 움직임 검색을 위한 정합의 식으로 쓰면 식 8과 같이 쓸 수 있다.

$$Diff(Q,D) = w_1 Diff(S_q, S_d) + w_2 Diff(D_q, D_d) \tag{8}$$

여기에서 $Diff(Q, D)$는 질의로 주어진 움직임 벡터와 데이타베이스의 움직임 벡터 사이의 거리를 의미하고 $Diff(Sq, Sd)$는 질의로 주어진 움직임 벡터와 데이타베이스의 움직임 벡터에서 움직임 크기의 차이를, $Diff(Dq, Dd)$는 질의로 주어진 움직임 벡터와 데이타베이스의 움직임 벡터에서 움직임 방향의 차이를 각각 의미한다. w_1과 w_2는 가중치이다. 식 8은 그 동안 동영상의 움직임 분석에서 널리 사용되었던 작은 속도 변화 제약을 비디오 검색에서 적용할 수 있도록 하는 이론적 근거를 제시하는데, 움직임 검색 코드는 식 8의 개념을 정확히 반영하므로 비디오 검색에 적합한 정합의 척도라고 할 수 있다.

한편 표 3의 움직임 검색 코드는 3/7의 가중치를 갖는 객체의 움직임 크기와 4/7의 가중치를 갖는 움직임 방향의 두 성분으로 이루어져 있는데, 만약 움직임 크기와 움직임 방향에 대한 가중치를 다르게 조정해야할 경우에는 표 4, 5와 같은 방법으로 각 항에 대한 비트의 수를 조정하면 된다.

개념적으로 움직임 검색 코드는 변형된 체인 코드(modified chain code)라고 볼 수 있다. 즉, 움직임 방향만 표현했던 기존의 체인 코드를 움직임 크기도 동시에 표현할 수 있도록 확장하였으며, 효율적인 정합을 위해 움직임 검색 코드 자체에 검색을 위한 정합 척도를

포함하도록 하여 비트 연산에 의한 정합이 가능하도록 하였다.

3.2.5 Look-up table을 이용한 속도 향상

움직임 벡터를 양자화 하여 표현할 경우 두개의 움직임 벡터에 대해 가능한 모든 경우의 차이(dissimilarity)를 미리 계산할 수 있다. 따라서 움직임 검색 코드에 의한 비트 연산보다 더욱 고속의 연산을 수행할 수 있는 방법으로 look-up table을 이용한 정합을 수행할 수 있다. 본 논문에서는 보다 고속의 정합을 위해 look-up table을 이용한 정합을 수행하였는데, look-up table을 이용한 정합은 비디오 데이타베이스의 대용량성을 고려했을 때 무시할 수 있을 정도의 추가 기억 공간 부담으로 계산 속도를 향상시킬 수 있는 장점이 있다.

4. 실험 및 성능 평가

본 논문에서는 제안하는 알고리즘을 Dagtas 등의 방법[14], Lee 등의 방법[16] 등 기존의 방법들과 비교, 실험하였다. 이를 위하여 Che 시퀀스, Yard 시퀀스, 그리고 Walkmen 시퀀스 등 MPEG-7 테스트 시퀀스로 공인된 비디오를 이용하여 데이타베이스로 구성하였는데 데이타베이스에 저장되어 있는 시퀀스들은 352×240 해상도로서 총 13,309 프레임에, 68개의 이동 객체를 포함하고 있다. 이 시퀀스들은 MPEG-7 표준화 작업에서 움직임 궤적 기술자(motion trajectory descriptor)의 성능 테스트를 위해 제작된 것으로서 자연스러운 움직임 궤적을 갖는 자동차들과 다양한 움직임을 갖는 사람들, 그리고 우마차들로 이루어져 있으나 인위적으로 만들어진 복잡한 형태의 움직임은 포함하지 않는다[16].

비디오 검색의 정확도에 대한 성능 평가는 영상 데이타베이스의 성능 평가에 널리 쓰이는 recall과 precision을 이용하였다.

4.1 움직임 속도에 무관한 검색의 성능 평가

움직임 속도에 무관한 검색을 수행하기 위해서는 그림 4에서 보이는 것과 같이 데이타베이스를 구축할 때 각 샷의 이동 객체의 움직임에 대한 거리 영상을 생성하여 이를 각 샷과 함께 저장해야 한다. 객체의 움직임 방향과 경로를 고려하여 6개의 질의를 만들어 사용하였는데(query by sketch), 실세계의 객체가 가질 수 있는 일반적인 움직임을 고려하여 다양한 방향의 직선과 곡선을 포함하도록 하였으나 인위적으로 만들어 낼 수 있는 복잡한 움직임은 포함하지 않도록 하였다.

각 질의에 대하여 같거나 비슷한 궤적이 어떤 것인가를 컴퓨터 공학을 전공하는 대학원생들에게 설문 조사하여 과반수 이상이 같거나 비슷하다고 답한 데이타베이스의 궤적을 ground truth로 하였다.

4.1.1 공간 위치를 고려한 검색

4.1.1.1 계산의 정확도

Dagtas 등의 방법은 질의 혼적과 데이타베이스에 있는 혼적 사이에 겹쳐지는 영역이 없는 경우에는 두 혼적 영상 사이의 거리 정보를 제공할 수 없으며 객체의 크기에 따라 정합의 결과가 달라지는 단점이 있다. 제안하는 방법과 Lee 등의 방법은 질의로 주어진 객체의 크기에 무관한 검색을 수행한다. 한편 Lee 등의 방법에서 다항식 계수의 비교에 의한 비디오 검색은 다항식 계수의 차이가 물리적인 궤적의 차이와는 관련이 없기 때문에, 단순히 4개의 다항식 계수를 비교해서는 정확한 검색 결과를 얻기 힘들며, 움직임의 일부분이 비슷한 움직임을 찾을 수 없기 때문에 응용 범위가 제안하는 방법에 비해 좁다.

그림 11은 제안하는 방법과 Dagtas 등의 방법, 그리고 Lee 등의 방법을 이용하여 움직임 속도에 무관하고 공간 위치를 고려한 비디오 검색을 수행한 성능을 보이고 있는데, 제안하는 방법은 기존의 방법에 비해 recall은 최대 14%, precision은 최대 17% 성능이 향상된다.

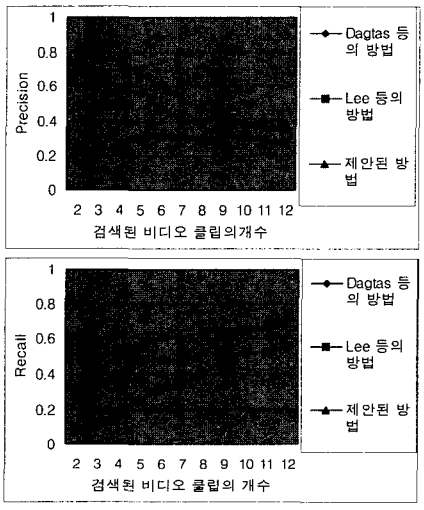


그림 11 움직임 속도에 무관하고 공간 위치를 고려한 검색의 성능 평가

4.1.1.2 수행 시간

Dagtas 등의 방법은 객체의 공간 위치를 고려한 검색을 수행하기 위하여 너비와 높이가 각각 N인 영상에

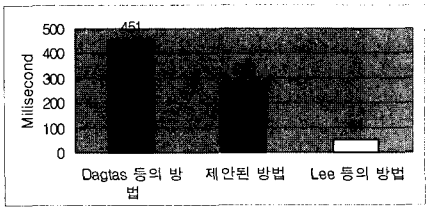


그림 12 움직임 속도에 무관하고 공간 위치를 고려한 검색의 수행 시간 비교

대하여 $O(N^2)$인 계산이 필요하고, Lee 등의 방법의 계산량은 상수로 주어진다. 제안하는 방법을 이용하여 검색을 수행하기 위한 계산량은 n개의 점으로 구성되어있는 궤적의 경우 $O(n)$으로 주어진다. 그림 12는 한 개의 질의를 이용하여 검색을 수행했을 때 검색 시스템이 응답하기까지 걸리는 시간을 보이고 있다. 수행 시간의 비교를 위해 질의로 사용된 비디오 샷은 MPEG-7 테스트 시퀀스로 공인된 비디오 중 Road 시퀀스에 있는 샷으로써 349개의 프레임으로 이루어져 있다.

4.1.2 공간 위치에 무관한 검색

4.1.2.1 계산의 정확도

공간 위치에 무관한 검색을 위하여 Dagtas 등은 그들의 공간 위치를 고려한 검색 방법을 그대로 이용하여 확장하였는데, 제안하는 방법에 비해 정확성이 떨어진다. 한편 Lee 등은 움직임 혼적에서 첫번째 점의 좌표와 마지막 점의 좌표로부터 움직임 방향을 계산하여 정합에 이용였는데, 움직임이 직선일 경우에는 효율적으로 사용될 수 있으나 곡선을 포함하는 움직임에 대한 검색은 그 정확도가 떨어진다. 그림 13은 제안하는 방법과 기존의 방법들의 검색 성능을 나타낸 것인데, 제안하는 방법이 기존의 방법에 비해 recall은 최대 5%, precision은 최대 17%의 성능 향상이 있음을 보이고 있다.

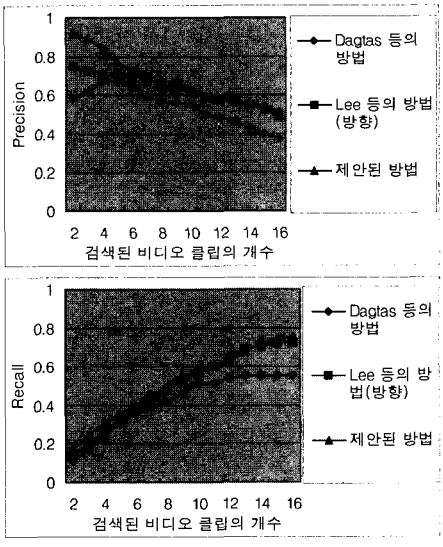


그림 13 움직임 속도에 무관하고 공간 위치에 무관한 비디오 검색의 성능 평가

4.1.2.2 수행 시간

본 논문에서 제안하는 방법으로 객체의 공간 위치에 무관(spatial translation-invariant)한 검색을 위한 계산량은 n 개의 점으로 구성되어있는 궤적의 경우 $O(n^2)$으로 주어지며, Dagtas 등의 방법은 고속 퓨리에 변환을 수행해야 하므로 영상의 폭과 높이가 각각 N일 경우 $O(N^2\log N)$의 계산량이 필요하다. 한편 Lee 등의 방법의 계산량은 상수로 주어진다. 그림 14는 한 개의 질의에 대한 응답 시간을 보이고 있다.

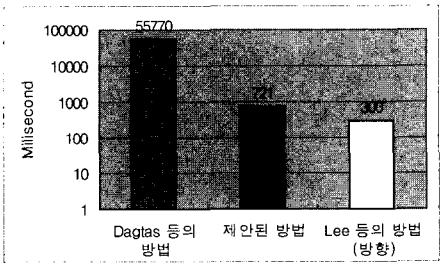


그림 14 움직임 속도에 무관하고 공간 위치에 무관한 비디오 검색의 수행 시간 비교.

4.2 움직임 속도를 고려하고 공간 위치에 무관한 검색의 성능 평가

움직임 속도를 고려한 검색을 수행하기 위해서는 먼저 각 샷에 대한 움직임 검색 코드를 생성하여 각 샷과 함께 데이타베이스에 저장해야 한다. 움직임 속도를 고려하고 공간 위치에 무관한 검색의 성능 평가를 위하여 본 논문에서는 데이타베이스를 구성하고 있는 궤적 중 7개를 질의로 선택하여 사용하였다. 질의로 사용된 궤적은 도로를 운행 중인 자동차, 행인, 자전거, 그리고 우마차 등의 움직임을 표현하고 있는데, 움직임의 속도와 움직임의 형태를 다양하게 선택하여 성능 평가에 적합하도록 하였다.

4.2.1 계산의 정확도

Dagtas 등은 움직임 속도를 고려하고 공간 위치에 무관한 비디오 검색을 수행하기 위하여 유클리디언 거리를 정합 척도로 사용하였는데, 유클리디언 거리는 인간 시각 특성을 반영하지 못할 뿐 아니라 움직임에 관

한 정합 척도가 아니기 때문에 직관적으로 수긍하기 힘든 결과를 출력할 수 있다. 한편, Lee 등의 방법에서 객체의 움직임 속도만 이용한 검색 기법은 객체가 이동해 나가는 중간 경로에 관한 고려가 없기 때문에 사용자의 직관과 전혀 다른 결과를 출력하는 경우가 많았고, 따라서 본 실험에서는 움직임 속도 정보와 함께 다항식 근사화를 통해 얻은 다항식 계수를 함께 사용함으로써 객체의 움직임 경로에 관한 고려를 할 수 있도록 하였다.

본 논문에서는 움직임 속도를 고려하고 공간 위치에 무관한 비디오 검색을 수행하기 위하여 움직임 검색 코드를 제안하였는데, 움직임 검색 코드는 객체의 움직임 방향과 크기에 대하여 작은 속도 변화 제약을 반영하여 설계되었다. 그러므로 기존의 체인 코드를 이용한 검색이나 유클리디언 거리를 정합 척도로 사용한 방법에 비해 정확한 결과를 얻을 수 있다.

그림 15는 제안된 방법을 Dagtas 등의 방법, Lee 등이 제안한 방법과 비교한 결과를 recall과 precision로 나타낸 것으로서, 제안된 방법이 기존의 방법에 비해 recall은 최대 14%, precision은 최대 15% 향상되었음을 보이고 있다.

한편, 움직임 검색 코드 생성 과정에 양자화 오류가 존재할 수 있지만, 양자화 과정에서 궤적의 잡음 제거 효과도 동시에 얻을 수 있다. 따라서 양자화는 그림 15에서 보이는 바와 같이 검색 성능을 저하시키지 않는다.

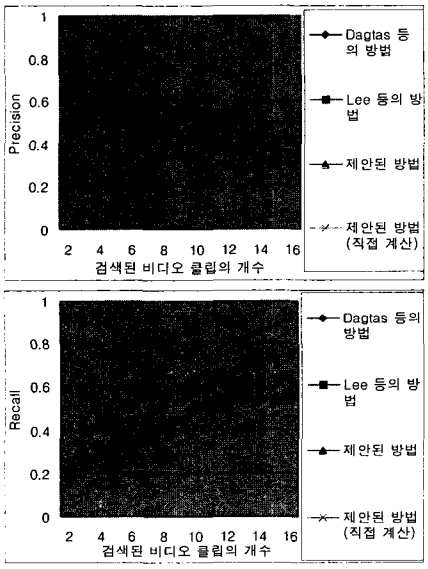


그림 15 움직임 속도를 고려하고 공간 위치에 무관한 비디오 검색의 성능 평가

또한 본 논문에서는 움직임 크기에 대한 표현을 효과적으로 하기 위해 데이타베이스로 주어진 비디오 영상들에 존재하는 모든 객체의 움직임 크기를 누적 히스토그램으로 나타낸 다음, 누적 히스토그램의 빈(bin)의 값이 전체 움직임 벡터 개수의 1/4에 해당 할 때를 기준으로 객체의 움직임 크기를 양자화 하였다. 그러므로 질의로 주어진 움직임이 데이타베이스로 주어진 비디오의 중요한 움직임(frequent motion)과 유사할 경우에는 정확한 검색이 가능하지만, 질의로 주어진 움직임이 데이타베이스로 주어진 비디오의 중요한 움직임과 차이가 클 경우에는 움직임 크기에 대한 양자화 간격이 커져서 움직임 검색 코드의 변별력이 떨어질 수 있다. 그러나 통계학적으로 이런 경우는 많지 않다.

4.2.2 수행 시간의 비교

Dagtas 등의 방법을 이용한 정합의 계산 요구량은 데이타베이스의 궤적과 질의 궤적이 각각 n, m 개의 점으로 이루어져 있을 경우, $O(nm)$의 곱셈이 필요하다. 한편 Lee 등의 방법은 질의로 주어진 궤적과 데이타베이스의 궤적 사이의 움직임 속도, 다항식 계수 등을 이용하였기 때문에 검색을 위한 계산량은 프레임이나 궤적의 길이에 관계없이 상수로 주어진다. 제안하는 알고리즘을 이용한 정합은 움직임 검색 코드 자체가 정합 척도를 내포하고 있기 때문에 정합하고자 하는 두 움직임 검색 코드에서 실제 정합은 서로 다른 비트의 수만 계산하면 된다. 이때 부분 궤적(sub trajectory)에 대한 정합을 위해 궤적을 이루는 각 점에 대해 한 점씩 이동(shift) 시켜나가면서 계산을 수행하기 때문에 제안하는 알고리즘을 이용한 정합의 계산 요구량은 데이타베이스의 궤적과 질의 궤적이 각각 n, m 개의 점으로 이루어져 있을 경우, $O(nm)$의 비트 연산이 필요하고, look up table을 사용할 경우에는 $O(nm)$의 look up table 참조가 필요하다.

한편, 제안된 방법은 데이타베이스를 구성할 때 주어진 궤적에 대한 움직임 검색 코드를 미리 생성해서 저장해야 하며, Lee 등의 방법은 주어진 궤적에 대한 다항식 근사화와 객체의 이동 방향을 계산하여 저장해야 한다. 또한 제안된 방법은 실제 검색을 수행할 때 질의 궤적에 대한 움직임 검색 코드를 생성하기 위한 시간이 필요하고, Lee 등의 방법은 질의로 주어진 궤적에 대한 다항식 근사화 및 이동 방향을 계산하기 위한 시간이 필요하다. 그러나 제안하는 방법과 Lee 등의 방법에서 전처리 시간은 주어진 질의에 대해 단 한번만 수행되기 때문에 비디오 데이타베이스의 대용량성을 고려했을 때 무시할 수 있을 정도의 짧은 시간이라고 할 수 있다. 한편 Dagtas 등의 방법은 별도의 전처리가 필요하지 않다.

그림 16은 본 논문에서 제안한 방법과 기존의 다른

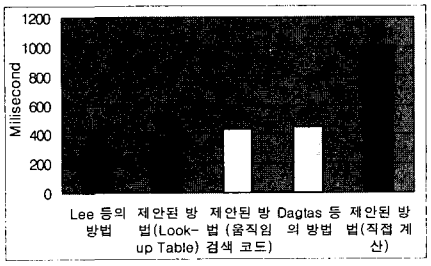


그림 16 움직임 속도를 고려하고 공간 위치에 무관한 비디오 검색의 수행 시간 비교

방법들의 수행 시간을 비교하였는데, 주어진 질의에 대한 전처리 과정을 포함한 전체 수행시간을 보이고 있다.

5. 결 론

본 논문에서는 객체의 움직임 정보를 이용한 효율적인 비디오 검색 기법을 제안하였다. 제안된 기법은 움직임 속도에 무관한 검색(temporal scale-invariant retrieval)과 움직임 속도를 고려한 검색(temporal scale-absolute retrieval)으로 나누어 검색을 수행하는데, 움직임 속도에 무관한 검색은 공간 위치를 고려한 검색(spatial translation-absolute retrieval)과 공간 위치에 무관한 검색(spatial translation-invariant)을, 움직임 속도를 고려한 검색은 공간 위치에 무관한 검색을 포함한다.

움직임 속도에 무관한 검색을 수행하기 위하여 데이타베이스에 있는 혼적을 거리 변환하여 거리 영상을 생성한 후, 거리 영상을 정합에 이용함으로써 정확하고 빠른 검색을 수행할 수 있도록 하였다. Recall과 precision에 의한 성능 평가 결과, 공간 위치를 고려한 검색에 대하여 기존의 방법보다 recall은 최대 14%, precision은 최대 17% 성능이 향상되었고, 공간 위치에 무관한 검색 기법은 기존의 방법에 비해 recall은 최대 5%, precision은 최대 17%의 성능 향상이 있음을 확인하였다.

또한, 움직임 속도를 고려하고 공간 위치에 무관한 검색을 수행하기 위해 움직임 검색 코드(Motion Retrieval Code)를 제안하였다. 움직임 검색 코드는 작은 속도 변화 제약(small velocity change constraint)이 반영되도록 설계하여 인간 시각 특성을 반영하였고, 효율적인 정합을 고려하여 비트 연산이 가능하도록 설계하였다. 그러므로 제안하는 방법은 정확하면서도 고속의 정합을 수행할 수 있다. 움직임 검색 코드를 이용한 실험 결과, 제안하는 방법은 기존의 방법에 비해 recall은 최대 14%, precision은 최대 15% 성능이 향상됨을 확인할 수 있었다.

향후 사용자의 요구 사항을 더욱 유연하게 반영할 수 있는 비디오 검색 방법에 대한 연구가 필요하며, 주어진 궤적을 분할하여 정합 할 수 있는 방법에 관한 연구가 필요하다.

참 고 문 헌

[1] V. E. Ogle, "Chabot : Retrieval from a Relational Database of Images," IEEE Computer, vol. 28, no. 9, pp. 40-48, Sep. 1995.

[2] A. Mojsilovic, J. Hu, "A Method for Color Content Matching of Images," Proc. of the 2000 Int. Conf. on Multimedia and Expo, vol. 2, pp. 649-652, Jul. 2000.

[3] B. S. Manjunath and W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data," IEEE Trans. on Pattern Anal. Machine Intell.," vol. 18, no. 8, pp. 837-842, Aug. 1996.

[4] M. Flickner, M. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hanfner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and Video Content : The QBIC System," IEEE Computer, vol. 28, no. 32, pp. 23-32, Sep. 1995.

[5] R. Brunelli, O. Mich, and C. M. Modena, "A Survey on the Automatic Indexing of Video Data," J. of Visual Comm. and Image Rep., vol.2, no. 10, pp. 78-112, Jun. 1999.

[6] E. Ardizzone and M. L. Cascia, "Video Indexing Using Optical Flow Field," Proc. of Int. Conf. on Image Processing, vol. 3, pp. 831-834, Sep. 1996.

[7] C. Colombo, A. D. Bimbo, and P. Pala, "Semantics in Visual Information Retrieval," IEEE Multimedia, vol. 6, no. 3, pp. 38-53, Sep. 1999.

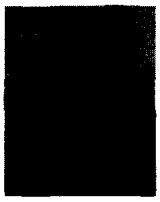
[8] D. Zhong and S. F. Chang, "Region Feature Based Similarity Searching of Semantic Video Objects," Proc. of Int. Conf. on Image Processing, vol. 2, pp. 111-115, 1999.

[9] J. R. Smith and S. F. Chang, "VisualSEEk : A Fully Automated Content-Based Image Query System," In ACM Multimedia Conf., pp. 87-98, Nov. 1996. [Online]. Available WWW : http://www.ctr.columbia.edu/ VisualSEEk.

[10] A. Hanrapur, A. Gupta, B. Horowitz, C. F. Shu, C. Fuller, J. Bach, M. Gorkani, and R. Jain, "Virage Video Engine," Proc. of SPIE in Storage and Retrieval for Image and Video Databases V, vol. 3022, pp. 188-197, Feb. 1997.

[11] H. D. Wactlar, T. Kanade, M. A. Smith, S. M. Stevens, "Intelligent Access to Digital Video :

Informedia Project," IEEE Computer, vol. 29, no. 5, pp. 46-52, May 1996.
[12] A. Pentland, R. W. Picard, and S. Sclaroff, "Photo book : Content-Based Manipulation of Image Database," Int. J. of Computer Vision, vol. 18, no. 3, pp. 233-254, 1996.
[13] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A Fully Automated Content-Based Video Search Engine Supporting Spatiotemporal Queries," IEEE Trans. Circuits and Sys. for Video Tech., vol. 8, no. 5, pp. 602-615 Sep. 1998.
[14] S. Dagtas, W. Al-Khatib, A. Ghafoor and R. L. Kashyap, "Models for Motion-Based Video Indexing and Retrieval," IEEE Trans. on Image Processing, vol. 9, no. 1, pp. 88-101, Jan. 2000.
[15] A. Yoshitaka, Y. Hosoda, M. Yoshimitsu, "VIOLONE : Video Retrieval by Motion Example," J. of Visual Languages and Computing, vol. 7, no. 4, pp. 423-443, 1996.
[16] K. W. Lee, W. S. You and J. Kim, "Video Retrieval based on the Object's Motion Trajectory," Proc. of SPIE in Visual Comm. and Image Processing, vol. 4067, pp. 114-124, 2000.
[17] W. Chen and S. F. Chang, "Motion Trajectory Matching of Video Objects," Proc. of SPIE, on Storage and Retrieval for Media Databases, vol. 3972, pp. 544-553, 2000.
[18] Z. Aghbari, K. Kaneko, and A. Makinouchi, "Modeling and Querying Videos by Content Trajectories," Proc. of 2000 IEEE Int. Conf. on Multimedia and Expo, vol. 1, Aug. 2000.
[19] N. H. AbouGhazaleh, Y. E. Gamal, "Compressed Video Indexing Based on Object Motion," Proc. of SPIE in Visual Comm. and Image Processing, vol. 4067, pp. 986-993, 2000.
[20] S. Panchanathan, F. Golshani and Y. C. Park, "VideoRoadMap : A System for Interactie Classification and Indexing of Still and Motion Pictures," Proc. IEEE Instrumentation and Measurement Tech. Conf., pp. 18-21, 1998.
[21] V. Singla, Y. C. Park, S. Panchanathan, F. Golshani, "Video Composition and Retrieval," Proc. of Int. Conf. on Multimedia and Expo, vol. II, pp. 1163-1166, 2000.
[22] G. Borgefors, "Hierarchical Chamfer Matching : A Parametric Edge Matching Algorithm," IEEE Trans. on Pattern Anal. Machine Intell., vol. 10, no. 6, pp. 849-865, Nov. 1988.
[23] N. Dimitrova and F. Golshani, "Motion Recovery for Video Content Classification," ACM Trans. on Information Sys., vol. 13, no. 4, pp. 408-439, Oct. 1995.
[24] 김소연, 노영만, "압축 영역에서 객체 움직임 맵에 의한 효율적인 비디오 인덱싱 방법에 관한 연구", 한국정보처리학회 논문지, 제7권 제5호, pp. 1570-1578, 2000년 5월.
[25] D. H. Ballard, C. M. Brown, Computer Vision, Prentice-Hall, 1982.
[26] I. K. Sethi and R. Jain, "Finding Trajectories of Feature Points in an Monocular Image Sequence," IEEE Trans. on Pattern Anal. Machine Intell., vol. 9, no. 1, pp. 56-73, Jan. 1987.
[27] K. Rangarajan and M. Shah, "Establishing Motion Correspondence," CVGIP : Image Understanding, vol. 54, no. 1, pp. 56-73, 1991.



정 종 면
1992년 한양대학교 전자계산학과 졸업(공학사). 1994년 한양대학교 전자계산학과 졸업(공학석사). 2001년 한양대학교 전자계산학과 졸업(공학박사). 2001년 ~ 현재 한국전자통신연구원 방송미디어연구부 선임연구원. 관심 분야는 데이터 방송, 대화형 방송, 컴퓨터 비전, 내용기반 비디오 검색, 디지털 워터마킹 등

문 영 식
1980년 서울대학교 전자공학과 졸업(공학사). 1982년 한국과학기술원 전기및전자공학과 졸업(공학석사). 1990년 University of California, Irvine 컴퓨터공학과 졸업(공학박사). 1982년 ~ 1985년 한국전자통신연구소 연구원. 1989년 ~ 1990년 InnoVision Medical(미국) 선임연구원. 1990년 ~ 1992년 생산기술연구원 선임연구원. 1992년 ~ 현재 한양대학교 컴퓨터공학과 부교수. 관심분야는 내용기반 멀티미디어 정보검색, 비디오 인덱싱, 객체기반 멀티미디어 통신, 멀티미디어 저작권 보호, 얼굴 및 제스쳐 인식, 병렬 컴퓨팅, 자동 시각 검사 등