

Optical flow 이론을 이용한 움직이는 객체의 자동 추출에 관한 연구

정철관, 김경수, 김중규⁰
⁰성균관대학교 전기전자컴퓨터공학부
0331-290-7199 / 0331-290-7191

A study on automatic extraction of a moving object using optical flow

C.K.Jung, K.S.Kim, J.K. Kim
The School of Electrical and Computer Science Engineering, SKKU
Email : ckjung@ece.skku.ac.kr

Abstract

In this work, the new algorithm that automatically extracts moving object of the video image is presented. In order to extract moving object, it is that velocity vectors correspond to each frame of the video image. Using the estimated velocity vector, the position of the object are determined. the value of the coordination of the object is initialized to the seed, and in the image plane, the moving object is automatically segmented by the region growing method. As the result of an application in sequential images, it is available to extract a moving object.

1. 서론

MPEG-1, MPEG-2, H.261이나 H.263과 같은 기존의 비디오 부호화 표준(block-based video coding standards)은 블록을 기반으로 하기 때문에 영상의 의미나 내용에 대해서는 다루지 않았다. 이러한 기존의 블록 기반 부호화 표준들은 고압축률을 가지고 있기 때문에 비디오 CD, 디지털 TV 방송, 영상통신, 영상전화 등에 널리 이용되었다. 최근 멀티미디어의 중요성이 강조됨에 따라 ISO(the International Organization of Standardization)에서는 멀티미디어 서비스를 위한 부호화 표준으로 MPEG-4를 제정했다. MPEG-4는 사람이 의미와 내용이 있는 객체를 부호화하는 객체 기반 부호화(object based coding) 방식을 채택하고 있다. 그러므로 영상에서 객체(object)의 분할(segmentation)은 매우 중요한 과정이

라 할 수 있다. 효과적이고 정확한 객체 분할 알고리즘은 객체 기반 부호화 방식에 중요한 전제조건이 된다. [1][2]

본 논문에서는 비디오 영상의 시공간적인(spatio-temporal) 특성을 이용해 움직이는 객체를 자동으로 추출하는 알고리즘을 제안하였다. 제안한 알고리즘은 크게 움직임 추정(motion estimation)과 영역 확장(region growing)으로 구성된다. 움직임 추정을 통해 비디오 영상에서의 움직이는 객체에 대한 시간적인(temporal) 정보를 얻고, 영역 확장을 이용해 공간적인(spatial) 정보를 얻게 된다. 이를 통해 얻어진 시공간적인 정보를 이용해 최종적으로 영상에서 움직이는 객체를 분할한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 제안하는 알고리즘을 설명하고, 제 3장에서는 실험결과를 보이며, 제 4장에서 결론을 맺는다.

2. 움직이는 객체의 자동추출 알고리즘

본 논문에서는 연속영상에서 움직이는 객체를 자동으로 추출하는 알고리즘을 제안하였다. 동영상에서 움직이는 객체는 움직임계(motion field)의 시간적인 일관성(temporal coherence)과 intensity의 공간적인 동질성(spatial homogeneity)을 가지고 있다. 본 논문에서 제안된 알고리즘은 움직이는 객체의 두 가지 성질을 이용해 자동으로 영역 분할을 수행한다. 제안된 알고리즘의 전체적인 구조는 그림 1과 같다.

그림 1에서 보는 바와 같이 실제 영상의 분할은 영역확장에 의해 수행된다. 그 전처리 과정으로 optical flow를 이용해 영역 확장에 필요한 초기 좌

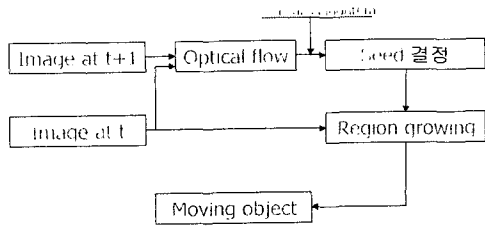


그림 1 제안된 자동추출 알고리즘

표값, 즉 seed를 얻게 된다. Optical flow에 의해 얻어진 seed 값을 이용해서 자동으로 움직이는 객체를 분할할 수 있다.

2.1 Optical flow

영상에서의 움직임(motion)에는 두 가지 종류가 있다. 카메라의 움직임에 의해 발생하는 전역 움직임(global motion)과 움직이는 객체에 의해 발생하는 지역 움직임(local motion)이다. 움직이는 객체를 추출하기 위해서는 전역 움직임의 보상이 선행되어야 한다. 전역 움직임이 보상되면 영상에서의 움직임은 객체의 움직임에 의해 발생하는 것이다. 본 논문에서는 이러한 전역 움직임은 없는 것으로 가정한다.

움직이는 객체에 의한 지역 움직임은 optical flow에 의해 추정될 수 있다. Optical flow란 움직이는 객체에 의해 형성되는 광도패턴(brightness pattern)의 이동을 의미한다. 이는 관찰자나 물체의 이동에 의해 영상에서 velocity field로 나타낼 수 있다.

시간 t 에서, 영상에서의 한 점 (x, y) 의 intensity를 $I(x, y, t)$ 라고 표현하면, 연속영상에 대하여 intensity $I(x, y, t)$ 의 변화율은 0가 되어 (1)식과 같이 나타낼 수 있다. [3]

$$\frac{d}{dt} I(x, y, t) = 0 \quad (1)$$

(1)식은 chain rule에 의해 (2)식과 같이 된다.

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (2)$$

여기서 $I = I(x, y, t)$ 이다.

$u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$ 라고 하면, (3)식과 같이 된다.

$$I_x u + I_y v + I_t = 0 \quad (3)$$

여기서 $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $I_t = \frac{\partial I}{\partial t}$ 이고, u 와 v 는 optical flow vector를 이루는 구성성분들로서 velocity vector들이다.

(3)식에서 u , v 를 계산하기 위해 Horn과 Schunck는 variation of calculus를 이용해 오차함수(error function)를 최소화시키는 방법을 제안하였다. 속도계(velocity field)가 평탄하다(smooth)고 가정하면, (3)식은 variational method에 의해 해를 계산해 낼 수

가 있다.

속도계의 평탄성에 대한 척도(smoothness measure : E_s)와 (3)식에서 발생하는 오차 (E_t)는 각각 다음 식들로 표현된다.

$$E_t = \int \int (I_x u + I_y v + I_t)^2 dx dy \quad (4)$$

$$E_s = \int \int (u_x^2 + u_y^2 + v_x^2 + v_y^2) dx dy \quad (5)$$

가중상수(weighting constant) λ 를 이용하여 (4),(5)식을 종합하여 오차함수 E_t 를 표현하면

$$E_t = E_t + \lambda E_s \quad (6)$$

이 된다. E_t 를 최소화함에 의해 u , v 를 구하면

$$\nabla^2 u = \lambda (I_x u + I_y v + I_t) I_x \quad (7)$$

$$\nabla^2 v = \lambda (I_x u + I_y v + I_t) I_y \quad (8)$$

와 같다.

여기서 $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ 이다.

2.2 K-means algorithm에 의한 seed 결정

2.1절에서 optical flow를 이용해 속도벡터를 구할 수 있었다. 속도벡터가 존재한다는 것은 움직이는 객체에 의한 intensity의 변화가 있다는 뜻이다. 따라서 속도벡터가 존재하는 픽셀을 찾아내면 영역확장(region growing)의 시작점인 seed 값을 구할 수 있다. 이때 seed가 되는 좌표값을 결정하는 과정이 필요한데, K-means 알고리즘은 이러한 seed를 결정하는데 사용된다. K-means 알고리즘은 두 점 사이의 거리의 유사도가 높은 좌표값들을 묶어 좌표값의 집합인 cluster를 형성하는 알고리즘이다. K-means 알고리즘은 다음과 같은 과정을 거친다.[4]

Step 1. Cluster의 개수와 각각의 cluster의 초기 중심점을 선택한다.

$$\{z_1(l), z_2(l), \dots, z_k(l)\} \quad (9)$$

여기서 z 는 cluster의 중심점을 의미하고, k 는 cluster의 개수, l 은 반복회수를 나타낸다.

Step 2. 각각의 좌표들이 해당하는 cluster를 다음 조건에 의해 찾아낸다.

$$x^{(l)} \in S_j(l) \text{ if } \|x^{(l)} - z_j(l)\| < \|x^{(l)} - z_i(l)\| \text{ for all } i=1,2,\dots,K, i \neq j \quad (10)$$

$S_j(l)$ 은 l 번 반복시, j 번째 cluster의 집합을 의미한다.

Step 3. 이전의 Step 2에서 형성된 새로운 클러스터의 좌표값들을 평균하여 새로운 중심점을 계산한다.

새로운 중심점 $z_j(l+1)$ 은 (11)식과 같이 된다.

$$z_j(l+1) = \frac{1}{N_j} \sum_{x^{(a)} \in S_j(l)} x^{(a)} \quad (11)$$

여기서 N_j 는 S_j 에 소속된 좌표값의 개수를 나타낸다.

Step 4. (12)식과 같이 cluster의 중심점이 변화되지 않는다는 조건을 점검하여 최종적인 cluster의 중심점을 구한다.

$$z_j(l+1) = z_j(l) \quad j=1, 2, \dots, K \quad (12)$$

(12)식을 만족하지 않으면 step 2 를 반복한다.

(12)식에 의해 결정된 z_j 값이 영역확장에 사용될 seed 이다.

2.3 영역확장(region growing)

일반적으로 영상에서 동일한 객체의 영역에 소속된 픽셀들은 서로 유사한 통계적인 특성을 가지고 있다. 영역확장은 영상에서의 이러한 특성을 이용하여 초기 시작점, 즉 seed로부터 시작하여 유사한 영역을 확장시키는 방법이다. 이러한 seed 값은 사용자가 설정을 해주어야 한다. 그림 2는 영역확장을 설명하는 그림이다. 그림에서 보는 바와 같이 seed 픽셀과 이웃한 픽셀들의 intensity 값을 비교하여 유사도가 높은 경우에 영역을 확장해 나간다. [5]

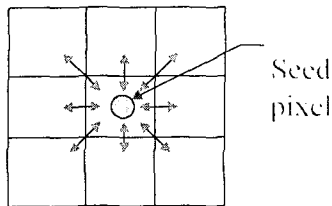


그림 2 영역확장

영상에서 영역의 동질성(homogeneity)을 검사하는 식은 아래와 같다.

$$P(R_i^{(k)} \cup \{x\}) = TRUE \quad (13)$$

여기서 $P(\cdot)$ 는 논리 연산자를 나타내고 $R_i^{(k)}$ 는 seed 픽셀과 이웃한 k 개의 픽셀들을 의미한다. 만약 seed 픽셀과 이웃한 영역이 서로 동질성(homogeneity)을 갖는다면 (13)식과 같이 되어 영역을 확장할 수 있다.

본 논문에서는 사용자가 설정해 주어야 하는 seed 값을 2.2절에 의해서 결정된 좌표값에 의해 자동으로 설정해 준다. 또한 이 seed 값은 움직이는 객체의 중

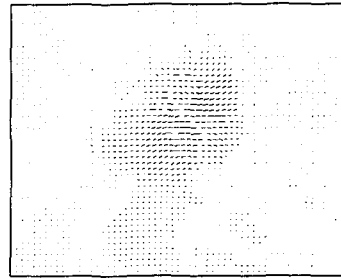
심점이기 때문에 움직이는 객체의 영역을 분할하게 된다.

3. 실험결과

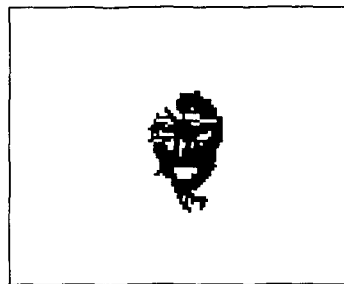
본 절에서는 제안한 알고리즘의 효율성을 확인하기 위해 연속영상에 대해 모의실험을 한 결과이다. 제안 알고리즘을 Akiyo, Foreman 영상에 각각 적용하여 얻은 실험결과이다. 실험환경은 Celeron 300MHz CPU, 32MB의 메모리를 가지는 IBM호환 PC상에서 수행하였으며 OS는 Windows 98, 개발툴은 Visual C++ 6.0을 사용하였다.



(a) 입력 영상



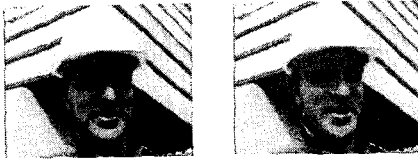
(b) Optical flow에 의한 seed 결정



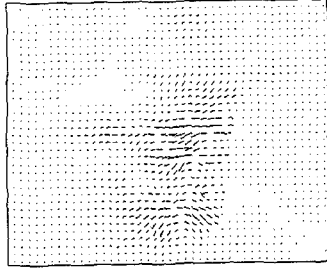
(c) 분할된 영상

그림 3 Akiyo 영상

그림 3-(a)에서 보는 바와 같이 실험영상은 얼굴만이 이동하고 있으며, 속도벡터를 구한 결과가 그림 3-(b)이다. 그림 3-(c)는 최종적으로 분할된 영상이며 움직이는 객체에 해당하는 얼굴부분을 찾아낸다.



(a) 입력영상



(b) optical flow



(c) 분할된 영상
그림 4 Foreman 영상

그림 4는 제안알고리즘을 foreman 영상에 적용하여 얻은 결과이다. Akiyo 영상과 같이 움직이는 객체가 얼굴이므로 그림 4-(c)와 같이 영상을 분할 할 수 있었다.

4. 결론

본 논문에서는 동영상에서 움직이는 객체를 자동으로 추출하는 알고리즘을 제안하였다. 움직이는 객체를 자동으로 추출하기 위해 optical flow 이론을 이용하여 움직임이 존재하는 영역의 좌표값을 구하고 그 점을 seed로 하여 영역확장을 수행하였다. 모의실험결과 사용자의 수작업 없이도 자동으로 움직이는 객체를 분할할 수 있었다.

참고문헌

- [1] J. Fan G. Fujita M. Furuie T. Onoye I. Shirakawa L. Wu, "Automatic moving object extraction toward compact video representation", Opt. Eng. 39(2) pp 438-452 February 2000
- [2] M. K. Kim J. G. Choi D. Kim H. Lee M. H. Lee C. Ahn Y. S. Ho, "A VOP Generation Tool : Automatic Segmentation of Moving Objects in Image Sequences Based on Spatio-Temporal Information", IEEE Trans. Circuits and Systems for Video Technology vol. 9 No. 8 December 1999
- [3] B. K. P. Horn B.G. Schunck, "Determining Optical Flow", Artificial Intelligence 17 pp 185-203 1981
- [4] A. S. Pandya R. B. Macy, "Pattern Recognition with Neural Networks in C++", IEEE Press pp 213-230 1996
- [5] I. Pitas, "Digital Image Processing Algorithms", Prentice Hall pp 254-274 1993