

## 영상 시퀀스의 계층 분리를 위한 움직임 분할

장 정 진, 오 정 수, 홍 현 기, 최 중 수  
중앙대학교 첨단영상전문대학원 영상공학과  
전화 : 02-826-2505 / FAX : 02-814-5404

### Motion Segmentation for Layer Decomposition of Image Sequences

Joung-Jin Chang, Joung-Su Oh, Hyun-Gi Hong, Jong-Su Choi  
Dept. of Image Eng., The Graduate School of  
Advanced Imaging Science, Multimedia, and Film, Chung Ang Univ.  
E-mail : ciel@candy.ee.cau.ac.kr

#### Abstract

This paper proposes a motion segmentation algorithm for layer decomposition of image sequences. The proposed algorithm segments an image into initial regions by using its color and texture and computes a motion model of each initial region. Each pixel assigns one of the motion represented by the models or a motion except them, which segments the image into the motion regions. The proposed algorithm is applied image sequences and the segmented motion is shown.

#### I. 서론

움직임은 영상에서 대상 물체를 추출하거나 구분할 수 있는 가장 강력한 단서로서 이에 기반한 움직임 분할은 영상의 해석에 있어서 중요한 역할을 한다. 따라서 움직임 추적 및 검출, 그리고 광류(optical flow)의 계산 등의 여러 가지 움직임 기반 처리 기법들과 연관되어 컴퓨터 비전 및 영상 압축 분야에서 활발히 연구되고 있다. 정확한 움직임 분할을 위해선 움직임에 따라 구분이 되는 적절한 특징(feature)의 선택이 무엇보다 중요하다. 직관적으로 움직임을 구분할 수 있는 특징으로는 광류를 들 수 있다. 광류는 영상에서 밝기(intensity)로 표현되는 각 화소(pixel)들의 움직임을 나타내며, 서로 다른 움직임을 가지는 독립적인 물체들은 각각 서로 다른 광류를 가지므로 단순한 움직임을 가지는 균일한 물체에 대해서는 쉽게 움직임을 분할을 수행할 수 있다[2][3][4]. 그러나 회전하는 물체나 원근감을 가지고 움직이는 물체에 대해서는 정확한 움직임 분할을 수행할 수 없으므로, 직접적인 특징으로

서 사용되지 않는다. 움직임 분할을 위해 사용되는 일반적인 특징은 움직임 모델로서, 서로 다른 각각의 움직임에 대해 알맞은 모델이 정의될 수 있으므로 가장 많이 사용된다. 기존의 움직임 분할 방법들은 움직임에 맞는 모델을 구하기 위하여 프레임간의 차(difference)를 이용하거나, 광류를 이용한다[2]. 이 중에서 프레임간의 차를 이용하는 방법은 움직임이 있는 영역과 없는 영역을 구분하기 때문에, 영상 전체에 움직임이 있는 경우 움직임 분할에 어려움이 있었다. 반면 광류를 이용하는 방법은 광류의 계산에 따르는 일반적인 문제로 인해 움직임 경계에서의 분할을 정확히 수행하지 못하므로 최적화를 위한 반복수행과 복잡한 후처리 과정으로 인한 계산상의 부담이 컸다[1].

본 논문에서는 색상과 질감에 따른 초기 영역 분할과 함께 4쌍의 대응점을 계산하여 움직임 모델을 계산하고, 각 모델에 맞는 영역을 할당함으로써 움직임 분할을 수행하는 알고리즘을 제안한다. 이 때, 광류를 사용하지 않음으로써 최적화과정의 필요성을 제거하여 계산상의 부담을 줄이며, 영상 전체의 움직임도 분할할 수 있는 장점이 있다.

#### II. 움직임 모델

##### 2.1 움직임 모델의 종류

동차 좌표(homogeneous coordinates)를 사용하여 점을 표현할 때, 일반적으로 2-D 평면 변환을 식(1)처럼 행렬의 곱 형태로 나타낸다[6].

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad (1)$$

이러한 변환의 종류에는 원근감을 표현할 수 있는 사영변환(projective transforms)과 깊이값을 고정시킨

어파인 변환 (affine transforms)이 있으며, 그리고 이동과 회전에 크기의 변환을 나타낼 수 있는 유사변환 (similarity transforms)과 크기를 고려하지 않고 이동과 회전만을 표현할 수 있는 변환 (rigid transforms)이 있다. 정사각형에 대한 각각의 변환 예가 그림 1과 같다.

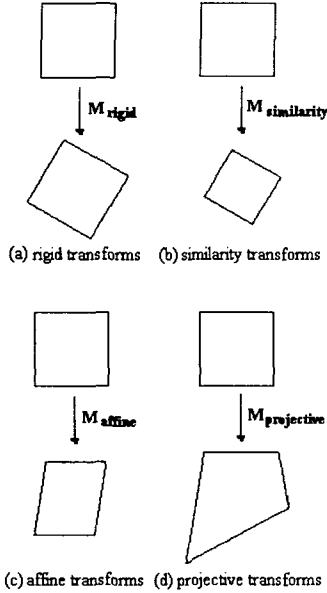


그림 1 Rigid, similarity, affine and projective

움직임 분할을 위한 특징으로서 적절한 모델을 선택하기 위해서는 영상내 물체의 다양한 움직임을 고려하여야 한다. 단순한 이동이나 회전, 또는 크기의 변화만으로는 깊이와 관련된 움직임과 같은 다양한 움직임을 나타낼 수 없기 때문에 기존의 움직임 분할 방법에서는 어파인 변환을 사용하였다[1]. 어파인 변환은 정확한 원근감은 표현하지 못하나 영상내 움직이는 물체들이 카메라로부터 멀리 떨어져 있다고 가정했을 때 각각의 움직임을 효과적으로 정의할 수 있다. 어파인 변환은 다음 식(2)에서처럼 6개의 파라미터를 필요로 하며, 이 파라미터들을 계산하기 위해 식(3)을 사용하였다.

$$\mathbf{M}_{\text{affine}} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$\begin{aligned} V_x(x, y) &= (m_{00} - 1)x + m_{01}y + m_{02}w \\ V_y(x, y) &= m_{10}x + (m_{11} - 1)y + m_{12}w \end{aligned} \quad (3)$$

여기서  $V_x(x, y)$ 와  $V_y(x, y)$ 는 영상에서 각 화소에 대해 구한 광류의  $x$ 축,  $y$ 축 방향으로의 속도성분이다. 이 방법으로 움직임 모델을 구하는 경우에는 첫째, 광류의 계산 결과에 민감하며 둘째, 움직임에 해당하는 영역을 정확히 모르는 경우 정확한 모델을 계산하기 위해 반복적인 계산이 불가피하다는 단점이 있다. 한편, 사영변환은 정확한 원근감의 표현이 가능하기 때문에

움직임 모델로서 적합할 수 있으나 8개의 파라미터를 필요로 하므로 위와 같은 관계를 사용할 경우 계산상의 부담이 커지는 어려움이 있다. 본 논문에서는 좀더 정확한 움직임을 이용하기 위하여 사영 변환을 사용하며 계산량을 줄이기 위해 4쌍의 대응점을 이용하여 직접적으로 8개의 파라미터를 계산하는 방법을 사용한다.

## 2.2 사영변환의 모델 파라미터 계산

사영변환의 변환 행렬은 다음과 같다[6][7].

$$\mathbf{M}_{\text{projective}} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \quad (4)$$

식(4)에서  $m_{22}$ 을 정규화하여 식(1)과 같이 식(5)로 나타낼 수 있다.

$$\begin{bmatrix} x'_k \\ y'_k \\ w'_k \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} \quad (5)$$

식(5)에서 영상 평면상의 이동된 점의 좌표  $p(x'_{ki}, y'_{ki})$ 는  $x'_{ki} = \frac{x_k}{w'_k}$ ,  $y'_{ki} = \frac{y_k}{w'_k}$ 의 관계에 의해 다음과 같이 나타낼 수 있으며,

$$\begin{aligned} x'_{ki} &= \frac{m_{00}x_k + m_{01}y_k + m_{02}}{m_{20}x_k + m_{21}y_k + 1} \\ y'_{ki} &= \frac{m_{10}x_k + m_{11}y_k + m_{12}}{m_{20}x_k + m_{21}y_k + 1} \end{aligned} \quad (6)$$

이 식을 정리하면,

$$\begin{aligned} m_{00}x_k + m_{01}y_k + m_{02} - m_{20}x_k x'_{ki} - m_{21}x'_{ki} y_k &= x'_{ki} \\ m_{10}x_k + m_{11}y_k + m_{12} - m_{20}x_k y'_{ki} - m_{21}y'_{ki} x_k &= y'_{ki} \end{aligned} \quad (7)$$

이 된다. 식(7)에서 미지수는 8개 이므로 기존 프레임에서 한 움직임에 대한 4개의 점과 다음 프레임에서 이동된 4개의 점의 좌표를 알면 8개의 파라미터를 구할 수 있으며, 그 행렬식이 다음과 같다.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 x'_{1i} & -y_1 x'_{1i} \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1 y'_{1i} & -y_1 y'_{1i} \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2 x'_{2i} & -y_2 x'_{2i} \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2 y'_{2i} & -y_2 y'_{2i} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4 x'_{4i} & -y_4 x'_{4i} \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4 y'_{4i} & -y_4 y'_{4i} \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{20} \\ m_{21} \end{bmatrix} = \begin{bmatrix} x'_{1i} \\ y'_{1i} \\ x'_{2i} \\ y'_{2i} \\ \vdots \\ x'_{4i} \\ y'_{4i} \end{bmatrix} \quad (8)$$

## III. 샘플점 선택 및 대응점의 계산

### 3.1 초기 영역의 분할

하나의 움직임에 대응되는 4개의 점을 선택하기 위해서는 적절한 영역의 구분이 중요하다. 이때 영역을 구분할 수 있는 특징으로 색상의 분포나 질감을 사용할 수 있으며, 이것은 움직임이 있는 대상이 특정한 색상 분포나 질감을 가지고 있기 때문에 유효하다[5].

또한 이렇게 얻은 색상과 질감에 대한 정보는 구해진 영역이 확실한 영역이라는 가정을 한다면 영역 할당 과정에서도 사용될 수 있다. 초기 영역을 분할하기 위해 본 논문에서 사용한 방법은 일반적인 칼라 분할 기법에서와 같이 우선, 기준 영상을 양자화 시켜 칼라의 수를 감소시켰으며, RGB 각각의 채널에 대해 히스토그램에 기반하여 색상을 크게 몇 개의 집합으로 나누었다[8]. 이렇게 구해진 분할 영상들을 AND 연산을 통하여 큰 영역만 남겨두고 제거하였다. 이것은 각 채널의 밝기 영상이 상호간에 유사한것에 근거한 것으로서 예를 들어 하나의 채널이 어떤 집합으로 나누어지면, 대상에 따라 다른 채널에서도 유사한 집합이 형성되게 된다. 따라서 모든 채널에 대해 AND연산을 수행하여 겹치는 영역을 골라내고, 큰 영역에 대해서 결과를 정리하면 그 영역은 어떤 대상에 대한 확실한 영역을 형성하게 된다.

### 3.2 샘플점의 선택 및 대응점의 계산

4개의 점을 선택할때는 각 점들이 일직선상에 놓여서는 안되며 너무 작은 영역에 걸쳐서 점이 선택되어서도 안된다. 따라서 앞의 과정에서 구해진 영역내의 사각형을 고려하여 네 꼭지점을 선택하는 방법을 사용하였다. 그러기 위해서 우선 분할된 영상에 대해 각 축에 대한 투영(projection)을 하고 결과의 평균을 중심점으로 선택했다. 그리고 중심점을 기준으로 사각형을 형성하고 이를 확장시킴으로서 영역내에 포함되는 사각형을 구할 수 있다. 대응점 계산은 선택된 4개의 점에 대하여 블록 정합 기법을 사용하여 다음 프레임에서의 대응되는 4개의 점을 찾을 수 있다. 이 때 사용된 블록 정합 기법은 검색 영역을 인접되어 있는 화소에 한정시킴으로서 계산에 걸리는 부하를 최소화할 수 있다[2][7].

## IV. 영역 할당 및 후처리 과정

### 4.1 영역의 할당

불필요한 계산을 줄이기 위해 초기 영역의 분할 과정에서 이미 정해진 영역은 그대로 할당을 한다. 나머지 정해지지 않은 영역에 대하여 앞서 과정에서 구해진 움직임 모델을 적용하여 식 (6)에서와 같이 이동된 점을 예측할 수 있다. 따라서 기준 프레임의 하나의 점에 대하여 모델의 수만큼 경우의 수가 생겼으며, 또한 초기 영역 분할 과정에서 나누어지지 않은, 어느 모델에도 포함되지 않는 영역이 있을 가능성이 존재한다. 우선 모델에 대응되는 정확한 이동점을 구분하기 위하여 이동된 점과 다음 프레임의 같은 위치의 점과의 오차를 계산한다. 이 때 하나의 화소에 대한 오차를 계산할 경우 영상내에서 특정 화소의 값이 고유한 값을 가질 수 없으므로 정확한 결과를 얻을 수 없다. 따라서 이동된 화소를 중심으로 일정 영역내의 화소까지 고려하여 계산하여야 한다. 기준 프레임에서의 하

나의 점  $p(x, y)$ 와 각 모델  $M_k$ 에 따른 이동된 점  $p'(x_k, y_k)$ 와의 관계를  $M_k\{p(x, y)\} = p'(x_k, y_k)$ 라 할때, 영역을 할당하기 위한 식이 다음과 같다.

$$L(x, y) = \underset{k}{\text{Arg min}} \left\{ \sum_{(i, j) \in B} \sqrt{(I(x_k + i, y_k + j) - I(x + i, y + j))^2} \right\} \quad (10)$$

단,  $B = \{-2, -1, \dots, 2\}$

이 때, 만약 오차가 가장 적은 점도 큰 오차값을 가지고 있다면 어느 모델도 영역으로서 할당할 수 없으므로 이 때는 다른 값을 할당한다. 이 과정을 이미 할당된 영역의 화소를 제외한 모든 화소에 대해 처리하게 되면 영역의 할당과정이 끝나게 된다.

### 4.2 후처리 과정

영역의 할당 과정이 초기 분할 과정에서 이미 정해진 영역을 제외하고 화소단위로 이루어 지기 때문에 고립되거나 작은 영역이 존재할 가능성이 있다. 일반적으로 고립되거나 크기가 작은 영역은 영상내에서 의미있는 정보나 대상이 아니므로 비슷한 성질을 가지는 보다 큰 영역과의 병합이 필요하다[1]. 본 논문에서는 주위의 영역중 가장 큰 영역과 합치는 방법으로 단순히 일정 크기의 블록을 사용해서 각 영역의 크기를 계산하고 그 블록내 가장 큰 크기를 갖는 영역으로 중심의 점을 병합시키는 방법을 사용했다.

## V. 제안된 알고리즘의 모의실험

제안한 알고리즘의 전체 구성도가 그림 2와 같다. 시뮬레이션을 위해 사용된 영상은 그림 3에서처럼 MPEG 형식의 "flower garden" 시퀀스의 1,2 프레임을 사용하였으며, 해상도는 352×240이다. 그림 5는 초기 영역분할 결과와 그에 따른 4개의 샘플점을 선택한 결과이다. 어떻게 보이는 영역이 초기 분할된 영역이며

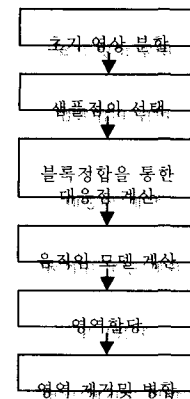


그림 2 구성도

투영결과로부터 얻어낸 중심점과, 그로부터 확장된 사각형이 있다. 그림 5은 선택된 4개의 점에 대해서 블록정합을 통해 대응점을 찾아낸 결과이다. 그림 6은 최종적인 움직임 분할 결과로서 움직임 모델을 통해 찾아낸 2개의 영역과 두 개의 모델에 속하지 않는 하나의 영역까지 총 3개의 영역을 찾아내었다.



그림 3 the "Flower garden" sequence



그림 4 초기 영역과 선택된샘플점



그림 5 계산된 4개의 샘플점



그림 6 움직임 분할 결과

## VI. 결론

본 논문에서는 색상 및 질감을 고려하여 초기 영역을 분할하고, 각 영역에 맞는 4개의 점과 블록정합을 통해 찾아낸 4개의 대응점을 이용하여 움직임 모델을 찾아내어 움직임 분할을 수행하는 알고리즘을 제안하였으며, 이를 모의실험해서 움직임 분할이 수행되는 것을 확인했다. 제안된 방법은 최적화를 위한 반복과정이 필요 없으며 몇 개의 점만으로 정확한 모델의 계산이 가능하다. 하지만 초기 영역의 분할 단계에서 분할된 영역의 수가 적으며, 모델을 이용한 영역 할당 후의 과정에서 경계를 침식시킴으로써 세밀한 정보를 많이 잃어버려서 이에 대한 보완이 필요하다.

이러한 자동적인 움직임 분할방법을 통한, 계층적인 표현은 영상내에서 원하지 않는 물체, 전경, 배경등의 제거나 새로운 영상의 합성에 이용될 수 있으며, 분할을 통해 얻어진 객체는 영상 압축, 영상 합성 등을 위한 데이터베이스에 저장되어 응용될 수 있다.

## 참고문헌

- [1] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE trans. On Image Proc.*, vol. 3, pp. 625-638, Sep. 1994.
- [2] A. M. Tekalp, *Digital Video Processing*, Prentice-Hall PTR Upper Saddle River, NJ 07458, 1995.
- [3] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185-203, 1981.
- [4] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Image Understanding Workshop*, 1981, pp 121 -130.
- [5] Y. Weiss and E. H. Adelson, "A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models," *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, June 1996.
- [6] R. Szeliski. "Image Mosacing for Tele-Reality Application," Technical Report CRL94/2, The Cambridge Research Lab, May 1994.
- [7] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Addison Wesley Longman 1992.
- [8] Y. W. Lim and S. U. Lee, "On The Color Image Segmentation Algorithm Based on the Thresholding and the Fuzzy c-Means Techniques", *Pattern Recognition*, vol. 23, No. 9, pp. 935-952, 1990.