

Moving image segmentation에 관한 연구

*이 원희, 변재웅, 김재영, 정 진현
 광운 대학교 제어계측 공학과

A study on the moving image segmentation

*Wonhee Lee, Chaegung Byun, Jaeyoung Kim, Chinhyun Chung
 Control & Instrumentation Engineering Kwangwoon Univ.

ABSTRACT

Most real image sequences contain multiple moving objects or multiple motions. In this paper, we segmented the moving objects with optical flow. Motion estimation by this method can estimate and compress the image sequences better than other methods such as block matching method. And, especially, we can make new image sequences by synthesizing the segmented objects. But, it takes too much time for motion estimation. And, it is not easy for a hardware implementation.

1. 서론

실제 image에는 여러 가지 움직임을 가지는 object들을 가지는데, motion segmentation이란 유사한 움직임을 갖는 pixel들을 묶어서 앞의 여러 가지 object들을 표현하는 것이다. 예를 들면, MPEG에서 쓰여지는 block matching 방법은 임의의 block내의 motion에 대해서만 spatiotemporal coherence를 고려 하지만, segmentation은 image전체에 대해서 고려하여 유사한 움직임을 가지는 영역에 대해서 하나의 움직임만을 부여함으로써 움직임의 coding에 있어 앞의 방법보다는 더 정확하고 향상된 성능을 가지게 된다. 즉, image sequence의 segmentation은 video data의 spatiotemporal coherence를 image 전체에 대해 정의하는 것이다. 우선, 움직임추정 방법으로 optical flow를 이용하였고 translation, rotation, zoom, shear등의 움직임을 나타내기 위해 affine motion model을 사용한다. 그리고, 유사한 움직임을 갖는 model들을 clustering해 주게 된다. 여기서 clustering을 위해 model에 대한 가정과 그 가정의 검토, 영역 할당을 하게 된다. 이 영역이 적당하게 할당되어 질 때까지 반복적으로 이작업을 수행함으로써 image sequence를 segmentation할 수 있다. 더 나아가 segmentation작업이 성공적으로 이루어졌을 때에는 이로부터 각각의 object를 layer표현으로 구성하여 단 몇장으로 부터 overlapping시킴으로써 여러 프레임의 sequence를 복원해 낼 수 있다. 즉, 이미지의 압축측면에서 매우 큰 효과를 가져 올 수 있다.

2. Motion segmentation

2.1. 전체 블록도

Image sequence의 segmentation을 위해 먼저 움직임추정을 수행하고 이를 여러 가지 motion에 대해 표현할 수 있는 affine model motion으로 mapping시킨다. 이

model에 대해 유사한 움직임을 가지는, 즉, 같은 parameter를 가지는 영역을 clustering시켜 줌으로써 각각의 object영역을 얻어낸다. 이때, affine model을 estimation하기 위해 초기에는 영역을 우리가 임의로 가정하여 픽셀들을 clustering하고 이를 검토함으로써 영역이 알맞게 할당되어 질때까지 위의 작업을 반복해서 영역을 할당한다. 그리고, 그 다음 프레임에 대한 작업은 앞의 결과를 초기치로 해서 이 작업을 반복하게 된다. 그림 1은 전체 알고리즘에 대한 그림이다.

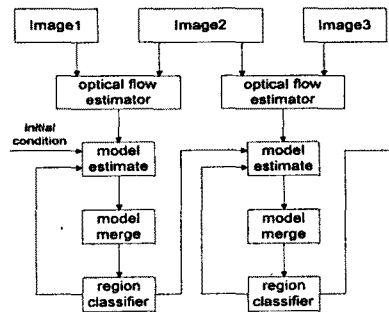


그림 1. motion segmentation의 알고리즘 블록선도

2.2. Motion estimation : optical flow estimation

다양한 움직임을 갖는 object가 존재하는 image에 대해서 배경으로부터 각각의 다양한 움직임을 갖는 영역을 추출해 내기 위해서, 움직임추정시 반복적인 알고리즘을 쓴다. 그리고 같은 움직임을 갖는 영역에 대해서는 하나의 움직임 vector를 부여한다. 그리고 일단, 한 영역에 대한 움직임벡터가 주어지면 다음 차례의 추정에서는 전체의 image에서 이 영역을 뺀 나머지 부분에 대해서만 추정을 하게 된다. 즉, 기존의 다양한 움직임추정 방법보다는 픽셀단위로 움직임을 찾아내어 움직임벡터를 구하는 optical flow estimation방법이 다양한 움직임으로부터 추출해 낸 영역이 서로 섞이는 것을 막아 줌으로써 segmentation을 수행하는데 적합하다. 일단, optical flow estimation방법으로 구한 움직임으로 image의 여러 영역에 대한 motion을 나타내는 affine model을 결정한다. 즉, optical flow estimation방법으로 구한 움직임벡터를 affine parameter와 matching시켜야 한다.

●Optical flow estimation methods
 시간 t와 t+1 프레임에 대해서 t+1 프레임은 t 프레임에 대해 아래와 같이 표현된다.

$$S_{t+1} = S_t(x - V_x(x, y), y - V_y(x, y)), \quad (1)$$

여기서, 우리는 velocity vector $V_x(x, y)$ 와 $V_y(x, y)$ 를 구하면 $t+1$ 프레임은 t 프레임에서 쉽게 얻을 수 있는데 이는 optical flow equation을 이용하여 구할 수 있다. optical flow equation은 식(2)와 같이 표현 된다.

$$\frac{dS(x, y, t)}{dt} = 0 \quad (2)$$

여기서 S 는 각 픽셀의 intensity를 나타낸다. 또 식(2)는 시간의 변화에 따른 image-plane 상의 각 pixel의 intensity 변화율을 나타낸다. 위 식(2)로부터 우리는 displacement vector를 구할 수 있다. 식(2)를 다시 표현하면,

$$\frac{\partial S(x, y, t)}{\partial x} V_x(x, y, t) + \frac{\partial S(x, y, t)}{\partial y} V_y(x, y, t) + \frac{\partial S(x, y, t)}{\partial t} = 0 \quad (3)$$

이 된다. 여기서, $V_x(x, y, t)$ 와 $V_y(x, y, t)$ 를 구하면 되는데 여기에는 한 식에 미지수가 두 개 존재하므로 바로 결정 되어 질 수 없다. 그래서 displacement vector를 구하기 위해서 (3)식을 다시 한번 미분 하여 얻어 낸다. 이를 $V_x(x, y, t)$ 와 $V_y(x, y, t)$ 로 표현 하면,

$$\begin{bmatrix} V_x(x, y, t) \\ V_y(x, y, t) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 S(x, y, t)}{\partial x^2} & \frac{\partial^2 S(x, y, t)}{\partial y \partial x} \\ \frac{\partial^2 S(x, y, t)}{\partial x \partial y} & \frac{\partial^2 S(x, y, t)}{\partial y^2} \end{bmatrix}^{-1} \begin{bmatrix} -\frac{\partial S(x, y, t)}{\partial t} \\ -\frac{\partial S(x, y, t)}{\partial t} \end{bmatrix} \quad (4)$$

가 된다. 이젠, 식(4)에서 두 번 편미분된 값만 구하면 우리가 구하고자 하는 displacement vector를 구할 수 있다. 이는 Lucas-Kanade에 의해 제안된 방식을 이용하여 얻는데, 이 방식은 임의의 블록R에서는 모든 픽셀이 같은 움직임벡터를 가지고 있다는 가정에 근거한다. 그리고, 계산이 비교적 간단한 finite difference방법과 polynomial fitting방법을 이용하여 위의 미분 값을 구한다. Finite difference방법은 전 프레임과 현재 프레임 사이의 이웃하는 pixel 들의 차분 값으로 위의 미분값을 근사화해서 구한 것이다. Polynomial fitting방법은 image픽셀세기를 기본다항식을 이용하여 근사화한 다음 위의 미분값을 구한다.

2.3. Model motion의 가정

Motion data의 segmentation은 영역 분류를 위한 model motion을 가정함으로써 출발 한다. 만약에, motion model을 알고 있다면 쉽게 영역을 분류 할 수 있지만 그렇지 않은 경우에는 segmentation의 기초가 되는 가장 중요한 작업이다. 보통 처음으로 두 프레임 사이에 대해 영역을 분류해 낼 때는 이를 알 수가 없는데, 이때 model motion가정이 필요하다. 첫번째로, 가능한 모든 움직임에 대한 가정을 할 수 있는데 이는 너무 복잡하고 시간이 많이 소요 되므로 image data의 sampling에 의해 나타날 수 있는 움직임에 대해 서만 서술한다. 일단 image를 여러개의 조각으로 나누고 model estimator를 이용하여 이들 영역에 대한 model parameter값을 계산한다. 결과로써 각 영역에 대한 motion의 가정이 이루어진다. 또, 주어진 motion data로부터 affine parameter값들이 추정된다. Affine model이란 translation, rotation, zooming, shear등을 나타낼 수 있는 parameter를 가진 model이다. 다음 식(5)는 각각의 displacement vector에 대한 affine mapping식이다.

$$\begin{aligned} V_x(x, y) &= a_{x0} + a_{xx}x + a_{xy}y = \Phi^T a_{xi} \\ V_y(x, y) &= a_{y0} + a_{yx}x + a_{yy}y = \Phi^T a_{yi} \end{aligned} \quad (5)$$

여기서, $\Phi^T = [1 \ x \ y]$ 이다. 따라서, 각 영역의 motion은 이들 6개의 parameter로 나타낼 수 있다. 식(5)는 다시 $\Phi^T = [1 \ x \ y]$ 를 이용하여 식(6)과 같이 표현된다.

$$[a_{xi} \ a_{yi}] = [\sum \Phi \Phi^T]^{-1} \sum \Phi [V_x(x, y) \ V_y(x, y)] \quad (6)$$

우리가 분석하려는 초기 영역크기는 안정한 model estimation을 위해 가능한 한 작게 설정하는 것이 좋다.

2.4. Model clustering

Model estimation이 끝나면 model merge가 유사한 움직임 갖는 영역들을 조합하여 region을 할당하는 단계에서 계산을 줄이기 위해 하나의 표현으로 나타낸다. 초기의 영역 크기에 상관 없이 그 영역이 image의 boundary를 포함하므로 affine model의 가정이 잘 못 설정 될 수 있다. 이것을 해결하기 위해 residual error값을 넘어가는 pixel값들은 그 영역에서 버리고 분석을 하게 된다. Residual error는 다음 식(7)과 같다.

$$\sigma_i^2 = \frac{1}{N_i} \sum (V(x, y) - V_{ai}(x, y))^2 \quad (7)$$

여기서, N_i 는 우리가 분석하고자 하는 영역의 pixel의 갯수이다. 유사한 움직임을 가지고 있는 object들은 affine motion parameter space에서 clustering알고리즘에 의해 그룹지어진다. 그리고, 각각의 그룹에 대해 표현 할 수 있는 model을 유도해 낸다. Model clustering알고리즘으로는 다음 식(8)과 같이 k-means알고리즘을 쓴다.

$$\begin{aligned} D_m(a_1, a_2) &= [(a_1 - a_2)^T M (a_1 - a_2)]^2 \\ M &= \text{diag}(1 \ \text{dim} \ \text{dim} \ 1 \ \text{dim} \ \text{dim}) \end{aligned} \quad (8)$$

여기서, dim은 image의 dimension이다. 식(8)에서 $D_m(a_1, a_2)$ 은 parameter space에서 다른 component들의 거리를 나타내는 scale factor이다. Scale factor는 image의 경계에서 임의의 단위 displacement에 대응하는 parameter space상에서의 unit distance로 선택한다. 위의 알고리즘에서 cluster중심은 초기값으로 주어진다. 각각의 affine model의 가정은 가장 가까운 clustering중심에 할당 되어진다. 할당 후 clustering중심은 새로 바뀌어진다. 반복적으로 clustering의 중심은 더 이상 중심이 바뀌지 않을 때 까지 수정되어진다. 그런데, 두 cluster를 merging할 때 두 cluster사이의 거리가 미리 정해 놓은 거리보다 적을때에는 두 개를 합쳐주고 그렇지 않을 때는 각각 다른 하나의 cluster로 남겨진다. 이 k-means알고리즘은 예측을 최소로 줄이면서 적은 cluster갯수로 image를 표현 하려는 데 목적이 있다.

2.5. Motion 가정의 test와 영역 할당

이 작업은 우리가 가정했던 model을 테스트하고 그 model에 대응 하는 motion영역을 다시 결정, 할당해 주는 단계이다. 영역을 할당 할 때 우리는 다음 함수 $G(i(x, y))$ 를 최소로 하는 image를 가진 영역들로 할당한다. 여기서 $G(i(x, y))$ 는 cost/distortion 함수이다.

$$G(i(x, y)) = \sum (V(x, y) - V_{ai}(x, y))^2 \quad (9)$$

여기서 $i(x, y)$ 는 (x, y) 좌표에 할당되어진 model을 나타내고 $V(x, y)$ 는 optical flow estimation에 의한 motion field를 나타내고, $V_{ai}(x, y)$ 는 i 번째 affine motion을 가정

에 대응하는 affine motion이다. 위 식에서 $G(i(x, y))$ 는 affine parameter가 영역의 motion을 정확하게 서술 할 때 가장 작은 값인 0을 갖게 된다. 그러나, noise와 같은 신호가 섞였을 때에는 그렇지 않다. 이 경우에는 error를 가장 작게 하는 값을 취하게 된다. 이는 각 위치에서의 다음 함수 $[V(x, y) - V_a(x, y)]^2$ 를 최소로 하는 값을 취하게 된다. 그러나, optical flow estimation 방법에 의한 모든 픽셀들이 영역을 할당 받는 것이 아니다. 왜냐하면, optical flow estimation에 의한 방법은 object의 경계부분에서 affine model을 사용하여 정확하게 표현 할 수 없기 때문이다. 이런 경계부분을 포함하는 영역은 제대로 할당되어 지지 않는다. 그래서, 이런 부분에 대해서는 어떤 threshold 값을 주어 영역을 다시 할당해 주어야 한다. 이 threshold 값은 실험적으로 얻어 사용한다. 이와 같이, image sequence의 segmentation은 여러 단계의 과정을 거쳐 이루어 지게 된다. 그러므로, 계산 측면에서 굉장한 부하가 생기게 되는데 이는 위의 반복적 알고리즘에서의 반복횟수를 줄이고 초기치를 적절하게 부여함으로써 해결 할 수 있다.

3. The layered representation

서로 다른 프레임에서 대응하는 영역은 단지 다른 affine parameter값을 갖게 된다. 원래의 image의 모든 프레임에 transform들을 적용함으로써 다른 프레임 에서 그에 상응하는 영역은 그영역의 motion에 의해 표현 될 수 있다. 그러므로, 한 영역의 motion이 정확하게 얻어지면 다른 프레임에서 그 영역에 대한 image를 얻어낼 수 있다. 위의 segmentation에서 얻어낸 각 영역에 대해서 각각 그 영역의 속도와 intensity성분을 뽑아 내면 이 성분 만으로 여러 장의 image를 복원 해 낼 수 있다. 만약에 1초에 30프레임을 갖는 image sequence가 있다면 대략 3프레임의 image로 부터 30프레임 전체의 image를 복원 할 수 있는데, image sequence압축에 상당히 뛰어난 효과를 가져 올 수 있다. 다음 그림은 affine model을 적용하기 위한 optical flow를 이용한 움직임벡터를 나타낸 것이다. 실험 image sequence로는 ping pong sequence와 mobile and calendar sequence를 사용하였다.



그림 1(a) finite difference방식



그림 1(b) 복원된 영상



그림 2(a)polynomial fitting 방식



그림 2(b) 복원된 영상



그림 3(a) finite difference방식



그림 3(b) 복원된 영상

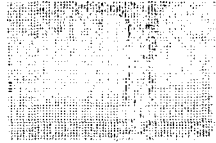


그림 4(a)polynomial fitting방식



그림 4(b) 복원된 영상

4. 결론

우리는 image sequence의 segmentation 구현해 보았는데 이 작업은 임의로 우리가 초기치를 주어 결과를 얻어내는 반복적 알고리즘을 사용하여 여러 가지 움직임을 갖는 object를 표현 하였다. 위 그림에서 알 수 있는 바와 같이 ping pong sequence에 대해서는 아주 정확한 움직임벡터를 구하는 것을 알 수 있다. 일반적으로 많이 쓰이는 block matching 방법과 비교해보면, 경계부분에서의 블록화 현상이 두드러지게 줄어들음을 볼 수 있다. 그러나, mobile and calendar sequence에 대해서는 그다지 정확하게 움직임벡터를 추정하지 못하는데, 전처리과정을 거쳐 smoothing을 수행한 후 움직임벡터를 추정하여야 하겠다. Optical flow를 이용하여 구한 움직임벡터로 부터 affine motion model을 적용하여 반복적으로 object를 segmentation하는데, 반복 횟수가 늘어감에 따라 object에 대한 윤곽이 들어나게 된다. 그러나, 특히, 이미지의 경계부분에 대한 처리가 큰 문제가 되는데, merging 알고리즘을 이용하여 각각의 object의 경계부분을 처리하였다. 앞으로 segmentation을 이용하여 image sequence를 압축에 응용하게 되면 큰 효과가 있을것으로 기대된다. 그리고, 계산시간이 오래걸리는 점과, image내의 object가 mobile and calendar sequence와 같이 아주 복잡하게 있을 경우에는 clustering하는데 어려움이 있었다. 특히, occlusion 경계부분과 같은 경우에 문제점이 많았다.

5. 추후 과제

본 논문에서는 video data에 대한 기존의 segmentation 알고리즘을 구현해 보았는데 이 과정에서 각 단계마다 새로운 알고리즘을 추가해 보고 그 효과를 이 알고리즘과 비교 검토해 보면 좋은 결과를 얻을 수 있을 것이다. 또한 앞에서도 언급한 바와 같이 segmentation을 이용하여 각각의 object를 layer 표현으로 이미지를 압축하는 알고리즘을 구현 하면 더욱더 효과적인 이미지 코딩을 행할 수 있을 것이다. 그리고, occlusion경계부분에 대한 개선이 있어야 할 것으로 본다.

참고 문헌

- [1]A. murat tekalp, "Digital video processing," Prentice Hall Signal Processing Series, 1995
- [2]Tohn Y. A Wang and Edward H. Adelson, "Spatio-Temporal Segmentation of Video Data," M.I.T. Media Laboratory Vision and Modeling Group, Technical Report No.262, February 1994
- [3]G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," IEEE Trans. Pattern Anal. Mach. Intell., vol.7, pp384-401, 1985
- [4]M. M. chang, M. I. Sezan, and A.M.Tekalp, "An algorithm for simultaneous motion estimation and scene segmentation," Proc. Int. Conf. ASSP, Adelaide, Australia, April 1994
- [5]H. Nicolas and C. Labit, "Region-based motion estimation using deterministic relaxation schemes for image sequence coding," IEEE ICASSP, volume 3, pages 265-268, San Francisco, California, April 1992