

## 가우시안 분포를 기반으로 한 얼굴 추적

박순영<sup>o</sup> 송영섭 김항준

경북대학교 컴퓨터공학과 인공지능연구소

{syPark<sup>o</sup>, yssong, hjkim}@ailab.knu.ac.kr

### Gaussian Distribution-Based Face Tracking

Soon Young Park<sup>o</sup>, Young Sub Song, Hang Joon Kim

Dept. of Computer Engineering, Kyungpook National University

#### 요 약

본 논문에서는 연속 영상에서 가우시안 분포를 사용하여 사람의 얼굴을 추적하는 방법을 제안한다. 영상은 여러 개의 동일한 영역들로 이루어지고, 이 영역들 중 얼굴 영역이 있다고 가정하였다. 영상에 있는 모든 영역들을 가우시안 분포로 표현하였으며, 이들의 집합을 가우시안 분포의 혼합 모델로 표현하였다. 제안된 방법에서는 이전 프레임에서 가우시안 분포들을 찾고, 찾아진 이전 프레임의 가우시안 분포들을 이용하여 현재 프레임의 영역들을 찾는다. 이 영역들 중, 초기에 주어진 얼굴 영역이 있으며, 현재 프레임의 영역들에 의해 가우시안 분포는 갱신되고 이 과정을 반복함으로써 얼굴을 추적한다. 가우시안 분포의 개수를 다양하게 변화시켜 실험을 하였고, 이를 통해 가우시안 분포의 혼합 모델로 얼굴을 추적할 수 있음을 보였다.

#### 1. 서 론

비디오 안에서 물체를 검출하고 이를 추적하는 것은 컴퓨터 비전 분야에서 중요한 문제로 대두되었다. 그리고 지난 몇 년 동안 많은 연구자들은 물체를 더욱 빠르고 정확하게 추적하는 방법에 대해서 연구해왔다. 특히 얼굴을 따라다니는 일은 HCI(Human Computer Interaction), 스마트 홈, 보안, 전자 회의, 화상 채팅 등 여러 분야에 사용할 수 있어 더욱 흥미 있는 연구 주제이다.

얼굴 추적의 목적은 비디오 내에서 얼굴들을 따라다니는 것이다. 이를 위해 시간이 흐름에 따라서 얼굴의 위치가 어떻게 변하는가에 대한 연구를 많이 하고 있다. 지금까지 제안된 물체를 따라다니는 방법은 크게 영역기반, 윤곽선 기반, 망사 기반 등으로 나눌 수 있다[1]. 첫 번째 방법인 영역 기반은 사용자가 미리 추적하고자 하는 물체를 지정한다. 그 후, 연속적인 프레임 시퀀스에서 매 프레임이 입력될 때마다 프레임들을 분할한다. 그리고 이전 프레임과 현재 프레임의 각 영역들의 짝을 맞추므로써 지정한 물체들을 추적한다[2]. 윤곽선 기반은 추적하고자 하는 물체 내의 모든 정보를 사용하는 것이 아니라 물체의 외곽선 쪽의 경계선 부근에 대한 정보만을 사용한다[3]. 마지막으로 망사 기반은 물체를 추적하는 방법은 추적하고자 하는 물체의 내부에 대한 정보와 경계선에 대한 정보를 모두 이용하여 물체를 추적한다[4]. 위에서 설명한 물체 추적 방법들은 주로 추적하고자 하는 물체에 대한 정보만 이용하여 물체를 추적한다. 그러나 본 논문은 추적하고자 하는 물체뿐만 아니라 그 이외의 정보도 함께 이용하여 물체를 추적한다. 즉, 우리는 추적하고자 하는 물체와 이미지 내에 존재하는 다른 물체들과의 관계에 대한 정보도 이용한다. 이를 위해서 이전에 이미지를 분할하는데 사용되었던 문맥 기반 접근 방법을 이용하여 얼굴을 추적한다[5].

본 논문은 문맥 기반 접근 방법으로 프레임 내에 있는 하나의 얼굴을 추적한다. 입력 프레임은 여러 개의 영역들로 이루어져 있고 이들은 가우시안 분포를 이용하여 표현한다. 각 가우시안 분포는 얼굴 같은 정도를 가진다. 그리고 이 가우시안 분포들이 표현한 영역들의 위치를 이용하여 현재 프레임에서의 얼굴 위치를 알 수 있다. 따라서 본 논문에서 제안한 방법은 다음 두 단계를 거쳐서 얼굴을 추적한다. 먼저 이전 프레임의 가우시안 분포들을 찾고, 이 가우시안 분포들을 이용하여 현재 프레임의 얼굴의 위치를 찾

는다.

얼굴 추적을 위한 본 논문의 구성은 다음과 같다. 2절에서는 입력 영상을 가우시안 분포를 이용하여 모델링하고 3절에서는 모델링된 가우시안 분포들을 이용하여 얼굴을 추적하는 방법을 제안한다. 4절에서는 제안한 방법을 이용하여 여러 입력 영상에 있는 얼굴을 추적한 실험 결과를 보이고 5절에서는 결론을 보인다.

#### 2. 가우시안 분포를 이용한 이미지 모델링

본 논문에서는 프레임은 여러 개의 영역들의 집합이고 얼굴은 이 여러 개의 영역들 중 몇 개의 영역이라고 본다. 프레임 내에 존재하는 각 영역들은 하나의 가우시안 분포로 표현한다. 이 때 각 가우시안 분포는 얼굴에 해당하는지에 대한 라벨을 가진다. 따라서 모든 영역들의 집합인 이미지는 MoG(Mixture of Gaussian)로 표현한다.

영역을 표현한 가우시안 분포는 위치 정보와 색깔 정보를 가지는 특징 공간 내에 존재하며 각 가우시안 분포는 특징 벡터들의 밀도를 나타낸다. 프레임에 있는 하나의 픽셀은 특징 공간에 있는 하나의 특징 벡터에 매핑 된다. 그러므로 특징 벡터들은 프레임 내에 있는 픽셀의 정보를 표현한다. 하나의 특징 벡터는 픽셀의 위치 정보와 색깔 정보로 이루어져 있다. 특징 벡터의 위치 정보는 특징 벡터가 표현하는 픽셀의 프레임 내에서  $(x, y)$  좌표를 가지며 색깔 정보는 픽셀의 3차원 색 공간 안에서 추출한 색을 가진다. 본 논문에서는  $(r, g, b)$ 의 색 공간을 사용한다. 그러므로 프레임 내에서 각 픽셀은 하나의 5차원 특징 벡터  $[p, p, p, p, p]$ 로 표현한다.

5차원 특징 공간 내에 있는 특징 벡터들을 그룹화 시킴으로써 프레임 내에 있는 픽셀들을 여러 개의 영역으로 그룹화 한다. 따라서 프레임 내에서 위치와 색깔이 비슷한 픽셀들이 하나의 영역에 속하게 된다. 이 영역은 5차원 특징 공간의 가우시안 분포로 표현한다. 즉, 하나의 영역은  $N(\mu, \Sigma)$ 로 표현한다. 각 특징 벡터는 여러 개의 가우시안 분포들 중에서 그 특징 벡터가 속할 가능성이 제일 큰 가우시안 분포에 지정된다. 즉, 하나의 특징 벡터는 유사도가 가장 큰 가우시안 분포에 지정된다. 유사도는 특징 벡터와 가우시안 분포 사이의 거리가 멀어질수록 낮아진다. 이 때, 거리와 유사도는 정반비례의 관계는 아니며, 유사도를 구할 때 사용

하는 거리는 특징 벡터와 가우시안 분포 사이의 거리이다. 그리고 가우시안 분포는 영역의 표현이기 때문에 특징 벡터와 가우시안 분포 사이의 거리는 유클리디안 거리가 적당하지 않다. 이에 대한 자세한 설명은 3.2절에서 하겠다. 이 방법으로 각 특징 벡터들을 하나의 가우시안 분포에 지정함으로써 프레임 내의 모든 픽셀들은 여러 개의 영역들로 그룹화 한다.

그림 1은 주어진 입력 영상을 가우시안 분포들을 이용하여 모델링한 예를 보인다. 그림은 혼합된 가우시안 분포를 여러 개의 타원들로 보인다. 각 타원은 프레임 내의 특징 가우시안 분포의 위치와 평균 색깔을 나타낸다. 그림 1.b의 MoG 모델을 이용하여 그림 1.c와 같이 입력 영상을 여러 개의 영역들로 분할한다. 영역들은 각 픽셀을 하나의 가우시안 분포에 지정함으로써 구분한다.



그림 1. (a) 입력 영상. (b) MoG를 이용한 영상 모델링. (c) 모델을 이용한 프레임의 영역 구분.

### 3. 얼굴 추적 방법

본 논문은 매 프레임마다 얼굴인 영역을 확인함으로써 얼굴을 추적한다. 얼굴인 영역은 프레임에서 가우시안 분포의 얼굴 같은 정도를 보고 구할 수 있다. 즉, 프레임을 모델링 한 MoG를 보고 프레임 내에서의 얼굴인 영역을 확인하여 얼굴을 찾는다. 그러나 실제로 프레임의 MoG는 구하는 것은 어렵기 때문에 그 프레임의 MoG와 비슷한 MoG를 이용한다. 비디오 내에서 얼굴을 추적할 경우, 연속된 프레임 사이에는 프레임 내의 상태 변화가 매우 적다. 따라서 본 논문에서는 현재 프레임의 MoG와 비슷한 유사 MoG로 이전 프레임의 MoG를 사용한다. MoG 모델을 사용하여 얼굴인 영역을 추적하기 위해서는 다음 두 단계를 거친다. 먼저 이전 프레임의 MoG를 구한다. 그리고 구한 MoG를 이용하여 프레임의 얼굴 영역을 확인하여 얼굴을 추적한다. 이에 대한 자세한 과정은 다음과 같다.

#### 3.1 MoG 구하기

하나의 프레임과 그 프레임에 해당하는 분할된 영역이 주어지면 프레임의 MoG를 구할 수 있다. 프레임의 MoG는 프레임에 대한 영역들의 가우시안 분포들을 구함으로써 찾는다. 각각 각각의 영역들에 대해서 다음과 같은 방법으로 가우시안 분포를 구한다. 먼저 각 영역에 속하는 픽셀들을 5차원 특징 공간의 특징 벡터들로 매핑 시키고, 이들에 대한 평균과 분산을 구한다. 프레임 내의 모든 영역을 가우시안 분포로 표현하면, 프레임은 MoG로 표현된다. 각 영역에 대한 가우시안 분포는 다음과 같이 구한다.

프레임 내에 전체 k개의 영역이 있고, n개의 픽셀이 있을 때, 영역 R<sub>i</sub>에 해당하는 가우시안 분포는 다음과 같다.

$$\mu_i = \frac{\sum_{r=1}^n w_{ri} \mathbf{p}_r}{n_i},$$

$$\Sigma_i = \frac{\sum_{r=1}^n w_{ri} (\mathbf{p}_r - \mu_i)^T \mathbf{I} (\mathbf{p}_r - \mu_i)}{n_i} \quad (1)$$

이 때,

$$w_{ri} = \begin{cases} 1, & \mathbf{p}_r \in R_i \\ 0, & \text{others} \end{cases}$$

이고,

$$n_i = \sum_{r=1}^n w_{ri}$$

이다.

즉, w<sub>ri</sub>는 i번째 픽셀이 i번째 영역에 속하는지에 대한 여부를 나타내며, n<sub>i</sub>는 i번째 영역에 속하는 픽셀의 개수이다. 만약 i번째 영역에 속하는 픽셀의 개수 n<sub>i</sub>가 기준치 이하이면, 그 가우시안 분포는 제거한다.

위의 방법으로 프레임에 있는 모든 영역들에 대해 가우시안 분포를 구한다. 이 때, 각 영역에 대한 가우시안 분포는 독립적으로 구한다. 프레임에 있는 모든 영역에 대해 가우시안 분포를 구하여 프레임의 MoG 모델을 찾는다.

#### 3.2 얼굴 영역 찾기

프레임에서 얼굴은 프레임 안에 있는 얼굴인 영역을 확인함으로써 찾는다. 즉, 프레임을 나타내는 영역들의 집합에서 얼굴에 해당하는 영역이 얼굴이다. 프레임 안에 있는 각 영역은 프레임과 그 프레임을 모델링 한 MoG를 이용하여 구한다. 그러나 현재 프레임의 MoG는 구하기가 어렵기 때문에 현재 프레임과 비슷한 MoG를 사용한다. 즉, 현재 프레임과 유사 MoG를 이용하여 현재 프레임의 각 영역들을 구한다. 그리고 이 영역들 중 얼굴인 영역들을 확인한다.

프레임의 영역들은 다음과 같이 구한다. 프레임에 있는 모든 픽셀에 해당하는 특징 벡터들을 주어진 MoG 중 하나의 가우시안 분포에 지정한다. 이 때, 각 특징 벡터는 유사도가 가장 큰 가우시안 분포에 지정된다. 유사도는 특징 벡터와 가우시안 분포 사이의 거리를 이용하여 구한다. 유사도를 구할 때 사용되는 거리는 평균 유클리디안 거리에 대한 특징 벡터와 가우시안의 평균 위치 사이의 유클리디안 거리의 비율을 이용한다. 즉, 하나의 픽셀과 하나의 가우시안 분포 사이의 유사도는 다음과 같이 정의한다[6]. 프레임이 d차원의 가우시안 분포 k개로 이루어졌을 때, 하나의 픽셀 **p**와 i번째 가우시안 분포의 유사도는

$$f_i(\mathbf{p}|\alpha_i, \mu_i, \Sigma_i) = \alpha_i \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left\{-\frac{1}{2}(\mathbf{p} - \mu_i)^T \Sigma_i^{-1} (\mathbf{p} - \mu_i)\right\} \quad (2)$$

이다. 이 때, α<sub>i</sub>는 i번째 가우시안 분포가 MoG 중에서 차지하는 비중이다. 그리고 다음과 같이 각 픽셀은 k개의 가우시안 분포들 중에서 유사도가 가장 큰 값을 가지는 가우시안 분포에 속한다.

$$\text{Label}(\mathbf{p}) = \arg \max_i f_i(\mathbf{p}|\alpha_i, \mu_i, \Sigma_i) \quad (3)$$

이 경우, 어느 한 픽셀이 i번째 가우시안 분포에 속할 확률은 다음과 같다.

$$p(\text{Label}(\mathbf{p}) = i) = \frac{f_i(\mathbf{p}|\alpha_i, \mu_i, \Sigma_i)}{\sum_{j=1}^k f_j(\mathbf{p}|\alpha_j, \mu_j, \Sigma_j)} \quad (4)$$

이 과정을 프레임에 있는 모든 픽셀들에 대해서 수행한다. 즉, 프레임의 각 픽셀들은 하나의 가우시안 분포에 속하게 되고, 그 결과 프레임의 영역들을 구한다. 이 때, 얼굴에 해당하는 가우시안 분포에 지정된 픽셀들로 이루어진 영역들이 얼굴이다.

4. 실험 및 결과

본 논문에서 제시한 얼굴 추적 방법의 성능을 평가하기 위하여 다음과 같이 실험하였다. 실험에 사용된 PC 사양은 Pentium 4의 2.4GHz이며, Window XP 운영체제였다. 제안한 얼굴 추적 방법은 Visual C++를 사용하여 구현하였으며 실험 영상의 입출력을 위해서 OpenCV에서 제공하는 함수들을 사용하였다. 실험에 사용된 영상은 컴퓨터 채팅 중인 사람의 얼굴을 사용하였다. 영상은 삼성사의 anycam 카메라를 이용하여 320×240 크기의 24비트 영상을 초당 15프레임으로 입력 받았다.

이와 같은 실험 환경으로 각 실험 영상에 대해 얼굴을 추적한 결과는 그림 3과 같았다. 실험은 입력 영상을 모델링하는 가우시안 분포의 개수를 2개, 3개, 5개로 다르게 하여 얼굴을 추적하였다.

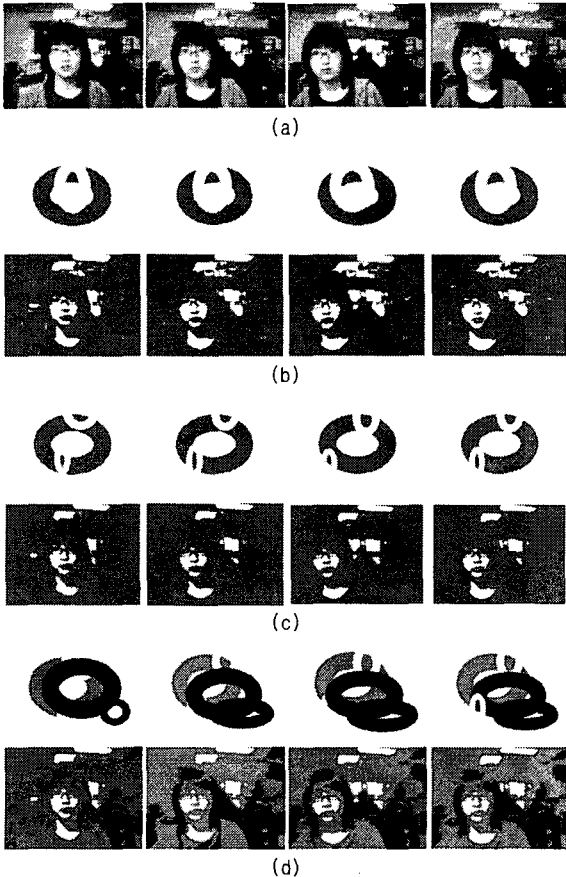


그림 2. 실험 결과  
(a) 입력 영상의 프레임 (17, 37, 57, 77번 프레임) (b) 얼굴 추적 결과(k=2) (c) 얼굴 추적 결과(k=3) (d) 얼굴 추적 결과(k=5)

위의 실험 영상에 대한 실험 결과로는, 프레임을 모델링하는 가우시안 분포의 개수가 많아질수록 얼굴을 잘 추적하였다. 이는 다음 두 가지 결과를 보고 알 수 있었다. 첫 번째로는 얼굴에 해당하는 가우시안 분포의 움직임이 영상 내에서 실제 얼굴의 움직임과 더 유사하였다. 그림 2.b에서는 얼굴에 해당하는 가우시안 분포가 실제와 다르게 움직임이 거의 없었다. 이에

비해서 그림 2.d에서는 얼굴에 해당하는 가우시안 분포의 움직임이 실제 얼굴의 움직임과 유사하였다. 두 번째로는 얼굴이 아닌 부분을 얼굴이라고 잘못 판단하는 픽셀의 개수가 줄었다. 그림 2.b를 보면 얼굴 오른쪽 위의 밝은 부분에 위치하는 픽셀들이 얼굴 영역이라 지정되어있다. 그러나 그림 2.c의 실험에서는 그 밝은 부분을 하나의 배경 영역이라 추가하였고 이를 통해 실제 배경인 픽셀을 얼굴 영역이라 판단하는 개수가 줄어들었다.

5. 결론

얼굴 추적은 HCI(Human Computer Interaction) 등에서 사용되는 중요한 문제 중에 하나이다. 본 논문에서는 이전에 영상 분할에 사용되었던 방법을 사용하여 얼굴을 추적하는 방법을 제시하였다. 제안한 방법은 입력 영상을 여러 개의 영역들의 집합으로 보고 이를 MoG(Mixture of Gaussian)로 표현하여 얼굴을 추적하였다. 얼굴을 추적하기 위해서 매 프레임마다 얼굴에 해당하는 영역을 확인하였다. 이 때, 얼굴 영역들은 사전에 구한 이전 프레임의 MoG와 현재 프레임을 이용하여 구했다. 제안한 방법의 얼굴 추적 효과를 확인하기 위해서 채팅을 하고 있는 사용자의 얼굴을 추적하여 보았다. 실험 결과, 가우시안 분포의 혼합 모델로 얼굴을 추적할 수 있음을 보였다. 이 때, 가우시안 분포의 개수가 많아질수록 노이즈가 줄어들어 얼굴을 잘 추적하였다. 그러나 얼굴인 부분을 얼굴 영역이 아니라고 판단하는 문제점이 있었다. 이는 얼굴을 하나의 영역으로 가정하였기 때문으로 판단된다. 따라서 얼굴을 여러 개의 영역으로 확장할 필요가 있다.

참고 문헌

[1] Karthik Hariharakrishnan and Dan Schonfeld, "Fast Object Tracking Using Adaptive Block Matching," IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 7, NO. 5, 2005  
 [2] MICHAEL J. BLACK and ALLAN D. JEPSON, "Eigenttracking: robust matching and tracking of articulated objects using a view-based representation," Int. J. Comput.Vis., vol. 26, no. 1, pp. 63-84, 1998.  
 [3] Yue Fu, A. Tanju Erdem, and A. Murat Tekalp, "Tracking visible boundary of objects using occlusion adaptive motion snake," IEEE Trans. Image Process., vol. 9, no. 12, pp. 2051-2060, 2000.  
 [4] Yucef Altunbasak and A. Murat Tekalp, "Occlusion-adaptive, content-based mesh design and forward tracking," IEEE Trans. Image Process., vol. 6, no. 9, pp. 1270-1280, 1997.  
 [5] Jacob Goldberger, Hayit Greenspan, "Context-Based Segmentation of Image Sequences" IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 28, NO. 3, 2006.  
 [6] Hayit Greenspan, Jacob Goldberger, "Context Dependent Image Segmentation and Image Matching via EMD Flow," J. Computer Vision and Image Understanding, vol. 93, no. 1, pp 86-109, 2004.