

## Lucas-Kanade 특징 추출을 이용한 실시간 얼굴 추적

김기상<sup>0</sup> 김계영<sup>1</sup> 최형일<sup>0</sup>

<sup>0</sup>송실대학교 일반대학원 미디어학과

<sup>1</sup>송실대학교 일반대학원 컴퓨터학과

illusion1004@ssu.ac.kr, gykim11@ssu.ac.kr, hic@ssu.ac.kr

### Face tracking in real-time using Lucas-Kanade feature tracker

Ki-Sang Kim<sup>0</sup> Gye-Young Kim<sup>1</sup> Hyung-Il Choi<sup>0</sup>

<sup>0</sup>School of Media, Soongsil University, Seoul, Korea

<sup>0</sup>School of Computing, Soongsil University, Seoul, Korea

본 논문에서는 특징기반 방법에 기초하여 실시간으로 회전에 강건하게 얼굴 영역 검출 및 추적을 하였다. 먼저, 회전에 강건한 얼굴 영역의 특징을 검출하기 위해 KLT(Kanade-Lucas-Tomasi)를 이용하여 특징점을 추출한다. 그리고 추출된 특징 점들과 Lucas-Kanade 특징 추적을 이용하여 실시간으로 얼굴 영역을 추적한다.

배경으로부터 얼굴 영역을 추출한 후, 얼굴 영역을 추적하기 위해 사용되는 특징을 추출하여야 한다. 얼굴 영역 추적에 사용되는 특징으로는 점, 모양, 질감 등이 있다. 본 논문에서는 특징을 추출하기 위해 회전에 강건한 KLT 알고리즘을 사용하여 특징 점을 추출하였다. KLT 알고리즘은 일정한 윈도우 안의 영상을 식 (1)과 같이 x축과 y축으로 각각 미분하고, 전치행렬을 식(2)와 같이 곱한다.

$$g = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \nabla I \quad (1)$$

$$gg^T = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \begin{bmatrix} g_x & g_y \end{bmatrix} = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} \quad (2)$$

그리고 윈도우 영역의 모든 행렬값을 식 (3)과 같이 더한다. 매트릭스 Z는 순수한 텍스처 정보를 보유하고 있어, Z의 고유 값을 분석하면 영역 안의 특징을 분류할 수 있다.

$$Z = \int \int_w \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} w dx \quad (3)$$

식(3)에서의 w는 가중치 함수를 나타낸다. 작은 두 개의 고유 값은 윈도우 영역에서 일정한 명암값 패턴을 가지고 있으며, 하나는 작고 다른 하나는 큰 고유 값은 선형 패턴이 포함되었음을 나타낸다. 그리고 두 개의 큰 고유 값은 특징 점으로 큰 명암 값 패턴을 가지고 있다.

그림 1은 KLT 알고리즘을 사용하여 특징점을 추출한 결과를 보여준다.

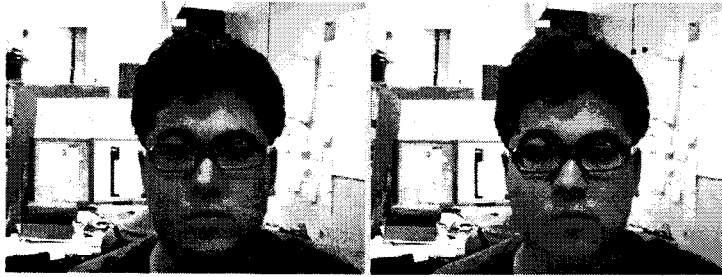


그림 1. 특징 점 추출 결과

다음으로 얼굴 영역의 특징 점을 추적하기 위하여 Lucas-Kanade 특징 추출을 사용하였다. 실시간으로 움직이는 얼굴 모션은 공간 도메인과 시간 도메인에 깊은 관련이 있다. 즉, 실시간으로 움직이는 얼굴은 한 번에 멀리 움직이는 것이 아니라, 여러 프레임에 걸쳐 움직이기 때문에 각 프레임에 따른 얼굴 이동 간격은 조금씩 움직인다. Lucas-Kanade 특징 추출은 광류를 계산할 때, 반복적인 계산을 제거하기 때문에 실시간으로 구현이 가능하다. 두 개의 2차원 이미지  $I$ 와  $J$ 가 있다고 가정하자. 이것을 다르게 표현하면  $I(x) = I(x, y)$ 와  $J(x) = J(x, y)$ 로 표현 될 수 있는데,  $x$ 와  $y$ 축으로 구성된 2차원 좌표인  $x = [x, y]^T$ 로 표현되는 이미지 좌표에서의 색상 값을 갖는다. 이미지  $I$ 는 첫 번째 이미지라 하고, 이미지  $J$ 는 두 번째 이미지라고 참조한다.

첫 번째 이미지  $I$ 의 특징 점을  $u = [u_x, u_y]^T$ 라고 했을 때, 두 번째 이미지  $J$ 의 특징 점  $v = v + d = [u_x + d_x, u_y + d_y]^T$ 을 추적하는 것이 목표이다. 이 두 특징 점  $I(u)$ 와  $J(v)$ 는 비슷한 특징이 있다. 벡터  $d = [d_x, d_y]^T$ 는 이미지를 추적할 물체의 움직인 정도로 특징 점의 움직임 나타낸다. 윈도우의 크기를 정수인  $\omega_x$ 와  $\omega_y$ 로 나타낼 때, 식 (4)와 같이  $\epsilon$ 값을 최소화 한다.

$$\begin{aligned} \epsilon(d) &= \epsilon(d_x, d_y) \\ &= \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} (I(x, y) - J(x + d_x, y + d_y)) \end{aligned} \quad (4)$$

그림 2은 Lucas-Kanade 특징 추적을 이용한 결과를 보여준다.



그림 2. 특징 점 추적 결과

실험 결과를 통하여 얼굴의 특징 점 추출과 추적이 잘 되는 것을 확인하였다. 그러나, 얼굴이 다른 것에 의해 가려지거나, 영상 밖으로 나갔다가 안으로 들어왔을 경우에 대한 얼굴 영역 추출이 향후 연구 과제로 필요하다.