

# 가림현상에 대처한 실시간 이동 물체 추적

김학희\*, 윤한경\*\*

## Handled in real-time tracking of moving object occlusion

Hag-Hee Kim\*, Han-Kyung Yun\*\*

### 요 약

일반적으로 이동 물체 추적은 움직임 및 회전, 크기에 견고한 루카스-카나데 특징 추적 방법이 많이 사용된다. 그러나 이 방법은 추적하던 이동 물체가 배경이나, 다른 물체 등으로 인해 추적하는 물체가 가려질 경우, 이동 물체를 추적하지 못하고 배경이나 다른 물체 등을 추적하거나, 추적이 종료된다. 따라서 본 논문에서는 이러한 문제점을 해결하기 위해 루카스-카나데 특징 추적 방법에 평가 함수와 이동 물체 추적 예측을 도입하여, 가림현상에도 이동 물체를 추적하는 방법을 제안한다.

### ABSTRACT

Generally, moving object tracking used Lucas-Kanade feature tracking method which is strong in movement, rotation and size. But this method is very weak of occlusion by background or another object and so on. In this case, this method tracks backgrounds or another objects instead a moving object, or a tracking is finished. In order to solve this problem, we proposes Lucas-Kanade feature tracking method which introduce a destimation function and prediction function.

**Key Words** : object tracking, optical flow, lucas-kanade, occlusion, harris corner

### 1. 서 론

연속된 비디오 영상에서 움직임과 형태 분석을 통한 2차원 이동 물체 추적의 문제는 많은 응용 분야에서 그 중요성이 증가하고 있다. 특히, 연속된 영상에서의 비정형 객체에 대한 추출과 추적[1]은 비디오 감시 시스템의 가장 중요한 기능이 된다.

이동 물체 검출과 추적 기술에는 많은 방법들이 있다. 단순하지만 많이 사용 하는 방법으로 차영상 추적 알고리즘[4]이 있다. 형태 기반 추적에서는 이

동 물체의 대한 정보를 사전에 학습하여 객체의 윤곽과 가장 유사한 부분을 찾는 방법이 있다. 이러한 방법으로는 윤곽 기반 방법[7]과 상태 공간 표본 추출 접근 방법인 응축 알고리즘[8]등이 알려져 있다. 또한, 특징 기반 추적은 이동 물체의 특징을 추출하여 이 특징을 추적하는 방법으로 Lucas-Kanade 특징 추적 방법[9]이 있다.

기존의 특징점 기반 추적은 가림현상에 대해 미흡하다는 단점을 가지고 있다.

실시간 환경에서 구현이 가능하고 가림현상에

\* 한국기술교육대학교 컴퓨터공학과 박사 (hhkim@kut.ac.kr)

\*\* 교신저자 : 한국기술교육대학교 컴퓨터공학부 교수 (hkyun@kut.ac.kr)

접수일자 : 2011년 08월 09일, 수정일자 : 2011년 08월 26일, 심사완료일자 : 2011년 08월 30일

대해 물체 추적을 위해서, 본 논문에서는 특징점 기반 평가 함수와 예측 기법을 제안한다.

본 논문의 구성은 5장으로 구성된다. 제2장에서는 기존의 이동 물체 검출 방법 및 추적 방법을 소개하고, 제3장에서는 Lucas-Kanade 특징 추적의 문제점을 해결하기 위하여 본 논문에서는 가림현상에 강건하기 위한 평가 함수와 예측 기법을 제안한다. 제4장에서는 본 논문에서 제안한 평가 함수와 예측 기법을 사용한 특징점 추적의 실험 결과에 대해 기술한다. 마지막으로, 제5장에서는 결론 및 향후 연구 과제에 대해 기술한다.

## II. 관련 연구

### 1. 차영상을 이용한 객체 검출 방법

차영상은 두 영상에서 동일한 위치에 해당하는 화소의 픽셀 값의 차이를 의미 한다. 차영상은 식(1)과 같이 표현된다.

$$\delta I(x,y) = |I_1(x,y) - I_2(x,y)| \quad (1)$$

식(1)에서 는 좌표  $x, y$ 에 위치한 화소의 픽셀 값의 차이를 의미하며,  $I_1(x,y)$ 와  $I_2(x,y)$ 는 차영상을 할 두 개의 이미지이다. 단순히 차영상만을 가지고는 객체를 검출할 수 없으므로 이진화를 통하여 객체를 검출한다. 이진화 과정은 식(2) 처럼 표현된다.

$$D(x,y) = \begin{cases} 1, & \delta I(x,y) > Th \\ 0, & otherwise \end{cases} \quad (2)$$

이 방법을 사용하면 배경에 상관없이 객체를 검출할 수 있다. 그러나 특징점 기반 추적에서 특징점을 추출하기 위해서는 정확한 이동 물체 검출이 필요한데 이 방법은, 정확한 이동 물체 검출을 하기 힘들다. 그림 1과 같이 이전 영상에 있었던 객체까지 동시에 검출이 되기 때문이다.



그림 1. 차영상을 이용한 객체 검출  
Fig. 1 Object detection using frame difference



그림 2. 차영상을 이용한 특징점 검출  
Fig. 2 Feature detection using frame difference

여기서 검출된 이동 물체 범위를 이용하여 특징점을 검출하면 그림 2와 같은 결과를 보여준다. 쓸모없는 특징점을 생성으로 인하여 제대로 된 객체 추적을 할 수 없다.

이러한 단점을 극복하기 위해 배경영상을 미리 획득한 후, 현재영상과 차영상을 통해 이동 물체를 감지하는 방법으로 보다 정확한 이동 물체 검출을 하기 위해 그림 3과 같은 방법을 사용한다.

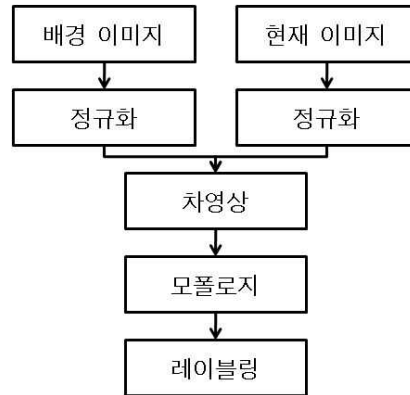


그림 3. 이동 물체 검출 전체 시스템 구조도  
Fig. 3 The entire system structure of moving object detection

그림 4는 차영상을 통하여 이동 물체 검출 결과를 보여준다.

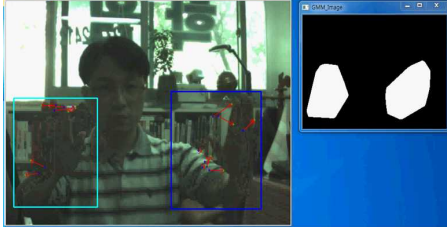


그림 4. 배경영상과 현재영상의 차영상을 통한 이동 물체 검출  
 Fig. 4 Background image and the image of a moving object detection by frame difference

### 2. Harris Corner Detector를 이용한 객체의 특징점 검출

배경으로부터 이동 물체를 검출 후, 이동 물체를 추적하기 위해 특징점 추적을 사용한다. 그러기 위해서는 우선 이동 물체에 대한 특징을 추출하여야 한다. Harris Corner Detector는 일정한 윈도우 안의 영상을 식(3)과 같이 x축과 y축으로 각각 미분하고, 위치행렬을 식(4)와 같이 곱한다.

$$g = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \nabla I \quad (3)$$

$$gg^T = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \begin{bmatrix} g_x & g_y \end{bmatrix} = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} \quad (4)$$

그리고 윈도우 영역의 모든 행렬 값을 식(5)와 같이 더한다. 매트릭스 Z는 순수한 텍스처 정보를 보유하고 있어, Z의 고유 값을 분석하면 영역 안의 특징을 분류할 수 있다.

$$Z = \iint_W \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} w dx \quad (5)$$

식(5)에서의 W는 가중치 함수를 나타낸다. 그림 5는 Harris Corner Detector를 이용하여 이동 물체의 특징점을 추출 한 결과 영상이다.

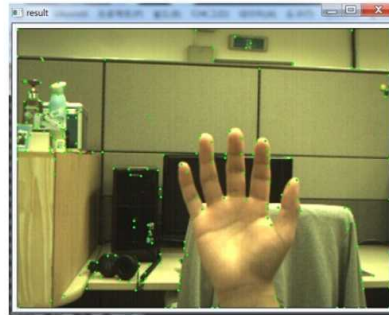


그림 5. Harris Corner Detector를 이용한 특징점 추출 결과  
 Fig. 5 Feature extraction using Harris Corner Detector

### 3. Lucas-Kanade 특징 추적을 이용한 객체 추적

Lucas-Kanade 특징 추적은 광류를 계산할 때, 반복적인 계산을 제거함으로써 실시간 구현이 가능하다.

특징추적 방법은 두 개의 연속되는 영상  $I_t(x) = I_t(x, y)$ 와  $I_{t+1}(x) = I_{t+1}(x, y)$ 인 t번째 영상과 그 다음 영상이 있다.  $I_t$ 의 영상에서의 특징점의 위치를  $f_t = [f_x^t, f_y^t]^T$ 라고 표현 했을 때, 다음 영상  $I_{t+1}$ 에서의 특징점의 위치  $f_{t+1} = f_t + d_t = [f_x^t + d_x^t, f_y^t + d_y^t]^T$ 을 추적하는 것이 목표이다. 여기서의 벡터  $d_t = [d_x^t, d_y^t]^T$ 는 영상을 추적할 물체의 움직임 정도로 특징점의 움직임을 나타낸다.

이 때, 두 특징점  $I_t(f_t)$ 와  $I_{t+1}(f_{t+1})$ 은 유사한 특징 값을 가지고 있다. 이 특징점의 위치를 구하기 위해 식(6)의 추적 값  $\epsilon$ 점 최소화 하여  $f_{t+1}$ 을 구하게 된다. 추적 값  $\epsilon$ 은 식(6)에서와 같이 계산한다.

$$\begin{aligned} \epsilon(d_t) &= \epsilon(d_x^t, d_y^t) \quad (6) \\ &= \sum_{x=f_x^t-w_x}^{f_x^t+w_x} \sum_{y=f_y^t-w_y}^{f_y^t+w_y} (I_t(x, y) - I_{t+1}(x+d_x, y+d_y))^2 \end{aligned}$$

그러나 Lucas-Kanade 특징 추적은 항상 올바른 추적을 하는 것은 아니다. 그림 6에서 보듯, 이동

물체를 추적하다가 일시적인 가림현상으로 인하여 올바른 특징점 추적을 하지 못하고 특징점들이 흐트러지는 것을 볼 수 있다. 이러한 문제점을 해결 방안을 3장에서 제안한다.



그림 6. L-K 특징 추적이 가림에 의한 오류 발생  
Fig. 6 LK feature tracking errors due to the occlusion

### III. 제안하는 특징점 추적 방법

제안하는 특징점 추적 방법은 기존의 추적 방법인 Lucas-Kanade 특징 추적 방법에 기반을 둔 추적 방법이다. 그러나 단순히 Lucas-Kanade 특징 추적 방법을 사용하게 되면 이동 물체의 가림에 의하여 제대로 된 추적을 할 수 없으므로, 이에 대한 보완 방법을 제안한다. 제안하는 방법은 크게 두 가지로 분류된다. 하나는 잘못된 특징점 검출 방법이다. 특징점 추적에 있어서 가림현상에 의해 잘못된 추적이 되었을 경우 이를 바로 검출할 수 있게 하기 위해서 잘못된 특징점을 우선적으로 검출할 필요가 있다. 또 다른 하나는 특징점 이동 예측방법이다. 잘못된 추적이 감지되었다면 바른 추적을 하기 위해서 특징점 이동 예측을 통하여 다시 이동 물체가 나타났을 때 다시 감지할 수 있도록 하기 위하여 사용되는 방법이다.

#### 1. 잘못된 특징점 검출 방법

Lucas-Kanade 특징 추적으로 인한 특징점이 잘 추적되었는지 확인하기 위해서 평가 함수를 사용하였다. 그림 7은 Lucas-Kanade에서 평가 함수를

사용한 구조도를 나타낸다.

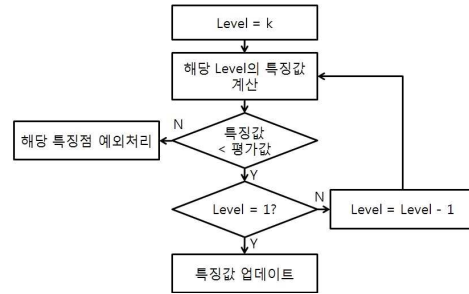


그림 7. 잘못된 특징점 검출 시스템 구조도  
Fig. 7 Incorrect feature detection system structure

그림 7에서 Level은 Lucas-Kanade 특징 추적에서의 Pyramidal Level을 나타낸다. 보통 k는 3 혹은 4의 값을 갖는다. 해당 Level의 특징값 계산은 예측된 위치의 Harris Corner Detector의 값을 말한다. 제일 처음 특징점을 찾은 방법이 Harris Corner Detector 값을 기준으로 찾았기 때문에, 이 값을 기준으로 비교 하였다. 식(7)은 올바른 특징점인지 찾는 평가 함수를 나타낸다.

$$e_i^t = e(x_i^t, y_i^t) = \tag{7}$$

$$\sum_{x=-w}^w \sum_{y=-w}^w (I^{t-n}(x_i^{t-n} + x, y_i^{t-n} + y) - (I^t(x_i^t + x, y_i^t + y)))$$

식(7)에서  $i$ 는  $i$ 번째 특징점을 표현하고,  $t$ 는 현재 영상을 나타낸다.  $t-n$ 은  $t$ 번째 영상으로부터  $n$ 만큼 뒤의 영상이다. 소개될 특징점 이동을 예측하게 되면,  $t-1$ 번째 영상이 항상 올바른 추적을 했다고 볼 수 없다. 그러므로 가장 최근 올바른 추적된  $t-n$ 번째 영상을 기준으로 비교하게 된다.  $w$ 는 특징점 자체만 비교하는 것이 아니라 주위까지 비교하기 위해 윈도우를 씌웠다. 보통  $w$ 의 값은 1에서 2값을 갖는다. 식(7)에서의 평가 값  $e_i^t$ 을 기준으로 특징 값을 업데이트 하게 된다. 식(8)은 특징 값을 업데이트 하는 방법을 나타낸다.

$$f_i^t = \begin{cases} f_i^t & \text{if } e_i^t \leq Th \\ f_i^{t-1} & \text{otherwise} \end{cases} \tag{8}$$

식(8)에서  $f_i^t$ 는  $t$ 번째 영상에서  $i$ 번째 특징 값을

나타낸다. 이 특징 값은 특징점 위치의 Harris Corner Detector의 값을 말한다.  $T_h$ 는 임계값을 말하며,  $e_i^t$ 가 임계값보다 작으면 업데이트 하지만, 임계값보다 클 경우, 업데이트를 하지 않는다. 임계값보다 클 경우 업데이트를 하지 않는 이유는 기존의 특징과 많이 다른 특징이기 때문에 잘못된 추적이라 판별하고, 이 잘못된 추적의 기준으로 업데이트 할 경우, 잘못된 추적을 계속하기 때문이다.

그림 8은 그림 6과 달리 가려진 현상이 되었으면, 특징점이 사라지는 것을 보여준다. 이를 통해 잘못된 특징점이 잘 판별되는지를 알 수 있다.



그림 8. 가림현상에 대한 검출 결과  
Fig. 8 detection results for occlusion

## 2. 특징점 이동 예측 방법

### 1) 부분적 가림에 대한 이동 예측 방법

부분적으로 가려진 경우 가림이 일어나지 않은 특징점의 이동을 이용하여 나머지 부분을 예측하는 방법이다. 이동 물체의 움직임은 크게 봤을 때, 부분적으로 움직이지 않고 전체적으로 움직인다. 이러한 가정에 의하여 가림현상이 일어나지 않은 부분을 이용하여 가려진 부분의 움직임을 예측할 수 있다. 이해를 돕기 위해 그림 9는 예를 보여준다.

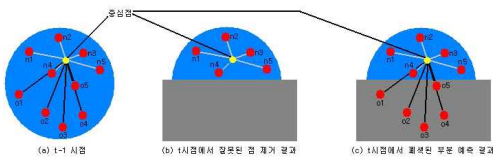


그림 9. 부분적 가림에 대한 특징점 예측 예제  
Fig. 9 Feature prediction examples for partial occlusion

그림 9에서  $n1 \sim n5$ 는 가림현상이 일어나지 않은 특징점들이고,  $o1 \sim o5$ 는 가림현상이 일어난 특징점들을 말한다. 가려진 특징점들의 위치를 예측하기 위해서 중심점으로부터 제대로 찾은 특징점들의 거리와 각도를 구하게 된다. 중심점을 구하는 이유는 특징점들이 한쪽으로 쏠림을 방지하기 위해서 중심점을 구한다. 이 중심점을 이용하여 나머지 부분의 이전 영상에서의 중심과의 거리와 각도를 구하여 특징점의 위치를 예측하게 된다. 식(9)은 가림현상이 일어나지 않은 부분의 중심점을 구하는 수식이다.

$$n_c^t = \sum_{k=1}^l n_k^t \quad (9)$$

식(9)에서  $n_k^t = (x_{n_k}^t, y_{n_k}^t)$ 는  $t$ 번째 영상의 가려진 현상이 되지 않은  $k$ 번째 특징점의 위치 좌표를 나타내고,  $n_c^t = (x_{n_c}^t, y_{n_c}^t)$ 는 가려지지 않은 특징점들의 중심을 말한다. 가려지지 않은 특징점들의 중심을 구한 후 가려지지 않은 특징점들의 평균 거리 비율 변화와 각도 변화를 계산한다. 식(10)은 거리 비율 변화를 계산하며, 식(11)은 각도 변화 비율을 계산한다.

$$r_a = \frac{1}{n} \sum_{k=1}^n \frac{\sqrt{(x_{n_k}^t - x_{n_c}^t)^2 + (y_{n_k}^t - y_{n_c}^t)^2}}{\sqrt{(x_{n_k}^{t-1} - x_{n_c}^{t-1})^2 + (y_{n_k}^{t-1} - y_{n_c}^{t-1})^2}} \quad (10)$$

$$\theta_a = \frac{1}{n} \sum_{k=1}^n \theta_k^t - \theta_k^{t-1} \quad (11)$$

식(10)에서  $r_a$ 는 정규화 된 거리 변화의 평균을 나타낸다. 거리 변화는 가려짐이 일어난  $t$ 번째 영상의 특징점과 중심점 사이의 거리와  $t-1$ 번째 영상에서  $t$ 번째 영상으로 연결되는 특징점들과 그 점들의 중심점 사이의 거리로부터 정규화를 통해 거리 변화를 계산한다. 식(11)에서  $\theta_a$ 는  $t$ 번째 영상에서의 각도와  $t-1$ 번째 영상에서의 각도 차를 이용한 각도 변화 평균을 나타낸다.  $\theta_k^t$ 는  $t$ 번째 영상에서  $k$ 번째 특징점이 중심으로부터의 각을 나타낸 것으로 현재 각도와 이전 영상의 각도의 차이를 구해 이동 물체가 전체적으로 어느 정도 각도가 변화하였는지 계산한다. 이렇게 식(10)과 식(11)로 거

리와 각도 변화를 계산한 다음 가려진 부분의 거리와 각도를 계산한다. 가려진 점의 예측되는 점은 식(12)과 식(13)에 의하여 계산된다.

$$o_k^t = o_k^{t-1} \times r_a \quad (12)$$

$$\theta_{o_k}^t = \theta_{o_k}^{t-1} \times \theta_a \quad (13)$$

식(12)에서  $o_k^t = (x_{o_k}^t, y_{o_k}^t)$ 는 t번째 영상에서 k번째 가려진 특징점이 가려지지 않은 특징점들의 중심으로부터의 거리 값이다. 또한 수식(13)에서  $\theta_{o_k}^t$  가려진 특징점과 가려지지 않은 특징점들의 중심과의 각도를 나타낸다. 이 두 개의 수식을 이용하여 가려진 특징점들의 위치를 예상할 수 있다.

## 2) 전체적 가림현상에 대한 이동 예측 방법

이동 물체가 다른 배경이나 물체에 의해서 완전히 가려졌을 경우 다소 정확성이 떨어지지만, 이동 물체의 속도와 가속도, 그리고 이동 방향을 이용하여 이동 물체의 위치를 예측 할 수 있다. 그러나 이 예측 방법은 정확한 예측보다는 다시 물체가 나타났을 경우, 제대로 추적할 수 있도록 도와주기 위한 방법으로 다소 간단한 방법으로 예측을 한다.

이동을 예측하기 위해서 이동 물체의 속도와 가속도를 구한다. 이동 물체의 속도와 가속도를 구하기 이전 식(14)에서와 같이 이동 물체의 특징점들의 중심점을 구한다. 그리고 속도는 식(15)을 이용하여 구하고, 가속도는 식(16)을 사용한다.

$$c_c^t = \frac{1}{n} \sum_{k=1}^n c_k^t \quad (14)$$

$$v^t = \sqrt{(x_c^t - x_c^{t-1})^2 + (y_c^t - y_c^{t-1})^2} \quad (15)$$

$$a^t = |v^t - v^{t-1}| \quad (16)$$

식(14)에서  $c_c^t = (x_c^t, y_c^t)$ 는 t번째 영상에서 특징점들의 중심이다. 또한  $v^t$ 와  $a^t$ 는 이동 물체가 t시점에서의 속도와 가속도를 표현한 것이다. 길이 변화를 예측했다면 이번에는 각도를 예측하기 위해

이전 영상과 현재 영상간의 이동각도를 이용하였다. 이 각도를 구하기 위해 식(17)을 사용했다. 그러나 단순히 각도 변화뿐만 아니라 변화율을 계산하기 위해 식(18)을 이용하여 더 정확한 각도를 예측하였다.

$$\theta^t = \tan^{-1} \frac{y_c^t - y_c^{t-1}}{x_c^t - x_c^{t-1}} \quad (17)$$

$$\theta_a^t = \theta^t \times \theta^{t-1} \quad (18)$$

$x_c^t$ 와  $y_c^t$ 는 t번째 영상에서 특징점들의 중심 x, y 좌표를 나타낸다. 또한  $\theta_a^t$ 는 각도 변화량을 나타낸다. 길이 예측과 각도 예측을 통하여 가려졌을 때, 현재 영상에서의 특징점들의 중심을 예측할 수 있다. 최종적인 위치는 식(19)과 식(20)을 통해 계산될 수 있다.

$$x_c^t = (c_c^{t-1} + v^{t-1} + a^{t-1}) \cos(\theta^{t-1} + \theta_a^{t-1}) \quad (19)$$

$$y_c^t = (c_c^{t-1} + v^{t-1} + a^{t-1}) \sin(\theta^{t-1} + \theta_a^{t-1}) \quad (20)$$

중심점을 예측 후, 나머지 특징점들은 중심으로 부터 이전 영상에 위치한 거리와 동일한 위치 거리를 갖는다.

## IV. 실험 및 결과

본 논문에서는 제안한 알고리즘을 실험하고, 기존의 Lucas-Kanade 방법과 정확성을 비교한다. 실험을 위해 사용한 컴퓨터는 IntelPentium-4 3.0GHz의 CPU와 3GB의 RAM을 사용하였고, 사용언어는 Microsoft의 Visual C++ 2008을 사용하였다. 실험을 위한 640×480해상도와 32비트 칼라 값을 갖고 초당 20프레임의 동영상을 사용하였다.

초기 위치를 구하기 위해 그림 12과 같이 그림 11의 배경영상과 그림 10의 초기 프레임의 차영상을 이용하여 초기 영역을 구하였다. 단순한 차영상만을 이용하지 않고 15×15 크기의 열림을 이용하여 잡음을 제거하였다.

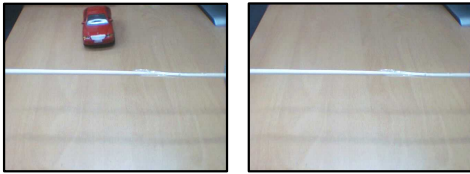


그림 10. 이동 모형 자동차의 초기 동영상  
Fig. 10 Go early model car videos

그림 11. 이동 모형 자동차의 배경 영상  
Fig. 11 Move the background image model cars

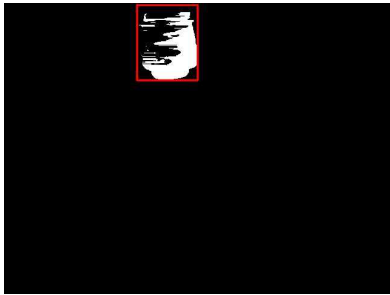


그림 12. 차영상을 이용한 모형 자동차의 초기 영역 추출  
Fig. 12 Using the model, the initial extraction of the car frame difference

잡음 제거 후, Harris Corner Detector를 이용하여 이동 물체의 특징점을 검출하였다. 특징점 검출에 있어 개수는 제한이 없이 특징값이 일정값 이상인 경우, 검출하게 하였다. 그림 13는 특징점을 검출한 결과를 나타낸다.



그림 13. Harris Corner Detector를 이용한 모형 자동차 특징 추출  
Fig. 13 Harris Corner Detector feature extraction using a model car

Harris Corner Detector로 검출한 특징점을 그림 14와 그림 15에서는 추적하였다. 그림 14는 기본적으로 Lucas-Kanade 특징 추적 알고리즘만을 사용 하였을 때, 가려짐에 대한 특징점 추적이 잘

안됨을 보여준다. 그래서 여기서 제안한, 평가 함수와 특징점 예측 기법을 이용하여 가려짐 현상에 강건함을 그림 15에서 보여준다.

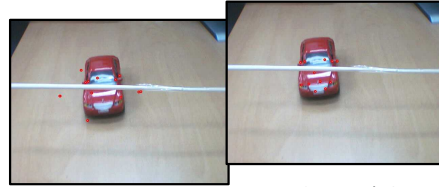


그림 14. 기존 알고리즘 이동 물체 추적  
Fig. 14 Conventional moving object tracking algorithm

그림 15. 제안 알고리즘 이동 물체 추적  
Fig. 15 The algorithm of moving object tracking

특징점 예측 기법 사용에 있어서, 부분적 가림에 대한 이동 예측방법을 이용할지, 전체적 가림에 대한 이동 예측방법을 사용할지는 가림현상에 정도에 따라 달라진다. 여기서 실험한 것으로 볼 때 특징점 전체 중 65%이상이 가려짐이 일어나면 전체적 가림현상에 대한 이동 예측 방법을 사용하는 것이 좋다는 것을 그림 16의 그래프를 통해 알 수 있다. 그림 16은 가림현상 이후 특징점들이 원 위치에 오는 특징점들의 개수를 말한다. 실험에 쓰였던 특징점의 총 개수는 16개였다.

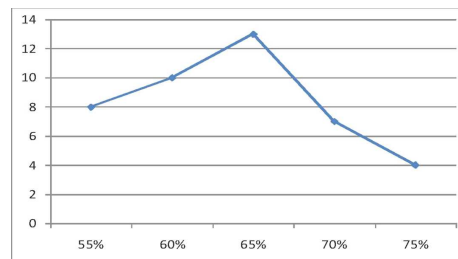


그림 16. 가림현상 이후 원 위치로 오는 특징점의 개수

Fig. 16 Original position after occlusion followed by the number of minutiae

## V. 결론 및 향후 연구 과제

본 논문에서는 가려진 현상에 대한 추적을 강건하게 하기 위해 평가 함수와 특징점 이동 예측을 제안하였다. 앞의 실험결과에서 알 수 있듯이 본 논문의 제안방법이 이동 물체 추적에 있어서 가려진 현상에 대해 강건함을 알 수 있다. 그러나 시작 프레임으로부터 짧은 간격으로 일어나는 가려진 현상에 대해서는 예측하기 어려웠다. 예를 들어 시작 프레임을 1프레임이라고 했을 때, 가림현상이 4프레임쯤 일어났을 경우, 기존의 예측 알고리즘을 사용했을 경우, 그 간격이 너무 짧아 제대로 된 예측을 할 수 없었다. 예측을 하더라도 가림현상 이후 특징점들이 제대로 자기 위치를 찾지 못하는 결과를 보였다. 이러한 문제점에 대해서는 향후 연구가 더 필요하다고 생각된다.

## 후 기

이 논문은 한국기술교육대학교 교육연구진흥비 지원 프로그램의 (일부) 지원에 의하여 수행되었습니다.

## 참 고 문 헌

- [1] S. McKenna, Y. Raja, and S. Gong, "Tracking Contour Objects Using Adaptive Mixture Models," *Image and Vision Computing*, pp. 225-231, 1999
- [2] R. Plankers and P. Fua, "Tracking and Modeling People in Video Sequence," *Computer Vision and Image Understanding*, pp. 285-302, 2001
- [3] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 25, No. 5, pp. 564-577, May 2003.
- [4] C. R. Wren, A. Azerbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 780-785, July 1997.
- [5] D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, vol. 2, Hilton Head Island, SC, pp. 142-149, June 2000.
- [6] A. M. Baumberg, "Learning deformable models for tracking human motion," Ph. D. dissertation, School of Comput. Studies, Univ. of Leeds, U. K., Oct 1995.
- [7] T. J. Cootes, C. J. Taylor, D. H. Cooper, and J. Gragam, "Training models of shape from sets of examples," in *British Machine Vision Conference*, pp. 9-18, September 1992.
- [8] J. Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm", Intel Corporation, Microprocessor Research Labs, 2000.
- [9] X. Wei, Z. Zhu, L. Yin, and Q. Ji, "A real-time face tracking and animation system," *Proceedings of the CVPR Workshop on Face Processing in Video*, 2004.
- [10] J. Shi, and C. Tomasi, "Good features to track", *IEEE Conference on CVPR Seattle*, pp. 593-600, 1994.



