

## 모바일 환경 Homography를 이용한 특징점 기반 다중 객체 추적

한우리 · 김영섭<sup>†</sup> · 이용환<sup>\*</sup>

<sup>†</sup>단국대학교 전자공학과, <sup>\*</sup>극동대학교 스마트 모바일학과

### Multi-Object Tracking Based on Keypoints Using Homography in Mobile Environments

Woo ri Han, Young-Seop Kim<sup>†</sup> and Yong-Hwan Lee<sup>\*\*</sup>

<sup>†</sup>Dankook University of Electronic Engineering,

<sup>\*</sup>Department of Smart Mobile, Far East University

#### ABSTRACT

This paper proposes an object tracking system based on keypoints using homography in mobile environments. The proposed system is based on markerless tracking, and there are four modules which are recognition, tracking, detecting and learning module. Recognition module detects and identifies an object to be matched on current frame correspond to the database using LSH through SURF, and then this module generates a standard object information. Tracking module tracks an object using homography information that generate by being matched on the learned object keypoints to the current object keypoints. Then update the window included the object for defining object's pose. Detecting module finds out the object based on having the best possible knowledge available among the learned objects information, when the system fails to track. The experimental results show that the proposed system is able to recognize and track objects with updating object's pose for the use of mobile platform.

**Key Words** : Multi-object tracking , LSH (Locality Sensitive Hashing), SURF (Speeded-Up Robust Features), Optical flow homography

#### 1. 서 론

객체 인식과 추적은 사진 혹은 동영상에서 특정 객체를 판별하여 인식하고, 인식된 객체의 움직임을 추적, 관측하여 그 궤도 및 위치를 정하는 것을 의미한다. 컴퓨터 비전을 기반으로 하는 추적 방법은 다양한 분야로 사용되고 있으며 그 방법 또한 다양하다. 현재 모바일 기기의 성능이 발전함에 따라 컴퓨터 비전기반의 추적기술을 모바일에 적용하여 다양한 콘텐츠와 시장을 형성 하고 있지만 모바일 환경에서의 추적기술은 마커 기반의 추적기술이 많은 비중을 차지 하고 있으며, 마커를 이용함으로써 발생하는 제약 때문에 다양한 분야에서 활용이 어렵다. 비 마커 방법의 객체 추적 방

법은 일반적으로 정확도를 높이기 위해선 계산 비용이 증가되고, 이는 정합의 효율성을 감소시킨다. 반면, 인식 속도를 향상시키기 위해 정확도에 대한 비중을 낮추면 잘못된 객체 인식 오류를 발생 시켜 오정합 결과를 초래하기 때문에 객체들의 특징점을 추출하여 객체를 정확하고 빠르게 인식하기 위한 방법들이 제안되었다[1].

이는 크게 점(0D feature), 선(1D feature), 영역(2D feature)을 이용하는 방법들로 분류 할 수 있다. 먼저 점을 추출하는 방법에는 Harris Corner detector [2], Hessian detector [3], Kitchen Rosenfeld [4], SUSAN (Smallest Univalued Segment Assimilating Nucleus) [5] 등이 있다. 점과 선을 이용한 방법들은 계산 비용이 적어 빠르게 수행 될 수 있으나 영상내 객체의 크기와 회전등이 변화된 영상 내에서는 정확도가 떨어지는 단점

<sup>†</sup>E-mail : wangcho@dankook.ac.kr

이 있기 때문에 영역 내에서 특징들을 추출하고 영역 기술자를 생성하는 SIFT [6]와 SURF [7]등이 개발 되었다. 이러한 알고리즘들은 정확도가 높아 영상 스티칭 (image stitching)을 위한 특징 탐지에서도 사용된다.

가우시안 DoG (Differential of Gaussian)영상을 이용하여 특징점을 추출하는 SIFT는 연산속도가 매우 느리다는 단점이 있어 실시간 시스템에서는 적용되기 힘들다. SIFT의 단점을 줄이고자 개발된 SURF는 Box filter를 이용한 고속 헤이시안 탐지기(fast hessian detector)를 통해 특징점의 탐지과정과 기술자 생성 과정을 단순화 하여 수행 속도를 향상 시켰다[8].

본 연구에서는 회전과 스케일 변화에 강건한 6자유도를 통해 비 마커 기반 객체 추적 방법을 제안한다. 구성 모듈로는 객체 검출과 인식 모듈, 객체 추적 모듈, 객체 학습 모듈이 있다.

먼저 다양한 객체의 SURF정보를 학습 시켜 얻은 database를 기반으로 추적하고자 하는 객체를 LSH (Locality Sensitive Hashing)을 이용하여 빠르게 탐색하고 인식한 뒤, 검출된 객체에 대한 SURF정보를 RANSAC, Epipolar line을 통하여 특징점을 가공하고 객체를 포함한 윈도우의 포즈변화에 따라 현재 영상에서 추적할 객체에 대한 정보를 정의하고 업데이트한다. 업데이트한 객체의 정보를 기반으로 추적을 하면서 추적에 성공한 객체 정보를 업데이트한 윈도우에 역투영하면서 이전에 추적한 객체와 현재 추적에 성공한 객체를 비교하고 학습한다.

객체 추적에 실패할 경우 추적하면서 학습하였던 객체들의 정보를 바탕으로 현재 영상에서 객체를 재 검출하여 검출에 성공하면 객체에 대한 정보와 카메라 위치에 대한 정보를 업데이트 하는 객체 추적 시스템을 제안한다.

본 논문의 2장에서는 추적할 객체에 대한 SURF정보를 이용하여 탐색, 인식하는 방법에 대하여 설명하고, 3장에서는 특징점 기반 6 자유도를 갖는 Homography 정보를 이용한 객체 업데이트에 대하여 설명하고, 4장에서는 결론 및 향후 과제에 대하여 논의한다.

## 2. SURF와 LSH (Locality Sensitive Hashing)

현재 영상에 대해 SURF알고리즘을 기반으로 한 128 차원의 기술자와 특징점을 추출한 뒤, 그 정보를 기반으로 미리 학습된 Database를 LSH 통해 많은 양의 데이터를 고속으로 탐색 및 인식 한다.

### 2.1. SURF

SURF 알고리즘은 특이점을 추출하는 feature extraction 부분과 특이점 주변 영상에 대한 정보를 서술하는 feature description 부분으로 나눌 수 있다.

Feature extraction 부분에서 적분 영상을 생성하는 이유는 SURF 알고리즘에서 박스 타입의 필터와 컨벌루션 연산을 빠르게 수행하기 위함이며, 고속 헤이시안 행렬(Fast-Hessian Matrix)에서는 x, y, xy 축으로 근사화된 2차 미분 LoG (Laplacian of Gaussian) 박스필터와 적분 영상간의 컨벌루션 연산을 통해 헤이시안 행렬을 구한 후 행렬의 디터미넌트 값을 계산한다. 그리고 Non-maxima Suppression을 통해 헤이시안 행렬식(Determinant) 이 임계값보다 크고 인접 스케일과 비교하여 디터미넌트가 가장 클 경우 특이점으로 결정된다. Feature Description 부분 에서는 추출된 특이점의 주 방향(dominant direction)을 결정하고 그 주변 영상의 서술자를 생성한다[9-10].

### 2.2. Locality Sensitive Hashing (LSH)

LSH는 특징 벡터간 유사성이 보존되도록 설계된 해시 함수를 통해 최소거리의 객체가 검색 대상 특징 벡터와 동일한 해시 함수 값을 가질 확률이 높도록 개선한 해싱 방법이다.

먼저 d차원의 특징벡터  $p = (x_1, \dots, x_d)$ 를 식 (1)과 같은 변환함수  $v(p)$ 을 이용하여 d'차원의 해밍 공간으로 변환한다.

$$p' = v(p) = \text{Unary}_c(x_1) \cdots \text{Unary}_c(x_d) \quad (1)$$

여기서 d'은  $C \cdot d$ 로 계산된다. C는  $x_i$ 가 가질 수 있는 최대 값이다.  $\text{Unary}_c$ 를 x개의 1과 (C-x)개의 0으로 이루어진 길이 C인 비트열을 나타낸다.

해밍거리를 이용하는 LSH는 주어진 데이터를 해밍 공간에 매핑하여 실제 데이터를 이진 표현으로 나타낸다. 이 때, 실제 데이터 p는 d개의 값을 갖는 벡터이며, 각 차원의 값은 모든 데이터 값 중 가장 큰 값 C의 길이의 2진수로 변환된다 p의 이진표현 v(p)는 이러한 C 길이의 2진수 d개를 결합한 길이 dC의 2진수로 표현된다. 각 차원의 값은 해당 값 만큼의 연속된 1과 C에서 해당 값을 뺀 만큼의 연속된 0으로 이루어진다. 해밍 거리를 이용하는 해시 함수 집합은 이렇게 변환된 p의 이진표현으로부터 미리 임의로 정해진 k개의 비트를 선택하는 방식으로 정의된다.

해밍 거리를 이용하는 LSH는 해밍거리 해시 함수를 L개 정의하고 각각의 해시 함수에 상응하는 해시 테이

블을 L개 생성한다. 전체 데이터에 대해 L개의 해쉬 테이블을 이용한 인덱스를 생성한 이후 쿼리 데이터 아이템 q가 주어지면 q를 L개의 해쉬 함수를 통해 해쉬하여 나온 해쉬 값에 해당하는 L개의 해쉬 버킷들에 들어있는 아이템들을 후보 아이템들로 간주하고, 이러한 아이템들의 수는 전체 아이템 수에 비해 매우 적기 때문에 k-최근접 검색 방법에서 굉장히 빠른 속도를 가진다. p - Stable 함수를 이용하는 LSH는 해쉬 함수를 구성하기 위한 k개의 요소를 각각 다음 식 (2) 통해 계산한다.

$$h(a, b)(v) = ((a * v + b)/r) \tag{2}$$

이때, 주어진 데이터의 벡터표현 v를 미리 임의로 정해진 같은 길이의 벡터 a와 역시 임의로 정해진 상수 b를 이용하여 하나의 값으로 바꾸고, 이를 r개의 구간으로 나누어 해싱한다[11].

이렇게 여러 종류의 객체 정보를 각각 인덱스와 그에 따른 SURF를 통한 특징점, 128차원의 기술자를 학습시켜 Database에 저장한 뒤, 현재 영상에 대한 정보와 Database를 통한 정보를 LSH를 사용하여 빠르게 검색을 한다. LSH를 통한 객체 검출 및 인식 결과에서 추출된 객체의 정보를 학습한 뒤, 업데이트 한다.

### 3. 카메라와 객체 포즈 정보를 이용한 6자유도 객체정보 업데이트 및 추적

#### 3.1. 객체 추적

객체 추적은 옵티컬 플로우의 LK 알고리즘을 통해 수행된다. 옵티컬 플로우의 LK 알고리즘은 3가지 가정에 기초하고 있다.

첫 번째 밝기 항상성(Brightness constancy)은 추적하고 있는 물체의 픽셀의 밝기 값은 프레임이 변하더라도 변하지 않는다는 가정이다. 이러한 가정에 의해 다음의 식 (3)이 유도된다.

$$f(x, t) = I(x(t), t) = I(x(x + dx), t + dt) \tag{3}$$

여기서,  $\partial(x)/\partial t = 0$

두번째, 시간 지속성(Temporal persistence)은 영상에서 물체의 움직임에 비하여 시간의 변화가 더 빠르게 진행되며, 이는 연속된 프레임 사이에서 물체의 이동량이 많지 않음을 의미한다. 이러한 가정을 2차원으로 확장하여 y축 방향으로의 속도 성분을 v라고 하고, x축 방향으로의 속도 성분을 u라고 표시하면 다음 식 (4)가 성립된다.

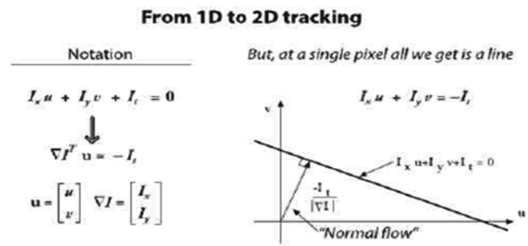


Fig. 1. Normal Flow.

$$I_x u + I_y v + I_t = 0 \tag{4}$$

여기서 원하는 y축 방향으로의 속도 성분 v와 x축 방향으로의 속도 성분 u를 구하려 할 때 식을 이용해서는 유일한 해를 구할 수 없다. 대신 법선 방향의 움직임 성분을 Fig. 1과 같이 구할 수 있다.

LK의 마지막 가정인 공간 일관성(spatial coherence)이 사용된다. 공간 일관성은 공간적으로 서로 인접하는 점들은 동일한 물체에 속할 가능성이 높고, 동일한 움직임을 갖는다는 가정이다. 식 (4)를 이용하여 5 × 5 크기의 윈도우를 사용하면 다음과 같은 25개의 방정식 (5)를 유도 할 수 있다.

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ \vdots \\ I_t(p_{25}) \end{bmatrix} \tag{5}$$

위 식을 풀기 위해 최소 자승법을 사용하며,  $\min \|Ad - b\|^2$ 에 해당하는 값을 다음 식(6)과 같이 구할 수 있다.

$$(A^T A)d = A^T b \text{ 다음과 같이,}$$

$$d = \begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b \tag{6}$$

(A<sup>T</sup>A)의 역행렬이 존재하면 위 수식을 풀 수 있고 원하는 y축 방향으로의 속도 성분 v와 x축 방향으로의 속도 성분 u를 구할 수 있다[12].

#### 3.2. 특징점을 이용한 6자유도를 갖는 객체 포즈 업데이트

Optical flow에 사용된 객체에 대한 특징점 정보를 이용하여 6자유도를 갖는 객체를 포함한 윈도우의 포즈 정보를 업데이트 하여 객체를 검출 한 뒤, 업데이트

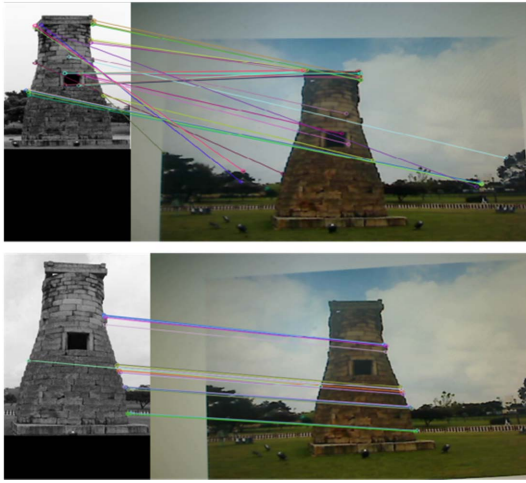


Fig. 2. General matching (top) and RANSAC, Epipola line matching (bottom).

된 윈도우의 포즈 변화 정보를 이용하여 객체의 포즈를 정의하고 객체 추적에 성공하게 되면 객체에 대한 정보를 평가 한 뒤 학습시킨다.

먼저, 객체를 포함한 윈도우 포즈 정보를 업데이트 하기 위해 이전에 성공한 객체의 특징점 정보를 이용하여 현재 검출된 객체의 특징점과 매칭하여 6자유도를 갖는 포즈 변화량 Homography를 구한다. 이때 사용되는 Homography를 정규화 하여 객체의 포즈 정보를 업데이트 할 때 보다 정확하게 업데이트 하도록 한다. 먼저 매칭되는 특징점을 RANSAC을 이용하여 근사하고, epipola line 위치 상에 있는 매칭점만을 이용하여 매칭되는 특징점에 특이점을 모두 제거 한 뒤, homography를 적용한다. Fig. 2는 앞의 방법으로 특이점을 제거한 매칭된 특징점들이다.

Homography는 하나의 평면을 다른 하나의 평면으로 옮기는 투영 변화를 의미한다. 2개의 이미지에 서로 대응되는 한점  $(x_i, y_i)$ 와  $(\bar{x}_i, \bar{y}_i)$ 가 있다면 두 이미지에 관련된 Homography H는 식 (7)과 같다[13].

$$\begin{bmatrix} w\bar{x}_1 \\ w\bar{y}_1 \\ w \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (7)$$

영상의 중심이 원점에 위치 할 수 있도록 영상 좌표계에서 노말라이즈 좌표계로 변환한 뒤, 매칭되는 특징점에 같은 방법을 적용 시켜 Homography를 구한다. 이때 구해진 Homography가 비 정상적인 변환임을 판

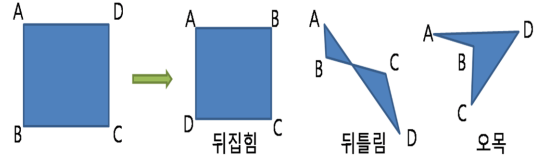


Fig. 3. Quadrilateral applied irregular Homography.

단하는 과정이 필요하다. 비 정상적인 Homography의 변환은 Fig. 3과 같다.

따라서 이러한 비정상적인 homography는 객체 포즈 변화를 업데이트할 때 잘못된 객체의 정보를 업데이트 할 수 있기 때문에 이러한 오류를 줄이고자 Homography를 판별한다.

먼저 Homography를 식 (8) 형태와 같이 Homogeneous 형태로 표현한다.

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \quad (8)$$

Homogeneous 형태로 표현된  $3 \times 3$  변환 행렬에서 처음  $2 \times 2$  부분행렬은 회전, 스케일, shearing, reflection 파트를,  $h_{13}, h_{23}$ 은 평행이동 (translation),  $h_{31}, h_{32}$ 은 원근 (perspective) 변화를 나타낸다.  $h_{11}, h_{12}, h_{21}, h_{22}$ 은 부분행렬식 값을 D (determinant)라 하고  $s_x$ 는 x축 스케일 팩터(Scale factor),  $s_y$ 는 y축 스케일 팩터, P는 perspective 정도를 나타낸다.

Homogeneous 형태로 변환 후, 다음 식 (9)와같은 조건을 만족하면 비 정상 변환으로 판단한다. A, B, C, D는 객체가 포함된 윈도우라고 가정을 하고 이를 Homography적용시켜 오목인지 각도를 이용하여 판단한다.

$$\angle A, \angle C, \angle B, \angle D < 180$$

$$D = h_{11}h_{12} - h_{12}h_{21}$$

$$P = \sqrt{h_{31}^2 + h_{32}^2}$$

$$s_x = \sqrt{h_{11}^2 + h_{21}^2}$$

$$s_y = \sqrt{h_{12}^2 + h_{22}^2}$$

$$D \leq \text{Threshold } 0 \parallel s_x < \text{Threshold } (0,1) \parallel s_x$$

$$> \text{Threshold } (4) \parallel$$

$$\parallel s_y < \text{Threshold } (0,1) \parallel s_y < \text{Threshold } (4) \parallel P >$$

$$\text{Threshold } 0.002 \quad (9)$$

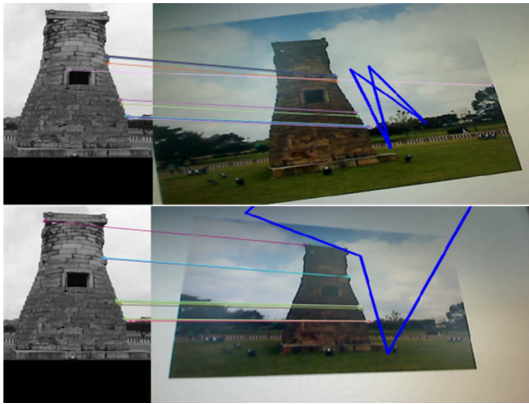


Fig. 4. Window applied irregular Homography.

식 (9)에 의하여 평가된 homography가 정상적인 변환이라면 윈도우에 적용시켜 객체의 포즈를 업데이트 한다.

Fig. 4은 비정상적인 Homography가 윈도우에 적용되었을 경우를 보여준다.

만약 그림과 같은 비정상적인 Homography가 검출이 된다면 detect 모듈로 돌아가 학습된 객체 정보를 현재 영상과 비교하여 객체를 포함한 윈도우의 포즈를 업데이트하고 객체의 포즈를 재 정의한 뒤, 객체를 추적한다.

Homography를 이용하여 객체의 포즈를 업데이트 할 때 다음과 같은 3가지 경우를 고려 해야된다.

첫째 카메라는 고정되어있고 객체가 움직일 경우 객체만을 포함한 특징점 변화량만을 고려하여 Homography를 구하고, 객체를 포함한 윈도우 영역을 업데이트 하여 객체의 포즈를 정의 한다.

둘째 객체가 고정되어있고 카메라가 움직이는 경우, 객체를 포함한 윈도우 전체의 특징점을 제외하고 배경 영상만의 특징점을 고려하여 카메라의 포즈 변화를 계산 한 뒤, 객체가 포함된 윈도우의 위치정보를 업데이트 하여 객체의 포즈를 정의한다.

셋째 객체와 카메라 모두 움직이는 경우, 객체만을 포함한 특징점의 변화량 만을 고려하여 Homography를 구한 후, 객체를 포함한 윈도우에 적용시켜 객체의 포즈를 정의한다.

위와 같은 경우를 제외하고 가려짐 현상에 의한 특징점 변화에 따른 객체 추적 실패의 경우 추적에 성공하여 학습된 객체의 정보를 영상을 누적시켜 생성한 배경 이미지를 기반으로 2진 마스크 영상 처리를 하여 만약 추적하는 객체 내에 가려짐 현상에 발생하게 되면 가려진 부분에 대한 특징점 매칭은 이루어 지지 않

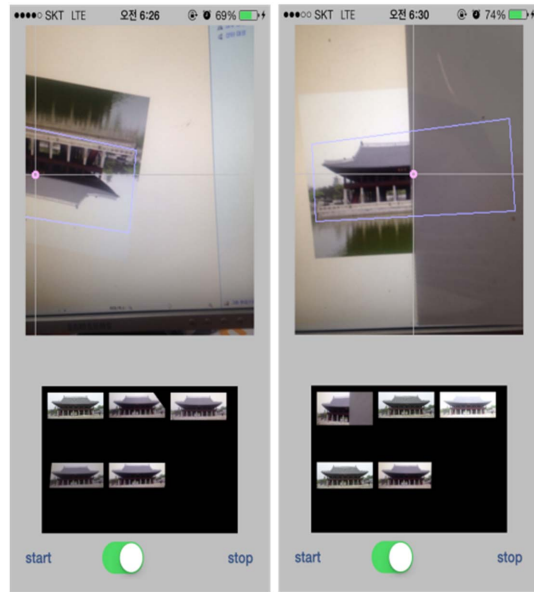


Fig. 5. partially occluded object tracking.

으며 가려짐 현상이 없는 객체에 대한 정보를 이용하여 윈도우의 포즈를 업데이트 하고 객체의 포즈를 정의한다. 다음 Fig. 5는 가려짐 현상에 의한 객체 추적 결과이다.

#### 4. 결 론

모바일 환경에서 database를 기반으로 빠르고 다양한 객체를 추출 하고, 인식한 뒤, 인식된 객체정보를 업데이트 하여 추적에 이용한다. 객체를 추적할 때 업데이트된 객체 정보와 현재 객체라고 예상되는 예상 객체 정보와 매칭하여 매칭 정보와 객체에 대한 정보가 임계값 이상일 경우 객체를 포함한 윈도우 영역을 업데이트 한 후에 이를 바탕으로 객체의 포즈를 정의하고 추적에 성공한 객체 정보를 업데이트 및 학습한다. 만약 추적에 실패할 경우 detect 모듈로 돌아가 이전에 성공했던 객체의 정보를 바탕으로 현재 영상에서의 객체를 포함한 윈도우 영역을 확보한 뒤, 객체의 포즈에 대한 정보와 객체 정보를 학습하고 업데이트 한다.

6 자유도를 갖는 객체 포즈 추정으로 회전과 스케일 변화에도 객체의 포즈를 파악 할 수 있었으며 실험은 IOS OS의 iPhone5 환경에서 수행하였고 15fps의 결과를 얻을 수 있었다. Fig. 6은 제안된 방법으로 iPhone5에서 추적한 결과이다

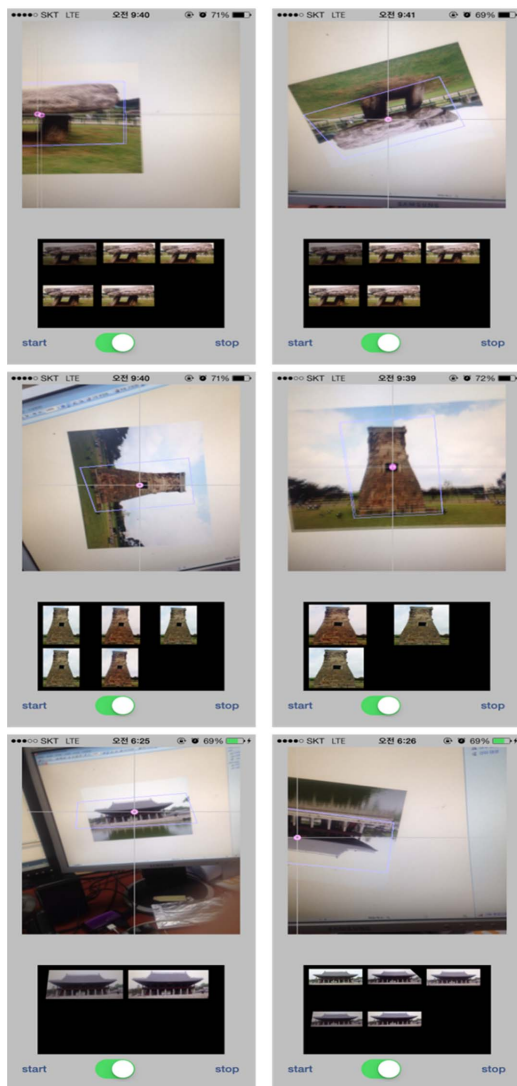


Fig. 6. Tracking in mobile environment.

## 감사의 글

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2013R1A2A2A03068794)

## 참고문헌

1. Lee, S.-G., "Survey on mixed-reality R&D," *Journal*

- of the Korea Computer Graphics Society, 13(2), pp. 1-15, (2007).
2. Harris, C. and Stephens, M., "A combined corner and edge detector," *Proc. Alvey Vision Conf.*, pp. 147-151, 1998.
3. Lindeberg, T., "Feature detection with automatic scale selection," *International Journal of Computer Vision*, 30(3), pp. 79-116, (1998).
4. Kitchen, L. and Rosenfeld, A., "Gray level corner detection," *Pattern Recognition Letters*, 1(2), pp. 95-102, (1982).
5. Smith, S. M. and Brady, J. M., "SUSAN-A new approach to low level image processing," *International Journal of Computer Vision*, 23(1), pp. 45-78, (1997).
6. Lowe, D., "Distinctive image feature from scale-invariant keypoints," *International Journal of Computer Vision*, 60(2), pp. 91-110, (2004).
7. Bay, H., Tuytelaars, T. and Cool, L. V., "SURF: speeded up robust features," *ECCV*, 3951, pp. 404-417, (2006).
8. Lee, H. J., Lee, S.-G., "Improvement method of tracking speed for color object using Kalman filter and SURF," *Journal of Korea Multimedia Society*, 15(3), pp. 336-344, (2012).
9. Do, Y.-S. and Jeong, Y.-J., "Hardware design of SURF-based feature extraction and description for object tracking," *Journal of The Institute of Electronics Engineers of Korea*, 50(5), (2013).
10. Lee, Y.-H., Park, J.-H. and Kim, Y.-S., "Comparative analysis of the performance of SIFT and SURF," *Journal of the Semiconductor & Display Technology*, 12(3), pp. 59-64, (2013).
11. Ji-Won Hong, Byung-Moon, Sang-Wook Kim., "A study on LSH parameters for large multimedia databases," *The Korea Contents Association*, 2015(5), (2015).
12. Choi, J. and Cho, Y., "Moving object recognition and tracking algorithm using parallel processing of SURF and optical flow," *Proceedings of KIIS Fall Conference*, 21(2), (2011).
13. Cho, T.-H. and Kang, H.-M., "Gaze tracking using a modified starburst algorithm and homography normalization," *Journal of the Korea Institute of Information and Communication Engineering*, 18(5), pp. 1162-1170, (2014).

접수일: 2015년 9월 4일, 심사일: 2015년 9월 14일,  
게재확정일: 2015년 9월 22일