

광류를 사용한 빠른 자연특징 추적

배 병 조[†] · 박 종 승^{††}

요 약

시각기반 증강현실을 구현하기 위한 추적 방법들은 정형 패턴 마커를 가정하는 마커 추적기법과 영상 특징점을 추출하여 이를 추적하는 자연특징 추적기법으로 분류된다. 마커 추적기법은 빠른 마커의 추출 및 인식이 가능하여 모바일 기기에서도 실시간 처리가 가능하다. 한편 자연특징 추적기법의 경우는 입력 영상의 다양성을 고려해야 하므로 계산량이 많은 처리과정을 거쳐야 한다. 따라서 저사양의 모바일 기기에서는 빠른 실시간 처리에 어려움이 있다. 기존의 자연특징 추적에서는 입력되는 카메라 영상의 매 프레임마다 특징점을 추출하고 패턴매칭 과정을 거친다. 다수의 자연특징점들을 추출하는 과정과 패턴매칭 과정은 계산량이 많아 실시간 응용에 많은 제약을 가하는 요인으로 작용한다. 특히 등록된 패턴의 개수가 증가될수록 패턴매칭 과정의 처리시간도 증가하게 된다. 본 논문에서는 이러한 단점을 해결하고자 자연특징 추적 과정에 광류를 사용하여 모바일 기기에서의 실시간 동작이 가능하도록 하였다. 패턴매칭에 사용된 특징점들은 다음의 연속 프레임에서 광류추적 기법을 적용하여 대응점들을 빠르게 찾도록 하였다. 또한 추적 과정에서 소실되는 특징점의 수에 비례하여 새로운 특징점들을 추가하여 특징점의 전체 개수는 일정 수준으로 유지되도록 하였다. 실험 결과 제안하는 추적 방법은 자연특징점 추적 시간을 상당히 단축시킬 뿐만 아니라 카메라 자세 추정 결과도 더욱 안정시킴을 보여주었다.

키워드 : 자연특징 추적; 광류; 특징점 매칭; 증강현실

Fast Natural Feature Tracking Using Optical Flow

Byung-Jo Bae[†] · Jong-Seung Park^{††}

ABSTRACT

Visual tracking techniques for Augmented Reality are classified as either a marker tracking approach or a natural feature tracking approach. Marker-based tracking algorithms can be efficiently implemented sufficient to work in real-time on mobile devices. On the other hand, natural feature tracking methods require a lot of computationally expensive procedures. Most previous natural feature tracking methods include heavy feature extraction and pattern matching procedures for each of the input image frame. It is difficult to implement real-time augmented reality applications including the capability of natural feature tracking on low performance devices. The required computational time cost is also in proportion to the number of patterns to be matched. To speed up the natural feature tracking process, we propose a novel fast tracking method based on optical flow. We implemented the proposed method on mobile devices to run in real-time and be appropriately used with mobile augmented reality applications. Moreover, during tracking, we keep up the total number of feature points by inserting new feature points proportional to the number of vanished feature points. Experimental results showed that the proposed method reduces the computational cost and also stabilizes the camera pose estimation results.

Keywords : Natural Feature Tracking, Optical Flow, Feature Matching, Augmented Reality

1. 서 론

시각기반 증강현실 응용 시스템을 구현하기 위한 방법은 정형화된 형태의 패턴을 사용하는 마커 기반과 사용자 임의의 형태를 사용하는 자연특징 기반의 증강현실 응용 시스템

으로 분류될 수 있다. 이동성이 보장된 모바일 증강현실 구현을 위하여 많은 연구가 진행되어왔다. back-pack 형태의 시스템, 모바일 기기에서 클라이언트-서버 방식으로 네트워크를 사용하는 방식 등의 여러 연구가 있었다[1-4]. 최근의 증강현실 시스템은 임의의 형태를 가지는 패턴을 사용하는 자연특징 기반의 증강현실 응용 시스템을 개발하는데 많은 노력을 기울이고 있다[5-7]. 자연특징 기반 응용 시스템을 구현하기 위하여 사용되는 자연특징을 추출하는 알고리즘은 대표적으로 Scale-Invariant Feature Transform (SIFT)[8]와 Speeded-Up Robust Features (SURF)[9]가 있다. 자연특

* 이 논문은 2008년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한 국학술진흥재단의 지원을 받아 연구되었음(KRF-2008-331-D000560).

† 준 회 원 : 인천대학교 컴퓨터공학부 석사과정

†† 종 신 회 원 : 인천대학교 컴퓨터공학부 부교수(교신저자)

논문접수: 2010년 5월 24일

수 정 일: 2010년 9월 2일

심사완료: 2010년 9월 6일

징을 추출하는 것은 복잡한 문제이며 많은 계산 과정을 거친다. 이러한 자연특징 기반 증강현실 응용 시스템의 동작을 위하여 일반적으로 강력한 컴퓨팅 성능이 필요하며 이때문에 저사양의 모바일 기기에서 자연특징 기반 증강현실 응용 시스템을 구현하는데 어려움이 따른다.

(그림 1)은 SURF 알고리즘을 사용하고 등록된 패턴의 개수를 각각 1개, 5개, 10개로 하고 카메라 입력영상을 동일하게 하여 추적하는 과정에서의 카메라로부터 입력 영상을 받아오는 시간, 입력된 카메라 영상에서 실시간으로 특징점을 추출하는 시간, 그리고 카메라 영상과 등록된 패턴과의 매칭을 하는 과정의 시간, 마지막으로 간단한 3D 객체를 삽입하여 생성된 증강공간을 렌더링하는 시간을 측정하여 비교한 것이다. 전체 처리시간에서 입력되는 카메라 영상에서 실시간으로 특징점을 추출하는 시간은 50% 이상을 차지하며 전체 처리시간에서 큰 비율을 차지하고 있음을 확인할 수 있다. 또한 등록된 패턴의 개수가 증가하면서 패턴의 매칭시간이 증가하고 있으며 이것은 등록된 패턴의 개수가 늘어날수록 전체시간은 비례하여 늘어나는 것을 알 수 있다.

본 논문에서는 이러한 문제점을 해결하고자 광류(optical flow)를 사용하여 모바일 기기에서 자연특징 기반 증강현실 응용 시스템을 구현하는 방법을 제안한다. 증강공간을 구현하기 위한 카메라 자세 추정 과정을 위하여 기존의 자연특징 기반 증강현실 응용 시스템에서는 자연특징 추출 및 패턴매칭 과정을 반복한다. 이러한 과정은 처리시간이 많은 문제점을 가지고 있으며 저사양의 모바일 기기에서는 부적절한 방법이다. 따라서 반복적으로 이루어지는 자연특징 추출 및 패턴매칭 과정의 처리 과정을 단축하는 방안을 모색하게 되었다. 패턴매칭 과정이 성공하게 되면 패턴의 정보 및 매칭된 특징점 집합을 얻을 수 있다. 매칭된 특징점 집합의 위치 정보를 이용하게 되면 반복적인 자연특징 추출 및 패턴매칭 과정을 생략할 수 있다. 이는 특징점 기반으로 동작하는 자연특징 기반 증강현실에서 기존의 매칭된 패턴 정보와 매칭된 특징점의 위치 정보를 기반으로 증강공간을 구성하기 위한 카메라 자세를 추정할 수 있는 특징을 가지고 있기 때문이다. 이러한 특징을 사용하여 자연특징 추출 및 패턴매칭 과정에 비해 빠른 처리속도를 나타내는 특징점을 추적하기 위한 광류 추적 기법을 사용하여 패턴을 추적하도록 하였다. 패턴추적 기법을 사용하여 증강공간을 제공하는 형태의 증강현실 시스템은 사용자가 빈번한 패턴 인식을 요구하지 않는다는 가정 아래에서 적절한 접근이 될 수 있다. 사용자는 증강된 정보를 관찰하기 위하여 일정시간 이상의 시선을 유지하는 것이 일반적이기 때문에 이러한 가

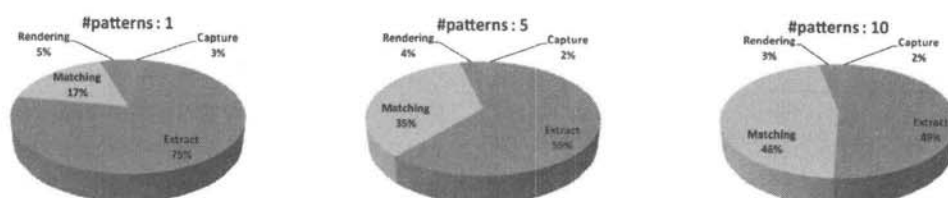
정은 현실적이라 할 수 있다.

제안하는 패턴추적 과정의 기본적인 동작은 기존의 자연특징 추출 및 패턴매칭 과정을 반복하는 것이 아니라 패턴매칭이 성공하게 되면 더 이상의 패턴매칭 과정을 중단하고 광류 추적 기법을 사용하여 매칭된 특징점 집합의 위치 정보를 갱신함으로써 패턴추적 과정을 빠르게 반복하는 방식이다. 이때 사용되는 광류를 사용한 특징점 추적 기법은 비교적 우수한 성능을 나타내는 것으로 알려진 고전적인 Lucas-Kanade 광류 알고리즘[10-11]을 사용하였다. 추가적으로 특징점을 추적하는 과정에서 카메라의 이동 또는 추적 중인 패턴의 움직임에 따라 점차적으로 추적에 실패하는 특징점이 생기게 되는데 이는 정확한 카메라 자세 추정을 할 수 없게 만드는 요인이 된다. 이렇게 정확한 패턴추적을 할 수 없는 경우 패턴추적 과정을 중단하고 다시 패턴매칭 과정을 통하여 새롭게 매칭된 특징점 정보를 얻어야 한다. 이를 위하여 카메라 영상 전체를 다시 패턴매칭 과정을 통하여 매칭하는 것이 아니라 패턴 영역에 특징점 분포에 따라 패턴을 분할하여 분할된 패턴 영역만을 패턴매칭 과정을 통하여 매칭하도록 하였다. 이처럼 분할된 패턴매칭 기법을 사용하여 패턴추적의 정확성을 유지하도록 하였다. 또한 패턴추적 과정이 성공적으로 이루어졌는지를 판단하는 기준을 설정하여 새로운 패턴매칭 과정을 통하여 정확한 증강공간을 유지시킬 수 있도록 하였다.

논문의 구성은 다음과 같다. 2장에서는 자연특징 추출 및 패턴매칭을 위하여 사용된 SURF 알고리즘과 특징점 추적을 위하여 사용된 Lucas-Kanade 광류 추적 기법에 대한 소개한다. 3장에서는 기존의 자연특징 기반 증강현실 시스템의 처리과정과 본 논문에서 제안한 자연특징 추적 과정에 대하여 기술한다. 4장에서는 기존의 자연특징 기반의 증강현실 시스템과 본 논문에서 제안한 광류를 사용한 자연특징 기반 증강현실 시스템을 두 가지 기종의 UMPC에서 비교 테스트한 결과를 제시하고 마지막으로 5장에서 결론을 맺는다.

2. 자연특징 추출 알고리즘과 광류

본 논문에서는 패턴매칭 과정에서 사용되는 자연특징점을 추출하기 위한 방법으로 SURF 알고리즘을 사용하였으며 패턴추적 과정에서 사용되는 특징점 추적 알고리즘은 Lucas-Kanade 광류 추적 기법을 사용하였다. 자연특징 추출 알고리즘인 SURF를 통하여 카메라 영상에서 특징점을 추출하고 등록된 패턴과의 매칭 과정을 거치게 된다. 패턴매칭 과



(그림 1) SURF 알고리즘에서의 프레임 처리시간 비율

정이 성공하게 되면 매칭된 특징점 집합을 얻게 되며 이를 광류 추적 기법을 통하여 지속적으로 추적하게 되고 이를 통하여 카메라 자세를 추정하게 된다. 이 장에서는 본 논문에서 사용하는 SURF 알고리즘과 Lucas-Kanade 광류 추적 기법에 대하여 간략하게 소개한다.

2.1 SURF 특징점 추출

SURF(Speeded-Up Robust Features) 알고리즘은 SIFT와 유사한 알고리즘으로 강건한 특징점을 추출하는 알고리즘이며 기존의 SIFT에 비하여 빠른 처리속도를 보인다. 이 알고리즘은 SIFT 알고리즘과는 다르게 근사화된 Hessian 검출기를 사용하여 빠른 속도로 동작하도록 하고 있다. 또한 SURF는 적분영상(integral images)을 사용한다. 적분영상 $I_{\Sigma(x)}$ 는 식 (1)의 간단한 연산을 통하여 구할 수 있다. 여기서 $\mathbf{x} = (x, y)^T$ 는 이미지에서의 픽셀 좌표를 의미한다. 생성된 적분 영상을 사용하여 크기에 상관없이 사각 영역에 대한 픽셀 값의 합을 쉽고 빠르게 얻을 수 있다.

$$I_{\Sigma(x)} = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (1)$$

Hessian 검출기는 식 (2)에 정의된 Hessian 행렬식에 기반을 둔 특징점 추출 알고리즘으로 속도와 정확도면에서 좋은 성능을 보인다.

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \quad (2)$$

식 (2)에서 $L_{xx}(X, \sigma)$ 는 이미지 I 에서의 점 $X = (x, y)$ 를 Gaussian 함수의 x 방향 2차 미분값 $\frac{\partial^2}{\partial x^2}g(\sigma)$ 으로 회전한 값을 의미하고, L_{xy} 는 xy 방향으로, L_{yy} 는 y 방향으로의 2차 미분값을 나타낸다. 근사화된 Hessian 검출기는 Hessian 행렬식을 사용하는 대신 근사화된 사각필터를 이용하여 근사화된 Hessian 행렬식을 사용한다.

또한 스케일에 강건한 특징점을 얻기 위해 입력되는 영상의 크기를 축소 및 확대하는 것이 아니라 사각필터의 크기를 변화시켜 특징점을 추출한다. 결과적으로 사각필터의 회전연산을 진행하는데 앞서 얻어진 적분영상을 이용함으로써 사각 영역의 크기에 상관없이 빠르게 Hessian 행렬식을 구성하고 특징점을 찾아낸다.

회전에 강건한 특성을 가지도록 하기 위하여 특징점의 주방향을 계산한다. 주방향을 계산하기 위하여 특징점이 찾아진 스케일 공간 s 에서 특징점을 중심으로 $6s$ 반경 내의 이웃하는 픽셀들에 대하여 x, y 방향으로의 Haar Wavelet 연산을 거치게 되며 이 연산을 통하여 특정 방향을 가리키는 특징점의 주방향을 결정한다.

2.2 광류 추적

광류란 두 개의 연속적인 영상 사이에서 특정한 객체가

가지는 특징점의 움직임에 대한 정보를 나타낸다. 영상에서 추적하고자 하는 객체의 특징점이 가지는 영상 밝기값이 이전 영상에서의 위치와 현재 영상에서의 위치를 비교하여 두 영상에서의 변위량을 계산하는 방법이다[12]. 본 논문에서는 특징점의 움직임을 추적하기 위하여 작은 사각 영역을 사용하여 추적하는 광류 추적 기법으로 빠른 처리속도를 내는 Lucas-Kanade 광류 방법[13]을 사용하였다.

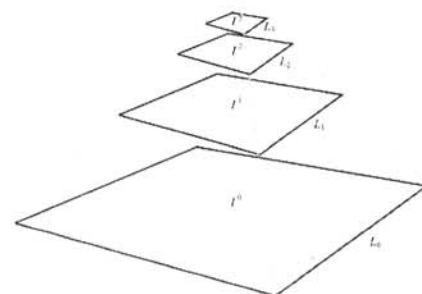
광류를 계산하기 위해서는 카메라 영상에서 추적하고자 하는 객체의 특징점이 가지는 밝기값은 시간이 지나도 일정하다는 것을 가정으로 한다. 광류식은 식 (3)과 같이 나타낼 수 있다.

$$I(\mathbf{x}, t) = I(\mathbf{x} + \mathbf{u}, t + 1) \quad (3)$$

식 (3)은 추적중인 특징점의 위치 $\mathbf{x} = (x, y)^T$ 에서의 영상 밝기값을 $I(\mathbf{x}, t)$ 로 나타내며, t 는 시간을 나타내는 변수이다. $\mathbf{u} = (u_1, u_2)^T$ 는 평면상에서의 추적중인 특징점이 이동하는 속도를 나타낸다[14]. 또 다른 하나의 가정은 추적중인 객체가 시간의 흐름에 비하여 작은 움직임을 갖는다는 것이다. Lucas-Kanade 광류 추적 방법은 카메라 영상에서 특징점을 추적을 위하여 일정한 크기의 적분 윈도우 영역을 사용하여 특징점의 움직임을 계산한다. 특징점의 움직임을 추적하기 위하여 윈도우 패치를 사용할 경우 어퍼처 문제(aperture problem)가 발생할 수 있다. 이 때문에 본 논문에서는 (그림 2)과 같이 영상 피라미드를 사용하여 어퍼처 문제를 해결하고 빠른 카메라 움직임에서도 정확한 특징점의 추적이 가능하도록 피라미드를 사용한 Lucas-Kanade 광류 추적 기법[10]을 사용하였다.

2.3 영상 피라미드를 사용한 Lucas-Kanade 광류 추적 기법

영상 피라미드를 사용한 Lucas-Kanade 광류 추적 기법은 2.2절에서 설명한 바와 같이 특징점 추적을 위하여 윈도우 패치를 사용하여 특징점의 움직임을 계산할 때 발생하는 어퍼처 문제를 해결한다. 영상 피라미드는 가장 높은 해상도의 영상에서부터 절반씩 감소시켜 축소된 영상을 생성하게 된다. 예를 들어 640×480 사이즈의 가장 높은 해상도의 영상을 I^0 라고 한다면 다음 레벨의 영상은 I^0 로부터 1/2배 축소된 320×240 사이즈의 영상 I^1 을 생성하고, 다음 레벨은



(그림 2) 영상 피라미드의 구성

I^1 으로부터 1/2배 축소된 160×120 사이즈의 영상 I^2 를 생성하게 된다. 다음 레벨은 위와 같은 방법을 반복하여 80×60 사이즈 I^3 , 40×30 사이즈 I^4 를 생성하게 된다.

(그림 2)와 같이 생성된 피라미드 영상 I^0, I^1, I^2, I^3, I^4 에서 특징점을 추적하게 된다. I^0 에서의 특징점의 위치를 $u = [u_x, u_y]$ 이라 하고 피라미드 영상의 레벨 $L = 0, \dots, L_m$ 에서의 특징점 위치는 $u^L = \frac{u}{2^L}$, $u^L = [u_x^L, u_y^L]$ 로 나타낼 수

있다. 특징점 u 는 가장 낮은 피라미드 레벨 I^4 에서부터 계산되며 계산된 결과는 한 단계 상위 레벨인 I^3 으로 전달된다. 이러한 과정을 통하여 가장 상위 레벨인 I^0 에서의 특징점 위치를 계산하여 특징점을 추적한다.

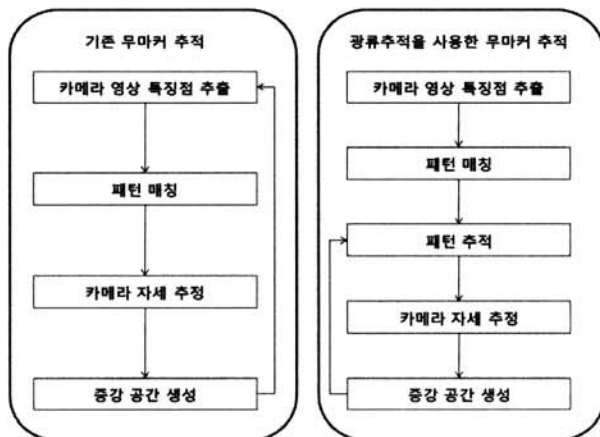
3. 광류를 사용한 무마커 추적

3.1 기존의 자연특징 추적 과정

기존의 자연특징 추적방법은 (그림 3)의 좌측의 그림과 같이 연속적으로 입력되는 카메라 영상에서 매 프레임 자연특징점을 추출하고 이를 등록된 패턴의 특징점과 비교하는 패턴매칭 과정을 거친다. 패턴매칭을 통하여 생성된 매칭된 특징점 집합을 사용하여 카메라 자세를 추정하고 이를 바탕으로 증강공간을 생성한다.

본 논문에서 제안하는 광류추적을 사용한 추적 방법은 (그림 3)의 오른쪽 그림과 같다. 제안하는 방법은 패턴매칭 과정까지는 기존의 자연특징 추적방법과 동일하게 진행한다. 그러나 기존 방법과는 다르게 한번 패턴매칭이 성공하게 되면 매 프레임마다 패턴매칭 과정을 반복하는 것이 아니라 더 이상의 패턴매칭 과정을 중단하고 빠른 광류 추적을 사용하여 매칭된 특징점 집합을 추적한다. 첫 번째 매칭된 프레임 이후는 광류 추적을 통하여 위치 정보가 갱신된 매칭된 특징점 집합을 사용하여 카메라의 자세를 추정한다.

(그림 4)를 통하여 SURF 알고리즘을 사용하여 증강현실을 시스템을 구현하는 과정을 자세하게 설명한다. 기존의



(그림 3) 기존 자연특징 추적 방법 흐름도(좌), 제안된 방법을 사용한 자연특징 추적 방법 흐름도(우)

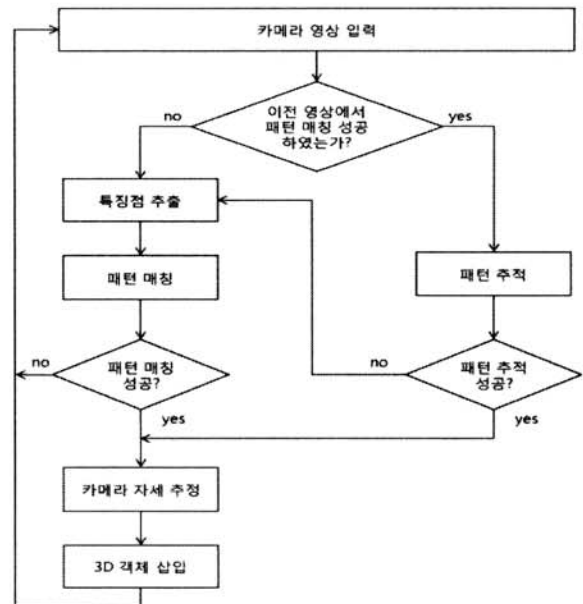
- 단계 1. 카메라로부터 영상을 입력받는다.
- 단계 2. SURF 알고리즘을 통하여 입력된 카메라 영상의 자연특징점을 추출한다.
- 단계 3. 미리 등록된 패턴의 특징점과 카메라 영상의 특징점을 유사도에 따른 비교를 통하여 패턴매칭을 한다. 이 과정에서 매칭된 특징점 집합을 얻는다.
- 단계 4. 매칭된 특징점 집합을 통하여 카메라 자세를 추정한다.
- 단계 5. 패턴에 지정된 3D 객체를 삽입하여 증강현실 영상을 생성한다.
- 단계 6. 단계 1~5의 과정을 반복한다.

(그림 4) 기존의 자연특징 추적 과정

방법은 카메라 영상에서 계속적으로 자연특징점을 추출하는 과정을 반복하여 매번 새로 입력되는 카메라 영상마다 패턴매칭을 진행하기 때문에 정확도 면에서는 높은 수준을 유지할 수 있다. 그러나 패턴매칭 과정은 특징점을 추출하는 과정은 많은 계산량을 필요로 하며 더불어 등록된 패턴과의 비교과정을 거쳐 패턴매칭을 하기 때문에 등록된 패턴의 개수가 늘어날수록 더 많은 시간이 걸린다. 이 때문에 패턴매칭 과정을 반복적으로 지속하는 것은 하드웨어 성능이 떨어지는 모바일 기기에서는 적합하지 않다.

3.2 광류를 사용한 자연특징 추적 과정

본 논문에서는 광류를 사용하여 유사한 수준의 추적 정확성을 유지하면서 속도가 빠른 방법을 제시한다. (그림 5)는 본 논문에서 제안한 광류 추적을 사용하는 자연특징 추적과정의 간략한 흐름도를 나타낸 것이다. 가장 먼저 특징점을 추출하여 특징점의 유사도를 비교하여 패턴매칭이 성공하게 되면 광류 추적을 위하여 초기 특징점 집합을 지정하게 되고 이를 Lucas-Kanade 광류 추적 기법을 사용하여 추적하게 된다. 광류 추적 기법을 사용하여 지속적인 추적을 유지하는 동안 가려짐 또는 입력영상에서의 사라지는 경우 정확한 매칭 결과를 얻을 수 없기 때문에 이를 해결하기 위하여



(그림 5) 제안된 추적 기법의 흐름도

광류 추적의 지속 여부를 판단하기 위한 기준을 마련하여 광류 추적을 중단하고 새로운 패턴매칭 과정을 위하여 카메라 영상의 특징점 추출 과정부터 다시 수행하게 된다.

(그림 6)은 본 논문에서 제안하는 기존의 SURF 알고리즘에 광류 추적 기법을 추가적으로 사용한 방법을 나타낸 것이다. 기존의 방법과의 차이점은 단계 4~5에서 매칭된 특징점 집합을 Lucas-Kanade 광류 추적 기법을 사용하여 추적하는 것이다. 이를 통하여 처리량이 많은 SURF 알고리즘을 통한 특징점 추출과정을 거치지 않게 된다.

매칭된 특징점 집합을 추적하는 과정은 연속적으로 입력되는 t 번째 카메라영상에서 패턴매칭이 성공하게 되면 n 개의 매칭된 특징점 집합을 얻을 수 있으며 이를 식 (4)와 같이 나타낼 수 있다. 다음으로 입력되는 $t+1$ 번째 카메라영상에서는 t 번째 카메라영상에서 매칭된 특징점 집합의 특징점을 광류 추적 기법을 사용하여 위치 정보가 갱신된 특징점 집합을 얻는다. 이렇게 갱신된 특징점 집합은 식 (5)과 같이 나타낼 수 있다. 만약 t 번째 카메라 영상에서 매칭된 특징점 집합의 일부 특징점이 $t+1$ 번째 카메라 영상에서 추적이 실패하였다면 식 (5)에서와 같이 null 값을 가지게 된다. 식 (5)에서와 같이 특징점의 추적과정 중 추적에 실패하여 null 값을 가진 특징점을 제거하게 되면 특징점 m 개를 가지는 특징점 집합을 얻게 되며 식 (6)과 같이 나타낼 수 있다.

$$F(t) = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)\} \quad (4)$$

$$F'(t+1) = \{(x'_1, y'_1), (x'_2, y'_2), null, \dots, (x'_n, y'_n)\} \quad (5)$$

$$F(t+1) = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}, m \leq n \quad (6)$$

식 (6)과 같이 갱신된 특징점 집합을 사용하여 호모그래피 행렬의 생성과 카메라 자세 추정과정을 거쳐 패턴과 관련된 3D 객체를 삽입하여 증강공간을 구성하게 된다. 그러나 이러한 특징점의 위치를 추적하는 광류 추적 기법 추적을 과정에서는 가려짐 또는 카메라 영상의 빠른 움직임으로 인하여 점차적으로 추적이 실패하는 특징점의 개수가 늘어

나게 되고 이는 올바른 카메라 자세 추정 및 정확한 호모그래피 행렬의 생성이 이루어지지 않게 하는 요인이 된다. 이러한 현상을 방지하기 위하여 추적중인 특징점의 개수가 최소한의 개수를 유지하도록 하고 이를 만족하지 못할 경우 새로운 패턴매칭과정을 통하여 정확성을 유지하도록 하였다.

3.3 패턴추적 지속 여부 판단 기준

패턴매칭이 성공하면 추적 특징점 집합을 매칭된 특징점 집합으로 초기화를 하고 Lucas-Kanade 광류 추적 방법을 사용하여 추적을 시작한다. 추적 과정에서 추적에 실패하는 특징점이 점차적으로 늘어나 모두 사라지게 되면 더 이상의 추적이 불가능하게 되고 카메라의 자세 추정 또한 불가능하게 된다. 이때 패턴의 추적을 중단하고 새로운 패턴매칭을 시도하게 된다. 증강공간을 떨림없이 안정적으로 유지하기 위해서는 추적의 연속성을 보장하는 기준이 필요하다. (그림 7)은 패턴의 추적을 지속할지의 여부를 결정하는 기준을 나타내고 있다. 이러한 판단 기준을 검사하기 위하여 매칭된 특징점 집합이 패턴의 영역에 어떻게 분포하는 지를 파악할 필요가 있다. 이를 위하여 본 논문에서는 패턴 분할 레벨을 결정하여 판단하도록 하였다.

(그림 8)은 SURF 알고리즘을 통하여 패턴매칭을 성공하여 매칭된 카메라 영상에서 패턴의 영역 내부에 포함되어 있는 특징점의 위치를 X표시를 통하여 나타낸 것이다. 분할된 영역 내에 하나 이상의 특징점이 위치해 있는 것은 음영으로 나타내었다. 패턴매칭 영역을 4, 9, 16개의 영역으로 분할하여 각 분할된 모든 영역에 적어도 하나 이상의 특징점이 위치하고 있는 최고의 레벨을 선택한다. (그림 8)와 같은 특징점 분포의 경우는 Level-2를 선택하게 된다.

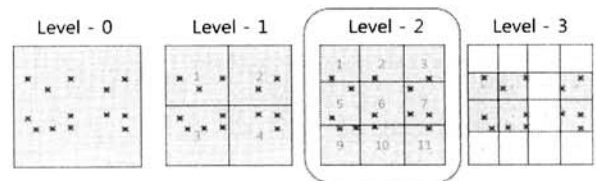
(그림 9)는 패턴이 왼쪽 하단에서 우측 상단으로 이동하는 모습을 나타내고 있으며 결정된 패턴 레벨의 영역을 나누는 각 지점별 좌표를 나타내고 있다. 외곽의 검은 사각영역은 카메라 영상을 나타내는 것이며 카메라영역을 벗어난 영역의 각 지점을 빨간색으로 나타내고 있다. 특징점을 추적하는 중에 가려짐에 의하여 분할된 각각의 영역에 특징점이 사라지는 경우가 발생하게 된다. 이러한 경우 특징점이

- 단계 1. 카메라로부터 영상을 입력받는다.
- 단계 2. SURF 알고리즘을 통하여 입력된 카메라 영상의 자연 특징점을 추출한다.
- 단계 3. 미리 등록된 패턴의 특징점과 카메라 영상의 특징점을 유사도에 따른 비교를 통하여 패턴매칭을 한다. 이 과정에서 매칭된 특징점 집합을 얻는다.
- 단계 4. 매칭된 특징점 집합에서 카메라 영상의 특징점 좌표를 Lucas-Kanade 광류 추적 기법을 사용하여 추적한다.
- 단계 5. 광류 추적 중 가려짐 등에 의하여 추적에 실패하는 특징점의 개수 및 정확도를 유지하고자 패턴 분할 매칭 기법을 적용하여 새로운 매칭이 필요한지와 지속적인 특징점의 추적을 유지하는지를 결정한다.
- 단계 6. 추적을 통하여 갱신된 특징점 집합을 통하여 카메라 자세 추정을 한다.
- 단계 7. 패턴에 지정된 3D 객체를 삽입하여 합성된 증강공간 영상을 생성한다.
- 단계 8. 광류추적의 지속여부를 판단하는 기준에 부합한다면 단계 5~7을 반복한다. 그렇지 않다면 다시 단계 1로 돌아가 새로운 패턴매칭 과정을 시작한다.

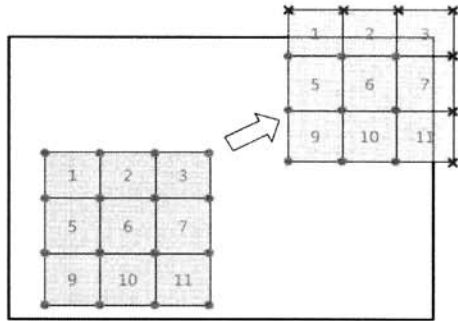
(그림 6) 제안하는 자연특징 추적 방법

- 기준 1. 처음 매칭이 성공하여 매칭된 특징점으로 초기화 되었을 때의 특징점의 총 개수보다 50% 이하로 떨어졌을 경우 추적을 중단한다.
- 기준 2. 매칭된 패턴의 코너점이 가지는 좌표가 카메라 영상의 2배를 넘어가는 범위에 있다면 추적을 중단한다.
- 기준 3. 레벨에 따라 분할된 영역에 최소 1개 이상의 특징점이 분포하지 않을 경우 추적을 중단한다.

(그림 7) 패턴의 추적 지속 여부 판단 기준



(그림 8) 패턴 레벨 결정 과정

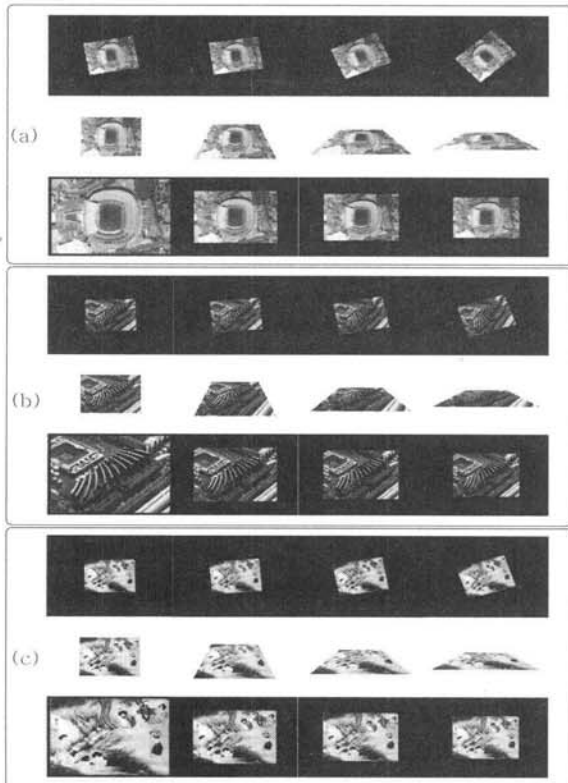


(그림 9) 분할된 영역의 부분 매칭을 위한 기준

사라진 분할 영역의 특징점을 새롭게 SURF 알고리즘을 통하여 추출하게 되고 부분적으로 패턴매칭을 시도하게 한다. 이때 분할된 영역을 지정하는 4개의 코너점이 카메라 영상 내부에 모두 위치해 있다면 정상적으로 새로운 부분 패턴매칭을 시도한다. 그러나 모든 코너점이 존재하지 않는다면 카메라 영역을 벗어난 것이므로 부분매칭을 시도하지 않고 나머지 부분 영역에서 추적중인 특징점만을 사용하여 호모그래피 행렬을 생성하고 패턴매칭 영역을 추정한다.

4. 실험 결과

여러 실영상 데이터 집합을 사용하여 제안하는 추적 기법의 정확성과 처리 시간을 분석하였다. 기존의 SURF 알고리즘을 단독으로 사용한 방식과 본 논문에서 제안한 기법을



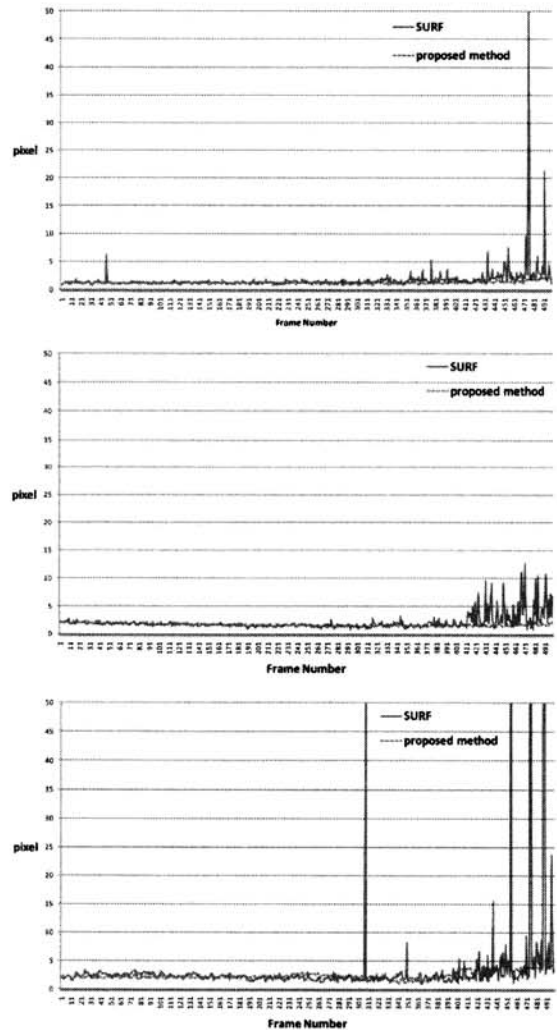
(그림 10) 테스트 영상 집합. (a) Test_set_1, (b) Test_set_2, (c) Test_set_3

적용한 방식의 처리시간의 비교를 위하여 데스크탑 PC(Intel i7 920)와 2종의 UMPC를 사용하였다. 실험에 사용된 UMPC 기종은 Sony VGN-UX57LN(UMPC1)와 Samsung NT-Q1u(UMPC2)를 사용하였다.

4.1 정확성 분석

미리 등록된 한 패턴의 모습을 카메라가 움직이며 촬영하는 상황에서 추적의 정확성을 측정하였다. 카메라가 움직이므로 패턴의 모습은 카메라의 자세에 따라서 회전, 기울임, 크기 변화가 적용된다.

(그림 10)은 정확성 실험을 위하여 사용된 회전, 기울임, 크기 변화를 준 실험용 테스트 영상 집합들이다. 각 영상 집합에서 가장 위의 줄은 카메라 회전이 적용되는 경우이고, 중간 줄은 기울임(tilt)이 적용되는 경우이고, 아래 줄은 줌(zoom)이 적용되는 경우이다. 회전 영상들은 목표 영상을 회전시켜 350장의 영상을 생성하였다. 기울임 영상들은 x축을 기준으로 90°부터 0°까지 기울어지는 영상 500장을 생성하였다. 줌 영상들은 패턴 본래 크기의 영상에서 약 1/10배



(그림 11) 기울임에 대한 평균 오차: 상(Test_set_1), 중(Test_set_2), 하(Test_set_3)

축소한 영상까지 크기를 변화시킨 영상 170장을 생성하였다. 기존의 SURF 알고리즘만을 사용한 추적 방법과 본 논문에서 제안한 추적 기법을 각각 실험용 영상 (a), (b), (c)에 적용하였다. 이후 추적 결과로부터 추정되는 패턴의 코너 좌표를 비교하였다. 미리 알고 있는 실험용 영상의 패턴 코너 좌표와 패턴매칭 및 패턴추적 과정을 거쳐 출력된 코너 좌표의 각각의 차를 구하여 평균 오차를 구하였다.

(그림 11)은 기울임 영상 집합을 사용하여 측정된 실험 결과를 나타내고 있다. 기존의 SURF 알고리즘을 사용한 방법은 약 45°의 기울임을 적용한 영상부터 오차가 급격하게 커짐을 확인할 수 있다. 한편 본 논문에서 제안한 방법을 사용한 추적 결과는 오차가 점차 증가하지만 기존의 방법과 같이 큰 오차를 보이지 않았다.

(그림 12)는 회전 영상 집합을 사용하여 측정한 평균 오차이다. 정면을 기준으로 회전한 영상들에 SURF 알고리즘을 적용하는 경우에는 정확성 측면에서 전반적으로 우수한 결과를 나타냈다. 그러나 급격히 오차가 증가하는 프레임들이 존재하였다. 한편 제안한 방법의 경우는 점차적으로 오차가 증가하지만 갑작스런 오차 발생의 결과는 발생하지 않는 것을 확인할 수 있었다. 이러한 결과는 제안 방법이 이

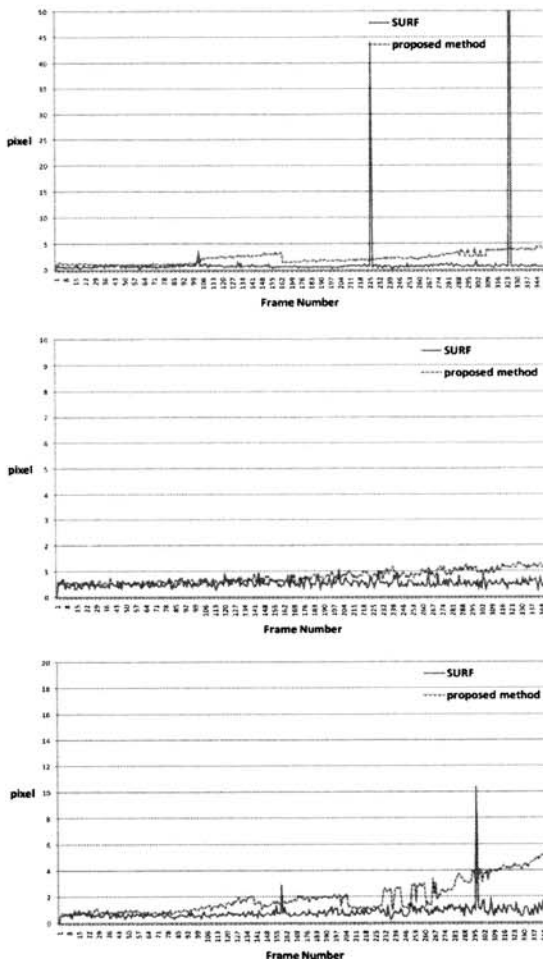
전 프레임에서의 매칭에 사용된 특징점들을 다음 프레임에서 우선적으로 추적하는 특성에 기인한 것이다.

(그림 13)은 줌 영상들을 사용하여 측정한 평균 오차를 나타내고 있다. 기존의 방법을 적용한 결과 제안된 방법에 비하여 정확도 측면에서는 전반적으로 우수한 결과를 보였다. 또한 패턴매칭 실패가 불규칙적으로 발생하는 결과를 보였다. 그러나 제안된 방법의 경우 패턴매칭의 정확도는 유사한 수준을 유지하면서 급격한 오차유발 프레임이 없이 안정된 결과를 보였다.

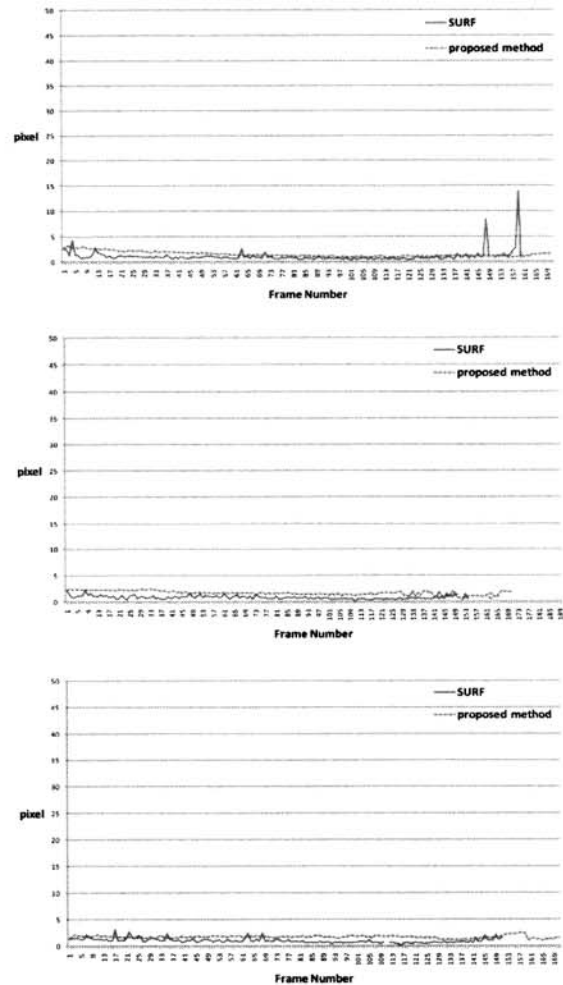
4.2 처리 시간 분석

실험 영상 집합들에 대해서 각 프레임의 처리시간을 비교 분석하였다. 처리시간의 비교 실험을 위하여 총 300개 프레임의 카메라 입력 영상을 사용하였다. 실험에서 등록된 패턴의 개수는 10개로 하였으며, 각각의 패턴영상에서 추출된 특징점의 개수는 평균 230개 정도를 가지도록 하였다.

(그림 14)는 기존의 방법을 사용하여 카메라 영상의 특징점을 추출 과정에서 걸리는 시간을 측정하였다. UMPC1은 평균 170ms, UMPC2는 평균 306ms의 처리 시간이 소요



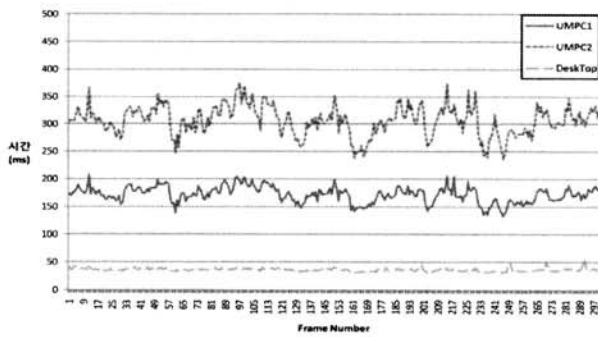
(그림 12) 회전에 대한 평균 오차: 상(Test_set_1), 중(Test_set_2), 하(Test_set_3)



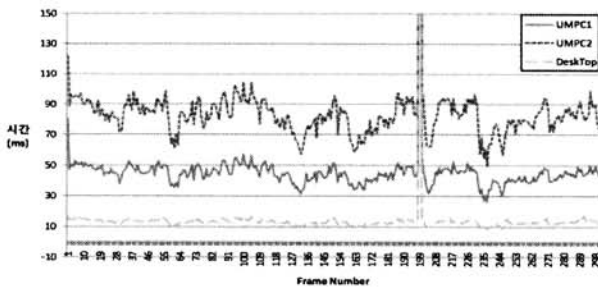
(그림 13) 줌에 대한 평균 오차: 상(Test_set_1), 중(Test_set_2), 하(Test_set_3)

되었다. (그림 15)는 패턴매칭 과정에서 걸리는 시간을 나타낸 것이다. UMPC1은 평균 50ms, UMPC2는 평균 90ms의 처리 시간이 소요되었다.

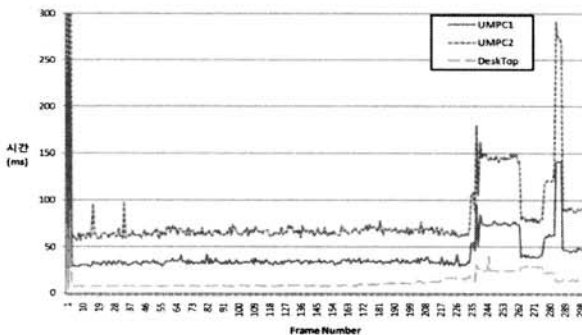
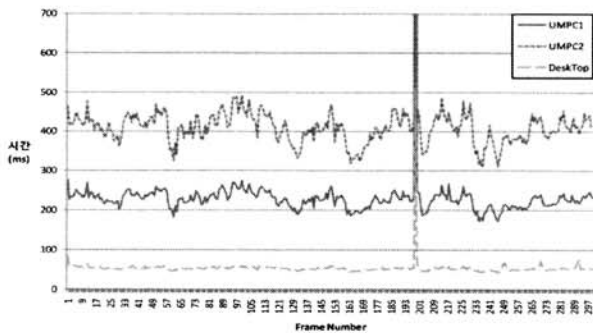
(그림 16)에서는 기존의 자연특징 추적 방법과 본 논문에서 제안한 방법을 사용하여 하나의 카메라영상 프레임을 처리하는데 걸리는 시간을 나타낸 것이다. 기존의 방법은 한 프레임을 처리하는데 UMPC1은 평균 230ms, UMPC2는 평



(그림 14) 실행 환경에 따른 기존 방법의 특징점 추출 시간 비교



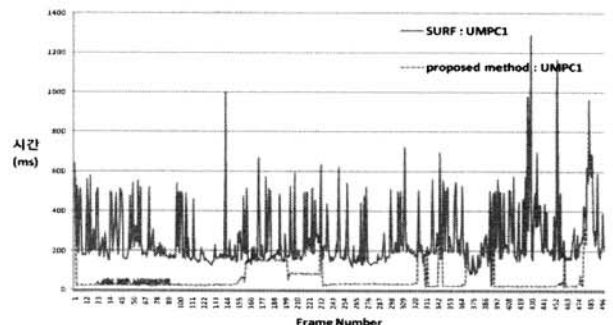
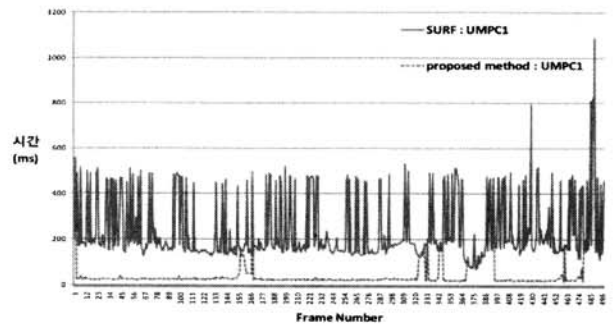
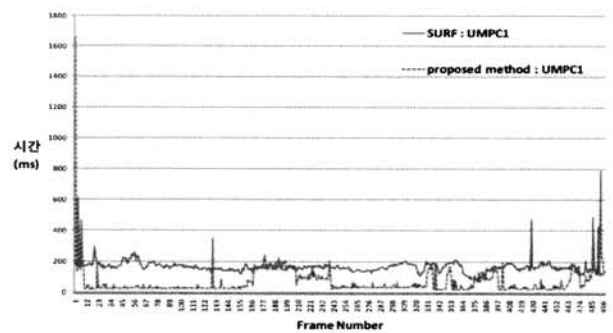
(그림 15) 실행 환경에 따른 기존 방법의 패턴매칭 시간 비교



(그림 16) 전체 처리 시간 비교: 기존의 방법(상), 제안된 방법(하)

균 400ms 전후의 처리시간이 걸리는 것을 확인할 수 있다. 반면 제안한 추적 기법을 적용한 경우에는 UMPC1은 평균 40ms, UMPC2는 평균 80ms의 처리시간을 나타내고 있다. 240번째 프레임과 290번째 프레임에서 처리시간이 급격히 증가함을 볼 수 있다. 이는 패턴추적이 실패하여 새로운 패턴매칭 과정을 수행하기 때문이다.

추가적으로 총 500프레임의 입력데이터를 사용하여 UMPC1에서의 전체 처리시간을 비교하였다. 패턴의 개수는 위의 실험과 동일하게 1개, 5개, 10개로 증가시키면서 진행하였다. (그림 17)은 UMPC1에서 기존의 방법과 제안된 방법으로 하나의 영상 프레임을 처리할 때 걸리는 전체 시간을 측정하여 비교한 것이다. 동일한 조건에서 패턴의 개수를 변경하여 실험을 진행하였다. 기존의 방법에서 패턴의 개수를 1개, 5개, 10개씩 변경하였을 때 각각 평균 160ms, 230ms, 260ms의 처리 시간이 소요되었다. 반면 제안된 방법을 사용하였을 경우 각각 평균 71ms, 52ms, 70ms의 처리 시간이 소요되었다. 기존의 방법에서는 패턴의 개수가 증가할수록 단일 프레임을 처리하는데 걸리는 시간이 늘어남을 확인할 수 있었다. 한편 제안된 방법에서는 패턴의 개수가 증가해

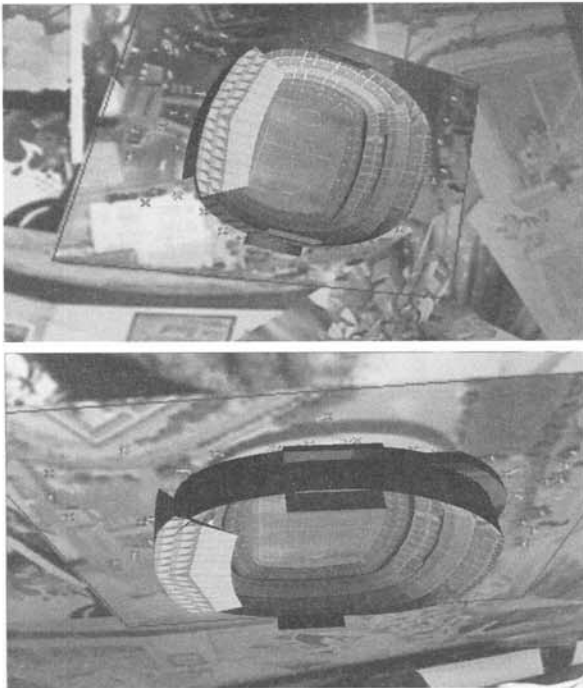


(그림 17) 패턴의 개수에 따른 프레임 처리 시간 비교: 1개(상), 5개(중), 10개(하)

도 처리시간이 크게 늘어나지 않았다. 이는 제안된 방법이 추적중인 패턴의 개수보다는 추적중인 특징점의 개수에 의존적이기 때문이다. 그래프에서 처리시간이 갑자기 증가하는 부분은 패턴추적이 실패한 뒤 패턴매칭의 과정을 다시 거치기 때문이다.

4.3 증강현실 응용

제안된 추적 방법은 실시간 증강현실 응용에 즉시 사용될 수 있다. (그림 18)은 본 논문에서 제안한 방법을 사용하여 패턴추적 과정을 거친 후에 3D 객체를 삽입한 화면이다. 축구경기장의 위성사진을 목표 패턴으로 하여 패턴 상에 실제 경기장과 유사한 경기장 3D 모델을 삽입하는 증강현실 응용시스템이다. 파란색으로 표시된 사각 영역은 특징점 추적으로 얻어진 패턴 영역을 나타내고 있다. 녹색으로 표시된 점은 추적 중인 특징점을 표시한 것이다. 매칭된 특징점에 대하여 광류를 사용하기 때문에 처음 매칭이 올바르게 이루어졌다면 큰 각도의 변화에도 유연한 카메라 자세 추정이 가능함을 확인할 수 있다.



(그림 18) 제안된 추적 방법을 사용한 실시간 증강현실 응용

5. 결론 및 향후연구

일반적인 자연특징 추적 기법은 연속적으로 입력되는 카메라 영상에서 특징점을 추출하고 등록된 패턴의 특징점과 유사도를 비교하여 매칭하고 매칭된 특징점 집합을 사용하여 카메라 자세를 추정하는 과정을 거친다. 이러한 반복적인 패턴매칭 과정으로 인한 처리시간의 증가는 모바일 기기와 같은 저사양의 컴퓨팅 환경에서는 실시간으로 동작하는데에 어려움이 있다. 이러한 문제를 해결하기 위하여 본 논

문에서는 패턴매칭 과정을 통하여 패턴이 인식된 이후에는 카메라 영상에서 새로운 특징점을 추출하여 매칭하는 것이 아니라 광류 추적 기법을 통하여 패턴을 추적하는 방법을 제안하였다. 실험에 사용한 자연특징 추출 알고리즘은 SURF 기법을 사용하였고 광류 추적을 위하여 Lucas-Kanade 광류 기법을 사용하였다.

패턴추적 과정은 먼저 자연특징 추출을 통한 패턴매칭 과정을 거치게 된다. 이 과정을 거치게 되면 매칭이 성공된 카메라 영상에서의 특징점 집합의 위치 정보를 가지게 된다. 이 매칭된 특징점 집합의 위치정보를 두 번째 과정인 광류 추적을 위한 초기값으로 전달하게 되고 이를 Lucas-Kanade 광류 기법을 사용하여 추적하며 카메라 자세 추정을 한다. 패턴추적 과정 중에 추적중인 특징점이 카메라 영상을 벗어나는 경우가 생기게 되는데 이는 카메라 자세 추정을 부정확하게 만든다. 이러한 추적이 실패하는 특징점이 점차적으로 늘어나고 정확히 일치된 증강공간을 유지하기 힘들어진다. 이러한 문제를 해결하기 위하여 추적 중인 패턴의 특징점 집합의 분포를 조사하여 분포에 따라 입력되는 카메라 영상을 분할하여 관리하는 방법을 사용하였다. 분할된 영역 내에 위치하는 특징점의 개수를 조사하여 최소 개수를 유지하지 못하는 영역은 새롭게 자연특징 추출 과정과 패턴매칭 과정을 거쳐 매칭된 특징점 집합을 얻도록 하였다. 여기서 얻어진 특징점 집합은 기존의 패턴 추적 과정에 사용되는 특징점 집합에 포함된다. 이러한 분할된 영역만을 매칭하는 과정을 통하여 처리 시간의 증가에 따른 화면의 끊김 현상을 줄일 수 있었다.

특징점 기반 증강현실에서 이상점(outlier)은 호모그래피 행렬의 부정확하게 하는 요인이 된다. 본 논문에서 사용한 SURF 알고리즘은 RANSAC[15]을 사용하여 이상점을 제거하는 과정을 거쳤지만 상당수의 이상점이 여전히 제거되지 않는 문제점이 있다. 또한 장시간 특징점 추적을 하게 되면 추적중인 특징점의 오차가 누적되어 부정확한 카메라 자세 추정이 이루어지는 문제점이 있다. 향후 이러한 문제점을 해결하여야 정확한 증강공간을 생성할 수 있을 것이다.

참 고 문 헌

- [1] R. Azuma, "A survey of augmented reality," Presence: Teleoperators and Virtual Environments, Vol.6, No.4, pp. 355-385, 1997.
- [2] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway, "Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System," IEEE Computers Graphics and Applications, Vol.23, No.6, pp.779-785, 1999.
- [3] C. Geiger, B. Kleinnjohann, C. Reimann, and D. Stichling, "Mobile AR4ALL," International Symposium on Augmented Reality 2001, pp.181-182, 2001.
- [4] M. Assad, D. J. Carmichael, D. Cutting, and A. Hudson, "A

Demonstration of Mobile Augmented Reality,” In OzCHI Demo Section, 2003.

[5] David G. Lowe, “Object recognition from local scale-invariant features,” In International Conference on Computer Vision, Corfu, Greece, pp.1150-1157, 1999.

[6] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, D. Schmalstieg, “Pose Tracking from Natural Features on Mobile Phones,” In Proc. of ISMAR 2008, pp.125-134, 2008.

[7] Wagner, D., Schmalstieg, D., Bischof, H., “Multiple target detection and tracking with guaranteed framerates on mobile phones”, ISMAR 2009. pp.57-64, 2009.

[8] David G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision, Vol.60(2), pp.91-110, 2004.

[9] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded Up Robust Features,” ECCV, pp.404-417, 2006.

[10] J.Y. Bouguet, “Pyramidal Implementation of the Lucas-Kanade Feature Tracker,” Technical Report, Intel Corporation, Microprocessor Research Labs, 2000.

[11] B. Lucas, T. Kanade, “An iterative image registration technique with an application to stereo vision,” In Proc. Seventh International Joint Conference on Artificial Intelligence, Vancouver, Canada, pp.674-679, 1981.

[12] Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., Schmalstieg, D., “Real-Time Detection and Tracking for Augmented Reality on Mobile Phones,” Visualization and Computer Graphics, Vol.16, pp.355-368, 2010.

[13] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt, “Performance of Optical Flow Techniques,” CVPR, pp. 236-242, 1992.

[14] G. Papagiannakis, G. Singh, N. Magnenat-Thalmann, “A survey of mobile and wireless technologies for augmented reality systems,” Computer Animation and Virtual Worlds, Vol.19, pp.3-22, 2008.

[15] M. A. Fischler, R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” ACM, Vol.24, pp 381-395, 1981.



배 병 조

e-mail : wings810@incheon.ac.kr

2010년 인천대학교 컴퓨터공학부(공학사)

2010년~현 재 인천대학교 컴퓨터공학부 석사과정

관심분야 : 동작기반 인터페이스, 증강현실, 카메라 추적



박 종 승

e-mail : jong@incheon.ac.kr

1992년 경북대학교 전자계산학과(이학사)

1994년 포항공과대학교 컴퓨터공학과(공학 석사)

1999년 포항공과대학교 컴퓨터공학과(공학 박사)

2004년~현 재 인천대학교 컴퓨터공학부 부교수
관심분야 : 영상처리, 3D 비전, 증강현실, 게임공학