

긴 비디오 프레임들에서의 강건한 2차원 특징점 추적

윤 종 현[†] · 박 종 승^{**}

요 약

비디오 영상 프레임들에서 2D 특징점들을 지속적으로 추적하는 문제는 프레임 간의 빈번한 특징점 매칭 실패로 인하여 어려움을 겪어왔다. 본 논문에서는 긴 비디오 프레임들에서 강건하게 2D 특징점들을 추적하는 기법을 제안한다. 이전 프레임까지 추적되어온 각 특징점에 대해 움직임 상태변수를 정의하고 이들 상태변수로부터 현재 프레임에서의 움직임을 예측한다. 예측된 움직임은 추적을 위한 탐색 윈도우를 설정을 위한 초기 위치로 지정된다. 유사성 검사를 통해서 탐색 윈도우 내에서 대응점을 결정한다. 측정 데이터를 반영하여 현재 프레임에서의 특징점의 움직임 상태 변수를 수정하는 과정을 갖는다. 특징점의 추적 결과는 오차를 포함하고 있고 잘못된 추적이 발생할 수 있다. 잘못 추적된 이상값들은 RANSAC 알고리즘을 적용하여 제거함으로써 정확한 특징점 추적이 지속될 수 있도록 한다. 실제 비디오 프레임들에 대해 특징점 추적을 실시한 결과 긴 비디오 프레임들에 대해서도 특징점 추적이 안정적으로 수행됨을 확인할 수 있었다.

키워드 : 특징점 추적, 이상점 제거, RANSAC

Robust 2D Feature Tracking in Long Video Sequences

Jong-Hyun Yoon[†] · Jong-Seung Park^{**}

ABSTRACT

Feature tracking in video frame sequences has suffered from the instability and the frequent failure of feature matching between two successive frames. In this paper, we propose a robust 2D feature tracking method that is stable to long video sequences. To improve the stability of feature tracking, we predict the spatial movement in the current image frame using the state variables. The predicted current movement is used for the initialization of the search window. By computing the feature similarities in the search window, we refine the current feature positions. Then, the current feature states are updated. This tracking process is repeated for each input frame. To reduce false matches, the outlier rejection stage is also introduced. Experimental results from real video sequences showed that the proposed method performs stable feature tracking for long frame sequences.

Key Words : Feature tracking, Outlier rejection, RANSAC

1. 서 론

특징점 추적은 영상 분석 및 컴퓨터 비전 분야에 있어서 정보 획득의 가장 기본 단계이고 선행 단계에 해당하는 분야중 하나이다. 비디오 영상에서의 추적 시스템은 초기에 주요 시설물이나 공공 장소의 보안을 목적으로 사용되기 시작하였다. 비디오센서, 영상처리, 인터페이스 기술 등이 급속히 발전하고 다양화되었다. 기존의 비디오 기반 추적 기술은 감시, 검색, 추적, 표현을 지능화시킨 비디오 해석 및 추적 기술로 진화하고 있다. 연속되는 영상이 입력으로 주어졌을 때, 각 영상으로부터 특징점을 추출한다. 이렇게 얻어진 특징점들을 사용하여 각 특징점의 추적과정을 거쳐게 되

면, 각 프레임 사이에서 특징점들의 대응관계를 알 수 있게 된다. 이러한 특징점들의 대응 관계는 카메라의 움직임을 추정에 사용되고, 영상에 나타나는 물체의 형태를 추정하기 위한 물체 복원의 기반 데이터로 사용된다. 따라서 연속되는 영상으로부터 대상 객체의 움직임 정보를 추정하기 위해 선행되어야 하는 필수 절차가 정확한 특징점 추적 데이터를 얻는 것이다.

본 논문에서는 긴 비디오 영상에 대하여 강건한 특징점 추적이 가능한 방법을 제안한다. 강건한 특징점 추적을 위해서는 안정적인 위치를 갖도록 특징점을 추출하는 것이 중요하다. 추적에서 사용되는 특징점으로 점이나 선분 또는 윤곽선 등이 모두 가능하나 점의 사용이 가장 기본적이다. 본 논문에서는 영상의 이동, 회전, 스케일링에 대해 안정적인 특징점 추출 방법[1]을 사용하여 추적할 특징점을 획득한다. 입력영상을 가우시안 영상 피라미드를 생성한 후 가우시안

[†] 준 회원 : 인천대학교 컴퓨터공학과 석사과정

^{**} 종신회원 : 인천대학교 컴퓨터공학과 조교수 (교신저자)
논문접수 : 2007년 9월 3일, 심사완료 : 2007년 11월 4일

영상들의 차(difference of Gaussian)를 구한다. 계산된 차이 값이 최대값이거나 최소값인 점들을 특징점으로 추출한다.

안정적인 위치를 갖는 특징점을 추출한 후 현재 프레임 이전까지 추적된 특징점의 상태값의 변화를 사용하여 현재의 상태값을 예측할 수 있고 예측된 상태값으로부터 현재 프레임에서의 특징점의 움직임 위치를 예측할 수 있다. 예상된 특징점의 위치는 현재 프레임에서 특징점의 위치를 결정하기 위한 시작 위치로서 사용된다. 예상된 위치로부터 이전 프레임들로부터 추출된 상태 변수를 사용하여 수정하는 방법으로 현재 프레임에서의 특징점의 위치를 추정한다. 일반적인 특징점 추적 방법과 마찬가지로 입력 데이터로는 비디오 영상 프레임을 사용한다. 출력 데이터로는 각 프레임별 특징점의 추적된 위치 및 연결 궤적 데이터를 얻을 수 있다.

본 논문에서 제안하는 방법에서는 카메라는 물리공간에서 자유로운 움직임을 허용한다. 그러나 물리공간상에서 객체는 움직이지 않는 정적 물체를 가정한다. 카메라와 물체가 같이 움직이게 되는 경우에는 짧은 영상 프레임에 대하여 객체에 대한 가려짐(occlusion) 현상이 발생할 수 있고 움직임의 애매성(ambiguity)이 발생하므로 문제가 훨씬 복잡해지게 된다[2]. 카메라와 독립적으로 움직이는 동적 물체가 있을 경우에는 물체를 탐지하고 특징점 추적 계산 과정에서 정적 물체와 별도로 처리되어야 정확한 추적결과를 얻을 수 있다.

2. 관련 연구

본 장에서는 2D 특징점 추적의 대표적인 방법인 KLT 방법[3]과 SIFT 방법[4]을 기술한다. 대부분의 2D 특징점 추적 방법들이 두 기법에 기반을 하거나 약간의 변형을 주어 제안되었다. 특징점 추출 및 추적 기법의 개선을 위해서는 대표적인 두 방법에 대한 성능 및 문제점을 인식하여야 한다.

2.1 KLT 특징점 추적

KLT(Kanade-Lucas-Tomasi) 추적기법의 특징점 추출에서는 좋은 윈도우를 정의하기 보다는 KLT 추적 방법에서 잘 추적될 수 있는 특징점을 찾는 것에 중점을 두고 있다[5]. KLT 특징점 추적 방법은 다음과 같다[3]. 연속되는 영상 프레임 집합을 $I(\mathbf{x}, t)$ 라 하자. 여기서 \mathbf{x} 는 영상점의 좌표로써 $\mathbf{x}=[u, v]^T$ 와 같이 표현한다. 만약 비디오 스트림에서 초당 프레임 수가 충분히 크다면, 영상에서 명암도의 변화가 거의 없고 영역들의 움직임이 작은 프레임들을 얻을 수 있다. 시간에 따른 영상 프레임의 밝기 변화는 식 (1)과 같이 표현된다.

$$I(\mathbf{x}, t) = I(\delta(\mathbf{x}), t + \tau) \quad (1)$$

여기에서 $\delta(\mathbf{x})$ 은 영상에서 특징점 \mathbf{x} 의 움직임 변환을 나타내는 모션 필드이다. 영상 프레임 간의 간격이 짧다는

가정이 유효하다면 움직임은 단지 이동(translation)에 가까운 움직임으로 표현할 수 있다. 그러므로 $\delta(\mathbf{x}) = \mathbf{x} + \mathbf{d}$ 라고 표현할 수 있고, 여기서 \mathbf{d} 가 \mathbf{x} 의 움직임 변위 벡터를 나타낸다. 특징점 추적 시에 연속되는 여러 영상에 대하여 선택된 특징점들의 더 정확한 대응점들 구하기 위하여, 신뢰성 높은 \mathbf{d} 를 구하는 것이 가장 중요한 문제이다[1].

식 (1)과 같은 영상 움직임 모델은 실제 영상의 경우와는 일반적으로 정확히 일치하지 못하다. 그 원인은 카메라나 물체의 움직임 또는 영상의 잡음이 있기 때문이다. 매칭에 따른 오차를 식 (2)와 같은 SSD(sum of squared difference) 오차로 정의하고 매칭은 이를 최소화하는 변위를 찾는 문제로 정의할 수 있다[6,7].

$$\epsilon = \sum_{\Omega} [I(\mathbf{x} + \mathbf{d}, t + \tau) - I(\mathbf{x}, t)]^2 \quad (2)$$

식 (2)에서 Ω 는 추적할 대상의 특징점을 중심으로 하는 영상에서 작은 영역 윈도우를 의미하고, \mathbf{d} 는 윈도우의 움직임 벡터이다. $(\mathbf{x} + \mathbf{d}, t + \tau)$ 에 1차 Taylor 식을 적용하면, 식 (3)과 같은 선형 방정식을 얻을 수 있다.

$$\mathbf{G}\mathbf{d} = \mathbf{e}, \quad \mathbf{G} = \sum_{\Omega} \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix}, \quad \mathbf{e} = -\tau \sum_{\Omega} I_t \begin{bmatrix} I_u \\ I_v \end{bmatrix}^T \quad (3)$$

여기서 I_u, I_v, I_t 는 I 의 u, v, t 에 대한 편미분 값들이다.

KLT 특징점 추적은 선형 시스템인 $\mathbf{G}\mathbf{d} = \mathbf{e}$ 에 기반하고 있고 적합한 두 영상이 입력으로 들어온다면 $\hat{\mathbf{d}} = \mathbf{G}^{-1}\mathbf{e}$ 로 계산된 $\hat{\mathbf{d}}$ 이 해가 되어, 새로운 프레임에서 특징점의 위치를 찾게 된다[3]. 위의 과정을 변위값 추정이 수렴할 때까지 Newton-Raphson schema에 따라서 반복하여 최적의 해를 얻게 된다.

이와 같은 특징점 추적 방법에 근거하여, 특징점 추출을 위해 $\mathbf{G}\mathbf{d} = \mathbf{e}$ 의 \mathbf{G} 에 대한 두 고유값(eigenvalue) λ_1, λ_2 를 구하고, 이를 사용하여 다음과 같이 각 픽셀 주위의 구조를 분류한다. 구조(structure)가 없는 경우에 $\lambda_1 \approx \lambda_2 \approx 0$ 의 값을 갖게 되고, 에지인 경우에는 $\lambda_1 \approx 0, \lambda_2 \gg 0$ 와 같은 값을 갖으며, 마지막으로 모서리인 경우에는 λ_1 과 λ_2 의 값이 서로 다른 큰 값을 갖는다. \mathbf{G} 로부터 계산된 λ_1, λ_2 가 $\min(\lambda_1, \lambda_2) > \lambda$ 를 만족할 경우 추적을 위한 특징점 윈도우로 결정한다. 여기서 λ 는 상수로 결정된 임계값을 의미한다.

2.2 SIFT 특징점 추적

SIFT의 특징점 추출은 자유로운 카메라 움직임에 강인하다는 장점을 가지고 있다. 특징점의 추출을 위해서 입력 영상에 대하여 가우시안 영상 피라미드를 생성한다. 각 레벨별 가우시안 영상의 차를 계산함으로써 그 값이 최대값 또

는 최소값인 점을 최초의 특징점으로 추출한다. 추출된 특징점 중 안정적인 특징점만을 추출한다. 이때 테일러 급수를 사용함으로써 대비가 작은 불안정한 특징점을 제거한다. 또한 특징점을 추출할 경우 에지 상에서 특징점들이 추출되는 경우가 많이 발생한다[8]. 이는 추적에 실패할 확률이 높은 특징점이기 때문에 제거하는 것이 바람직하다. 이와 같이 모서리 상에 존재하는 특징점의 제거를 위해서 Harris 모서리 검출기를 사용한다[8]. 그러므로 SIFT 특징점들은 노이즈에 영향을 덜 받는 장점을 가지고 있다. SIFT 특징점 추적의 세부 단계는 다음과 같다.

첫 번째 단계로써 카메라로부터 영상 프레임을 캡처하고 이를 국부 특징점 벡터(local feature vector)들의 큰 집합으로 변환한다. 이렇게 추출된 특징점 벡터들은 영상의 어떤 스케일링, 회전, 이동에도 불변하는 중요한 특징을 가지고 있다[5].

두 번째는 안정적인 특징점을 추출하는 단계이다. 이번 단계에서는 추출된 특징점 목록들로부터 낮은 대비(contrast)를 갖거나 에지 위에 위치하고 있는 점들을 제거한다. 추출된 특징점 중 안정적인 특징점만 추출하기 위해서 명암 대비가 작은 불안정한 특징점을 제거한다. 또한, 영상의 윤곽선 위의 점들은 Harris 모서리 검출기를 사용하여 제거된 안정적인 특징점만을 추출한다.

세 번째 단계에서는 특징점의 기준 방위(orientation)를 추출한다. 추출된 각 특징점에 대해서 대표 기울기를 추출한다. 픽셀의 방위는 36개의 각도로 양자화하여 히스토그램을 만들어 가장 높은 값을 갖는 각도를 특징점의 기준방위로 선택한다. 또한, 가장 높은 값의 80%내에 포함되는 값을 갖는 각도도 특징점의 기준 방위로 선택한다.

마지막으로 특징점의 특징벡터를 생성한다. 특징점을 중심으로 8개의 각도를 양자화시켜 4x4의 방위 히스토그램을 생성하고 특징벡터로 사용한다. 그러므로 한 특징점이 갖는 특징벡터는 4x4x8=128 차원의 벡터이다. 조명 변화에 강인하기 위하여 벡터들의 크기를 정규화시키고, 0.2보다 큰 값을 갖는 벡터 요소들은 0.2로 하여 다시 정규화한다.

3. 특징점 추적의 정확성 향상

특징점 추적의 정확성 및 안정성 향상을 위해서는 장기적으로 올바르게 추적될 수 있는 특징점들의 선정, 각 특징점의 이동에 대한 예측, 주어진 특징점에 대한 다음 프레임에서의 이동위치 계산 등의 세부 절차들이 강건하게 동작되어야 한다. 본 장에서는 특징점 추적의 정확성 향상을 위한 각 세부절차에 대해서 설명한다.

3.1 특징점 추출

초기 영상에서 신뢰성 있는 특징점을 추출하는 것은 강건한 특징점 추적을 위한 가장 기본이 된다. 특징점은 다른 점에 비해서 구분하기가 쉽고 다른 특징점들로부터 유일하게 구별될 수 있는 점으로 선정함이 바람직하다. 일반적으로

로 특징점은 모서리점들 중에서 선정한다. 특징점은 영상의 이동, 회전, 스케일링에 대해 안정적으로 찾을 수 있어야 하므로 특징점 추출 방법은 특징점 추적과 밀접히 연관된다.

특징점 추출을 위해서, 4단계의 필터링 방법을 적용한다. 첫 번째는 특징점을 추출하는 과정으로써 입력영상을 가우시안 영상 피라미드를 생성한 후 가우시안 영상들의 차(DOG; difference of Gaussian)를 구한다. 우선은 스케일 영역(scale space)을 정의한다. 스케일 영역은 입력 영상 $I(x,y)$ 와 스케일 가우시안 $G(x,y,\sigma)$ 를 사용하여 식 (4)와 같이 정의한다[9].

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (4)$$

스케일 영역에서 키포인트를 찾는 방법으로 가우시안 영상의 차(DOG)를 사용한다. DOG 필터는 $G(x,y,k\sigma) - G(x,y,\sigma)$ 같이 나타난다. DOG 필터와 영상으로 표현 과정은 식 (5)와 같다.

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ = L(x,y,k\sigma) - L(x,y,\sigma) \quad (5)$$

DOG 함수는 가우시안의 스케일 정규화된 라플라시안(scale-normalized Laplacian)인 $\sigma^2 \nabla^2 G$ 로써 근사치를 추정할 수 있다. 계산된 차이가 최대값이거나 최소값인 점들을 특징점으로 추출한다. $\sigma^2 \nabla^2 G$ 의 값이 최대값이거나 최소값일 경우에 다른 영상 함수들과 비교하여 안정적인 특징점을 생성한다.

3.2 특징점 추적을 위한 상태 변수 모델링

특징점 추적 환경은 객체가 고정되어 있고, 카메라가 움직이는 것을 가정한다. n 번째 영상에서 특징점들의 집합을 $X_n = (x_n, y_n)$ 라고 한다. 이 때 X_n 에서 X_{n+1} 로의 변화는 식 (6)과 같이 나타낼 수 있다.

$$X_{n+1} = f(X_n, P_n) \quad (6)$$

여기서 $P_n = (p_{n,1}, \dots, p_{n,m})$ 은 이전 프레임에서 현재 프레임으로의 특징점에 대한 상태를 나타내는 상태 벡터이다. 현재 프레임에서의 P_n 을 찾아내는 것이 특징점 추적에 있어서 가장 중요한 문제이다. 상태벡터의 가장 간단한 형태는 $P_n = (p_{n,1}, p_{n,2})$ 이다. 여기서 $(p_{n,1}, p_{n,2})$ 는 X_n 에서 X_{n+1} 으로의 이동 벡터를 나타낸다.

$$x_{n+1} = x_n + p_{n,1}, \quad y_{n+1} = y_n + p_{n,2}$$

P_n 을 n 번째 프레임의 상태벡터라고 하자. 현재 프레임에서의 상태벡터의 추정을 위하여 이전 프레임의 상태벡터를 사용한다. 이전의 $(n-1)$ 개의 상태벡터 P_1, P_2, \dots, P_{n-1} 를 가지고 있다. 현재 프레임에서의 특징점의 위치를 추정하기

위하여 첫 번째 프레임에서 $(n-1)$ 번째 프레임까지의 상태 벡터의 변화를 분석한다.

실시간 비디오 스트림에서는 짧은 프레임율(frame rate)을 갖기 때문에 현재 프레임까지 움직임 속도를 사용하여 탐색 윈도우의 초기 위치를 예측하는 것이 가능하다. 특징점의 위치는 이전 프레임에서 특징점의 위치와 현재 상태벡터 값을 사용하여 예측할 수 있다. 상태 벡터의 속도는 이전 프레임에서의 $V_{n-1} = P_{n-1} - P_{n-2}$ 에 의해서 근사치를 계산한다. 이렇게 계산된 n 번째 프레임에서 상태 벡터는 식 (7)과 같이 예측된다.

$$P_n = P_{n-1} + V_{n-1} \quad (7)$$

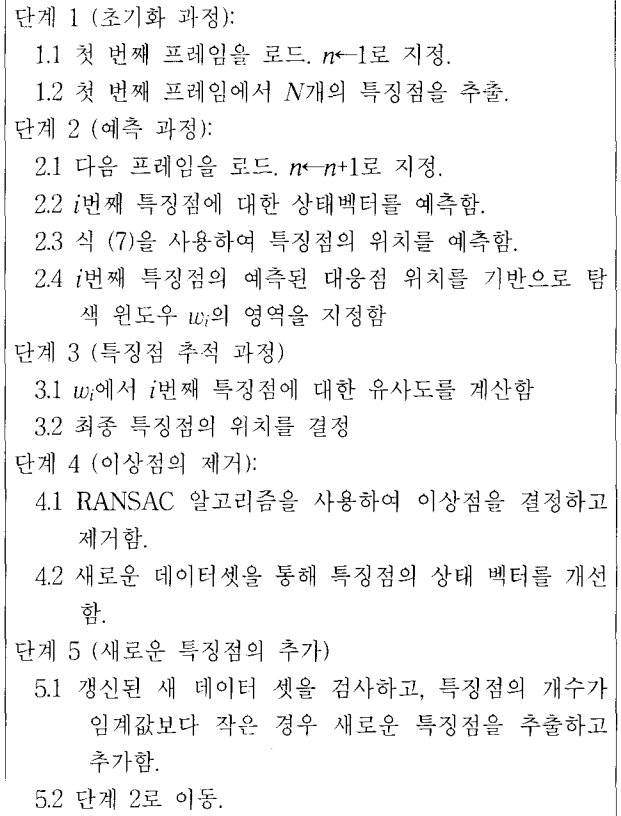
특징점의 위치는 P_n 을 사용하여 예측할 수 있다. 특징점의 예측 위치가 결정되면 그 위치에 탐색 영역을 지정한다. 지정된 탐색 영역을 중심으로 탐색창을 이동시키면서 대응점으로써의 유사도를 계산하고 가장 높은 유사도를 갖는 위치가 특징점의 대응 위치로써 결정된다. 탐색 영역의 크기는 이전 프레임에서 계산된 잔여에러(residual error)를 바탕으로 동적으로 결정된다.

3.3 이상점(outlier)의 제거

특징점들로부터 잘못 매치된 이상점(outlier)을 제거하기 위하여 RANSAC 알고리즘을 사용한다[10][11]. RANSAC 알고리즘은 이상점을 포함하고 있는 데이터를 판별하는데 강한 추정 기법이다. 기본 개념은 다음과 같다. 수집한 데이터 집합으로부터 기본 모델을 구성하는 최소 개수로 이루어진 데이터 모델을 형성한다. 무작위로 추출된 데이터들로 이루어진 모델은 계수를 추정하는데 사용된다. 모델의 추정 계수는 최소자승(least squares) 기법을 통해 계산된다. 모델이 선택되었다면 모델을 구성하는 요소들을 제외한 나머지를 적용한다. 나머지 데이터들을 모델에 적용한 결과값은 관측을 통해서 얻어낸 값과 비교한다. 그 차이가 미리 결정되어진 임계치(threshold)보다 작은 값을 갖는 개수를 계산한다. 여기서 사용하는 임계치는 큰 오차를 가지는 점(outlier)들로 구성된 모델과 적은 오차를 갖는 점(inlier)들을 결정하기 위하여 사용되는 값이다[12]. 위와 같은 과정은 최대 반복 횟수까지 반복되며, 모델로 계산된 값과 관측된 값의 차이가 오차의 임계치보다 작은 경우의 개수가 가장 많은 모델을 최종 모델로 선택한다. 최종 모델에 포함되어 있는 특징점 데이터들을 오차를 갖지 않는 데이터로 결정한다.

3.4 전체적인 특징점 추적 방법

본 장에서는 논문에서 제안한 특징점 추적의 각 수행 단계에 대하여 설명한다. 제안하는 특징점 추적 시스템의 전체적인 처리 단계가 (그림 1)에 있다. 기본적으로 첫번째 프레임에 대한 특징점을 추출하고 이를 다음 프레임에서 지속적으로 추적한다. 그리고 각 프레임에서 이상점을 제거하고



(그림 1) 전체적인 특징점 추적 절차

추적에 실패한 특징점의 수가 일정 수 이하이면 새로운 특징점을 추가한다. 이와 같은 단계를 기본으로 현재 프레임에서 특징점의 위치 추적을 위하여 이전 프레임까지의 상태 변수값을 사용한다. 상태변수값의 예측을 통하여 매칭 오차가 작은 대응점의 초기 위치를 계산할 수 있다. 입력 데이터로 들어오는 연속되는 영상들은 충분히 짧은 시간 간격을 가진 연속 영상이다. 따라서 현재 프레임에서 특징점의 움직임은 이전의 상태 변화나 이전 프레임에서 추적된 특징점의 위치에서 크게 벗어나지 않는다고 가정할 수 있다. 그러므로 특징점의 이전 움직임 히스토리를 고려함으로써, 오차가 작은 특징점의 위치 예측이 가능하다. 예측된 위치는 대응점을 찾기 위한 탐색 윈도우를 설정하기 위한 초기 위치로써 사용된다. 예측된 각 특징점의 위치는 측정값을 사용하여 수정된다. 측정값의 계산을 위하여 설정된 탐색 영역을 중심으로 유사도를 검사한다. 유사도 검사를 통해 가장 높은 유사값을 갖는 위치를 특징점의 대응점으로 결정한다. 위치가 결정된 모든 특징점들을 대상으로 이상점을 제거하기 위한 과정을 진행한다. 이상점들은 주로 폐색현상(occlusion) 때문에 발생한다. 이상점의 제거는 특징점 추적의 오차를 개선하는 중요한 과정이다. 이와 같은 과정을 거쳐 높은 신뢰성을 갖는 특징점을 계산할 수 있다.

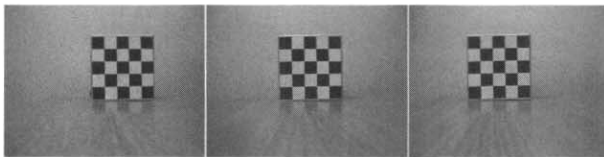
4. 실험 결과

실제 환경에서 촬영한 최소 100 프레임 이상을 포함하는

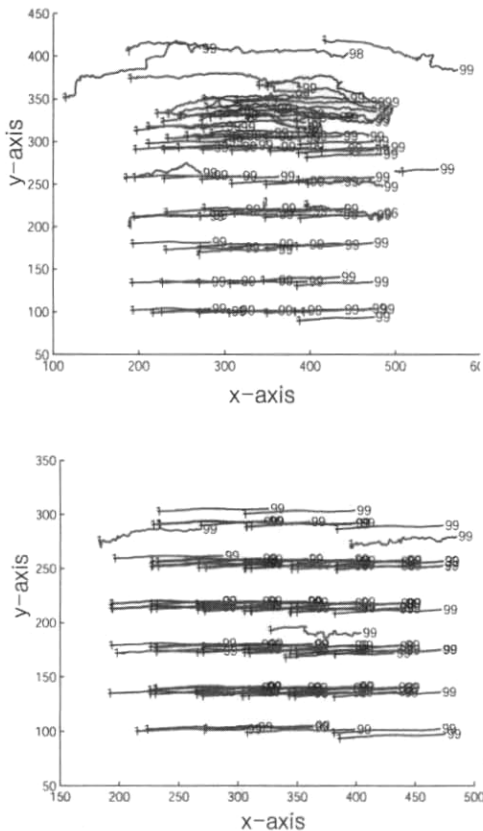
긴 비디오 프레임 데이터들에 대해 제안하는 특징점 추적 기법을 적용하였다. 비디오 프레임은 일반적인 실내의 환경에서 카메라를 이동하면서 정적 물체를 촬영하여 얻었다. 카메라의 움직임은 실험 결과의 분석을 위해서 일정한 방향으로 이동 및 회전하도록 하였다. 실험에 사용된 이미지는 대부분이 640x480의 이미지를 사용하였고 100 프레임을 기준으로 평균 적으로 4.5 FPS의 속도를 가졌다.

4.1. 카메라의 움직임이 적은 비디오 프레임에 대한 특징점 추적

(그림 2)는 패턴이 부착되어 있는 상자를 대상으로 평행 이동한 비디오 프레임들에 대한 입력 영상이다. (그림 3)은 (그림 2)의 특징점 추적 결과를 나타낸다. (그림 3)의 오른쪽 결과가 제안한 방법을 사용한 결과이고, 왼쪽의 실험 결과가 KLT 방법을 사용하여 특징점을 추적한 궤적이다. 두 실험결과를 비교해 볼 때 오른쪽 궤적에서 이상점이 많이 제거된 모습을 확인할 수 있다.



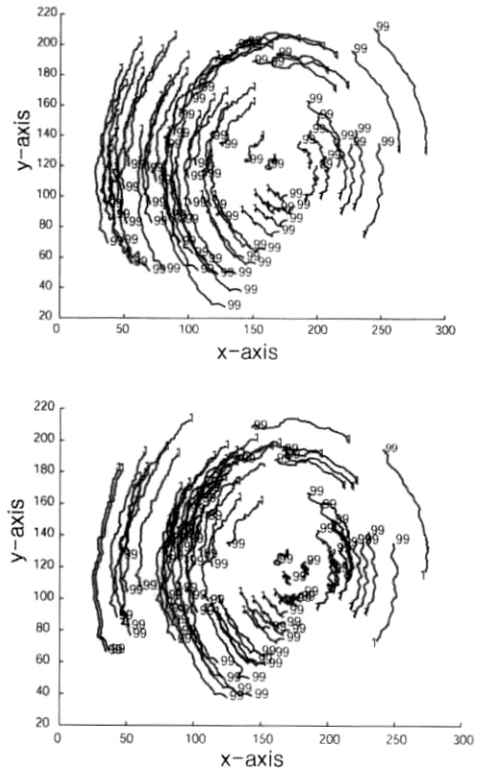
(그림 2) 카메라가 움직임이 작은 평행이동으로 획득된 영상 프레임



(그림 3) 패턴 상자 영상에 대하여 두 가지 방법으로 추적한 궤적



(그림 4) 카메라의 회전 움직임 실험에 대한 입력 영상

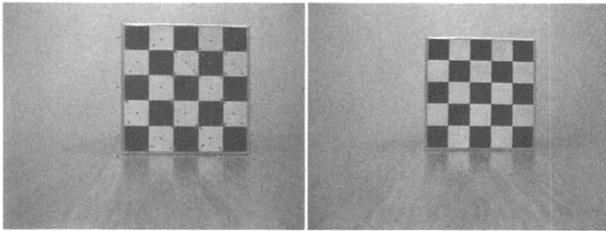


(그림 5) 카메라 회전 영상에 대하여 두 가지 방법으로 추적한 궤적

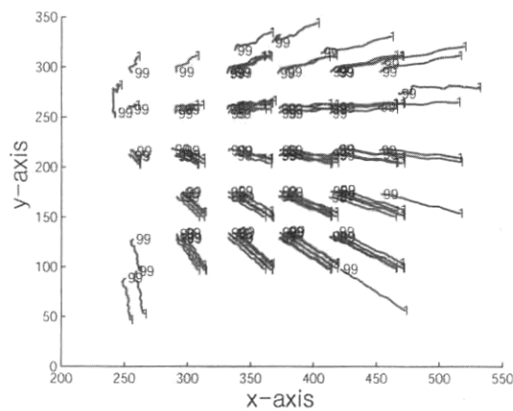
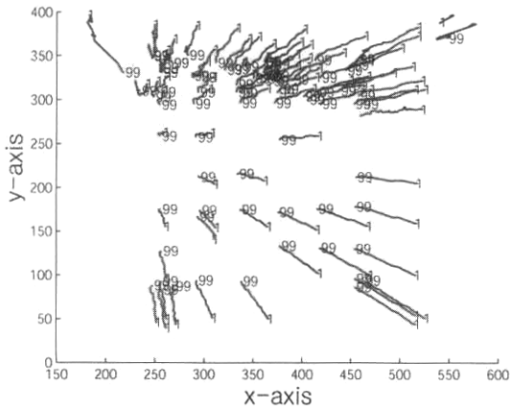
(그림 4)는 카메라의 회전에 대한 실험에 사용된 입력 영상이다. 왼쪽 이미지에서 처음 특징점치 추출된 결과를 확인할 수 있다. 회전 실험의 결과를 (그림 5)로부터 확인할 수 있다. 이동 실험과 같이 300개의 특징점을 추출하였고 100개의 프레임 동안 추적하였다.

(그림 6)은 물체로부터 카메라가 점점 멀어지는 움직임에 대한 실험의 입력 영상들이다. 제안된 본 논문에서 사용한 특징점 추출 방법은 회전과 스케일링에 있어 강력하다는 장점을 가지고 있다. 이는 (그림 7)의 KLT 방법으로 추적된 왼쪽 실험 결과와 제안된 방법으로 추적된 오른쪽 실험 결과의 비교를 통해 확인할 수 있다.

(그림 8)은 카메라의 움직임이 회전이 없는 평행 이동하는 (그림 3)의 실험에 대하여 특징점이 추적된 결과의 오차를 측정할 것이다. 평행이동을 하였으므로 모든 특징점이 99프레임동안 같은 x 좌표를 갖는다는 가정을 하고 오차를 계산하였다. 각 프레임마다 모든 특징점에 대하여 평균값을 계산한 후 그래프로 나타낸 결과이다. 그래프를 통해 확인



(그림 6) 스케일링 실험에 대한 입력 영상



(그림 7) 스케일링이 있는 영상에서 특징점의 궤적

할 수 있듯이 두 방법 모두 프레임이 이동하면서 오차가 증가하는 결과를 보였으나, 제안한 방법을 사용한 특징점 추적 결과가 더 작은 오차를 갖는 것을 확인할 수 있다.

4.2 복잡한 환경에서 카메라 움직임이 큰 비디오에 대한 특징점 추적

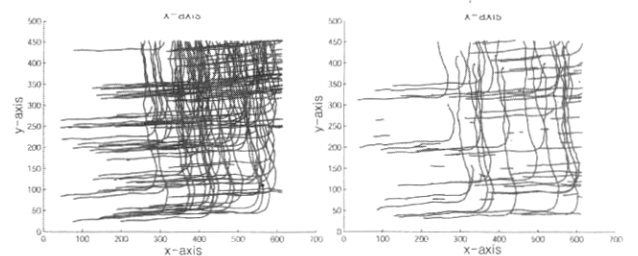
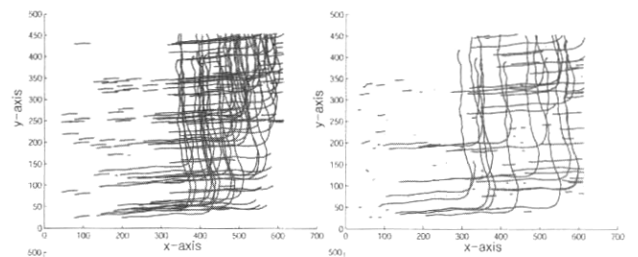
(그림 8)은 일반적인 실외 물체에 대하여 카메라의 자유로운 이동과 회전이 있는 실험에 대한 입력 영상이다. 실외의 이정표를 대상으로 특징점 추적 실험을 하였다. (그림 9)는 (그림 8)의 입력 영상을 대상으로 특징점을 추적한 결과이다. KLT의 특징점 추적 방법과 제안된 방법 두 가지를 비교하였다. (그림 9)의 위쪽 두 영상이 KLT 방법을 사용한 특징점 궤적을 나타내고 있고, 아래쪽 두 영상이 제안된 방법을 사용하여 특징점을 추적한 결과를 나타내고 있다. (그림 9)의 왼쪽 두 그림은 이상점이 제거되지 않은 결과이고,

오른쪽 두 그림은 왼쪽 결과에 대하여 이상점을 제거한 결과이다. 결과를 비교해 볼 때 제안된 방법을 사용한 결과에서 이상점이 상당수 제거된 특징점 궤적을 확인할 수 있다.

(그림 10)과 (그림 11)은 외부의 원거리 환경을 대상으로 실험이다. 높은 건물에서 볼 수 있는 일반적인 배경을 대상으로 입력 영상을 촬영하였고, 카메라는 이동과 회전의 자유로운 움직임을 갖는다. 실험 결과는 (그림 11)에서 확인할 수 있다. (그림 11)의 4개의 실험결과 중 왼쪽 두 그림은 KLT와 제안한 방법을 사용한 이상점을 제거하지 않은 결과이고, 오른쪽 두 그림은 왼쪽의 두 실험 결과에 대하여 이상점을 제거한 모습을 나타낸다. (그림 12)는 실내 책상을 배경으로 카메라의 회전과 이동이 함께 있는 실험이다. 입력으로 들어온 첫 번째 영상 프레임에서 특징점을 추출한 결과를 (그림 13)에서 확인할 수 있다. (그림 14)는 (그림 12)를 입력으로 한 특징점 추적의 결과를 나타내고 있다. 왼쪽 두 영상은 KLT와 제안한 방법의 이상점을 제거하지 않은 모습이고, 오른쪽 두 영상은 왼쪽 두 실험 결과에 대하여 이상점을 제거한 결과이다.



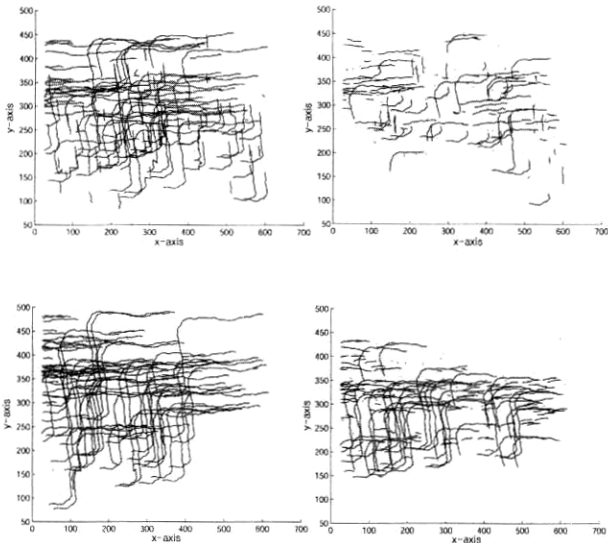
(그림 8) 여러 카메라 움직임을 포함하는 실외 환경에 대한 비디오 프레임



(그림 9) 특징점의 움직임 궤적과 이상점을 제거한 결과



(그림 10) 다양한 카메라의 움직임이 있는 외부 환경에 대한 입력 프레임



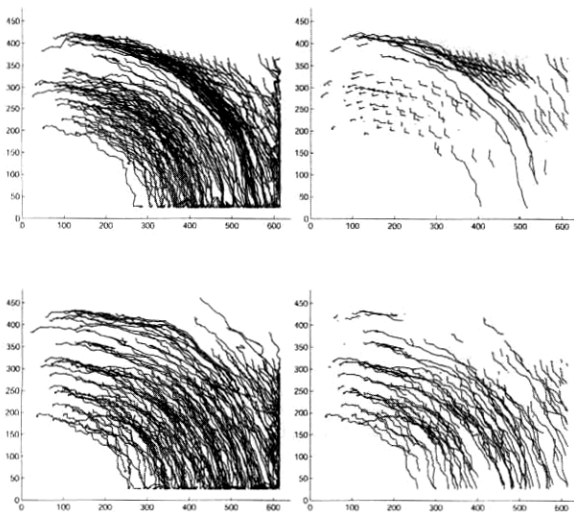
(그림 11) 특징점의 움직임 궤적과 이상점을 제거한 결과



(그림 12) 카메라의 회전 및 이동을 포함하는 움직임으로 촬영한 비디오 프레임



(그림 13) 키보드 비디오에서의 특징점 추출 결과



(그림 14) 특징점의 움직임 궤적과 이상점을 제거한 결과

4.3 Epipolar 거리 에러 분석

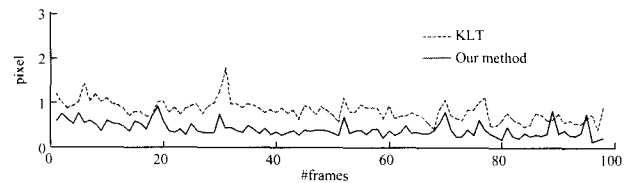
추적된 특징점들의 정확성 분석을 위해서는 추적 결과의 오차 측정이 수행되어야 한다. 각 프레임에서의 실제의 정확한 특징점 움직임 위치를 알 수 없으므로 오차를 에피폴라 거리(epipolar distance)로 정의한다. 에피폴라 기하는 두 영상 간의 기하관계를 명료하게 표현할 수 있도록 한다. 에피폴라 기하에서의 두 영상 사이의 기하관계는 기본행렬(fundamental matrix) F 로 표현된다. 행렬 F 를 사용하여 x_i 와 x_i' 사이의 관계가 표현된다[13]. Fx_i 는 다른 영상에서의 x_i 에 대응되는 에피폴라 라인으로써 x_i' 는 Fx_i 상에 존재해야 한다. 대응되는 점이 에피폴라 라인과 떨어진 거리를 대응점의 오차로 간주할 수 있다[14]. 에피폴라 거리의 잔여 에러(residual error)는 식 (8)과 같이 계산할 수 있다[15].

$$1/N \sum_i^N (d(x_i', Fx_i)^2 + d(x_i, F^T x_i')^2) \quad (8)$$

여기서 x_i 는 현재 영상에서의 특징점을 나타내고, x_i' 는 x_i 에 대응되는 다음 영상에서의 특징점을 나타낸다.

(그림 15)는 이와 같은 방법으로 잔여에러를 측정된 결과이다. 100장의 프레임동안 추적된 특징점의 대응관계를 사용하여 기본행렬(fundamental matrix)을 계산하였고, 잔여에러를 측정하였다. KLT와 제안한 방법에 대한 특징점 추적 결과를 사용하여 오차를 계산하였다. (그림 15)에서 확인할 수 있듯이 제안한 방법이 전체적으로 더 작은 오차를 갖는 것을 확인할 수 있다. (그림 15)의 실험에 대하여 평균값을 계산한 결과 KLT가 0.8288픽셀의 평균 오차를 가지고, 제안된 방법은 0.4133픽셀의 평균 오차를 갖는 결과를 확인할 수 있다.

<표 1>은 추적된 결과에 대하여 잔여에러를 측정을 통해 SIFT와 제안된 방법의 오차를 계산한 결과이다. 10프레임마다 값을 계산하였고, 제안한 방법이 SIFT의 방법보다 적은 오차를 갖는 것을 확인할 수 있다.



(그림 15) 추적된 특징점 궤적에서의 각 프레임마다 에피폴라 거리 잔여에러 (단위: 픽셀)

<표 1> 추적된 특징점 궤적에서의 매 10 프레임마다 에피폴라 거리의 잔여에러 (단위: 픽셀)

방법 \ 프레임	#10	#20	#30	#40	#50	#60
SIFT	1.3168	1.3824	1.1319	1.2480	1.3858	1.6074
제안한 방법	1.1197	1.2158	1.2201	1.1652	1.2398	1.2715

5. 결 론

본 논문에서는 긴 비디오 프레임들을 입력으로 강건한 특징점을 추적하고 이상점을 제거하는 방법을 제안하였다. 본 논문에서 제안한 알고리즘은 2차원적인 이동, 회전, 스케일링에 강인한 특징점을 추출하고, 특징점의 움직임 상태 변화의 히스토리를 고려하여 현재 프레임에서의 특징점을 추적함으로써 특징점 추적의 정확도를 향상하였다. 또한 이상점을 판단하고 제거함으로써 전체적인 오차를 향상시킬 수 있었다.

본 논문에서 제안한 기법은 증강현실 세계의 구현을 위하여 구조 및 객체의 복원의 정확도를 향상시키는데 유용할 것이다. 또한 3차원 객체 추적의 정확도를 향상시키는데도 활용될 수 있을 것으로 보인다. 정확한 특징점 추적이 주어진다면 카메라로 관찰되는 실세계의 3차원적인 구조의 추정이 가능하므로 다양한 응용에 사용될 수 있다.

특징점 추적 문제는 실 환경의 다양성 및 불규칙성에 의해서 항상 올바른 결과를 산출하지 못한다. 특징점 추적이 잘못된 매칭을 포함하고 있는 경우에 대해서 이를 제거할 수 있는 기법의 연구가 필요하다. 일반적으로 긴 프레임동안 추적되는 특징점의 수가 많을 수록 유용하지만 특징점의 수보다 그 정확도의 향상 및 이상점이 포함되지 않은 추적 결과가 더 유용한 응용이 많다. 향후 연구과제로 실제 환경의 물리적인 조건을 고려한 이상점이 없는 특징점 추적 연구가 필요하다. 또한 3차원 구조를 미리 알고 있는 객체에 대한 인식을 위해서 객체의 특징점을 안정적으로 추적하고 구조의 형태에 따른 카메라 자세를 추정하는 방법에 대한 연구가 필요하다.

참 고 문 헌

[1] J. Shi, C. Tomasi, "Good features to track," IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, 1994.
 [2] Timo Zinßer, Christoph Gräßl, Heinrich Niemann, "Efficient Feature Tracking for Long Video Sequences," DAGM, Vol 3175, pp. 326-333, 2004.
 [3] C. Tomasi, T. Kanade, "Detection and tracking of point features," Tech. Rep., CMU-CS-91132, Pittsburgh: Carnegie Mellon University School of Computer Science, 1991.
 [4] D. Lowe, "Object Recognition from Local Scale Invariant Features," International Conference on Computer Vision, pp. 1150-1157, 1999.
 [5] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm," Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999.
 [6] T. Tommasini, A. Fusiello, E. Trucco, V. Roberto, "Making good features track better," CVPR, pp. 178-183, 1998.
 [7] S. Sinha, J.-M. Frahm, and M. Pollefeys, "GPU-based Video Feature Tracking and Matching," Tech. Rep. TR06-012, University of North Carolina at Chapel Hill, May 2006.
 [8] C. G. Harris, M. J. Stephens, "A Combined Corner and Edge Detector," Proceedings of Fourth Alvey Vision Conference,

pp. 147-151, 1998.
 [9] D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 2003.
 [10] 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.
 [11] T. Tommasini, A. Fusiello, V. Roberto, and E. Trucco, "Robust feature tracking," Proceedings of the Joint Workshop 7of AI*IA and IAPR-IC, Ferrara, Università di Ferrara. pp. 93-98, 1998.
 [12] J. Matas, O. Chum, "Randomized RANSAC with Sequential Probability Ratio Test," IEEE International Conference on Computer Vision, Vol 2, pp. 1727-1732, 2005.
 [13] Zhengyou Zhang, Charles Loop, "Estimating the fundamental matrix by transforming image points in projective space," Computer Vision and Image Understanding, Vol 82, pp. 174-180, 2001.
 [14] Sun Yinan, Liu Weijun, Ma Yong Zhuang, Wang Yuechao, "A High-Accuracy Algorithm for Computing Fundamental Matrix," Proc. 2004 IEEE International Conference on Robotics and Biomimetics, pp. 733-736, 2004.
 [15] R. Hartley, A. Zisserman, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2000.



윤 중 현

e-mail : jhyoon@incheon.ac.kr
 2006년 인천대학교 컴퓨터공학과 (공학사)
 2006년~현재 인천대학교 컴퓨터공학과 석사과정
 관심분야: 동작기반 인터페이스, 증강현실게임, 카메라 추적



박 중 승

e-mail : jong@incheon.ac.kr
 1992년 경북대학교 전자계산학과 (이학사)
 1994년 포항공과대학교 컴퓨터공학과 (공학석사)
 1999년 포항공과대학교 컴퓨터공학과 (공학박사)
 1999년~2001년 한국전자통신연구원 가상현실연구부 선임연구원
 2001년~2003년 디지털아리아 시각정보처리연구소 책임연구원
 2004년~현재 인천대학교 컴퓨터공학과 조교수
 관심분야: 영상처리, 3D 비전, 증강현실, 게임공학