영상 시퀀스의 특징점에 대한 Outlier 보정

김재학* 박종성** 황지은* 한준희*
*포항공과대학교 컴퓨터공학과, **인천대학교 컴퓨터공학과
{ninyshio*, romeo, joonhan}@postech.ac.kr

Outlier correction from uncalibrated image sequence

Jae-Hak Kim* Jong-Seung Park** Ji-Woon Hwang* Joon H. Han*
* Computer Vision Lab, Dept. of Comp. Sci. & Eng., POSTECH
** Dept. of Comp. Sci. & Eng., University of Incheon

요 약
본 논문에서는 영상 시퀀스(image sequence)에서 얻은 특징점(feature point) 중 outlier 를 제거 및 보정할 수 있는 방법을 제시한다. 영상 시퀀스가 주어졌을 때, 우리는 이 영상에서 특징점 추적(tracking)을 하며, 영상의 중요한 정보를 이용한다. 이러한 자동적으로 얻어내는 특징점 추적 데이터는 올바르지 못하게 추적된 것이 있기 마련인데, 이렇게 올바르지 못한 데이터, 즉, outlier를 제거하기 위하여, 기존의 방법들은 tricocal tensor를 주로 사용하였다. 그러나 tricocal tensor 는 영상의 3차원 정보로 제어 되어, 인터페이스의 개별차를 줄이게 되는 단점이다. 따라서, 우리는 triangulation 방법을 이용하여, 3차 이상의 영상에서도, outlier 의 제거와 보정이 동시에 가능한 방법을 제시한다.

1. 서론
영상 시퀀스가 주어졌을 때, 우리는 각 영상에서 특징점을 추출해낸다. 이렇게 얻어진 특징점을 기반으로 전체 시퀀스에서 특징점에 해당하는 부분을 모두 추적해 내게 하게 되면, 전체 영상 시퀀스에서 특징점의 연속성을 이용한 사용가능하게 된다. 이러한 추적 결과는 기존의 추적 결과와 캐릭터의 움직임과 영상 내에 주어진 물체의 형변을 반영해 우리와 복잡하게 상호작용하게 한다. 따라서, 이러한 추적 결과는 캐릭터의 움직임 및 3차원 물체의 형변을 알기 위한 전체 시스템의 입력에 해당하는 역할을 담당하고 있다. 그리고 추정된 추적 결과는 일반적으로 잘 알려진 KLT tracker [1]을 이용하여 자동적으로 얻을 수 있다. 그러나, 자동적으로 얻어진 이 추적 결과는 일반적으로 올바르지 못한 데이터를 포함한다.

특징점 추적 결과가 알수 있을 때, 올바른 추적 결과를 얻기 위해서 outlier 탐지 알고리즘을 outlier라고 한다. outlier가 발생하는 이유는 영상의 가변성 및 경향 변화가 일정하지 않은 경우와 추적된 특징점이 다른 물체나 가려져서 추적 결과가 잘못 이루어져기 때문이다.

기존의 이러한 outlier 탐지 알고리즘은 tricocal tensor를 이용한 방법이었다. [2, 3] tricocal tensor는 3차 영상에 대하여, 기하학적 관계를 설명할 수 있으며, 추적된 추정된 추적 결과가 아님지를 판단할 수 있다. 그러나, 이 방법은 3차 이상의 영상에서만의 제한을 두며, outlier를 제거하기 하게 된다.

본 논문에서는 triangulation 방법을 이용하여 3차 이상의 영상에서 outlier의 제거와 보정을 할 수 있는 방법을 제시한다. triangulation 방법은 Hartley [4, 5]가 제시한 것으로 2차 이상의 영상에서 영상 대용량 및 카메라의 두고행렬(projection matrix)이 주어졌을 때, 두 공간간의 두 차원onto (projective space)에 있는 물체의 위치를 구하는 방법이다. 우리

의 방법은 이러한 triangulation 방법의 특성과 RANSAC [6] 알고리즘의 특성을 이용하여, 초래된 잘못 보정된 영상의 좌표값을 찾아보았을 때, 프로젝트 보정(projective reconstruction)과 동시에 outlier를 보정한다.

2. outlier 보정 알고리즘

Outlier를 보정하기 위하여 RANSAC 알고리즘을 이용한다. 먼저 RANSAC 알고리즘은 두번의 결정을 채택하게 되는데, 첫 번째로 주어진 영상 대용량들은 올바르지 못한 경우, outlier를 찾아내기 위한 사용이 된다. 영상 대용량들은 outlier로 분류가 되었을 경우, 영상 대용량을 구성하는 영상의 좌표터가 모두 outlier인 경우가 많으며, 올바른 영상의 좌표터가 없는 경우도 있다. 따라서, outlier로 분류가 되었을 경우, 영상의 좌표터는 outlier의 보정이 될 수 있다. 이것은 RANSAC의 뒷부분에 사용으로, outlier로 분류된 영상 대용량에서 올바른 영상의 좌표터를 찾아 outlier 보정을 하기 위하여 반복이 이루어진다.

그림 1은 outlier 보정이 이루어지는 과정을 보여주고 있다. 영상 3차가 주어졌고, 영상 영상 대용량 x', x''를 얻었을 때, 이것이 outlier로 판정되었다고 하자. outlier로 판정된 영상의 관계에 대하여, 만약 x와 x'는 올바른 것이라면, 이 영상 대용량을 모두 outlier로 판정하고 비디오에는 논의와의 위치가 날아다. 따라서, 올바른 x와 x'를 이용하여 triangulation을 하여, 올바른 X를 다시 세번째 카메라의 영상으로 투사시키며 접 x''을 연쇄화하여, 이는 x''의 새로운 보정된 영상이 된다. 그러므로, 여기에서의 문제는 outlier로 판정된 영상 대용량에서 올바른 영상의 좌표터가 없도록 되어 있는 것이 문제이다. 즉, 그림 1에서의 것과 같이, 일부 영상 대용량이 outlier로 판정되었으나, 3개의 영상에서 올바른 영상이 x와 x'를 찾는 것이 문제이다. 물론, 가장 간단한 방법으로는
그림 1: $x, x', x''$는 주어진 영상 대응점, $x''$는 triangulation 으로 outlier 보정이 이루어진 영상점.
3개의 영상점이 주어졌을 경우에는 3개의 조합만이 있을 뿐이므로, 이것을 조사하는 것이 가장 간단하다. 그러나, 3개 이상의 영상이 주어졌을 경우로 문제를 일반화 시키면 다음과 같다.
주어진 $m$개의 영상에서 얻어낸 영상 대응점이 있다고 하자. 그 영상 대응점이 outlier로 판명이 난다고 할 때, 그 영상 대응점에는 $m$개의 영상점이 있다. $m$개의 영상점 좌표 중 올바른 좌표점이 적어도 $m/2$개의 개수보다 많다는 가정을 하면, 우리는 RANSAC 알고리즘을 적용하여 올바른 좌표점 집합을 얻어낼 수 있다.

3. 프로젝티브 복원과 Outlier 보정
$m$개의 영상이 주어졌다고 하자. 또한 $n$개의 영상 대응점이 주어졌다고 할 때, 우리는 다음과 같이 Factorization 방법을 이용한 프로젝티브 복원을 할 수 있다[7].

\[ W = \begin{bmatrix} l_{1x} & l_{1x'} & \cdots & l_{1x''} \\ l_{2x} & l_{2x'} & \cdots & l_{2x''} \\ \vdots & \vdots & \ddots & \vdots \\ l_{nx} & l_{nx'} & \cdots & l_{nx''} \end{bmatrix} \]

\[ = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \]

이렇게 얻어낸 카메라의 투사행렬로부터 RANSAC 을 기반으로 한 outlier 영상 대응점을 찾아낼 수 있다.
1. 영상으로부터 임의적으로 샘플 점을 선택
2. Factorization 방법으로 프로젝티브 복원
3. 영상에서 선택하지 않은 점들에 대해서 다음을 실시함
   (a) 영상의 새로운 점 선택
   (b) Triangulation 으로 선택된 점의 3차원 좌표 복원
   (c) Reprojection 예리 계산
   (d) 선택점이 주어진 threshold를 만족하면, outlier set으로 토 저장
4. 2단계로 반복, 그러나 만약 반복 횟수 마지막에 오면 다음 단계로 갑
5. 가장 큰 outlier 집합을 찾아냄

6. Inlier 집합에 들어가지 못한 outlier를 찾아냄
7. Outlier 집합에 대해서 샘플 영상점들을 랜덤하게 선택
8. 영상점으로부터 triangulation 하여 이것을 reprojection
9. 이것을 계속 반복하여 reprojection 에러가 가장 적은 set 을 다시 끝내됨
10. Outlier 집합에서 올바른 영상 좌표점이 있을 경우,
11. 그 영상점에 대해서 triangulation 된 3차원 좌표를 바탕으로 outlier 보정을 함
12. 보정된 outlier를 outlier 집합에 추가

4. 실험 결과

실험은 우선 outlier를 찾아낸 후에, 이렇게 찾아낸 outlier를 보정할 여지가 있는 영상점들에서 outlier가 올바르게 되었는지를 확인하는 방식으로 수행되었다.

그림 2: (a) outlier를 찾아낸 3점의 영상, (b) outlier가 보정이 된 3점의 영상. 사각형표시: outlier, 십자표시: outlier

그림 2는 영상 시퀀스 중 3개의 영상에 대해서, outlier와 outlier를 찾아내어 표시한 결과이다. 그림 2-(a) 결과는 outlier를 찾기만 한 것으로 총 135개의 영상점 중 125개가 outlier로, 10개가 outlier로 판명되었다. 그림 2-(b) 결과는 outlier 보정을 한 결과로, 총 135개의 영상점 중 122개가 outlier로, 3개가 outlier 인체로 남았다. 즉, outlier 보정으로 총 10개의 outlier 중, 7개가 보정 가능하였다.

그림 3은 그림 2의 손부분을 확대하여 표시한 것이다. 그림 3-(a)는 outlier를 찾은 후의 영상이며, 여기서 보듯이 outlier로 판명된 영상 대응점에 해당하는 3개의 영상점은 적어도 2개가 올바른 위치를 가지고 있어 보정이 될 가능성이 충분히 있다. 그림 3-(b)는 outlier를 보정한 후의 결과로, 손가락에 있던 outlier들 모두 올바르게 보정되었음을 확인 할 수 있다.

그림 4는 복도 영상에 대해서 얻어낸 inlier 와 outlier를 표시한 것이다. 그림 4-(a)는 outlier를 판명해낸 것이며, 여기서는 총 409개의 영상점 중 399개가 outlier로 판명 되었고, 10개가 outlier로 판명되었다. 그림 4-(b)는 outlier를 보정한 결과로, 총 409개의 영상점 중 5개의 outlier만이 남아 10개의 outlier 중 5개의 점이 보정되었다.
그림 3: 2 영상중 공통의 손부분을 확대한 것, (a) outlier로 판별된 것, (b) outlier를 보정한 후, 사각형표시: outlier, 심사표시: inlier

그림 4: (a) outlier를 찾아낸 3장의 영상, (b) outlier가 보정이 된 3장의 영상. 사각형표시: outlier, 심사표시: inlier

그림 5: 4 영상중 북도의 길자‘D’ 부분을 확대한 것, (a) outlier로 판별된 것, (b) outlier를 보정한 후, 사각형표시: outlier, 심사표시: inlier
해결책을 찾아보는 것이 있을것이다.

참고문헌


