

# 영상 시퀀스의 특징점에 대한 Outlier 보정

김재학\*<sup>o</sup>, 박종승\*\*<sup>o</sup>, 황지운\*<sup>o</sup>, 한준희\*<sup>o</sup>

\*포항공과대학교 컴퓨터공학과, \*\*인천대학교 컴퓨터공학과  
{inyashio<sup>o</sup>, romeo, joonhan}@postech.ac.kr

## Outlier correction from uncalibrated image sequence

Jae-Hak Kim\*<sup>o</sup>, Jong-Seung Park\*\*<sup>o</sup>, Ji-Woon Hwang\*<sup>o</sup>, Joon H. Han\*<sup>o</sup>

\*Computer Vision Lab, Dept. of Comp. Sci. & Eng., POSTECH

\*\*Dept. of Comp. Sci. & Eng., University of Incheon

### 요약

본 논문에서는 영상 시퀀스(image sequence)에서 얻은 특징점(feature point) 중 outlier를 제거 및 보정할 수 있는 방법을 제시한다. 영상 시퀀스가 주어졌을 때, 우리는 이 영상에서 특징점 추적(tracking)을 하여, 영상의 중요한 정보로 이용한다. 이러한 자동적으로 얻어낸 특징점 추적 데이터는 올바르지 못하게 추적된 것이 있기 마련인데, 이렇게 올바르지 못한 데이터, 즉, outlier를 제거 하기 위하여, 기존의 방법들은 trifocal tensor를 주로 사용하였다. 그러나 trifocal tensor는 영상이 3장으로 제한 되어있다. 또한 outlier를 찾은 후에는 제거만 하게 되어, 입력 데이터의 개수를 줄이게 되는 단점이 있다. 따라서, 우리는 triangulation 방법을 이용하여, 3장 이상의 영상에서도, outlier의 제거와 보정이 동시에 가능한 방법을 제시한다.

### 1. 서론

영상 시퀀스가 주어졌을 때, 우리는 각 영상에서 특징점을 추출하게 된다. 이렇게 얻어진 특징점을 기반으로 전체 시퀀스에서 특징점에 해당하는 부분을 모두 찾는 추적을 하게 되면, 전체 영상 시퀀스에서 특징점의 연속된 대응관계를 알게 된다. 이러한 특징점 추적 결과는 카메라의 움직임과 영상 장면이 주어진 물체의 형태를 알아내기 위한 복원에 사용되게 된다. 따라서, 이러한 특징점 추적 결과는 카메라의 움직임 및 3차원 물체의 형태를 알기 위한 전체 시스템의 입력에 해당하는 역할을 담당하고 있다. 그러한 특징점의 추적은 일반적으로 잘 알려진 KLT tracker [1]를 이용하여 자동적으로 얻을 수 있다. 그러나, 자동적으로 얻게 되는 특징점 추적의 결과는 일반적으로 올바르지 못한 데이터를 포함한다.

특징점 추적 결과가 있을 때, 올바른 특징점 추적 결과를 inlier라 하고, 잘못 추적된 특징점들을 outlier라고 한다. Outlier들이 발생하는 이유는 영상 자체의 밝기 변화가 일정하지 않은 경우와 추적하는 특징점이 다른 물체에 가려져서 특징점 추적이 잘못 이루어지기 때문이다.

기존의 이러한 outlier를 제거하기 위한 연구로, trifocal tensor를 이용한 방법이 있었다[2, 3]. Trifocal tensor는 3장의 영상에 대하여, 기하학적 관계를 설명할 수 있어, 주어진 대응점이 올바른 특징점 추적의 결과인지 아닌지를 판단할 수 있다. 그러나, 이 방법은 3장의 영상에서만 얻을 수 있으며, outlier를 제거만 하게 된다.

본 논문에서는 triangulation 방법을 이용하여 3장 이상의 영상에서 outlier의 제거와 보정을 할 수 있는 방법을 제시한다. Triangulation 방법은 Hartley [4, 5]가 제시한 것으로 2장 이상의 영상에서 영상 대응점과 각 카메라의 투사행렬(projection matrix)이 주어졌을 때, 투사공간(projective space)에 있는 물체의 점의 위치를 구하는 방법이다. 우리

의 방법은 이러한 triangulation 방법의 특성과 RANSAC [6] 알고리즘의 특성을 이용하여, 프로젝티브 복원(projective reconstruction)과 동시에 outlier를 보정한다.

### 2. Outlier 보정 알고리즘

Outlier를 보정하기 위하여 RANSAC 알고리즘을 이용하였다. 먼저 RANSAC 알고리즘은 두번에 걸쳐 사용하게 되는데, 첫번째는 주어진 영상 대응점들중 올바르지 못한 즉, outlier를 찾아내기 위하여 사용이 된다. 어떤 영상 대응점들이 outlier로 분류가 되어졌을 경우, 영상 대응점을 구성하는 영상점의 좌표가 모두 outlier인 경우도 있으나, 올바른 영상점의 좌표를 가지는 경우도 있을 것이다. 따라서, 올바른 영상점을 갖는 좌표들이 어떤 것인지 알아낼 수 있다면, 비록 outlier로 분류가 되었지만, 내부를 구성하는 올바른 영상점 좌표들을 이용하여 outlier의 보정을 할 수 있게 된다. 이것이 바로 RANSAC의 두번째 사용으로, outlier로 분류된 영상 대응점에서 올바른 영상 좌표점을 찾아 outlier 보정을 하기 위하여 한번 더 이용된다.

그럼 1은 outlier 보정이 이루어지는 과정을 보여주고 있다. 영상 3장이 주어졌다고 가정하고, 영상 대응점  $x, x', x''$ 를 얻었을 때, 이것이 outlier로 판명 되었다고 하자. Outlier로 판명된 점들의 관계에서, 만약  $x$ 와  $x'$ 는 올바른 것이라면, 이 영상 대응점을 모두 outlier로 판단하고 버리기에는 수정의 여지가 남아있다. 따라서, 올바른  $x$ 와  $x'$ 를 이용하여 triangulation을 하여, 얻어낸  $X$ 를 다시 세번째 카메라의 영상으로 투사시키면 점  $x''$ 를 얻게되고, 이는  $x''$ 의 새롭게 보정될 점이 된다.

그렇다면, 여기에서의 문제는 outlier로 판명이 된 영상 대응점에서 올바른 영상점들이 어떤 것인지 찾는 것이 문제가 된다. 즉, 그림 1에서와 같이, 비록 영상 대응점이 outlier로 판명되었으나, 3개의 영상점중 올바른 영상점인  $x$ 와  $x'$ 를 찾는 것이 문제이다. 물론, 가장 간단한 방법으로는



그림 1:  $x, x', x''$ 는 주어진 영상 대응점,  $\bar{x}$ 는 triangulation으로 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 시켜면 다음과 같다.

주어진  $m$ 개의 영상에서 얻어낸 영상 대응점이 있다고 하자. 그 영상 대응점이 outlier로 판명이 났다고 할때, 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].

$$W = \begin{bmatrix} \lambda_1^1 x_1^1 & \lambda_2^1 x_2^1 & \dots & \lambda_n^1 x_n^1 \\ \lambda_1^2 x_1^2 & \lambda_2^2 x_2^2 & \dots & \lambda_n^2 x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^m x_1^m & \lambda_2^m x_2^m & \dots & \lambda_n^m x_n^m \end{bmatrix} \quad (1)$$

$$= \begin{bmatrix} p^1 \\ p^2 \\ \vdots \\ p^m \end{bmatrix} [X_1, X_2, \dots, X_n] \quad (2)$$

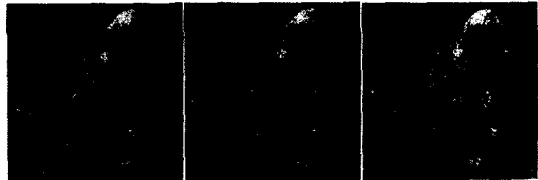
이렇게 얻어낸 카메라의 투사행렬로부터 RANSAC을 기반으로 한 outlier 영상 대응점을 찾아 낼 수 있다.

1. 영상으로부터 임의적으로 샘플 점을 선택
2. Factorization 방법으로 프로젝티브 복원
3. 영상에서 선택하지 않은 점들에 대해서 다음을 실시함
  - (a) 영상의 새로운 점을 선택
  - (b) Triangulation으로 선택된 점의 3차원 좌표 복원
  - (c) Reprojection 예러 계산
  - (d) 선택점이 주어진 threshold를 만족하면, inlier set으로 저장
4. 2단계로 반복, 그러나 만약 반복 회수 마지막에 오면 다음 단계로 감
5. 가장 큰 inlier 집합을 찾아냄

6. Inlier 집합에 들어가지 못한 outlier를 찾아냄
7. Outlier 집합에 대해서 샘플 영상점을 랜덤하게 선택
8. 영상점으로부터 triangulation하여 이것을 reprojection
9. 이것을 계속 반복하여 reprojection 예러가 가장 적은 set을 다시 골라냄
10. Outlier 집합에서 올바른 영상 좌표점이 있을 경우,
11. 그 집합에 대해서 triangulation된 3차원 좌표를 바탕으로 outlier 보정을 함
12. 보정된 outlier를 inlier 집합에 추가

4. 실험 결과

실험은 우선 outlier를 찾아낸 후에, 이렇게 찾아낸 outlier 중 보정될 여지가 있는 영상점들에서 대해서 outlier가 올바르게 되었는지를 확인하는 방식으로 수행되었다.



(a)



(b)

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

그림 2는 영상 시퀀스 중 3개의 영상에 대해서, inlier와 outlier를 찾아내어 표시한 결과이다. 그림 2-(a) 결과는 outlier를 찾지만 한 것으로 총 135개의 영상점중 125개가 inlier로, 10개가 outlier로 판명되었다. 그림 2-(b)의 결과는 outlier 보정을 한 결과로, 총 135개의 영상점중 132개가 inlier로, 3개가 outlier 인채로 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, 여기에서 보듯이, outlier로 판명된 영상 대응점에 해당하는 3개의 영상점은 적어도 2개가 올바른 위치를 가지고 있어서 보정이 될 가능성이 충분히 있다. 그림 3-(b)는 outlier를 보정한 후의 결과로, 손가락에 있던 outlier들이 모두 올바르게 보정되었음을 확인 할 수 있다.

그림 4는 복도 영상에 대해서 얻어낸 inlier와 outlier를 표시한 것이다. 그림 4-(a)는 outlier를 판명해낸 것이며, 여기에서는 총 409개의 영상점중 399개가 inlier로 판명되었고, 10개가 outlier로 판명되었다. 그림 4-(b)는 outlier를 보정한 결과로, 총 409개의 영상점중 5개의 outlier만이 남아 10개의 outlier 중 5개의 점이 보정되었다.

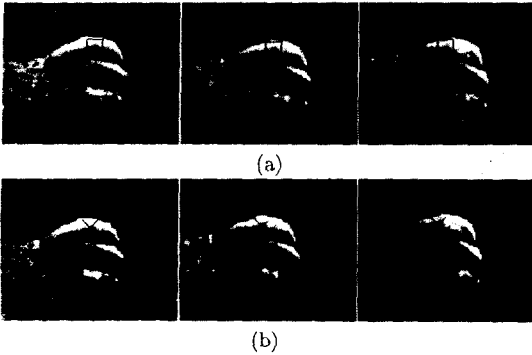


그림 3: 2 영상중 공통의 손부분을 확대한 것, (a) outlier로 판명된 것, (b) outlier를 보정한 후, 사각형표시: outlier, 십자표시: inlier

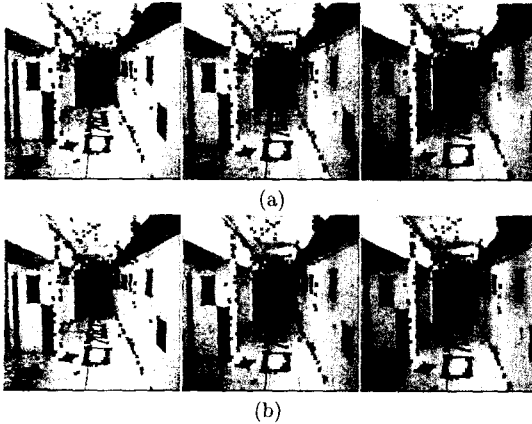


그림 4: (a) outlier를 찾아낸 3장의 영상, (b) outlier가 보정이 된 3장의 영상. 사각형표시: outlier, 십자표시: inlier

그림 5는 그림 4의 바닥에 있는 글자 'D'를 확대한 것으로, 결과에서 알 수 있듯이, 'D'글자의 오른쪽 상단 부분에서 outlier로 판명되었던 것이, outlier 보정을 거치면서 올바르게 위치 되었음을 알 수 있다. 또한 outlier 보정 과정을 거쳤음에도 불구하고, 여전히 outlier로 남아있는 것이 보여지고 있는데, 그 위치를 보면 3장의 영상에서 나타난 영상 점이 모두 일치 하지 않아, 그 중 어느것으로도 outlier를 보정할 참조점이 없기 때문이다. 이 결과에서 보듯이, outlier를 완전히 보정할 수 없는 조건에서는 outlier가 보정이 되지 않고 있음을 알 수 있다.

전체 실험 결과에서 알 수 있듯이, 영상에서 얻어낸 outlier를 찾는 것 뿐만이 아니라, outlier를 보정도 올바르게 할 수 있었다.

### 5. 결론

영상의 대응점이 주어졌을 때, triangulation과 RANSAC을 이용하여 outlier의 판단 및 보정하는 것이 가능했으며, 이는 기존에 trifocal tensor에서 제한되었던 영상의 개수와 outlier 제거로 인한 데이터 감소 문제를 해결할 수 있었다. 향후 연구 과제로는 outlier의 보정이 이루어진 데이터를 가지고, factorization 방법에서 missing 데이터에 대한

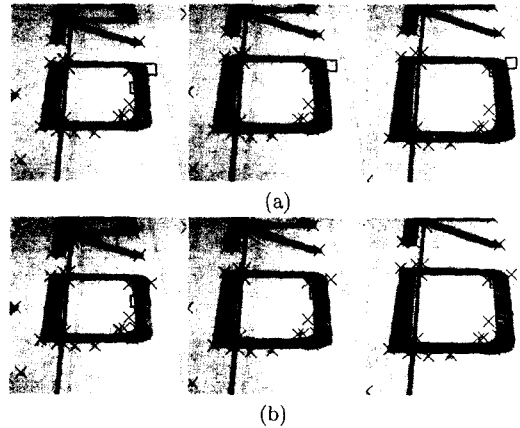


그림 5: 4 영상중 복도의 글자 'D'부분을 확대한 것, (a) outlier로 판명된 것, (b) outlier를 보정한 후, 사각형표시: outlier, 십자표시: inlier

해결책을 찾아보는 것이 있을 것이다.

### 참고문헌

- [1] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th Int. Joint Conf. on Artificial Intelligence*, Vancouver, 1981, pp. 674-679.
- [2] D. Martinec and T. Pajdla, "Outlier detection for factorization-based reconstruction from perspective images with occlusions," in *Proceedings of the Photogrammetric Computer Vision (PCV)*, Inst. f. Computer Graphics and Vision, TU-Graz, Sep 2002, pp. 161-164.
- [3] D. Q. Huynh and A. Heyden, "Outlier detection in video sequences under affine projection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, Kauai, Hawaii, Dec 2001, pp. 695-701.
- [4] R. I. Hartley and P. Sturm, "Triangulation," in *In Proc. Conference Computer Analysis of Images and Patterns*, Prague, Czech Republic, 1995.
- [5] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
- [6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," in *Communications of the Association for Computing Machinery*, vol. 24, no. 6, June 1981, pp. 381-395.
- [7] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *The 4th European Conference on Computer Vision*, 1996, pp. 709-720.