

## 교점의 추정 및 보간을 이용한 체커보드 검출

\*오상엽 \*\*조남익

서울대학교 INMC 전기정보공학부

\*syupoh@ispl.snu.ac.kr

## Finding locating checker board using corner detection and interpolation

\*Oh, Sang-yup \*\*Cho, Nam-ik

INMC, School of Electrical and Computer Engineering

Seoul National University

## 요약

카메라 캘리브레이션은 실제 세상인 3차원의 좌표와 카메라가 만든 영상의 2차원 좌표 사이에서 수학적 관계를 알기 위해서 필요하다. 보통 체커보드의 교점을 이용하여 2차원의 좌표를 정확하게 찾는 데 사용하며, 이는 카메라 캘리브레이션 계산으로 응용된다. 따라서 체커보드의 교점을 정확하게 찾아야만 카메라 캘리브레이션이 정상적인 성능을 낼 수 있다. 현존하는 체커보드 검출 방법은 입력 인수를 많이 필요로 하거나 정확도가 낮아 체커보드의 교점을 정확히 입력하지 못하면 좋지 않은 결과가 나타난다. 따라서 체커보드를 자동으로 검출하여 카메라 캘리브레이션 하는 방법은 아직 신뢰도가 낮은 편이다. 본 논문에서는 보다 안정적인 카메라 캘리브레이션을 위해서 체커 보드의 검출 성능을 높이고자 한다. 주위 픽셀들간의 미분 값을 기준삼아 검출된 교점들을 이용하여 체크 모양의 직선을 추측한다. 이 직선을 이용하면 장애물이 있거나 노이즈가 있어서 검출하기 어려운 교점들이 있는 경우에도 교점 보간 (point interpolation) 방법을 사용하여 나머지 교점들을 찾을 수 있다. 보간 과정을 통해서 검출에 방해가 되는 요소들이 있는 상황에서 체커 보드 교점 검출의 성능을 높이도록 하였다.

## 1. 서론

실제의 세상은 3차원이고 카메라가 세상을 찍으면 2차원의 이미지로 변하게 된다. 이런 상황에서 3차원의 점들이 사진 안에서 자리한 위치들을 기하학적으로 생각해 보면 이 위치들은 사진을 찍을 당시의 카메라의 위치와 방향에 의해서 결정이 된다. 실제 이미지는 사용된 렌즈, 렌즈와 이미지 센서와의 거리, 렌즈와 이미지 센서가 이루는 각 등 카메라 내부의 기구적인 부분에 의해서 크게 영향을 받는다. 따라서 3차원 점들이 영상에 투영된 위치를 구하거나 역으로 영상 좌표로부터 3차원 공간좌표를 복원할 때에는 이러한 내부 요인을 제거해야만 정확한 계산이 가능해진다. 이러한 내부 요인의 파라미터 값을 구하는 과정을 카메라 캘리브레이션이라 한다. 카메라 내부 파라미터로는  $x$ ,  $y$ 의 초점거리 (focal length) 각 2개,  $x$ ,  $y$ 의 주점(principal point) 각 2개, 비대칭 계수(skew coefficient)가 있다.

이러한 내부 파라미터 값들을 얻기 위해서는 여러 가지 방

법들이 연구되어 왔는데 그 중 Z. Zhang<sup>[1]</sup>의 동일 평면 (coplanar)을 여러 장 촬영하여 캘리브레이션을 하는 비 동일 평면(non-coplanar) 방식이 가장 널리 사용되고 있다. 그 중 체커 보드 패턴은 정사각형의 검은색 흰색 부분이 공존하는 패턴으로 교점이 특징점(feature point)로 사용되기 적당하여 캘리브레이션을 하는 패턴으로 자주 사용되고 있다. Z. Zhang의 방법의 경우 여러 장의 사진을 입력으로 넣어주면서 그 사진들의 체커 보드의 좌표를 알아야 하기 때문에 정밀한 좌표 값을 알지 못하면 정확하게 캘리브레이션이 되지 않는다.

카메라 캘리브레이션을 위한 입력 데이터로 체커보드의 위치 좌표를 얻는 방법으로는 사용자가 일일이 넣어주는 방법을 사용하기도 한다. MATLAB에서 배포한 캘리브레이션 toolbox<sup>[2]</sup>에서는 사용자가 직접 좌표를 지정하는 방법으로 좌표 정보를 얻는다. 사용자가 직접 좌표 정보를 입력하는 불편함을 없애기 위해서 자동으로 체커 보드 패턴을 찾는 연구도 여러 진행되고 있다.<sup>[3][4]</sup>

본 논문에서는 카메라 캘리브레이션을 위한 입력 인자로

사용되는 체커 보드의 세로, 가로 점의 개수, 체커 보드 이미를 가지고 체커 보드 교점의 좌표를 검출하고자 한다. 알고리즘은 체커 보드의 교점 좌표 추정, 긴 직선 추성, 교점 보간 세 가지 단계를 거쳐서 교점들의 좌표를 얻는다. 미분 공변 행렬(*gradient covariance matrix*)을 이용하여 기준 (*measure*)을 정하여 교점의 후보들을 정한다. 교점 후보들로 직선을 추정하여 일반적인 체커 모양을 예상한다. 마지막으로 이 직선을 이용하여 같은 직선상의 평균적인 점들의 거리를 알아내고 미처 검출하지 못한 교점들을 보간 (*interpolation*)한다. 첫 번째 단계에서 검출 못한 교점들을 보간 하여 최종 추정을 진행하기 때문에 체커 보드의 일부분이 장애물이나 노이즈에 의해 몇 개의 교점이 검출 안 되는 상황에서도 교점을 추정할 수 있다.

본 논문의 2절에서는 제안하는 체커 보드 교점 추출 알고리즘을 설명한다. 3절에서는 교점 추출 실험 결과를 제시하였고 4절에서는 결론과 향후 연구 방향을 정리하였다.

## 2. 체커 보드 교점 추출 알고리즘

### 2.1 체커 보드 교점 좌표 추정

체커 보드의 교점을 찾기 위해서 미분 행렬(*gradient matrix*)의 최소 고유값(*minimal eigenvalue*)을 계산하여 구하였다. 다음의 식을 통해 모든 픽셀 *p*에 대하여 블록 크기  $\times$ 블록 크기인 *S(p)*에 대하여 미분 값(*derivatives*)의 공변 행렬(*covariation matrix*)을 계산하였다.

$$M = \begin{bmatrix} \sum_{S(p)} \left(\frac{dl}{dx}\right)^2 & \sum_{S(p)} \left(\frac{dl}{dx} \frac{dl}{dy}\right) \\ \sum_{S(p)} \left(\frac{dl}{dx} \frac{dl}{dy}\right) & \sum_{S(p)} \left(\frac{dl}{dy}\right)^2 \end{bmatrix}$$

본 논문에서는 미분 값을 구하기 위해 *sobel operator*를 사용하였다.<sup>[5]</sup> *Sobel operator*로 구한 미분 값으로 행렬 *M*을 구한다.

행렬 *M*의 고유값(*eigen value*)와 고유 벡터(*eigen vector*)들을 구한 뒤, 그 중 작은 고유값을 취한다. 그 값을 체커 보드의 교점의 후보로써의 척도(*corner quality measure*)로 사용한다. *non-maximum suppression*<sup>[6]</sup>을 수행하여 일정 값 이하의 교점들을 추려낸다. 추려낸 교점들끼리 값이 높은 교점을 기준으로 일정 거리 이하로 가까운 교점들은 후보에서 제외된다.

### 2.2 체커 보드의 긴 직선 추정

체커 모양을 구성하는 직선을 추출하는 방법으로는 *RANSAC* 알고리즘을 사용한다. 즉, 여러 개의 후보점들을 뽑아서 랜덤으로 후보점들 중 임의의 점 4개를 뽑는다. 점 4개를 뽑아 의사 역행렬(*pseudo-inverse*)로 임의의 직선을 만들고 후보 교점들과의 거리를 측정한다. 그 중 거리가 가까운 점들의 집합을 구하고 그 점들로 다시 직선을 구한다. 이 과정을 반복하여 근접한 점의 개수가 가장 많은 직선을 선택하여 직선을 정한다. 앞에서 입력 받은 가로 점, 세로 점의 개수 정보를 바탕으로 가로 점, 세로 점 중 더 점의 개수가 많은 방향의 직선을 찾는다. 예를 들어 [그림 1]처럼 가로가 11개, 세로가 5개인 체커 보드가 입력으로 들어왔다고 하면, 더 길이가 길고 교점이 많은 세로 직선을 찾는다. 만약 직선이 근접하는 점의 개수가 아무리 많아도 5개를 넘지 못하거나 11개를 넘는 경우는 처음 교점들을 잘 못 추출한 경우이므로 버리고 처음부터 다시 추출하여 앞의 추출 과정을 반복한다. 직선의 추정이 완료되면 임의의 점들을 새로 추출하여서 다음 직선을 추정한다. 이때 추출하는 임의의 점들은 앞서 추정한 직선에 속하지 않는 임의의 점으로 하여 새로운 직선과 앞선 직선이 겹치지 않도록 한다.

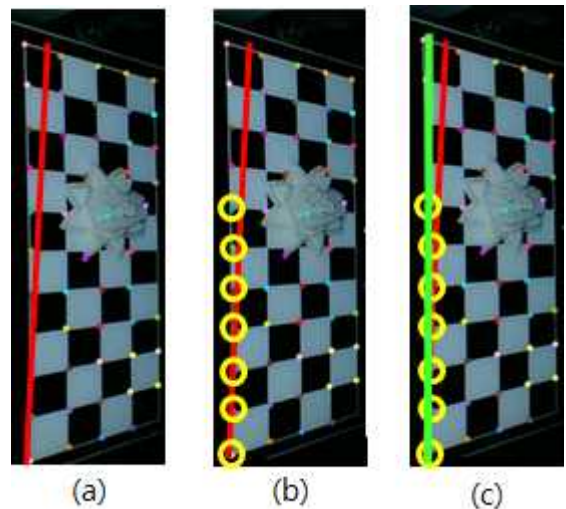


그림 1. RANSAC으로 직선 추출과정. (a) 랜덤한 4개의 점들로 직선을 추출 (b) 추출한 직선과 일정 거리 이하의 점들 (c) 일정거리 이하의 점들로 다시 구성한 직선

### 2.3 체커 보드의 교점 좌표 보간

마지막 단계에서는 보간(*interpolation*)을 하여 첫 번째 단계에서 추출하지 못한 교점들을 추정할 수 있다. 입력 인수로 받은 가로 점과 세로 점의 개수로 현재 직선

내에서 포함되지 못한 점들의 좌표를 추정할 수 있다. 체커 보드는 정사각형들이 모여서 만들어진 직사각형이므로 점들 간의 평균을 이용하여 추정하면 추출하지 못한 점의 좌표를 알 수 있다.

### 3. 실험 셋팅 및 결과

실험을 위해서 다음과 같이 설정하였다. 처음에 입력 받는 영상을 먼저 320x240으로 크기 조절을 한다. 첫 번째 임의의 교점을 추출하는 과정에서 찾을 수 있는 최대 교점의 개수(  $n_{max}$  )는 가로선과 세로선을 곱한 값으로 설정하였다 .

$$n_{max} = n_{liney} \times n_{linex}$$

추출한 교점들을 일정 거리 이하로 잘라낼 때 일정 거리(  $l$  )는 찾은 교점들의 개수(  $n_f$  )가 최대 교점의 개수가 되도록 일정 거리를 설정하였다.

$$\operatorname{argmax}_l n_f$$

두 번째 직선을 추정하는 과정에서는, 추정된 직선과 추출해낸 점들이 앞에서 설정한 일정거리의 0.2 이하

$distance(line_i, point_j) < 0.2l \rightarrow line_i \text{ include } point_j$  라면 속한다고 실험적으로 판단하였다.

찾은 직선이 점 몇 개 이상을 포함해야지 제대로 찾았다고 결정하는  $n_{correct}$  는 다음과 같다.

$$n_{correct} = n_{max} - \left( \frac{\alpha}{l} + \beta \right)$$

이상의 설정들을 사용하여 추출, 직선생성, 교점 보간을 진행한 실험의 결과는 다음과 같다.



그림 2. 복잡한 배경이 있을 때의 교점 검출

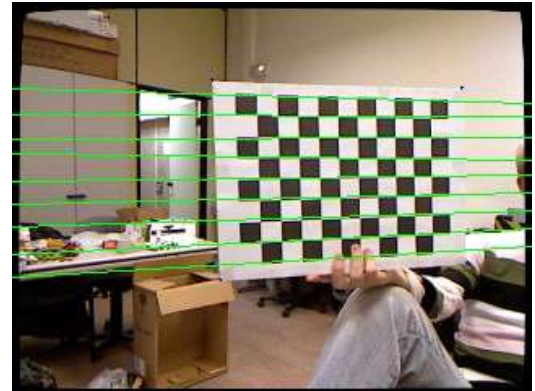


그림 3. 직선 추측이 완료된 영상



그림 4. 보간 작업을 통해 교점들을 추가한 영상

### 4. 결론 및 향후 연구 방향

카메라 영상에 정밀한 영상처리를 적용하기 위해서는 보다 정확한 카메라 캘리브레이션이 필수이다. 체커 보드를 사용하여 카메라 캘리브레이션을 하는 경우 만약 체커 보드 위에 있는 교점이 장애물이 있어서 가려지거나 노이즈에 의해 인식이 안 되는 문제가 생길 수 있다. 본 논문에서 제안한 직선의 추정을 이용한 교점 보간을 사용하면 이러한 문제들을 해결하고 가려진 교점들의 위치도 추정 가능하다.

하지만 추출해낸 교점들이 너무 적거나, 체커 모양 말고 엉뚱한 feature에서 교점들을 추출해 내는 경우가 있을 수도 있다. 추출해낸 교점들을 바탕으로 추정을 시작 하기 때문에, 첫 단계의 추출 성능이 좋지 못하다면 보간이 아무리 잘되어도 문제가 있는 결과가 나오게 된다. 이를 보완하기 위한 여러 방법들이 제안될 수 있는데, 추출한 점들끼리의 거리를 측정하여 refine 한 뒤에 직선을 만드는 방법이나, 아예 직선을 만드는 과정에서 직선을 refine 하는 방법 등이다. 이와 같은 refine 과정은 정말 필요한 정보들을 잃어버릴 수 있기 때문에 안정성 부분에서 좀 더 연구가 필요하다.

## 감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2016-R2718-16-0014)

## 참고문헌

- [1] Z. Zhang, "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations," in Proc. Seventh IEEE Int'l. Conf. on Comp. Vis., Vol. 1, pp. 20 - -27 1999.
- [2] J. Y. Bouguet, "Camera calibration toolbox for Matlab," [http://www.vision.caltech.edu/bouguet/tj/calib\\_doc/](http://www.vision.caltech.edu/bouguet/tj/calib_doc/).
- [3] C. Shu, A. Brunton, and M. Fiala, "Automatic grid finding in calibration patterns using delaunay triangulation," Technical Report NRC-46497/ERB-1104, Nat. Research Council, Inst. for Information Technology (2003)
- [4] Chunsheng Yu, Qingjin Peng "Robust recognition of checkerboard pattern for camera calibration"
- [5] Scharr, Hanno, 2000, Dissertation (in German), Optimal Operators in Digital Image Processing
- [6] Lindeberg, Tony "Edge detection and ridge detection with automatic scale selection", International Journal of Computer Vision, 30, 2, pp 117-154, 1998. (Includes the differential approach to non-maximum suppression.)