

# 고차 polynomial을 이용한 정밀한 카메라 캘리브레이션

## An Accurate Camera Calibration Using Higher-Order Polynomials

조태훈<sup>1</sup>

<sup>1</sup> 한국기술교육대학교 정보기술공학부  
E-mail: thcho@kut.ac.kr

### 요 약

카메라 캘리브레이션은 비전(vision) 시스템의 광학왜곡을 보정하기 위해, 영상 좌표계와 실세계 좌표계간의 변환관계를 정의해 주는 mapping을 구하는 과정으로 카메라를 이용한 측정, 검사, 위치보정 등의 응용에서 매우 중요하다. 카메라 캘리브레이션 방법으로 많이 사용되는 Tsai 알고리즘은 여러 카메라 내부 상수들을 필요로 하며, 적절한 활용을 위해서는 이에 대한 이해와 카메라와 렌즈왜곡의 모델에 대한 사전지식을 요한다. 본 논문에서는 카메라나 렌즈왜곡에 대한 모델이나 가정없이, 영상좌표와 실세계 좌표간의 변환을 고차(higher order) polynomial을 이용하여 구현하여 사용이 손쉬운 카메라 캘리브레이션 방법을 소개하고 성능을 평가하였다. 성능 평가 결과, 3차 polynomial을 이용한 카메라 캘리브레이션 방법이 Tsai알고리즘보다 정밀도에서 우수하였다.

**Key Words** : camera calibration, polynomial, vision system, optical distortion

### 1. 서 론

camera calibration의 목적은 모든 vision system에 존재하는 두 가지 유형의 왜곡을 제거하기 위함이다. 하나의 왜곡 유형은 "화소 왜곡(pixel distortion)"으로 화소(pixel)가 완전하게 정방형이 아닌 경우에 발생한다. 이 왜곡은 선형적(linear)이며, 영상의 전 범위에 걸쳐 예측 가능하다. 또 다른 유형의 왜곡은 "광학 왜곡(optical distortion)"으로 렌즈의 불완전한 특성같은 검사시스템 광학계의 비정형성(irregularities)에 의해 생긴다. 이 왜곡은 비선형적으로 영상에서의 위치에 따라 불균일하게 변화하여, 측정하기 어렵다. 특히, 영상의 가장 자리에 가까울수록 이 왜곡은 심하다. 대략적으로는 화소왜곡이 vision system 왜곡의 약 2/3 정도, 광학왜곡이 약 1/3 정도 차지한다고 볼 수 있다. 이 두 가지 유형의 왜곡을 보정하고, 영상의 해상도를 결정하면, 영상내 임의의 두 점간의 실제 거리(physical distance)를 정확하게 측정할 수 있다.

이러한 왜곡을 보정하기 위해서, camera calibration은 영상 좌표계와 실세계 좌표계간의 변환관계를 정의해 주는 mapping을 구하는 과정이다. 이를 위해 먼저, 이때 알고있는 실세

계 치수의 "특징"들로 구성되는 calibration plate의 영상을 취득하고, 특징(보통, dot 형상)들의 중심점들의 영상좌표를 추출한다. 그리고, 이 영상좌표들과 calibration plate의 dot 들의 실세계 좌표들간의 상호 변환관계를 구하게 된다. 따라서, camera calibration이 이루어지면, 영상좌표를 실세계좌표로, 실세계 좌표를 영상좌표로 변환할 수 있다. 그림 1은 수평, 수직방향으로 등간격인 격자형태로 놓인 흑점(black dots)들로 이루어진 calibration plate의 영상을 보여주고 있다.

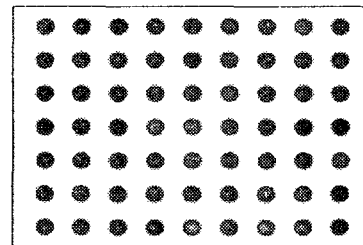


그림 1 calibration plate

calibration plate는 일반적으로 다음과 같은 사양이 바람직하다.

- 2차원 원형 dot grids
- dot 의 직경은 수평/수직 인접 dot간 거리의 1/2보다 작아야 한다.

- dot 의 직경은 적어도 10 pixels 이상  
 - calibration plate의 가장자리는 영상취득시 나타나지 않도록 한다.

카메라 캘리브레이션 방법에서 가장 많이 사용되는 방법으로 Tsai 알고리즘[1]과 Zhang 알고리즘[2]이 있다. Tsai카메라 모델은 3D-2D perspective projection과 1차 radial 렌즈왜곡의 pin-hole 모델에 기반하며, Zhang 알고리즘은 여러 장의 영상을 취득하는 방식으로 3차원 카메라 캘리브레이션에 많이 사용된다. 이 방법들은 카메라의 고유파라미터를 정확하게 추출하기 위해 최적화 기법을 사용하거나 여러 장의 영상을 얻어야 한다.

본 논문에서는 영상의 기하학적 왜곡[3]을 보정하기 위해 사용되는 선형변환 및 고차(higher order) polynomial 변환에 의한 2차원 카메라 캘리브레이션을 제안하고 성능평가를 하였다.

먼저, calibration plate 영상에서 추출된 dot들의 영상좌표가  $(I_1, J_1), (I_2, J_2), \dots, (I_N, J_N)$ 이고, 이에 대응되는 실세계 좌표들이  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ 라고 하자. 여기서, 영상좌표  $(I_i, J_i)$ 는 실세계좌표  $(x_i, y_i)$ 에 대응하며, N은 calibration point의 개수를 나타낸다.

## 2. 선형변환에 의한 카메라 보정

영상좌표 (I, J)와 실세계 좌표 (x, y)와의 관계를 선형변환(linear transformation)으로 나타내면,

$$I = ax + by + p \tag{1}$$

$$J = ex + fy + q \tag{2}$$

로 표시할 수 있다.

여기서, a, b, e, f는 rotation과 scaling에 관련된 파라미터이고, p, q는 translation 파라미터이다. N개의 영상좌표와 이에 대응되는 실세계좌표가 주어졌을 때, 식 (1), (2)을 최소자승오차(least square error)상 가장 정확하게 만족시키는 파라미터 a, b, p, e, f, q를 구한다. N=3이면, 유일해(unique solution)가 존재하고, N>3이면 overdetermined 선형방정식이 되어 pseudo-inverse 방법에 의해 해를 구한다.

N개의 영상좌표와 실세계좌표가 주어지면, 다음과 같은 행렬로 표시할 수 있다.

$$\begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_N \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ p \end{pmatrix}$$

즉,  $I = Ak$ ,

$$I = \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_N \end{pmatrix}, A = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{pmatrix}, k = \begin{pmatrix} a \\ b \\ p \end{pmatrix}$$

여기서,  $\sum_{i=1}^N (I_i - (ax_i + by_i + p))^2$  을 최소화하는 파라미터 a, b, p는 pseudo inverse method에 의해 아래와 같이 구할 수 있다.

$$k = (A^T A)^{-1} A^T I$$

$$= \begin{pmatrix} \sum_{k=1}^N x_k^2 & \sum_{k=1}^N x_k y_k & \sum_{k=1}^N x_k \\ \sum_{k=1}^N x_k y_k & \sum_{k=1}^N y_k^2 & \sum_{k=1}^N y_k \\ \sum_{k=1}^N x_k & \sum_{k=1}^N y_k & N \end{pmatrix}^{-1} \begin{pmatrix} \sum_{k=1}^N x_k I_k \\ \sum_{k=1}^N y_k I_k \\ \sum_{k=1}^N I_k \end{pmatrix}$$

같은 방법으로,  $\sum_{i=1}^N (J_i - (ex_i + fy_i + q))^2$  을 최소화하는 파라미터 e, f, q는 pseudo inverse method에 의해 아래와 같이 구해진다.

$$J = Al,$$

$$J = \begin{pmatrix} J_1 \\ J_2 \\ \vdots \\ J_N \end{pmatrix}, A = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{pmatrix}, l = \begin{pmatrix} e \\ f \\ q \end{pmatrix}$$

$$l = (A^T A)^{-1} A^T J$$

$$= \begin{pmatrix} \sum_{k=1}^N x_k^2 & \sum_{k=1}^N x_k y_k & \sum_{k=1}^N x_k \\ \sum_{k=1}^N x_k y_k & \sum_{k=1}^N y_k^2 & \sum_{k=1}^N y_k \\ \sum_{k=1}^N x_k & \sum_{k=1}^N y_k & N \end{pmatrix}^{-1} \begin{pmatrix} \sum_{k=1}^N x_k J_k \\ \sum_{k=1}^N y_k J_k \\ \sum_{k=1}^N J_k \end{pmatrix}$$

위와 같이, 파라미터 a, b, p, e, f, q를 구하면, 실세계좌표 (x, y)가 주어졌을 때 영상좌표 (I, J)를 계산할 수 있다. 또한, 실세계 좌표 (x, y)를 영상좌표 (I, J)로도 표시할 수 있다. 즉, 식 (1)과 (2)를 행렬로 표시하면,

$$\begin{pmatrix} I \\ J \end{pmatrix} = \begin{pmatrix} a & b \\ e & f \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} p \\ q \end{pmatrix}$$

$$af \neq be \text{ 이면, } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ e & f \end{pmatrix}^{-1} \begin{pmatrix} I-p \\ J-q \end{pmatrix}$$

$$= \frac{1}{af-be} \begin{pmatrix} f & -b \\ -e & a \end{pmatrix} \begin{pmatrix} I-p \\ J-q \end{pmatrix}$$

따라서, 영상좌표 (I, J)가 주어지면, 이에 대응하는 실세계 좌표 (x, y)도 구할 수 있게 된다

### 3. polynomial 변환에 의한 카메라 보정

선형변환(linear transformation)에 의한 카메라 보정방법은 간단하여 많이 사용되나, 광학계의 비선형 왜곡을 보상하지 못해 고정밀도를 얻기 어려운 단점이 있다. 이러한, 비선형 왜곡을 보상하여 보다 정밀도를 향상시키기 위해서 high-order polynomial 변환을 이용한 카메라 캘리브레이션을 제안한다.

영상좌표  $(I, J)$ 와 실세계 좌표  $(x, y)$ 와의 관계를 M차 polynomial transformation으로 표시하면,

$$x = \sum_{m=0}^M \sum_{n=0}^M k_{mn} I^m J^n, \quad y = \sum_{m=0}^M \sum_{n=0}^M l_{mn} I^m J^n$$

N개의 영상좌표와 이에 대응되는 실세계좌표가 주어졌을 때, 위식을 최소자승오차(least square error)상 가장 정확하게 만족시키는 즉,

$$\sum_{i=1}^N (x_i - \sum_{m=0}^M \sum_{n=0}^M k_{mn} I^m J^n)^2 \quad \text{을 최소화하는}$$

$$\text{파라미터 } k_{mn}, \quad \sum_{i=1}^N (y_i - \sum_{m=0}^M \sum_{n=0}^M l_{mn} I^m J^n)^2 \text{을}$$

최소화하는  $l_{mn}$ ,  $0 \leq m, n \leq M$  를 구한다.  $N=(M+1)^2$ 이면, 유일해(unique solution)가 존재하고,  $N > (M+1)^2$ 이면 overdetermined 선형방정식이 되어 pseudo-inverse 방법에 의해 해를 구한다.

1) M=2 일 때;

$$\mathbf{A} \mathbf{k} = \mathbf{x}$$

$$\mathbf{A} = \begin{pmatrix} 1 & J_1 & J_1^2 & I_1 & I_1 J_1 & I_1 J_1^2 & I_1^2 & I_1^2 J_1 & I_1^2 J_1^2 \\ 1 & J_2 & J_2^2 & I_2 & I_2 J_2 & I_2 J_2^2 & I_2^2 & I_2^2 J_2 & I_2^2 J_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & J_N & J_N^2 & I_N & I_N J_N & I_N J_N^2 & I_N^2 & I_N^2 J_N & I_N^2 J_N^2 \end{pmatrix}$$

$$\mathbf{k} = (k_{00} \ k_{01} \ k_{02} \ k_{10} \ k_{11} \ k_{12} \ k_{20} \ k_{21} \ k_{22})^t$$

$$\mathbf{x} = (x_1 \ x_2 \ \dots \ x_N)^t$$

여기서  $\mathbf{k}$  는 pseudo-inverse에 의해 아래와 같이 구해진다.

$$\mathbf{k} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{x}$$

또한,

$$\mathbf{A} \mathbf{l} = \mathbf{y}$$

$$\mathbf{l} = (l_{00} \ l_{01} \ l_{02} \ l_{10} \ l_{11} \ l_{12} \ l_{20} \ l_{21} \ l_{22})^t$$

$$\mathbf{y} = (y_1 \ y_2 \ \dots \ y_N)^t$$

$\mathbf{l}$ 을 pseudo-inverse method에 의해 구하면,

$$\mathbf{l} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}$$

2) M=3 일 때;

$$\mathbf{A} \mathbf{k} = \mathbf{x}$$

$$\mathbf{A} = \begin{pmatrix} 1 & J_1 & J_1^2 & J_1^3 & I_1 & I_1 J_1 & I_1 J_1^2 & I_1 J_1^3 \\ 1 & J_2 & J_2^2 & J_2^3 & I_2 & I_2 J_2 & I_2 J_2^2 & I_2 J_2^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & J_N & J_N^2 & J_N^3 & I_N & I_N J_N & I_N J_N^2 & I_N J_N^3 \end{pmatrix}$$

$$\left. \begin{matrix} I_1^2 & I_1^2 J_1 & I_1^2 J_1^2 & I_1^2 J_1^3 & I_1^3 & I_1^3 J_1 & I_1^3 J_1^2 & I_1^3 J_1^3 \\ I_2^2 & I_2^2 J_2 & I_2^2 J_2^2 & I_2^2 J_2^3 & I_2^3 & I_2^3 J_2 & I_2^3 J_2^2 & I_2^3 J_2^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ I_N^2 & I_N^2 J_N & I_N^2 J_N^2 & I_N^2 J_N^3 & I_N^3 & I_N^3 J_N & I_N^3 J_N^2 & I_N^3 J_N^3 \end{matrix} \right\}$$

$$\mathbf{k} = (k_{00} \ k_{01} \ k_{02} \ k_{03} \ k_{10} \ k_{11} \ k_{12} \ k_{13} \ k_{20} \ k_{21} \ k_{22} \ k_{23} \ k_{30} \ k_{31} \ k_{32} \ k_{33})^t$$

$$\mathbf{x} = (x_1 \ x_2 \ \dots \ x_N)^t$$

따라서,

$$\mathbf{k} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{x}$$

또한,

$$\mathbf{A} \mathbf{l} = \mathbf{y}$$

$$\mathbf{l} = (l_{00} \ l_{01} \ l_{02} \ l_{03} \ l_{10} \ l_{11} \ l_{12} \ l_{13} \ l_{20} \ l_{21} \ l_{22} \ l_{23} \ l_{30} \ l_{31} \ l_{32} \ l_{33})^t$$

$$\mathbf{y} = (y_1 \ y_2 \ \dots \ y_N)^t$$

그러므로,

$$\mathbf{l} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}$$

2차 polynomial을 사용하면, calibration parameter  $k_{mn}, l_{mn}$ 의 개수의 합이  $2*(2+1)^2=18$ 개이고, 3차 polynomial을 사용하면 파라미터 개수 합이  $2*(3+1)^2=32$ 개로 늘어나게 되지만, 3차 polynomial을 사용할 경우의 calibration 정밀도는 더 좋아진다. 물론, 파라미터 개수가 증가함에 따라, 요구되는 calibration point들의 개수도 증가한다.

위에서 영상좌표를 실세계좌표로 변환하는 polynomial 변환을 보였지만, 같은 방법으로 실세계 좌표를 영상좌표로 변환하는 polynomial 변환도 구할 수 있다.

### 4. 성능평가

위의 camera calibration 방법들의 성능의 평가하기 위하여 정밀도를 측정하였다. 초점거리 25mm인 CCTV 렌즈에 extension ring 2.5mm를 부착한 카메라 CVM-10(CCIR type)으로 768x574 크기의 calibration plate의 영상을 frame grabber 보드로 취득하였다. 사용된

calibration plate는 그림 1과 같이 직경 2mm 인 흑점(black dot)들이 수평, 수직 방향으로 3.3 mm 간격을 두고 grid 모양으로 배열된 것이다. 이 plate는 유리판위에 이러한 흑점들이 위치정밀도오차  $\pm 1\sim 2$   $\mu\text{m}$ 의 고정도로 인쇄되었다. 취득된 영상은 흑점이 수평, 수직방향으로 각각 8개, 6개 모두 48개가 들어 있고, 카메라의 FOV(Field of View)는 25x20mm 정도였다. 취득된 calibration plate 영상으로부터 connected component labelling[4] 알고리즘을 사용하여, 각 dot의 중심좌표를 추출하였다. 이에 대응되는 실세계좌표는 인접 dot간 거리가 3.3mm인 것을 이용하여 임의의 원점을 기준으로 하여 할당하였다.

48개의 calibration dot 들의 영상좌표와 실세계 좌표를 가지고, 선형변환, 2차, 3차 polynomial 변환, Tsai알고리즘(ccal: 기본 coplanar calibration, ccal\_fo: 최적화된 coplanar calibration)에 의한 calibration을 수행하고, calibration 오차를 계산한 결과를 표 1 과 표 2에 보인다. Tsai 알고리즘은 많이 사용되는 Wilson의 프로그램[5]을 가지고 테스트하였다. 여기서,  $ixy$ ,  $imax$  는 각각 영상좌표계에서의 거리(R) 방향 RMS(Root Mean Square) 오차, 거리방향 최대 오차를 가리킨다. 또한,  $wxy$ ,  $wmax$  는 각각 실세계 좌표계에서의 거리(R) 방향 RMS 오차, 거리방향 최대 오차를 가리킨다

표 1 영상좌표계에서의 RMS 오차 [pixel]

	$ixy$	$imax$
선형변환	0.44	1.33
2차 polynomial	0.29	0.40
3차 polynomial	0.04	0.09
Tsai (ccal)	0.26	1.06
Tsai (ccal_fo)	0.06	0.19

표 2 실세계좌표상에서의 RMS 오차 [mm]

	$wxy$	$wmax$
선형변환	0.016	0.048
2차 polynomial	0.010	0.014
3차 polynomial	0.002	0.003
Tsai (ccal)	0.009	0.038
Tsai (ccal_fo)	0.002	0.007

이 결과로부터, 선형변환에 의한 방법이 가장 정확도가 떨어지는 것을 알 수 있다. 또한,

2차 polynomial에 의한 방법은 선형변환에 의한 방법보다는 정밀하지만, 아직 오차가 큰 수준에 머무르고 있다. 반면에, 3차 polynomial에 의한 방법은 RMS 오차가 0.04 pixel 정도로 낮게 유지되며, 최대 오차가 0.09 pixel 인 고정밀의 calibration 방법임을 알 수 있다.

Tsai알고리즘은 최적화(full optimization)기능을 가진 ccal\_fo가 기본 기능의 ccal보다 성능이 낮지만, 3차 polynomial변환보다 예러가 크고, 2차 polynomial 변환과 성능이 비슷함을 볼 수 있다. 이는 Tsai알고리즘은 파라미터가 11개(카메라 내부파라미터 5개, 외부파라미터 6개)로 파라미터 갯수에서 3차 polynomial 변환보다 파라미터 갯수보다 적기 때문인 것으로 보인다. 따라서, 충분한 캘리브레이션 점들을 얻을 수 있다면, Tsai알고리즘보다 3차 polynomial 변환에 의한 캘리브레이션을 사용하는 것이 보다 정밀할 것으로 판단된다.

## 5. 결론

본 논문에서는 카메라나 렌즈왜곡에 대한 모델이나 가정없이, 영상좌표와 실세계 좌표간의 변환을 고차(higher order) polynomial을 이용하여 구현하여 사용이 손쉬운 카메라 캘리브레이션 방법을 소개하고 성능을 평가하였다. 성능 평가 결과, 3차 polynomial을 이용한 카메라 캘리브레이션 방법이 Tsai알고리즘보다 정밀도에서 우수하였다.

## 참 고 문 헌

- [1] R. Y. Tsai, "A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, pp. 323-344, August 1987
- [2] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 11, pp. 1330-1334, 2000.
- [3] J. C. Russ, *The Image Processing Handbook*, 4<sup>th</sup> ed., CRC Press, 2002.
- [4] R. Jain, R. Kasturi, and B.G. Schunck, *Machine Vision*, McGraw-Hill, 1995.
- [5] <http://www.cs.cmu.edu/~rgw/TsaiCode.html>