A Flexible New Technique for Camera Calibration

Zhengyou Zhang

December 2, 1998
(updated on December 14, 1998)
(updated on March 25, 1999)
(updated on Aug. 10, 2002; a typo in Appendix B)
(last updated on Aug. 13, 2008; a typo in Section 3.3)

Technical Report
MSR-TR-98-71

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

zhang@microsoft.com
http://research.microsoft.com/~zhang

A Flexible New Technique for Camera Calibration

Zhengyou Zhang
Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA
zhang@microsoft.com   http://research.microsoft.com/˜zhang

Contents

1 Motivations **2**

2 Basic Equations **3**
- 2.1 Notation 3
- 2.2 Homography between the model plane and its image 4
- 2.3 Constraints on the intrinsic parameters 4
- 2.4 Geometric Interpretation† 4

3 Solving Camera Calibration **5**
- 3.1 Closed-form solution 5
- 3.2 Maximum likelihood estimation 6
- 3.3 Dealing with radial distortion 7
- 3.4 Summary 8

4 Degenerate Configurations **8**

5 Experimental Results **9**
- 5.1 Computer Simulations 9
- 5.2 Real Data 10
- 5.3 Sensitivity with Respect to Model Imprecision‡ 14
  - 5.3.1 Random noise in the model points 14
  - 5.3.2 Systematic non-planarity of the model pattern 15

6 Conclusion **17**

A Estimation of the Homography Between the Model Plane and its Image **17**

B Extraction of the Intrinsic Parameters from Matrix $\mathbf{B}$ **18**

C Approximating a 3×3 matrix by a Rotation Matrix **18**

D Camera Calibration Under Known Pure Translation§ **19**

† added on December 14, 1998
‡ added on December 28, 1998; added results on systematic non-planarity on March 25, 1998
§ added on December 14, 1998, corrected (based on the comments from Andrew Zisserman) on January 7, 1999

A Flexible New Technique for Camera Calibration

Abstract

주제

We propose a flexible new technique to easily calibrate a camera. It is well suited for use without specialized knowledge of 3D geometry or computer vision. The technique only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. Either the camera or the planar pattern can be freely moved. The motion need not be known. Radial lens distortion is modeled. The proposed procedure consists of a closed-form solution, followed by a nonlinear refinement based on the maximum likelihood criterion. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques which use expensive equipment such as two or three orthogonal planes, the proposed technique is easy to use and flexible. It advances 3D computer vision one step from laboratory environments to real world use.

장점

Input

최소 2개의 다른 orientation에서 보여진 하나의 Plane pattern을 관찰하는 Camera가 필요하다.

Camera

Plane pattern

· Camera, planar pattern 을 움직이는 움직임이 자유로워도 된다.

· 움직임이 어떤지 알 필요도 없다.

Radial lens Distortion을 Modelling 한다.

Component
· closed-form solution
· Non-linear Refinement based on Maximum Likelihood criterion.

Index Terms— Camera calibration, calibration from planes, 2D pattern, absolute conic, projective mapping, lens distortion, closed-form solution, maximum likelihood estimation, flexible setup.

1 Motivations

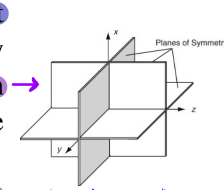
Camera Calibration의 필요성 : 2D Image로부터 Metric Information을 추출하기 위해 3D Computer vision 분야에서 필수적이다.

Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. Much work has been done, starting in the photogrammetry community (see [2, 4] to cite a few), and more recently in computer vision ([9, 8, 23, 7, 26, 24, 17, 6] to cite a few). We can classify those techniques roughly into two categories: photogrammetric calibration and self-calibration.

Camera Calibration의 분류
· Photogrammetric Calibration
· Self-Calibration

Photogrammetric calibration. Camera calibration is performed by observing a calibration object whose geometry in 3-D space is known with very good precision. Calibration can be done very efficiently [5]. The calibration object usually consists of two or three planes orthogonal to each other. Sometimes, a plane undergoing a precisely known translation is also used [23]. These approaches require an expensive calibration apparatus, and an elaborate setup.

Camera Calibration은 3차원공간내 Geometry를 매우 정밀하게 알고있는 Calibration Object를 관찰하여 수행된다.



· two or three planes orthogonal
· a precisely known translation plane

Self-calibration. Techniques in this category do not use any calibration object. Just by moving a camera in a static scene, the rigidity of the scene provides in general two constraints [17, 15] on the cameras' internal parameters from one camera displacement by using image information alone. Therefore, if images are taken by the same camera with fixed internal parameters, correspondences between three images are sufficient to recover both the internal and external parameters which allow us to reconstruct 3-D structure up to a similarity [16, 13]. While this approach is very flexible, it is not yet mature [1]. Because there are many parameters to estimate, we cannot always obtain reliable results.

Self-Calibration은 별도의 Calibration Object가 필요하지 않다.

↳ 정적인 Scene에서 Camera를 움직임.

↳ Scene의 rigidity는 Image 정보만을 사용하여 하나의 Camera의 변위로부터 Camera Internal Parameter에 대한 2가지의 Constraint를 제공한다.

고정된 Internal Parameter를 지닌 동일한 Camera로 사진을 촬영한다면, 3개의 Image 간의 대응점들은 3차원 구조를 Reconstruct 가능하도록 하는 Internal · External Parameter 모두와 Similarity 까지 복구하기에 충분하다.

Other techniques exist: vanishing points for orthogonal directions [3, 14], and calibration from pure rotation [11, 21].

Our current research is focused on a desktop vision system (DVS) since the potential for using DVSs is large. Cameras are becoming cheap and ubiquitous. A DVS aims at the general public, who are not experts in computer vision. A typical computer user will perform vision tasks only from time to time, so will not be willing to invest money for expensive equipment. Therefore, flexibility, robustness and low cost are important. The camera calibration technique described in this paper was developed with these considerations in mind.

본 연구는 DVS에 초점을 맞춤.

↳ 대중적.

↳ Expensive X

⇒ Flexibility, Robustness, low cost가 중요한 Property.

Principle point란?

Intrinsic Parameter란?

Extrinsic Parameter란?

다른 orientation에서 본 Planar Pattern을 관찰하는 Camera만 있으면 된다.
↳ Print하고 평면에 붙이기만 하면 된다.
↳ Camera · Plane 둘 중 하나를 이동해도 되고 Motion을 알 필요는 없다.

The proposed technique only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. The pattern can be printed on a laser printer and attached to a "reasonable" planar surface (e.g., a hard book cover). Either the camera or the planar pattern can be moved by hand. The motion need not be known. The proposed approach lies between the photogrammetric calibration and self-calibration, because we use 2D metric information rather than 3D or purely implicit one. Both computer simulation and real data have been used to test the proposed technique, and (장점) very good results have been obtained. Compared with classical techniques, the proposed technique is considerably (장점) more flexible. Compared with self-calibration, it gains considerable (장점) degree of robustness. We believe the new technique advances 3D computer vision one step from laboratory environments to the real world.

이 방법은 Photogrammetry Calibration과 Self-Calibration 방법 사이에 위치한다.
↳ 3D 또는 Purely Implicit 보다는 2D Metric Information을 사용하기 때문이다.

Note that Bill Triggs [22] recently developed (다른 방법1) a self-calibration technique from at least 5 views of a planar scene. His technique is more flexible than ours, but has difficulty to initialize. Liebowitz and Zisserman [14] described a technique of (다른 방법2) metric rectification for perspective images of planes using metric information such as a known angle, two equal though unknown angles, and a known length ratio. They also mentioned that calibration of the internal camera parameters is possible provided at least three such rectified planes, although no experimental results were shown.

The paper is organized as follows. Section 2 describes the basic constraints from observing a single plane. Section 3 describes the calibration procedure. We start with a closed-form solution, followed by nonlinear optimization. Radial lens distortion is also modeled. Section 4 studies configurations in which the proposed calibration technique fails. It is very easy to avoid such situations in practice. Section 5 provides the experimental results. Both computer simulation and real data are used to validate the proposed technique. In the Appendix, we provides a number of details, including the techniques for estimating the homography between the model plane and its image.

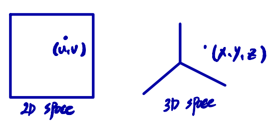
2 Basic Equations

단일 Plane을 관찰함으로서 제공되는 Camera의 Intrinsic Parameter에 대한 Constraint를 다룸.

We examine the constraints on the camera's intrinsic parameters provided by observing a single plane. We start with the notation used in this paper.

2.1 Notation

· 2D Point를 m이라고 표기 ⇒ $[u, v]^T$
· 3D Point를 M이라고 표기 ⇒ $[X, Y, Z]^T$



· 마지막 Element에 1을 추가한 Augmented Vector를 ~로 표기
↳ $\tilde{m}$: $[u, v, 1]^T$
$\tilde{M}$: $[X, Y, Z, 1]^T$

A 2D point is denoted by $\mathbf{m} = [u, v]^T$. A 3D point is denoted by $\mathtt{M} = [X, Y, Z]^T$. We use $\tilde{\mathbf{x}}$ to denote the augmented vector by adding 1 as the last element: $\tilde{\mathbf{m}} = [u, v, 1]^T$ and $\tilde{\mathtt{M}} = [X, Y, Z, 1]^T$. A camera is modeled by the usual pinhole: the relationship between a 3D point $\mathtt{M}$ and its image projection $\mathbf{m}$ is given by

Camera는 일반적인 Pinhole에 의해 Modeling.

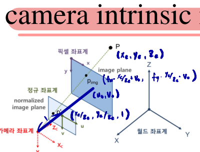
3D Point M과 이를 Image에 Projection한 m의 관계
: $S\tilde{m} = A[R\ t]\tilde{M}$

$$s\tilde{\mathbf{m}} = \mathbf{A}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\tilde{\mathtt{M}}, \tag{1}$$

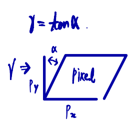
where s is an arbitrary scale factor, $(\mathbf{R}, \mathbf{t})$, called the extrinsic parameters, is the rotation and translation which relates the world coordinate system to the camera coordinate system, and $\mathbf{A}$, called the camera intrinsic matrix, is given by

$$\boxed{\mathbf{A} = \begin{bmatrix}\alpha & \gamma & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}}$$

· Focal length
⇒ lens의 중심과 Image sensor의 거리 (단위는 pixel)
↳ Focal length가 Image sensor의 Cell 크기에 대한 상대적인 값으로 표현



$\gamma = \tan K$.



· S: 임의의 Scale 계수
· R, t: World Coordinate와 Camera Coordinate 간의 Rotation과 Translation을 나타내는 Extrinsic Parameter.
· A: Camera Intrinsic Matrix

f_x f_y

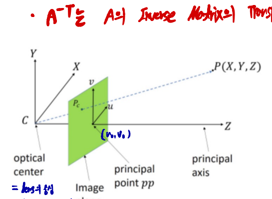
with (u_0, v_0) the coordinates of the principal point, α and β the scale factors in image u and v axes, and γ the parameter describing the skewness of the two image axes.

We use the abbreviation $\mathbf{A}^{-T}$ for $(\mathbf{A}^{-1})^T$ or $(\mathbf{A}^T)^{-1}$.

· Normalized Image Plane
⇒ Focal Point로부터 거리가 1인 Plane.
Camera Coordinate System의 한 점 (X_c, Y_c, Z_c)을 Image Coordinate System로 변환 시 X_c, Y_c를 Z_c로 나눔 ⇒ Normalized Image plane 상의 Coordinate로 변환.
↳ Focal length를 곱하고 u_0, v_0를 더하면 Image Plane 좌표가 된다.

· A^{-T}는 A의 Inverse Matrix의 Transpose / A의 Transpose Matrix에 대한 Inverse Matrix

· (u_0, v_0): Principal Point에 대한 좌표
· (α, β): Image의 u·v 축에 대한 Scale 계수
· γ: 두개의 Image Axes에 대한 Skewness



Homography : 두 plane 사이의 변환을 나타내는 수학적 개념

2.2 Homography between the model plane and its image

→ Model plane과 이것의 Image 간의 Homography

↳ Real world plane ↳ Camera 촬영한 Image plane

Without loss of generality, we assume the model plane is on $Z = 0$ of the world coordinate system. Let's denote the i^{th} column of the rotation matrix $\mathbf{R}$ by $\mathbf{r}_i$. From (1), we have

Generality를 잃지 않기 위해 Model plane을 World Coordinate system의 Z=0에 위치시킨다.

R이란 Rotation Matrix의 i번째 Column을 r_i로 표현한다.

Intrinsic Camera Matrix

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}\end{bmatrix}\begin{bmatrix}X\\Y\\0\\1\end{bmatrix}$$

임의의 scale factor ↳ 2D point (R,t) 3D point

Model plane의 점을 3D로 확장하기 위해 표현한다.

$$= \mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}\begin{bmatrix}X\\Y\\1\end{bmatrix}.$$

Z=0이기에 r_3에 대한 값을 없앤다.

→ 3차원 공간의 xy plane 위 Coordinate

→ Model plane 상 위의 Point를 나타내기 위해 M을 표기 → Z는 항상 '0'이므로 무시, $M=[X,Y]^T$로 나타낸다.

By abuse of notation, we still use $\mathtt{M}$ to denote a point on the model plane, but $\mathtt{M} = [X, Y]^T$ since Z is always equal to 0. In turn, $\widetilde{\mathtt{M}} = [X, Y, 1]^T$. Therefore, a model point $\mathtt{M}$ and its image $\mathbf{m}$ is related by a homography $\mathbf{H}$:

↳ $\tilde{M} = [x, y, 1]^T$로 표현 가능하다.

$s\tilde{m} = A[r_1, r_2, t]\tilde{M}$ 즉, Model Point M과 Image m에 대해 Homography를 나타낼 수 있다.

$$s\widetilde{\mathbf{m}} = \mathbf{H}\widetilde{\mathtt{M}} \quad \text{with} \quad \mathbf{H} = \mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}. \tag{2}$$

As is clear, the 3×3 matrix $\mathbf{H}$ is defined up to a scale factor.

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\alpha & \gamma & u_0\\0 & \beta & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & t_1\\r_{21} & r_{22} & t_2\\r_{31} & r_{32} & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\1\end{bmatrix}$$

Image Point Camera Intrinsic Matrix Z=0이기에 (R,t) model point

2.3 Constraints on the intrinsic parameters

Chessboard plane

Given an image of the model plane, an homography can be estimated (see Appendix A). Let's denote it by $\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix}$. From (2), we have

↳ H의 right singular vector

↳ Appendix A로 H를 구함. h_1과 h_2를 알고 있는 상태

$$\begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix} = \lambda\mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix},$$

↳ 임의의 scalar

→ $A^{-1}h_1 = r_1$, $A^{-1}h_2 = r_2$

where λ is an arbitrary scalar. Using the knowledge that $\mathbf{r}_1$ and $\mathbf{r}_2$ are orthonormal, we have

↳ X·Y에 대한 Rotation으로 서로 수직한 축이며 orthonormal하다. 적용

Intrinsic Parameter가 만족해야 하는 수식.

$$\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2 = 0 \tag{3}$$

$$\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_1 = \mathbf{h}_2^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2. \tag{4}$$

↳ $r_1^T r_2 = 0$ 과 $\|r_1\| = \|r_2\| = 1$

↳ (3)번 수식 ↳ (4)번 수식 → $r_1^T r_1 = \|r_1\| = 1$, $r_2^T r_2 = \|r_2\| = 1$

These are the two basic constraints on the intrinsic parameters, given one homography. Because a homography has 8 degrees of freedom and there are 6 extrinsic parameters (3 for rotation and 3 for translation), we can only obtain 2 constraints on the intrinsic parameters. Note that $\mathbf{A}^{-T}\mathbf{A}^{-1}$ actually describes the image of the absolute conic [16]. In the next subsection, we will give an geometric interpretation.

H는 8자유도, E·P는 6자유도 ⇒ H는 E·P와 I·P의 Matrix 곱.

$H = \begin{bmatrix}r_{11} & r_{12} & r_{13}\\r_{21} & r_{22} & r_{23}\\r_{31} & r_{32} & r_{33}\end{bmatrix}$ EP = 3Rotation 3Translation I·P에 대한 2개의 Constraint.

Euclidean Transformation에서 Conic을 거쳐서 보이는 Conic으로 보임을 의미한다. ↳ Translation · Rotation 모두 Invariant하다.

기하학적으로 설명하는 부분 : 100% 이해 X

2.4 Geometric Interpretation

We are now relating (3) and (4) to the absolute conic.

It is not difficult to verify that the model plane, under our convention, is described in the camera coordinate system by the following equation:

model plane을 Camera Coordinate system으로 나타낼 수 있다.

⇒ 카메라가 보는 Real world Plane을 Camera Coordinate System으로 나타낼 수 있다.

$r_3 = \begin{bmatrix}r_{13}\\r_{23}\\r_{33}\end{bmatrix}$ $r_3^T = [r_{13}\ r_{23}\ r_{33}]$ $t = \begin{bmatrix}t_1\\t_2\\t_3\end{bmatrix}$

$$\underbrace{\begin{bmatrix}\mathbf{r}_3\\\mathbf{r}_3^T\mathbf{t}\end{bmatrix}^T}_{1\times4}\underbrace{\begin{bmatrix}x\\y\\z\\w\end{bmatrix}}_{4\times1} = 0,$$

w가 '0'이면 ∞ point를 나타낸다.

where $w = 0$ for points at infinity and $w = 1$ othewise. This plane intersects the plane at infinity at a line, and we can easily see that $\begin{bmatrix}\mathbf{r}_1\\0\end{bmatrix}$ and $\begin{bmatrix}\mathbf{r}_2\\0\end{bmatrix}$ are two particular points on that line. Any point on it

Model Plane은 하나의 line에서 Plane at Infinity와 교차. → 직선 상의 2개의 Point.

is a linear combination of these two points, i.e., 해당 line 상의 어떤 점이든 linear Combination으로 얻을 수 있다.

$$\mathbf{x}_\infty = a\begin{bmatrix}\mathbf{r}_1\\0\end{bmatrix} + b\begin{bmatrix}\mathbf{r}_2\\0\end{bmatrix} = \begin{bmatrix}a\mathbf{r}_1 + b\mathbf{r}_2\\0\end{bmatrix}.$$

해당 line과 absolute conic이 교차하는 점을 찾는다.

Now, let's compute the intersection of the above line with the absolute conic. By definition, the point $\mathbf{x}_\infty$, known as the *circular point*, satisfies: $\mathbf{x}_\infty^T\mathbf{x}_\infty = 0$, i.e.,

circular point 특성.

absolute Conic은 Extrinsic Parameter와 Internal Parameter 사이의 관계를 나타낸다.

X_∞는 circular point.

$$(a\mathbf{r}_1 + b\mathbf{r}_2)^T(a\mathbf{r}_1 + b\mathbf{r}_2) = 0, \quad \text{or} \quad a^2 + b^2 = 0.$$

Absolute Conic과 교차하는 점

Absolute Conic = $A^T \cdot A$ (여기서 A는 Intrinsic parameter matrix)

The solution is $b = \pm ai$, where $i^2 = -1$. That is, the two intersection points are

복소수의 형태

$$\mathbf{x}_\infty = a\begin{bmatrix}\mathbf{r}_1 \pm i\mathbf{r}_2\\0\end{bmatrix}.$$

Their projection in the image plane is then given, up to a scale factor, by

Intersection Point를 Image Plane으로 Projection

$$\widetilde{\mathbf{m}}_\infty = \mathbf{A}(\mathbf{r}_1 \pm i\mathbf{r}_2) = \mathbf{h}_1 \pm i\mathbf{h}_2.$$

Intrinsic Parameter.

Point $\widetilde{\mathbf{m}}_\infty$ is on the image of the absolute conic, described by $\mathbf{A}^{-T}\mathbf{A}^{-1}$ [16]. This gives

$\widetilde{m}_\infty$는 Absolute Conic의 Image 위에 있다.

$$(\mathbf{h}_1 \pm i\mathbf{h}_2)^T\mathbf{A}^{-T}\mathbf{A}^{-1}(\mathbf{h}_1 \pm i\mathbf{h}_2) = 0.$$

Requiring that both real and imaginary parts be zero yields (3) and (4).

3 Solving Camera Calibration

This section provides the details how to effectively solve the camera calibration problem. We start with an analytical solution, followed by a nonlinear optimization technique based on the maximum likelihood criterion. Finally, we take into account lens distortion, giving both analytical and nonlinear solutions.

Analytical Solution은 x= ~로 나타낼수 있는 Solution을 의미한다.

이런 형태의 수식을 closed-form이라 한다.

3.1 Closed-form solution

Let

$$A = \begin{bmatrix}\alpha & \gamma & u_0\\0 & \beta & v_0\\0 & 0 & 1\end{bmatrix} \quad A^T = \begin{bmatrix}\alpha & 0 & 0\\\gamma & \beta & 0\\u_0 & v_0 & 1\end{bmatrix}$$

$$\mathbf{B} = \mathbf{A}^{-T}\mathbf{A}^{-1} \equiv \begin{bmatrix}B_{11} & B_{12} & B_{13}\\B_{12} & B_{22} & B_{23}\\B_{13} & B_{23} & B_{33}\end{bmatrix}$$

라고 한다.

$$A^{-1} = \begin{bmatrix}\frac{1}{\alpha} & \frac{-\gamma}{\alpha\beta} & -\frac{\beta u_0 + \gamma v_0}{\alpha\beta}\\0 & 1/\beta & -v_0/\beta\\0 & 0 & 1\end{bmatrix} \quad A^{-T} = \begin{bmatrix}1/\alpha & 0 & 0\\\frac{-\gamma}{\alpha\beta} & 1/\beta & 0\\-\frac{\beta u_0 + \gamma v_0}{\alpha\beta} & -v_0/\beta & 1\end{bmatrix}$$

$$= \begin{bmatrix}\frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta}\\-\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2}\\\frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1\end{bmatrix}. \tag{5}$$

Note that $\mathbf{B}$ is symmetric, defined by a 6D vector Symmetric Matrix

중복제거

h_{ij} 표기법??

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T. \tag{6}$$

Let the i^{th} column vector of $\mathbf{H}$ be $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$. Then, we have

$$H = \begin{bmatrix}h_{11} & h_{12} & h_{13}\\h_{21} & h_{22} & h_{23}\\h_{31} & h_{32} & h_{33}\end{bmatrix}$$

H의 i번째 Column을 h_i로 표기

$h_i = [h_{i1}\ h_{i2}\ h_{i3}]^T$로 정의

$$\mathbf{h}_i^T\mathbf{B}\mathbf{h}_j = \mathbf{v}_{ij}^T\mathbf{b} \tag{7}$$

$$\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T.$$

증명

$$\underbrace{\begin{bmatrix}h_{i1} & h_{i2} & h_{i3}\end{bmatrix}}_{h_i^T,\ 1\times3}\underbrace{\begin{bmatrix}B_{11} & B_{12} & B_{13}\\B_{21} & B_{22} & B_{23}\\B_{31} & B_{32} & B_{33}\end{bmatrix}}_{B,\ 3\times3}\underbrace{\begin{bmatrix}h_{j1}\\h_{j2}\\h_{j3}\end{bmatrix}}_{h_j,\ 3\times1} = \underbrace{\begin{bmatrix}h_{i1}B_{11} + h_{i2}B_{21} + h_{i3}B_{31} & h_{i1}B_{12} + h_{i2}B_{22} + h_{i3}B_{32} & h_{i1}B_{13} + h_{i2}B_{23} + h_{i3}B_{33}\end{bmatrix}}_{1\times3}\underbrace{\begin{bmatrix}h_{j1}\\h_{j2}\\h_{j3}\end{bmatrix}}_{3\times1}$$

$$= h_{j1}h_{i1}B_{11} + h_{j1}h_{i2}\underbrace{B_{21}}_{B_{12}} + h_{j1}h_{i3}\underbrace{B_{31}}_{B_{13}} + h_{j2}h_{i1}B_{12} + h_{j2}h_{i2}B_{22} + h_{j2}h_{i3}\underbrace{B_{32}}_{B_{23}} + h_{j3}h_{i1}B_{13} + h_{j3}h_{i2}B_{23} + h_{j3}h_{i3}B_{33}$$

$$= h_{j1}h_{i1}B_{11} + (h_{j1}h_{i2} + h_{j2}h_{i1})B_{12} + h_{j2}h_{i2}B_{22} + (h_{j3}h_{i1} + h_{j1}h_{i3})B_{13} + (h_{j2}h_{i3} + h_{j3}h_{i2})B_{23} + h_{j3}h_{i3}B_{33}$$

with

$$\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T .$$

Therefore, the two fundamental constraints (3) and (4), from a given homography, can be rewritten as 2 homogeneous equations in $\mathbf{b}$:

$H = A[r_1\ r_2\ \cancel{t}] = [h_1\ h_2\ h_3]$

$h_1 = Ar_1$

$n = 2$??

$$\mathbf{B} = \mathbf{A}^{-T}\mathbf{A}^{-1} \ \& \ \mathbf{h}_i^T\mathbf{B}\mathbf{h}_j = \mathbf{v}_{ij}^T\mathbf{b}$$

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} . \tag{8}$$

즉, $v_{ij}^T b = h_i^T A^{-T}A^{-1} h_j$ 가 되며, $\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2 = 0$, $\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_1 = \mathbf{h}_2^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2$. 를 의미한다.

두개의 Constraint로 얻을 수 있음.

If n images of the model plane are observed, by stacking n such equations as (8) we have

Model Plane에 대한 n개의 Image $\xrightarrow{\text{"n개의 식"}}$

$$\mathbf{V}\mathbf{b} = \mathbf{0} , \tag{9}$$

n이 3보다 크면, unique Solution을 얻을 수 있다. ⟶ Scale factor는 제외.

$\gamma = 0$이면,

$$\begin{bmatrix} \frac{1}{\alpha^2} & \cancel{-\frac{\gamma}{\alpha^2\beta}} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ \cancel{-\frac{\gamma}{\alpha^2\beta}} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1/\alpha^2 & 0 & -u_0\beta/\alpha^2\beta \\ 0 & 1/\beta^2 & -v_0/\beta^2 \\ -u_0\beta/\alpha^2\beta & -v_0/\beta^2 & (\cdots)^2/\alpha^2\beta^2 + v_0^2/\beta^2 + 1 \end{bmatrix}$$

where $\mathbf{V}$ is a $2n \times 6$ matrix. If $n \geq 3$, we will have in general a unique solution $\mathbf{b}$ defined up to a scale factor. If $n = 2$, we can impose the skewless constraint $\gamma = 0$, i.e., $[0, 1, 0, 0, 0, 0]\mathbf{b} = 0$, which is added as an additional equation to (9). (If $n = 1$, we can only solve two camera intrinsic parameters, e.g., α and β, assuming u_0 and v_0 are known (e.g., at the image center) and $\gamma = 0$, and that is indeed what we did in [19] for head pose determination based on the fact that eyes and mouth are reasonably coplanar.) The solution to (9) is well known as the eigenvector of $\mathbf{V}^T\mathbf{V}$ associated with the smallest eigenvalue (equivalently, the right singular vector of $\mathbf{V}$ associated with the smallest singular value).

n이 2라면, Skewless Constraint → $\gamma = 0$임.

n이 1이라면, u_0와 v_0를 안다, $\gamma = 0$이라 가정하여 α와 β를 구할 수 있다.

Vb=0 을 SVD를 이용하여 Solution을 구한다.

Once $\mathbf{b}$ is estimated, we can compute all camera intrinsic matrix $\mathbf{A}$. See Appendix B for the details.

b를 구하면, Camera Intrinsic Matrix를 구할 수 있다.

Intrinsic Parameter

$$\begin{aligned} v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \\ \lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \\ \alpha &= \sqrt{\lambda/B_{11}} \\ \beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\ \gamma &= -B_{12}\alpha^2\beta/\lambda \\ u_0 &= \gamma v_0/\beta - B_{13}\alpha^2/\lambda . \end{aligned}$$

Once $\mathbf{A}$ is known, the extrinsic parameters for each image is readily computed. From (2), we have

A를 구하면 각 Image에 대한 Extrinsic Parameter를 구할 수 있다.

$$\begin{aligned} \mathbf{r}_1 &= \lambda\mathbf{A}^{-1}\mathbf{h}_1 \\ \mathbf{r}_2 &= \lambda\mathbf{A}^{-1}\mathbf{h}_2 \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \lambda\mathbf{A}^{-1}\mathbf{h}_3 \end{aligned} \quad \longleftarrow \quad [\mathbf{h}_1 \ \ \mathbf{h}_2 \ \ \mathbf{h}_3] = \lambda\mathbf{A}[\mathbf{r}_1 \ \ \mathbf{r}_2 \ \ \mathbf{t}]$$

"수식유도 가능"

$\lambda = ??$

with $\lambda = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\| = 1/\|\mathbf{A}^{-1}\mathbf{h}_2\|$. Of course, because of noise in data, the so-computed matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ does not in general satisfy the properties of a rotation matrix. Appendix C describes a method to estimate the best rotation matrix from a general 3×3 matrix.

Data의 Noise로 인해 계산된 $R = [r_1\ r_2\ r_3]$는 Rotation Matrix의 특성을 만족하지 못한다.

3.2 Maximum likelihood estimation

The above solution is obtained through minimizing an algebraic distance which is not physically meaningful. We can refine it through maximum likelihood inference. // MLE를 통해 Refine하여 최종적인 A·R·t를 얻는다.

Model Plane에 대한 n장 Image

Model Plane에 대한 m개 Point

We are given n images of a model plane and there are m points on the model plane. Assume that the image points are corrupted by independent and identically distributed noise. The maximum likelihood estimate can be obtained by minimizing the following functional:

Independent: $f(x,y,z) = f(x)f(y)f(z)$: 각각의 사건이 서로 영향 X

Identical: 주사위 · 동전 던지기 처럼 동일한 확률분포를 가짐.

Image Point는 Independent · Identical Distribution Noise를 가진다. ⇒ i.i.d 임.

Random Variable이 독립 · 같은 Distribution을 가짐.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathrm{M}_j)\|^2 , \tag{10}$$

실제 영상에서 Point 위치

i Image에 M_j Point Projection.

M_j, A, R_i, t_i 기반 Point 추정 ⇒ M_j를 Image에 Projection 한 것

이를 최소화함으로써 MLE 진행 가능.

세 크기가 같지??

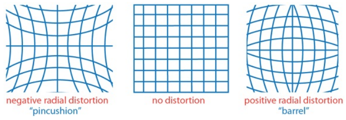
where $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathrm{M}_j)$ is the projection of point M_j in image i, according to equation (2). A rotation $\mathbf{R}$ is parameterized by a vector of 3 parameters, denoted by $\mathbf{r}$, which is parallel to the rotation axis and whose magnitude is equal to the rotation angle. $\mathbf{R}$ and $\mathbf{r}$ are related by the Rodrigues formula [5]. Minimizing (10) is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [18]. It requires an initial guess of $\mathbf{A}$, $\{\mathbf{R}_i, \mathbf{t}_i | i = 1..n\}$ which can be obtained using the technique described in the previous subsection.

A에 대한 초기값 설정 필요하다.

$R = [r_1\ r_2\ r_3]$

Rotation Axis를 나타냄. / 이들의 크기는 Rotation Angle과 같다.

· Radial Distortion ⇒



수식

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

· 방사왜곡은 볼록 Lens의 굴절률에 의해 생기는것으로 영상의 왜곡 정도가 중심에서의 거리에 의해 결정된다.

Lens의 Distortion은 Intrinsic parameter의 영향이 제거된 Normalized Image Plane에서 정의된다.

3차원의 Point → 2차원의 Normalized Image Plane 상의 한 점으로 Projection. → $\begin{bmatrix} u_n \\ v_n \end{bmatrix} = \begin{bmatrix} X_c / Z_c \\ Y_c / Z_c \end{bmatrix}$ (No Distortion)

Principle Point 기준 (No Distortion)

$$\begin{bmatrix} x_{n\_d} \\ y_{n\_d} \end{bmatrix} = \underbrace{(1 + k_1 r_u^2 + k_2 r_u^4 + k_3 r_u^6)}_{\text{Radial Distortion}} \begin{bmatrix} x_{n\_u} \\ y_{n\_u} \end{bmatrix} + \underbrace{\begin{bmatrix} 2p_1 x_{n\_u} y_{n\_u} + p_2(r_u^2 + 2x_{n\_u}^2) \\ p_1(r_u^2 + 2y_{n\_u}^2) + 2p_2 x_{n\_u} y_{n\_u} \end{bmatrix}}_{\text{tangential Distortion}}$$

3.3 Dealing with radial distortion

Lens Distortion을 고려하지 않았다. Desktop Camera는 Distortion이 심함 → Radial Distortion

Up to now, we have not considered lens distortion of a camera. However, a desktop camera usually exhibits significant lens distortion, especially radial distortion. In this section, we only consider the first two terms of radial distortion. The reader is referred to [20, 2, 4, 26] for more elaborated models. Based on the reports in the literature [2, 23, 25], it is likely that the distortion function is totally dominated by the radial components, and especially dominated by the first term. It has also been found that any more elaborated modeling not only would not help (negligible when compared with sensor quantization), but also would cause numerical instability [23, 25]. ⟶ 더 많이 해도 의미 없다.

본 논문에서는 k_2까지 고려한다.

다른 논문들에서 Radial Distortion이 지배적이고, 첫 term이 가장 큰 영향을 미친다

· Normalized Image Coordinate를 내부 Parameter의 영향 없이 실제 영상 Pixel 좌표로 계산한다.

"Pixel Image Coordinate"
u와 v는 이상적인 Point
$\breve{u}$와 $\breve{v}$는 왜곡이 있는 Point
"Normalized Image Coordinate"
x와 y는 이상적인 Coordinate
$\breve{x}$와 $\breve{y}$는 Distortion Normalized Image Coordinate.

Let (u, v) be the ideal (nonobservable distortion-free) pixel image coordinates, and $(\breve{u}, \breve{v})$ the corresponding real observed image coordinates. The ideal points are the projection of the model points according to the pinhole model. Similarly, (x, y) and $(\breve{x}, \breve{y})$ are the ideal (distortion-free) and real (distorted) normalized image coordinates. We have [2, 25]

Ideal Coordinate를 가지고 Real Coordinate를 구하는 식

$$\boxed{\begin{aligned} \breve{x} &= x + x[k_1(x^2 + \overset{r^2}{y^2}) + k_2(x^2 + y^2)^2] \\ \breve{y} &= y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \,, \end{aligned}}$$

방사왜곡의 중심은 Principle Point와 같다.

where k_1 and k_2 are the coefficients of the radial distortion. The center of the radial distortion is the same as the principal point. From* $\breve{u} = u_0 + \alpha\breve{x} + \gamma\breve{y}$ and $\breve{v} = v_0 + \beta\breve{y}$ and assuming $\gamma = 0$, we have

$$\boxed{\begin{aligned} \breve{u} &= u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] && (11) \\ \breve{v} &= v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \,. && (12) \end{aligned}}$$

$\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 1 & 1 \end{bmatrix}$

Estimating Radial Distortion by Alternation. As the radial distortion is expected to be small, one would expect to estimate the other five intrinsic parameters, using the technique described in Sect. 3.2, reasonable well by simply ignoring distortion. One strategy is then to estimate k_1 and k_2 after having estimated the other parameters, which will give us the ideal pixel coordinates (u, v). Then, from (11) and (12), we have two equations for each point in each image:

$$\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \breve{u}-u \\ \breve{v}-v \end{bmatrix} .$$

이상적인 Coordinate (u,v)를 가지고 나머지 Parameter를 estimation하여 이를 k_1, k_2 estimation 하는데 사용할 수 있다.

Given m points in n images, we can stack all equations together to obtain in total $2mn$ equations, or in matrix form as $\mathbf{Dk} = \mathbf{d}$, where $\mathbf{k} = [k_1, k_2]^T$. The linear least-squares solution is given by

$$\mathbf{k} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{d} \,. \qquad (13)$$

Once k_1 and k_2 are estimated, one can refine the estimate of the other parameters by solving (10) with $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)$ replaced by (11) and (12). We can alternate these two procedures until convergence.

Complete Maximum Likelihood Estimation. Experimentally, we found the convergence of the above alternation technique is slow. A natural extension to (10) is then to estimate the complete set of parameters by minimizing the following functional:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|\mathbf{m}_{ij} - \breve{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)\|^2 \,, \qquad (14)$$

*A typo was reported by Johannes Koester [johannes.koester@uni-dortmund.de] via email on Aug. 13, 2008.

where $\breve{\mathbf{m}}(\mathbf{A}, k1, k2, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)$ is the projection of point $\mathtt{M}_j$ in image i according to equation (2), followed by distortion according to (11) and (12). This is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [18]. A rotation is again parameterized by a 3-vector $\mathbf{r}$, as in Sect. 3.2. An initial guess of $\mathbf{A}$ and $\{\mathbf{R}_i, \mathbf{t}_i | i = 1..n\}$ can be obtained using the technique described in Sect. 3.1 or in Sect. 3.2. An initial guess of k_1 and k_2 can be obtained with the technique described in the last paragraph, or simply by setting them to 0.

3.4 Summary

The recommended calibration procedure is as follows:

1. Print a pattern and attach it to a planar surface;
2. Take a few images of the model plane under different orientations by moving either the plane or the camera;
3. Detect the feature points in the images;
4. Estimate the five intrinsic parameters and all the extrinsic parameters using the closed-form solution as described in Sect. 3.1;
5. Estimate the coefficients of the radial distortion by solving the linear least-squares (13);
6. Refine all parameters by minimizing (14).

4 Degenerate Configurations

We study in this section configurations in which additional images do not provide more constraints on the camera intrinsic parameters. Because (3) and (4) are derived from the properties of the rotation matrix, if $\mathbf{R}_2$ is not independent of $\mathbf{R}_1$, then image 2 does not provide additional constraints. In particular, if a plane undergoes a pure translation, then $\mathbf{R}_2 = \mathbf{R}_1$ and image 2 is not helpful for camera calibration. In the following, we consider a more complex configuration.

Proposition 1. *If the model plane at the second position is parallel to its first position, then the second homography does not provide additional constraints.*

Proof. Under our convention, $\mathbf{R}_2$ and $\mathbf{R}_1$ are related by a rotation around z-axis. That is,

$$\mathbf{R}_1 \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{R}_2 ,$$

where θ is the angle of the relative rotation. We will use superscript $^{(1)}$ and $^{(2)}$ to denote vectors related to image 1 and 2, respectively. It is clear that we have

$$\begin{aligned} \mathbf{h}_1^{(2)} &= \lambda^{(2)}(\mathbf{A}\mathbf{r}^{(1)}\cos\theta + \mathbf{A}\mathbf{r}^{(2)}\sin\theta) = \frac{\lambda^{(2)}}{\lambda^{(1)}}(\mathbf{h}_1^{(1)}\cos\theta + \mathbf{h}_2^{(1)}\sin\theta) \\ \mathbf{h}_2^{(2)} &= \lambda^{(2)}(-\mathbf{A}\mathbf{r}^{(1)}\sin\theta + \mathbf{A}\mathbf{r}^{(2)}\cos\theta) = \frac{\lambda^{(2)}}{\lambda^{(1)}}(-\mathbf{h}_1^{(1)}\sin\theta + \mathbf{h}_2^{(1)}\cos\theta) . \end{aligned}$$

Then, the first constraint (3) from image 2 becomes:

$$\begin{aligned} {\mathbf{h}_1^{(2)}}^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2^{(2)} = &\frac{\lambda^{(2)}}{\lambda^{(1)}}[(\cos^2\theta - \sin^2\theta)({\mathbf{h}_1^{(1)}}^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2^{(1)}) \\ &- \cos\theta\sin\theta({\mathbf{h}_1^{(1)}}^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_1^{(1)} - {\mathbf{h}_2^{(1)}}^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2^{(1)})] , \end{aligned}$$

6 Conclusion

In this paper, we have developed a flexible new technique to easily calibrate a camera. The technique only requires the camera to observe a planar pattern from a few (at least two) different orientations. We can move either the camera or the planar pattern. The motion does not need to be known. Radial lens distortion is modeled. The proposed procedure consists of a closed-form solution, followed by a nonlinear refinement based on maximum likelihood criterion. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques which use expensive equipment such as two or three orthogonal planes, the proposed technique gains considerable flexibility.

Acknowledgment

Thanks go to Brian Guenter for his software of corner extraction and for many discussions, and to Bill Triggs for insightful comments. Thanks go to Andrew Zisserman for bringing his CVPR98 work [14] to my attention, which uses the same constraint but in different form, and for pointing out an error in my discussion on the case of pure translation. Thanks go to Bill Triggs and Gideon Stein for suggesting experiments described in Sect. 5.3. Thanks also go to the members of the Vision Group at MSR for encouragement and discussions. Anandan and Charles Loop have checked the English.

3차원 내 checkerboard plane과 이것에 대한 Image 사이 H 추정

A Estimation of the Homography Between the Model Plane and its Image

$$s\widetilde{\mathbf{m}} = \mathbf{H}\widetilde{\mathbf{M}} \quad \text{with} \quad \mathbf{H} = \mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix} . \qquad (2)$$

→ 이상적인 경우, 이를 만족해야함.
but Noise로 인해 실제로는 만족 X.

There are many ways to estimate the homography between the model plane and its image. Here, we present a technique based on maximum likelihood criterion. Let $\mathtt{M}_i$ and $\mathbf{m}_i$ be the model and image points, respectively. Ideally, they should satisfy (2). In practice, they don't because of noise in the extracted image points. Let's assume that $\mathbf{m}_i$ is corrupted by Gaussian noise with mean $\mathbf{0}$ and covariance matrix $\mathbf{\Lambda}_{\mathbf{m}_i}$. Then, the maximum likelihood estimation of $\mathbf{H}$ is obtained by minimizing the following functional

↳ m_i가 평균이 '0'이고 공분산 행렬이 Λ_{m_i}인 Gaussian Noise에 영향을 받는다고 가정한다.

→ Multivariate Gaussian Distribution으로 부터 얻을수 있다.

$$\sum_i (\mathbf{m}_i - \hat{\mathbf{m}}_i)^T \mathbf{\Lambda}_{\mathbf{m}_i}^{-1} (\mathbf{m}_i - \hat{\mathbf{m}}_i) ,$$

H에 대한 MLE는 해당 수식을 최소화하면서 얻을수 있다.

$s\begin{bmatrix}u_i\\v_i\\1\end{bmatrix} = \begin{bmatrix}\bar h_1\\ \bar h_2\\ \bar h_3\end{bmatrix}^T \begin{bmatrix}X\\Y\\1\end{bmatrix}$ ↳ $s = \bar h_3^T \cdot M_i$

$s\cdot u_i = \bar h_1^T \cdot M_i$
$s\cdot v_i = \bar h_2^T \cdot M_i$
↳ $m_i = \frac{1}{s}\begin{bmatrix}\bar h_1^T \cdot M_i\\ \bar h_2^T \cdot M_i\end{bmatrix}$

where

$$\hat{\mathbf{m}}_i = \frac{1}{\bar{\mathbf{h}}_3^T \mathtt{M}_i} \begin{bmatrix}\bar{\mathbf{h}}_1^T \mathtt{M}_i\\ \bar{\mathbf{h}}_2^T \mathtt{M}_i\end{bmatrix} \quad \text{with } \bar{\mathbf{h}}_i \text{, the } i^{\text{th}} \text{ row of } \mathbf{H}.$$

$\begin{bmatrix}\cancel{r_{11}\ r_{12}\ r_{13}}\\ r_{21}\ r_{22}\ r_{23}\\ r_{31}\ r_{32}\ r_{33}\end{bmatrix} \to \bar h_1 \Rightarrow \begin{bmatrix}r_{11}\\r_{12}\\r_{13}\end{bmatrix}$로 표현된다.

• M_i : 실제 Image point
• $\hat m_i$: Estimation 값
• M_i : 실제 Point
• $\bar h_i$: H의 i번째 행.

Point들이 독립 ⇒ $Cov(x_i, x_j) = 0$

In practice, we simply assume $\mathbf{\Lambda}_{\mathbf{m}_i} = \sigma^2\mathbf{I}$ for all i. This is reasonable if points are extracted independently with the same procedure. In this case, the above problem becomes a nonlinear least-squares one, i.e., $\min_{\mathbf{H}} \sum_i \|\mathbf{m}_i - \hat{\mathbf{m}}_i\|^2$. The nonlinear minimization is conducted with the Levenberg-Marquardt Algorithm as implemented in `Minpack` [18]. This requires an initial guess, which can be obtained as follows.

↳ LMA는 수식을 점차 반복적으로 갱신하여 Parameter를 얻기에 초기값에 대해 고려할 필요가 있다.

Let $\mathbf{x} = [\bar{\mathbf{h}}_1^T, \bar{\mathbf{h}}_2^T, \bar{\mathbf{h}}_3^T]^T$. Then equation (2) can be rewritten as

$s\tilde m = H\tilde M$
$s\tilde m - H\tilde M = 0$, $H = \begin{bmatrix}\bar h_1^T\\ \bar h_2^T\\ \bar h_3^T\end{bmatrix}$
$s\begin{bmatrix}u\\v\\1\end{bmatrix} - \begin{bmatrix}\bar h_1^T \tilde M\\ \bar h_2^T \tilde M\\ \bar h_3^T \tilde M\end{bmatrix} = 0 \Rightarrow s - \bar h_3^T \tilde M = 0 \Rightarrow s = \bar h_3^T \tilde M$

$\bar h_3^T \cdot \tilde M \cdot u - \bar h_1^T \cdot \tilde M = 0$
$\bar h_3^T \cdot \tilde M \cdot v - \bar h_2^T \cdot \tilde M = 0$
으로 정리하면.

$$\begin{bmatrix}\widetilde{\mathtt{M}}^T & \mathbf{0}^T & -u\widetilde{\mathtt{M}}^T\\ \mathbf{0}^T & \widetilde{\mathtt{M}}^T & -v\widetilde{\mathtt{M}}^T\end{bmatrix} \mathbf{x} = \mathbf{0} .$$

↳ L

When we are given n points, we have n above equations, which can be written in matrix equation as $\mathbf{L}\mathbf{x} = \mathbf{0}$, where $\mathbf{L}$ is a $2n \times 9$ matrix. As $\mathbf{x}$ is defined up to a scale factor, the solution is well known

↳ $\tilde M^T$: 1x3 , 0^T : 1x3
1개 Point 당 2개의 Row
n개 ⇒ 2n Row.

↳ 해는 SVD를 이용
L의 right singular vector이다.

· SVD
· Right singular Vector

to be the right singular vector of $\mathbf{L}$ associated with the smallest singular value (or equivalently, the eigenvector of $\mathbf{L}^T\mathbf{L}$ associated with the smallest eigenvalue). L의 몇몇은 1, 몇몇은 pixel, 몇몇은 world coordinate, 몇몇은 이들의 곱

In $\mathbf{L}$, some elements are constant 1, some are in pixels, some are in world coordinates, and some are multiplication of both. This makes $\mathbf{L}$ poorly conditioned numerically. Much better results can be obtained by performing a simple data normalization, such as the one proposed in [12], prior to running the above procedure.

절차 과정을 수행하기전에 Data Normalization 수행.

B Extraction of the Intrinsic Parameters from Matrix B

Matrix $\mathbf{B}$, as described in Sect. 3.1, is estimated up to a scale factor, i.e., , $\mathbf{B} = \lambda\mathbf{A}^{-T}\mathbf{A}$ with λ an arbitrary scale. Without difficulty[†], we can uniquely extract the intrinsic parameters from matrix $\mathbf{B}$.

$$
\begin{aligned}
v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2)\\
\lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11}\\
\alpha &= \sqrt{\lambda/B_{11}}\\
\beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)}\\
\gamma &= -B_{12}\alpha^2\beta/\lambda\\
u_0 &= \gamma v_0/\beta - B_{13}\alpha^2/\lambda\ .
\end{aligned}
$$

C Approximating a 3×3 matrix by a Rotation Matrix

3x3 Q Matrix를 근사하여 Best Rotation Matrix R을 찾는다.

The problem considered in this section is to solve the best rotation matrix $\mathbf{R}$ to approximate a given 3×3 matrix $\mathbf{Q}$. Here, "best" is in the sense of the smallest Frobenius norm of the difference $\mathbf{R} - \mathbf{Q}$. That is, we are solving the following problem:

R-Q의 차이에 대한 Frobenius Norm을 최소화하는 방식으로 수행한다.

$$
\min_{\mathbf{R}} \|\mathbf{R} - \mathbf{Q}\|_F^2 \quad \text{subject to } \mathbf{R}^T\mathbf{R} = \mathbf{I}\ . \tag{15}
$$

$\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$

F-Norm을 최소화　　Rotation Matrix 성질　　⇒ Matrix의 크기를 측정할때 사용한다.

Since

$$
\begin{aligned}
\|\mathbf{R} - \mathbf{Q}\|_F^2 &= \text{trace}((\mathbf{R} - \mathbf{Q})^T(\mathbf{R} - \mathbf{Q}))\\
&= 3 + \text{trace}(\mathbf{Q}^T\mathbf{Q}) - 2\text{trace}(\mathbf{R}^T\mathbf{Q})\ ,
\end{aligned}
$$

$$
\begin{aligned}
\| R-Q \|_F^2 &= trace((R-Q)^T(R-Q))\\
&= trace(R^TR - 2R^TQ + Q^TQ)\\
&= trace(R^TR) - 2trace(R^TQ) + trace(Q^TQ)
\end{aligned}
$$

min‖R- Q‖$_F^2$는 결국 trace(R^TQ)를 Maximize 하는것이 된다.

problem (15) is equivalent to the one of maximizing $\text{trace}(\mathbf{R}^T\mathbf{Q})$.

Let the singular value decomposition of $\mathbf{Q}$ be $\mathbf{USV}^T$, where $\mathbf{S} = \text{diag}\,(\sigma_1, \sigma_2, \sigma_3)$. If we define an orthogonal matrix $\mathbf{Z}$ by $\mathbf{Z} = \mathbf{V}^T\mathbf{R}^T\mathbf{U}$, then Q를 SVD ⇒ USV^T　　S는 대각행렬

직교행렬 Z를 정의하고 Z= V^TR^TU 라고 한다.

수식?

$$
\begin{aligned}
\text{trace}(\mathbf{R}^T\mathbf{Q}) &= \text{trace}(\mathbf{R}^T\underbrace{\mathbf{USV}}_{Q}{}^T) = \text{trace}(\underbrace{\mathbf{V}^T\mathbf{R}^T\mathbf{U}}_{Z}\mathbf{S})\\
&= \text{trace}(\mathbf{ZS}) = \sum_{i=1}^{3} z_{ii}\sigma_i \le \sum_{i=1}^{3} \sigma_i\ .
\end{aligned}
$$

← SVD에서 나온 수식

Z가 I일때 Maximum.

It is clear that the maximum is achieved by setting $\mathbf{R} = \mathbf{UV}^T$ because then $\mathbf{Z} = \mathbf{I}$. This gives the solution to (15).

An excellent reference on matrix computations is the one by Golub and van Loan [10].

[†] A typo was reported in formula u_0 by Jiyong Ma [mailto:jiyong@cslr.Colorado.EDU] via an email on April 18, 2002.

The most important entries in the fundamental matrix are precisely those that are subject to the largest relative perturbation when enforcing the singularity constraint without prior normalization.

This condition is corrected if normalization of the image coordinates is carried out first, for then all entries of the fundamental matrix will be treated approximately equally, and none is more important than another in computing epipolar lines.

6 Normalizing transformations

The previous sections concerned with the condition number of the matrix $A^\top A$ indicate that it is desirable to apply a transformation to the coordinates before carrying out the 8-point algorithm for finding the fundamental matrix.

6.1 Isotropic Scaling

As a first step, the coordinates in each image are translated (by a different translation for each image) so as to bring the centroid of the set of all points to the origin. The coordinates are also scaled. The previous sections suggested that the best results will be obtained if the coordinates are scaled, so that on the average a point $\mathbf{u}$ is of the form $\mathbf{u} = (1, 1, 1)^\top$. Such a point will lie a distance $\sqrt{2}$ from the origin. Rather than choose different scale factors for each direction, an isotropic scaling factor is chosen so that the u and v coordinates of a point are scaled equally. The transformation is as follows :

1. The points are translated so that their centroid is at the origin.
2. The points are then scaled isotropically so that the average distance from the origin is equal to $\sqrt{2}$.

Such a transformation is applied to each of the two images independently.

Appendix A에 언급된 [12]의 Normalization 방법.

6.2 Non-isotropic Scaling

As an alternative to the isotropic scaling method just described, an affine transformation was tried, in which the centroid of the points was placed at the origin and the two principal moments of the set of points were both made equal to unity. Thus, the set of points are transformed to an approximately symmetric circular cloud of points of radius one about the origin. The results obtained using this type of transformation to the data were little different from those obtained using the isotropic scaling method.

D Camera Calibration Under Known Pure Translation

As said in Sect. 4, if the model plane undergoes a pure translation, the technique proposed in this paper will not work. However, camera calibration is possible if the translation is known like the setup in Tsai's technique [23]. From (2), we have $\mathbf{t} = \alpha \mathbf{A}^{-1}\mathbf{h}_3$, where $\alpha = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\|$. The translation between two positions i and j is then given by

$$\mathbf{t}^{(ij)} = \mathbf{t}^{(i)} - \mathbf{t}^{(j)} = \mathbf{A}^{-1}(\alpha^{(i)}\mathbf{h}_3^{(i)} - \alpha^{(j)}\mathbf{h}_3^{(j)}) \ .$$

(Note that although both $\mathbf{H}^{(i)}$ and $\mathbf{H}^{(j)}$ are estimated up to their own scale factors, they can be rescaled up to a single common scale factor using the fact that it is a pure translation.) If only the translation direction is known, we get two constraints on $\mathbf{A}$. If we know additionally the translation magnitude, then we have another constraint on $\mathbf{A}$. Full calibration is then possible from two planes.

References

[1] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *Proceedings of the 6th International Conference on Computer Vision*, pages 790–796, Jan. 1998.

[2] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[3] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration. *The International Journal of Computer Vision*, 4(2):127–140, Mar. 1990.

[4] W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41(12):1479–1486, 1975.

[5] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

[6] O. Faugeras, T. Luong, and S. Maybank. Camera self-calibration: theory and experiments. In G. Sandini, editor, *Proc 2nd ECCV*, volume 588 of *Lecture Notes in Computer Science*, pages 321–334, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[7] O. Faugeras and G. Toscani. The calibration problem for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15–20, Miami Beach, FL, June 1986. IEEE.

[8] S. Ganapathy. Decomposition of transformation matrices for robot vision. *Pattern Recognition Letters*, 2:401–412, Dec. 1984.

[9] D. Gennery. Stereo-camera calibration. In *Proceedings of the 10th Image Understanding Workshop*, pages 101–108, 1979.

[10] G. Golub and C. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, 3 edition, 1996.

[11] R. Hartley. Self-calibration from multiple views with a rotating camera. In J.-O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision*, volume 800-801 of *Lecture Notes in Computer Science*, pages 471–478, Stockholm, Sweden, May 1994. Springer-Verlag.

[12] R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision*, pages 1064–1070, Boston, MA, June 1995. IEEE Computer Society Press.

[13] R. I. Hartley. An algorithm for self calibration from several views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 908–912, Seattle, WA, June 1994. IEEE.

[14] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, Santa Barbara, California, June 1998. IEEE Computer Society.

[15] Q.-T. Luong. *Matrice Fondamentale et Calibration Visuelle sur l'Environnement-Vers une plus grande autonomie des systèmes robotiques*. PhD thesis, Université de Paris-Sud, Centre d'Orsay, Dec. 1992.

[16] Q.-T. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *The International Journal of Computer Vision*, 22(3):261–289, 1997.

[17] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *The International Journal of Computer Vision*, 8(2):123–152, Aug. 1992.

[18] J. More. The levenberg-marquardt algorithm, implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, Lecture Notes in Mathematics 630. Springer-Verlag, 1977.

[19] I. Shimizu, Z. Zhang, S. Akamatsu, and K. Deguchi. Head pose determination from one image using a generic model. In *Proceedings of the IEEE Third International Conference on Automatic Face and Gesture Recognition*, pages 100–105, Nara, Japan, Apr. 1998.

[20] C. C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, fourth edition, 1980.

[21] G. Stein. Accurate internal camera calibration using rotation, with analysis of sources of error. In *Proc. Fifth International Conference on Computer Vision*, pages 230–236, Cambridge, Massachusetts, June 1995.

[22] B. Triggs. Autocalibration from planar scenes. In *Proceedings of the 5th European Conference on Computer Vision*, pages 89–105, Freiburg, Germany, June 1998.

[23] 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*, 3(4):323–344, Aug. 1987.

[24] G. Wei and S. Ma. A complete two-plane camera calibration method and experimental comparisons. In *Proc. Fourth International Conference on Computer Vision*, pages 439–446, Berlin, May 1993.

[25] G. Wei and S. Ma. Implicit and explicit camera calibration: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):469–480, 1994.

[26] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, Oct. 1992.

[27] Z. Zhang. Motion and structure from two perspective views: From essential parameters to euclidean motion via fundamental matrix. *Journal of the Optical Society of America A*, 14(11):2938–2950, 1997.