

Autocalibration of a Projector-Screen-Camera System: Theory and Algorithm for Screen-to-Camera Homography Estimation

Takayuki Okatani and Koichiro Deguchi
Graduate School of Information Sciences, Tohoku University
980-8579 Sendai, Japan

Abstract

This paper deals with the autocalibration of a system that consists of a planar screen, multiple projectors, and a camera. In the system, either multiple projectors or a single moving projector projects patterns on a screen while a stationary camera placed in front of the screen takes images of the patterns. We treat the case in which the patterns that the projectors project toward space are assumed to be known (i.e., the projectors are calibrated), whereas poses of the projectors are unknown. Under these conditions, we consider the problem of estimating screen-to-camera homography from the images alone. This is intended for cases where there is no clue on the screen surface that enables direct estimation of the screen-to-camera homography. One application is a 6DOF input device; poses of a multi-beam projector freely moving in space are computed from the images of beam spots on the screen. The primary contribution of the paper is theoretical results on the uniqueness of solutions and a noniterative algorithm for the problem. The effectiveness of the method is shown by experimental results on synthetic as well as on real images.

1. Introduction

This paper deals with problems of calibrating projector-screen-camera systems. A projector-screen-camera system is a system composed of three components: a planar screen placed in a scene, projectors projecting patterns on the screen, and a stationary camera in front of the screen that takes the images of the patterns (see Fig.1).

This type of system has been well studied to date. For presentation using a LCD projector, methods for correcting the keystone distortion that occurs when the projector is placed obliquely toward a screen have been proposed [6]. Also, methods for integrating multiple images that are projected by different projectors to produce one large seamless image have been studied ([1] among many others).

Although its purpose is slightly different from the above systems, there is a system of computing the 3D pose of a projector that projects multiple laser beams to space. The projector can arbitrarily move in space while projecting the

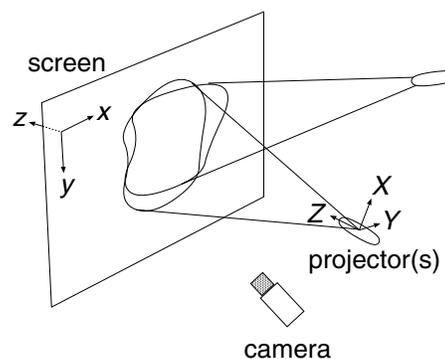


Figure 1. A projector-screen-camera system.

beams. This system can serve as a 6DOF input device and is expected to replace existing 6DOF input devices that are based on sensors measuring physical quantities, such as gyro sensors and magnetic sensors, in several cases where they are difficult to use. Also, when it is used for PC-based presentations, it can replace conventional laser pointers and serve, not just as a pointer, but as a 3D mouse assisting the presentation, which is expected to enrich presentation.

In this paper, for the projector-screen-camera systems described above, problems of estimating a geometrical relation between the screen and the camera are discussed. The relation can be represented by a homography that is given as a 3×3 matrix. We call this a screen-to-camera homography. Since in these systems we can observe only the images taken by the camera, it is first necessary to accurately estimate this screen-to-camera homography if the final goal is to obtain geometrical parameters between the screen and the projector. This applies to both the case of image correction/modification of LCD projectors and the case of pose estimation of the multi-beam projectors.

The screen-to-camera homography is easy to compute if there are markers or equivalent on the screen surface, e.g., points of known coordinates on the screen. In previous research on this type of system, the homography was determined in this or similar ways. In [6], it was proposed to use the fact that the screen used in presentation is usu-

ally square; the homography was computed by detecting the boundary of the screen.

In this paper, we develop a method for computing the screen-to-camera homography *without* such clues for the homography estimation on the screen surface. This enables us to deal with, for example, the following cases where the previous methods are difficult to use:

- an ordinary wall or floor is used as the screen whose surface does not have clues;
- a projector screen of known boundary shape is used but the boundary cannot be extracted because of difficult lighting conditions or small coverage of viewing angle of the camera lens.

In addition to these, the method has a potential application; the screen-to-camera homography obtained by the method might be useful in itself. For example, suppose multiple stationary cameras observing a floor to track human walking paths on the floor. Or suppose superimposing synthetic 2D patterns onto the image of some real planar surface. In order to compute accurate homographies between the target plane and the image planes of the cameras, a planar calibration pattern or equivalent is necessary, which could be difficult to prepare or find in several cases. Then the proposed method becomes useful. Although a multi-beam projector of known beam directions is necessary instead, it can be made compactly and used as a sort of “calibration apparatus” for these purposes.

It might seem impossible to determine the screen-to-camera homography without physical clues on the screen. However it can be done, as shown in what follows, if the projectors are calibrated and the patterns projected by the projectors toward space are known. Using knowledge of the patterns, the screen-to-camera homography can really be determined, up to freedom of choice of internal coordinate representation. In the case of pose estimation using multi-beam projectors, this becomes autocalibration. That is, it can be done without a prior calibration procedure.

The problem is very similar to the problems of autocalibration of cameras, which are to estimate (some of the) camera parameters from a given image sequence. With respect to problems of camera autocalibration, factorization of projective camera matrices plays a key role[3]. The counterpart in our problem is factorization of projector-to-camera homographies. Unfortunately, however, the same methodology as that used for camera autocalibration does not apply directly to our problem, since the forms of factorization are different. Thus, a new analysis is necessary.

In the rest of the paper we consider theories and methods for the above problem. Although they can potentially be used for various types of systems, for the purposes of explanation, we principally treat the system using multi-beam projectors. Section 2 describes the basic geometry of projector-screen-camera systems. In Section 3 it is shown how, and under what conditions, it is possible to determine

the screen-to-camera homography. In Section 4 an algorithm that is based on the results of Section 3 is presented. Experimental results are shown in Section 5.

2. Projector-screen-camera system

2.1. Relation between projectors and screen

The screen plane is assumed to be perfectly planar and to be on the xy plane of a coordinate frame $o-xyz$. We call this the screen coordinate frame. The projector has its own coordinate frame $O-XYZ$ called the projector coordinate frame, which is rigidly attached to the projector. The beams projected by the projector are assumed to emit from the origin of this coordinate frame (see Fig.1).

Now we derive a relation between each of the beams emitting from the projector and the point at which it arrives on the screen. Let $\mathbf{M} \equiv [X, Y, 1]^T$ be the orientation of a particular beam represented in the projector coordinates, and let $\mathbf{m} \equiv [u, v, 1]^T$ be homogeneous coordinates of the screen point at which the beam arrives, where $[u, v]$ are the x and y screen coordinates of that point. Then, the two 3-vectors \mathbf{m} and \mathbf{M} are connected by

$$\mathbf{m} \propto \mathbf{KRM}, \quad (1)$$

where \propto indicates equality up to scale and

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

where u_0 , v_0 , and f are variables such that $[u_0, v_0, -f]$ give the screen coordinates of the projector position; \mathbf{R} is a rotational component of the coordinate transform from the projector frame to the screen frame. Eq.(1) can be confirmed by considering that the projector and the screen form a system that has the same geometry as a pinhole camera; the position of the projector corresponds to the projection center and the screen corresponds to its image plane.

Conversely, suppose that we are given a 3×3 matrix \mathbf{H}_{ps} , or a *projector-to-screen homography*, such that

$$\mathbf{m} \propto \mathbf{H}_{ps}\mathbf{M}.$$

Applying a variant of the QR decomposition to \mathbf{H}_{ps} , \mathbf{H}_{ps} can be factorized into an upper triangular matrix \mathbf{K}' and a rotation matrix \mathbf{R}' :

$$\mathbf{H}_{ps} = \mathbf{K}'\mathbf{R}'. \quad (3)$$

By confining the signs of the diagonal elements of \mathbf{K}' to be either $[+, +, +]$ or $[-, -, +]$, the above factorization can be made unique up to these two cases. In order to guarantee $\det \mathbf{R}' = 1$, we normalize \mathbf{H}_{ps} so that $\det \mathbf{H}_{ps} > 0$. Then the position of the projector can be uniquely determined up to the sign of the z coordinate by comparing \mathbf{K}' with \mathbf{K} of Eq.(2), and the orientation can be correspondingly determined from \mathbf{R}' . In this way, the pointer pose can be calculated if \mathbf{H}_{ps} is provided.

2.2. Relation between screen and camera

However we can observe only the images of the projected patterns on the screen that are taken by the camera. Hence, in order to get the projector-to-screen homography \mathbf{H}_{ps} , we need to derive information on the screen plane from the observable images. Let \mathbf{n} be homogeneous coordinates of an image point. As is well-known, an image point and its corresponding point on the screen are connected by

$$\mathbf{n} \propto \mathbf{H}_{sc}\mathbf{m},$$

where \mathbf{H}_{sc} is the *screen-to-camera homography*. The substitution of $\mathbf{m} \propto \mathbf{H}_{ps}\mathbf{M}$ into the above equation yields

$$\mathbf{n} \propto \mathbf{H}_{sc}\mathbf{H}_{ps}\mathbf{M}. \quad (4)$$

By defining yet another homography, a *projector-to-camera homography*, as $\mathbf{H}_{pc} \equiv \mathbf{H}_{sc}\mathbf{H}_{ps}$, this can be rewritten as

$$\mathbf{n} \propto \mathbf{H}_{pc}\mathbf{M}. \quad (5)$$

The projector-to-camera homography, \mathbf{H}_{pc} , can be readily determined if the orientation \mathbf{M} of each beam is given and its corresponding image point \mathbf{n} can be identified. Since the screen-to-camera homography \mathbf{H}_{sc} is constant independently of the projector's pose, if \mathbf{H}_{sc} can be determined in advance by some method, the \mathbf{H}_{ps} of our interest can be simply computed as

$$\mathbf{H}_{ps} \propto \mathbf{H}_{sc}^{-1}\mathbf{H}_{pc}.$$

Thus, the problem is to obtain the screen-to-camera homography \mathbf{H}_{sc} . If there are several points of known coordinates on the screen surface, this can be easily done. If more than four point correspondences are given, \mathbf{H}_{sc} can be determined. In this paper we deal with the cases where no feature is available on the screen surface. As will be shown in the next section, even in those cases, it is possible to estimate \mathbf{H}_{sc} , provided that the orientations of the beams projected by the projector are known.

3. Screen-to-camera homography estimation

3.1. Problem formulation

We assume here that the projector-to-camera homography \mathbf{H}_{pc} can always be computed from the images. In the case of a multi-beam projector, this can be done, provided that (a) the orientation \mathbf{M} of each beam is known and its corresponding image point \mathbf{n} is identified, and (b) there are at least four such beams.

Then we consider the following problem.

Problem 3.1. *Given a sequence $\mathbf{H}_{pc,1}, \dots, \mathbf{H}_{pc,n}$, where $\mathbf{H}_{pc,i}$ is the projector-to-camera homography corresponding to i^{th} projector pose, determine the screen-to-camera homography \mathbf{H}_{sc} (and the projector poses).*

The following will be shown with respect to the solvability of this problem.

Proposition 3.1. *From a sequence $\mathbf{H}_{pc,1}, \dots, \mathbf{H}_{pc,n}$, \mathbf{H}_{sc} can be determined up to four free parameters. Three of them correspond to freedom of choice of the screen coordinate frame and the rest is scaling ambiguity.*

Suppose that for a given sequence $\mathbf{H}_{pc,1}, \dots, \mathbf{H}_{pc,n}$, a particular \mathbf{H}_{sc} enables factorization $\mathbf{H}_{sc}^{-1}\mathbf{H}_{pc,i} \propto \mathbf{K}_i\mathbf{R}_i$. Further suppose that there exist a 3×3 matrix \mathbf{T} such that, for any \mathbf{K} and \mathbf{R} , it is possible that $\mathbf{TKR} \propto \mathbf{K}'\mathbf{R}'$, where \mathbf{K}' is any matrix of the form of Eq.(2) and \mathbf{R}' is any orthogonal matrix. Then a matrix defined by $\mathbf{H}'_{sc} \propto \mathbf{H}_{sc}\mathbf{T}^{-1}$ should be another valid screen-to-camera homography. This is because

$$\mathbf{H}_{pc,i} \propto \mathbf{H}_{sc}\mathbf{K}_i\mathbf{R}_i \propto \mathbf{H}_{sc}\mathbf{T}^{-1}\mathbf{TK}_i\mathbf{R}_i = \mathbf{H}'_{sc}\mathbf{K}'_i\mathbf{R}'_i.$$

Thus, a key issue is if such \mathbf{T} exists and if so, what property it should have.

3.1.1 Difference from the problems of camera autocalibration from an image sequence

The above problem is quite similar to the problems of autocalibration of a camera [5, 3, 2]. They are problems of estimating camera parameters, including intrinsic ones, from a sequence of images, and their theoretical aspects have been already made clear. The camera autocalibration is to upgrade projective reconstruction to metric reconstruction, and it is done by factorizing projective camera matrices into a desired form. The main problem there was to clarify under what conditions the factorization is possible [2].

As described above, our problem here is to know whether it is possible to factorize the projector-to-camera homographies into our desired form and, if it is possible, how it may be achieved. Although the problem is seemingly the same as those of camera autocalibration, it differs in the following way: in the problems of camera autocalibration, the factorization form of $\mathbf{H} \propto \mathbf{KRT}$ is considered, whereas in our problem, the form of $\mathbf{H} \propto \mathbf{TKR}$ is considered, where in both cases, \mathbf{H} is a matrix obtained from observed data and \mathbf{T} is a matrix that needs to be determined. This slight difference results in that theories for camera autocalibration cannot be directly applied.

3.2. Uniqueness of solution

In order to prove Proposition 3.1 we first show two lemmas.

Lemma 3.2. *A 3×3 real matrix $\mathbf{V} \equiv [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]^T$ can be decomposed as $\mathbf{V} \propto \mathbf{KR}$ where \mathbf{K} is a matrix having the form of Eq.(2) and \mathbf{R} is an orthogonal matrix if and only if*

$$(\mathbf{v}_1 \times \mathbf{v}_3)^T(\mathbf{v}_2 \times \mathbf{v}_3) = 0 \quad (6)$$

and

$$|\mathbf{v}_1 \times \mathbf{v}_3| = |\mathbf{v}_2 \times \mathbf{v}_3|. \quad (7)$$

Proof. The QR (or RQ in this case) decomposition of \mathbf{V} is always possible as

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \mathbf{v}_3^\top \end{bmatrix} \propto \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \\ \mathbf{r}_3^\top \end{bmatrix}$$

where \mathbf{r}_1 , \mathbf{r}_2 , and \mathbf{r}_3 are the vectors forming an orthogonal system, and this decomposition is unique. In order for \mathbf{V} to be decomposed as described, it must hold that $a_{12} = 0$ and $a_{11} = a_{22}$, and vice versa. These equations can be represented using \mathbf{v}_1 , \mathbf{v}_2 , and \mathbf{v}_3 as Eqs. (6) and (7) \square

Lemma 3.3. *Let \mathbf{T} be a 3×3 matrix. For any matrix \mathbf{K} having the form of Eq.(2), multiplication \mathbf{TK} can be decomposed as $\mathbf{TK} \propto \mathbf{K}'\mathbf{U}$ where \mathbf{K}' is also a matrix of the form Eq.(2) and \mathbf{U} is any orthogonal matrix if and only if \mathbf{T} is given by*

$$\mathbf{T} \propto \begin{bmatrix} \cos \theta & -\sin \theta & p \\ \sigma \sin \theta & \sigma \cos \theta & q \\ 0 & 0 & r \end{bmatrix}, \quad (8)$$

where σ is either 1 or -1 .

Proof. We write \mathbf{T} and \mathbf{K} as

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} a & 0 & b \\ 0 & a & c \\ 0 & 0 & 1 \end{bmatrix}.$$

Then we apply the result of Lemma 3.2 here. By defining $\mathbf{V} \equiv \mathbf{TK}$ and substituting into Eqs.(6) and (7) we obtain two equations for the entries of \mathbf{T} . Those equations must hold for any \mathbf{K} , i.e., any a , b , and c . Then the coefficient of any order terms $a^i b^j c^k$ must be zero in the equations. Several equations are available, from which we obtain the following as independent ones:

$$t_{12}t_{31} + t_{11}t_{32} = t_{22}t_{31} + t_{21}t_{32} = 0, \quad (9a)$$

$$t_{11}t_{21} + t_{12}t_{22} = 0, \quad (9b)$$

$$t_{11}^2 + t_{12}^2 = t_{21}^2 + t_{22}^2. \quad (9c)$$

In order for \mathbf{T} not to be a trivial solution of $\mathbf{T} = \mathbf{0}$, it must hold that $t_{31} = t_{32} = 0$. Then it is easy to see that Eq.(8) is one parametrization that implicitly represents Eqs.(9a)-(9c).

In fact, if \mathbf{T} is given by Eq.(8), the following decomposition is possible:

$$\begin{aligned} \mathbf{TK} &\propto \begin{bmatrix} \cos \theta & -\sin \theta & p \\ \sigma \sin \theta & \sigma \cos \theta & q \\ 0 & 0 & r \end{bmatrix} \begin{bmatrix} a & 0 & b \\ 0 & a & c \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} a & 0 & b \cos \theta - c \sin \theta + p \\ 0 & a & b \sigma \sin \theta + c \sigma \cos \theta + q \\ 0 & 0 & r \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sigma \sin \theta & \sigma \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (10)$$

The first matrix on the right hand side has the desired form and the second is an orthogonal matrix. \square

Now we prove Proposition 3.1.

Proof of Proposition 3.1. Let \mathbf{H}_{sc} be the true screen-to-camera homography. Then $\mathbf{H}_{pc,i}$ can be decomposed as $\mathbf{H}_{pc,i} \propto \mathbf{H}_{sc}\mathbf{K}_i\mathbf{R}_i$. In order for \mathbf{T} to make possible a different decomposition: $\mathbf{H}_{sc}\mathbf{K}_i\mathbf{R}_i \propto \mathbf{H}_{sc}\mathbf{T}^{-1}\mathbf{TK}_i\mathbf{R}_i \propto \mathbf{H}'_{sc}\mathbf{K}'_i\mathbf{R}'_i$, it must be given in the form of Eq.(8), which has four free parameters, θ , p , q , and r . From Eq.(10) the coordinates $[b, c, a]$ of the pointer position are transformed by \mathbf{T} into $[b', c', a']$ as

$$\begin{bmatrix} b' \\ c' \\ a' \end{bmatrix} = \begin{bmatrix} (b \cos \theta - c \sin \theta + p)/r \\ \sigma(b \sin \theta + c \cos \theta + q)/r \\ a/r \end{bmatrix}.$$

Also, the pointer orientation is transformed as

$$\mathbf{R}' = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sigma \sin \theta & \sigma \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{R}.$$

Thus, it can be seen that the coordinates are transformed by 2D translation on the screen plane by $[p, q]$, rotation by angle θ around the screen z axis, and exchange between a left hand system and a right hand system by σ . These can be said to correspond to choice of the screen coordinate frame. It is also scaled by r . \square

Thus we have shown that \mathbf{H}_{sc} can be determined up to the described ambiguities. These can be resolved only by other means. An example of such resolution is to fix both the screen coordinate frame and the floating scale by setting

$$\mathbf{K}_1 \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{K}_2 \equiv \begin{bmatrix} * & 0 & 0 \\ 0 & * & * \\ 0 & 0 & 1 \end{bmatrix}, \quad (11)$$

which means that the 1st and 2nd pointer positions are set to $[0, 0, 1]$ and $[0, *, *]$, respectively. Then only $\mathbf{T} = \text{diag}[\pm 1, \pm 1, 1]$ is allowed for making the decomposition possible and we can determine \mathbf{H}_{sc} up to any combination of the signs. There are four combinations of the signs and thus we have four possible solutions.

4. Algorithms for computing the screen-to-camera homography

4.1. A noniterative method

In this section, we present an algorithm for solving Problem 3.1.

The problem was to derive \mathbf{H}_{sc} from a sequence $\mathbf{H}_{pc,1}, \dots, \mathbf{H}_{pc,n}$. We want to derive equations for \mathbf{H}_{sc} in as simple a form as possible. As in the case of camera autocalibration, orthogonality of rotation matrices is used first. Let $\mathbf{A}_i \equiv \mathbf{H}_{pc,i}\mathbf{H}_{pc,i}^\top$. The substitution of $\mathbf{H}_{pc,i} \propto \mathbf{H}_{sc}\mathbf{K}_i\mathbf{R}_i$ into $\mathbf{A}_i \equiv \mathbf{H}_{pc,i}\mathbf{H}_{pc,i}^\top$ yields

$$\mathbf{A}_i \propto \mathbf{H}_{sc}\mathbf{K}_i\mathbf{R}_i\mathbf{R}_i^\top\mathbf{K}_i^\top\mathbf{H}_{sc}^\top = \mathbf{H}_{sc}\mathbf{K}_i\mathbf{K}_i^\top\mathbf{H}_{sc}^\top, \quad (12)$$

where \mathbf{H}_{sc} and \mathbf{K}_i are unknowns. The fact that \mathbf{K}_i has the form of Eq.(2) places constraints on these unknowns, from which we can derive equations for \mathbf{H}_{sc} . For example, the elimination of entries of \mathbf{K}_i from Eq.(12) results in two polynomial equations of degree 4 for the entries of \mathbf{H}_{sc} . They are, however, quite difficult to solve analytically, due to their nonlinearity¹. Thus, another way must be found.

As described, there is freedom of choice of the screen coordinate system. By assuming \mathbf{K}_1 and \mathbf{K}_2 as in Eq.(11) and exploring the properties of the resulting equations, we can derive a comparatively simple algorithm as shown below.

Firstly, for the 1st pose, since $\mathbf{K}_1\mathbf{K}_1^T = \mathbf{I}$, we have $\mathbf{A}_1 \propto \mathbf{H}_{sc}\mathbf{H}_{sc}^T$. Since \mathbf{A}_1 is symmetry, its singular value decomposition (SVD) can be represented as $\mathbf{A}_1 = \mathbf{U}_1\mathbf{D}_1\mathbf{U}_1^T$ where \mathbf{U}_1 is an orthogonal matrix and \mathbf{D}_1 is a diagonal matrix. We use here the following known result; see [4] for proof.

Lemma 4.1. *For any given symmetry matrix \mathbf{A} , consider a square matrix \mathbf{X} satisfying $\mathbf{A} \propto \mathbf{X}\mathbf{X}^T$. There are many possible solutions. Let \mathbf{X}_0 be a particular solution. Then, all of the possible solutions can be represented by $\mathbf{X}_0\mathbf{Q}$ where \mathbf{Q} is any orthogonal matrix.*

By applying this, we can represent \mathbf{H}_{sc} without loss of generality as $\mathbf{H}_{sc} \propto \mathbf{U}_1\mathbf{D}_1^{\frac{1}{2}}\mathbf{Q}$, where $\mathbf{D}_1^{\frac{1}{2}}$ is a diagonal matrix whose elements are square roots of those of \mathbf{D}_1 , and \mathbf{Q} is an orthogonal matrix.

We next define for the 2nd pose

$$\mathbf{K}_2 = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & \beta \\ 0 & 0 & 1 \end{bmatrix}.$$

By substituting this along with $\mathbf{H}_{sc} \propto \mathbf{U}_1\mathbf{D}_1^{\frac{1}{2}}\mathbf{Q}$ into Eq.(12), we have

$$\mathbf{A}_2 \propto \mathbf{U}_1\mathbf{D}_1^{\frac{1}{2}}\mathbf{Q}\mathbf{W}\mathbf{Q}^T\mathbf{D}_1^{\frac{1}{2}}\mathbf{U}_1^T, \quad (13)$$

where

$$\mathbf{W} \equiv \mathbf{K}_2\mathbf{K}_2^T = \begin{bmatrix} \alpha^2 & 0 & 0 \\ 0 & \alpha^2 + \beta^2 & \beta \\ 0 & \beta & 1 \end{bmatrix}. \quad (14)$$

By calculation, it can be shown that the matrix \mathbf{W} has the following eigenvalues:

$$\begin{cases} \lambda_1, \lambda_3 = \frac{\alpha^2 + \beta^2 + 1 \pm \sqrt{((\alpha-1)^2 + \beta^2)((\alpha+1)^2 + \beta^2)}}{2} \\ \lambda_2 = \alpha^2 \end{cases}. \quad (15)$$

They are ordered as $\lambda_1 \geq \lambda_2 \geq \lambda_3$ independently of α and β , and the equalities hold only if $\alpha = 0$ or $\beta = 0$. Since it means that the pointer is exactly on the screen, $\alpha = 0$

¹As is often done in the methods for camera autocalibration [5], we can probably convert the nonlinear equations into linear ones by introducing new redundant variables. This requires a huge amount of new variables, however, due to the form of Eq.(12).

should not happen. Thus, the above eigenvalues coincide only when $\beta = 0$.

Now we want to determine the unknown \mathbf{Q} . In Eq.(13), we move known matrices \mathbf{U}_1 and \mathbf{D}_1 to the left hand side and define $\mathbf{A}'_2 \equiv \mathbf{D}_1^{-\frac{1}{2}}\mathbf{U}_1^T\mathbf{A}_2\mathbf{U}_1\mathbf{D}_1^{-\frac{1}{2}} (\propto \mathbf{Q}\mathbf{W}\mathbf{Q}^T)$. Its SVD can be represented as $\mathbf{A}'_2 \rightarrow \mathbf{U}'_2\mathbf{D}'_2\mathbf{U}'_2{}^T$, where \mathbf{U}'_2 is an orthogonal matrix and \mathbf{D}'_2 is a diagonal matrix. Since $\mathbf{A}'_2 \propto \mathbf{Q}\mathbf{W}\mathbf{Q}^T$ and $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$, known \mathbf{A}'_2 and unknown \mathbf{W} should have collinear eigenvalues. That is, letting τ_1, τ_2 , and τ_3 ($\tau_1 \geq \tau_2 \geq \tau_3$) be the eigenvalues of \mathbf{A}'_2 , $[\lambda_1, \lambda_2, \lambda_3]^T \propto [\tau_1, \tau_2, \tau_3]^T$. By solving this for α and β we have $\alpha = \pm\tau_2/\sqrt{\tau_1\tau_3}$ and $\beta = \pm\sqrt{(\tau_1 - \tau_2)(\tau_2 - \tau_3)/(\tau_1\tau_3)}$. (We again neglect the case of $\alpha = 0$ here.) As a result, there are four pairs of solutions for (α, β) corresponding to the signs, but effectively there are only two since α appears only in the form of α^2 .

By substituting these solutions for (α, β) into \mathbf{W} in Eq.(14), its SVD can be computed: $\mathbf{W} \rightarrow \mathbf{U}''_2\mathbf{D}''_2\mathbf{U}''_2{}^T$. The substitution of this into $\mathbf{A}'_2 \propto \mathbf{Q}\mathbf{W}\mathbf{Q}^T$ yields $\mathbf{A}'_2 \propto \mathbf{Q}\mathbf{U}''_2\mathbf{D}''_2\mathbf{U}''_2{}^T\mathbf{Q}^T$. Then we compare this with the SVD of \mathbf{A}'_2 , $\mathbf{A}'_2 \propto \mathbf{U}'_2\mathbf{D}'_2\mathbf{U}'_2{}^T$, that have been already computed. It is well known that for any matrix, its SVD, $\mathbf{U}\mathbf{D}\mathbf{V}^T$, is unique up to sign changes of any column vector of \mathbf{U} and \mathbf{V} , if the singular values are all different and sorted, say, in descending order. As described above \mathbf{W} has different eigenvalues whenever $\beta \neq 0$. Therefore, if singular values are sorted in descending order, it should hold that \mathbf{U}'_2 is equivalent to $\mathbf{Q}\mathbf{U}''_2$ up to sign changes of column vectors. Thus we can determine \mathbf{Q} as

$$\mathbf{Q} = \mathbf{U}'_2\text{diag}[\pm 1, \pm 1, \pm 1]\mathbf{U}''_2{}^T$$

There are eight solutions for \mathbf{Q} corresponding to the sign changes but effectively four since \mathbf{H}_{sc} has inherent scaling ambiguity. Thus, we have

$$\mathbf{H}_{sc} \propto \mathbf{U}_1\mathbf{D}_1^{\frac{1}{2}}\mathbf{U}'_2 \begin{bmatrix} \pm 1 & 0 & 0 \\ 0 & \pm 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{U}''_2{}^T.$$

There were two solutions for (α, β) . Hence there are, in total, eight ($2 \times 4 = 8$) solutions that satisfy all the constraints coming from the 1st and 2nd poses. According to the results of the last section, when data from a sufficient number of poses are given, there should be only four solutions. We can therefore choose four correct solutions by testing each of the eight solutions against the 3rd and more poses. The algorithm can be summarized as Fig.2, and as to the number of poses required, the following can be said.

Proposition 4.2. *If the parametrization by Eq.(11) is employed for the 1st and 2nd projector poses, eight solutions are obtained from the corresponding two images, unless the positions of the two projector poses are accidentally on a line perpendicular to the screen. They can be reduced to four if more than one image derived from generic projector poses is added.*

1. Compute the projector-to-camera homography $\mathbf{H}_{pc,i}$ for each pose i .
2. For the 1st pose, define $\mathbf{A}_1 = \mathbf{H}_{pc,1} \mathbf{H}_{pc,1}^\top$ and compute SVD of \mathbf{A} as $\mathbf{A}_1 \rightarrow \mathbf{U}_1 \mathbf{D}_1 \mathbf{U}_1^\top$.
3. For the 2nd pose, define $\mathbf{A}'_2 = \mathbf{D}_1^{-\frac{1}{2}} \mathbf{U}_1^\top \mathbf{A}_2 \mathbf{U}_1 \mathbf{D}_1^{-\frac{1}{2}}$ and compute its SVD as $\mathbf{A}'_2 \rightarrow \mathbf{U}'_2 \mathbf{D}'_2 \mathbf{U}'_2{}^\top$. All SVD's must be computed so that the diagonal elements are sorted in descending order. Let τ_1, τ_2 , and τ_3 ($\tau_1 \geq \tau_2 \geq \tau_3$) be the diagonal elements of \mathbf{D}'_2 . Compute α and β as

$$\alpha^2 = \tau_2^2 / \tau_1 \tau_3$$

$$\beta = \pm \sqrt{(\tau_1 - \tau_2)(\tau_2 - \tau_3) / (\tau_1 \tau_3)}.$$

4. Substitute the resulting α and β into \mathbf{W} in Eq.(14) and further compute its SVD: $\mathbf{W} \rightarrow \mathbf{U}_2'' \mathbf{D}_2'' \mathbf{U}_2''{}^\top$. Then, \mathbf{H}_{sc} is given by $\mathbf{H}_{sc} \propto \mathbf{U}_1 \mathbf{D}_1^{\frac{1}{2}} \mathbf{U}_2'' \text{diag}[\pm 1, \pm 1, 1] \mathbf{U}_2''{}^\top$.
5. From the eight candidates of solution, select four that are compatible with the 3rd and subsequent poses. This is done by computing the factorization $\mathbf{H}_{pc,3} \hat{\mathbf{H}}_{sc}^{-1} \rightarrow \mathbf{K} \mathbf{R}$, where $\hat{\mathbf{H}}_{sc}$ is a candidate of solution need to be tested, and then by checking if \mathbf{K} has the form of Eq.(2).

Figure 2. The proposed algorithm

The additional condition on the projector positions in the above statement is to guarantee $\beta \neq 0$.

4.2. Nonlinear iterative refinement: Bundle adjustment

The above noniterative method determines the parameters only from the 1st and the 2nd images. The rest of the images are used only for selecting solutions. It is clear that the more images take part in the parameter estimation, the more accurate the estimation. For the solution obtained by the above noniterative algorithm, we can refine the solution by maximum likelihood inference as is frequently done in many camera calibration methods. This is done by a nonlinear minimization of the so called reprojection errors: $\sum |p_i - \hat{p}_i(\theta)|^2 + |q_i - \hat{q}_i(\theta)|^2$, where (p_i, q_i) are the measured image coordinates of the i^{th} beam spots, $(\hat{p}_i(\theta), \hat{q}_i(\theta))$ are the functions representing the geometry, and θ is the parameter we want to estimate. Employing minimal parametrization by Eq.(11), we choose the following as the elements of θ : $\mathbf{H}_{sc}, \mathbf{R}_1, \dots, \mathbf{R}_n$, the elements of \mathbf{K}_2 other than (1,3), and $\mathbf{K}_3, \dots, \mathbf{K}_n$. This nonlinear method can improve the estimation accuracy, although it has a greater computational cost than does the proposed noniterative method.

5. Experimental results

The proposed algorithm has been tested on both synthetic and real data.

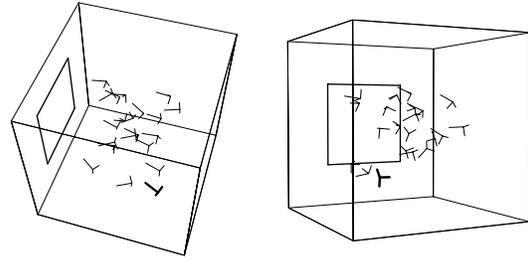


Figure 3. An example sequence of the projector poses used for the experiments; see text.

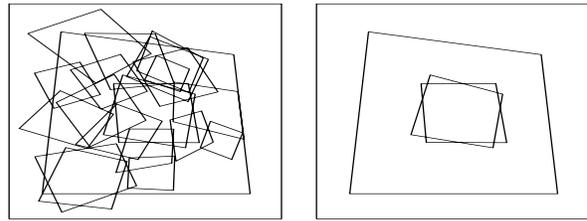


Figure 4. An example of the synthesized images overlaid into one image. Left: 20 images. Right: The 1st and 2nd images that are used as "key images" in the proposed algorithm.

5.1. Synthetic data

Experimental setup for synthetic data Figure 3 shows the setup used for the experiments. The quadrilateral on one side of the cube indicates a unit square on the screen plane. The triplets of thin lines indicate the projector poses, which are randomly chosen within a certain range of pose parameters. One triplet of thick lines indicates the pose of the camera. The proposed noniterative algorithm determines the parameter mostly from the 1st and the 2nd poses. Therefore, they are not randomly chosen, but chosen especially so that the resulting images become those shown on the right of Fig.4.

The projector used here has four discrete beams. The beams are along the edges of a square cone that has a ϕ diagonal angle. The resulting synthetic images are shown in Fig.4. The largest quadrilateral is the unit square on the screen, which is also shown in Fig.3. The small quadrilaterals scattered in the image represent the projections of the beams on the screen; their four corners are the image points of the beam spots. Then, Gaussian noise with mean 0 and variance σ^2 is added to those image points. The noise level σ is changed from 0.1 to 1.5 pixels, assuming the image size to be 500×500 pixels.

Accuracy measures of \mathbf{H}_{sc} The proposed noniterative algorithm and nonlinear refinement are applied to the images generated in the above way. Their estimation accu-

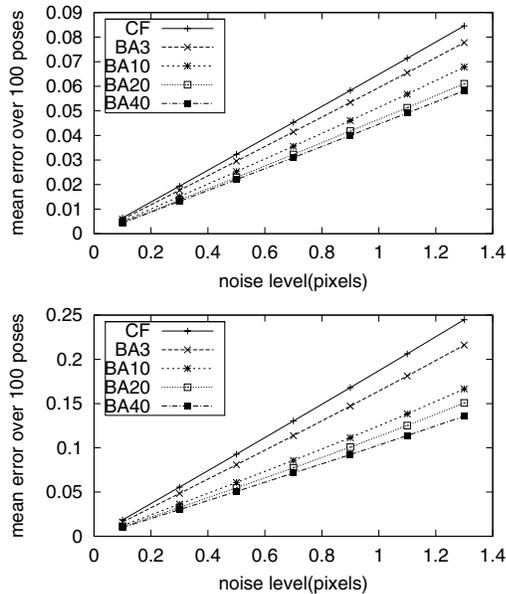


Figure 5. Errors vs. the noise level of the image points. Upper: Projector position. Lower: Orientation. 'CF' indicates the proposed noniterative algorithm and 'BA n ' indicates nonlinear refinement using n images.

racies are measured using recovery of the projector poses using estimated \mathbf{H}_{sc} 's. In order to distinguish the noise effects on the \mathbf{H}_{sc} estimation and those on the \mathbf{H}_{pc} decomposition, two image sequences are prepared. One is generated with noise and the other without. The former is used, of course, for estimating \mathbf{H}_{sc} and the latter is used for evaluating the accuracy of the estimated \mathbf{H}_{sc} . For the sequence without noise, the estimation accuracy is measured by errors of the recovery of the projector poses: (Error of position) = $\sqrt{\frac{1}{100} \sum_{i=1}^{100} |\mathbf{t}_i - \widehat{\mathbf{t}}_i|^2}$ and (Error of orientation) = $\sqrt{\frac{1}{100} \sum_{i=1}^{100} \|I - \mathbf{R}_i \widehat{\mathbf{R}}_i^T\|^2}$.

Performance w.r.t. the noise level In this experiment, we varied the noise level σ from 0.1 to 1.4 pixels. For each noise level, 100 trials were independently conducted by randomly choosing the image noise alone. The beam angle was set to $\phi = 20$ degrees. Figure 5 shows the results. It can be seen that for the case of $\sigma = 0.5$ pixels, which is considered to be a typical case, the relative errors for the noniterative algorithm are less than 10% for both the position and orientation. It can also be seen that the nonlinear refinement improves the estimation accuracy and that more images are used, the greater is the improvement in accuracy.

Performance w.r.t. the beam angle It is anticipated that we will get poor results with projectors with a small beam

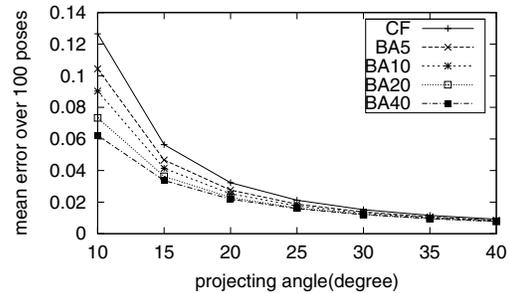


Figure 6. Error of the estimated projector position vs. the angle spanned by the beams.



Figure 7. Experimental setup.

angle. In this experiment we varied the beam angle ϕ from 10 to 40 degrees. The noise level σ was fixed at 0.5. The results are shown in Fig.6. To conserve space, only the results for the projector position are shown. The estimation errors seem to be inversely proportional to square of the beam angle, and it can be seen that the accuracy varies drastically from 10 to 20 degrees. It might be said that a beam angle of at least 20 degrees is desirable for applications that require accuracy.

5.2. Real data

The proposed algorithm was also tested on real images. The projector used here had four laser beams, which were made of four off-the-shelf single-beam type laser pointers. An ordinary whiteboard of approximately 1.5m \times 1.0m is used for the screen. On the screen, four marker points were attached for evaluating the accuracy of the results. A NIKON digital camera D1 was used, whose image size is 2000 \times 1312 pixels. The lens has a field angle of approximately 40 degrees. The setup is shown in Fig.7.

An example of the images acquired is shown in Fig.8. The beam spots appearing on the images were identified manually and then extracted automatically by color thresholding. Since each beam spot had an area, although it is small, a mean was computed to determine exact image coordinates.

Since it is difficult to get the ground truth of the pointer poses in this case, we measured the estimation accuracy in



Figure 8. An example of the image and its enlarged subimage of four beam spots.

the following way. Firstly, the screen-to-camera homography \mathbf{H}_{sc} was estimated from the four marker points attached on the screen. Let $\widehat{\mathbf{H}}_{sc,0}$ be the estimation. This is expected to be highly accurate due to the use of the special purpose marker points. Thus, we assumed $\widehat{\mathbf{H}}_{sc,0}$ to be accurate and thought of this as the ground truth for \mathbf{H}_{sc} .

The estimation by the proposed algorithm, $\widehat{\mathbf{H}}_{sc}$, was then compared with $\widehat{\mathbf{H}}_{sc,0}$. There is freedom of coordinate frame choice between $\widehat{\mathbf{H}}_{sc}$ and $\widehat{\mathbf{H}}_{sc,0}$, and thus direct comparison does not make sense. They are connected by the following relation if both are correct:

$$\widehat{\mathbf{H}}_{sc,0} \propto \widehat{\mathbf{H}}_{sc} \mathbf{T},$$

where \mathbf{T} is a matrix having the form of Eq.(8). In other words, if $\widehat{\mathbf{H}}_{sc}$ is correct, $\mathbf{T}' \equiv \widehat{\mathbf{H}}_{sc}^{-1} \widehat{\mathbf{H}}_{sc,0}$ should have the form Eq.(8). Thus, the estimation accuracy can be measured by checking the matrix \mathbf{T}' . Let $[\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3] \equiv \mathbf{T}'$. The conditions for \mathbf{T}' having the form Eq.(8) are given as Eqs.(9a)-(9c), which can be rewritten as $e_a \equiv 1 - [0, 0, 1]^T (\mathbf{t}_1 \times \mathbf{t}_2) = 0$, $e_b \equiv |\mathbf{t}_1| - |\mathbf{t}_2| = 0$, and $e_c \equiv (\mathbf{t}_1^T \mathbf{t}_2) / (|\mathbf{t}_1| |\mathbf{t}_2|) = 0$. If errors exist in the estimation, e_a , e_b , and e_c have non-zero values. Thus, we used e_a , e_b , and e_c as indexes for the estimation accuracy.

For 13 selected pairs of the images, we applied the proposed noniterative algorithm. The resulting indexes e_a , e_b , and e_c are shown in Table.1. It can be said that the results have considerably accuracy, assuming that $\widehat{\mathbf{H}}_{sc,0}$ is correct. An example of recovered projector poses is shown in Fig.9. In the same figure also poses recovered from $\widehat{\mathbf{H}}_{sc,0}$ is shown, to which the coordinate transform given by \mathbf{T}' were applied. The two recovery results almost coincide, and thus their difference is hardly seen².

6. Summary

We have shown that for the projector-screen-camera system, if the pattern the projector projects toward space is known, the screen-to-camera homography can be determined from multiple images of the patterns up to choice of the internal coordinate representation. In addition, we have

²If the reader is reading the paper on a PC display, he or she can see the difference by enlarging the figure.

Table 1. The indexes for the estimation accuracy over 13 pairs of images; see text.

	mean	std. dev.
e_a	-0.0033	0.020
e_b	0.0037	0.011
e_c	0.00031	0.00045

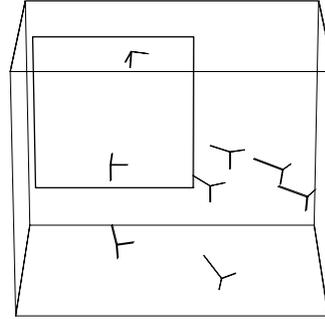


Figure 9. An example of projector pose recovery. Projector poses recovered using $\widehat{\mathbf{H}}_{sc,0}$ are also shown.

presented a noniterative algorithm that can directly provide solutions from basically more than two images. The performance of the method was confirmed by several experimental results.

Acknowledgment

We would like to thank Franck Galpin for helpful discussions and comments.

References

- [1] Y. Chen, D. W. Clark, A. Finkelstein, T. Housel, and K. Li. Automatic alignment of high-resolution multi-projector displays using an un-calibrated camera. In *IEEE Visualization*, 2000.
- [2] A. Heyden and K. Åström. Minimal conditions on intrinsic parameters for euclidean reconstruction. In *Proceedings of Asian Conference on Computer Vision*, 1998.
- [3] A. Heyden and K. Åström. Flexible calibration: Minimal cases for auto-calibration. In *Proceedings of IEEE International Conference on Computer Vision*, pages 438–443, 1999.
- [4] K. Kanatani. *Geometric computation for machine vision*. Oxford: Clarendon Press, 1993.
- [5] M. Pollefeys, R. Koch, and L. V. Gool. Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.
- [6] R. Sukthankar, R. G. Stockton, and M. D. Mullin. Smarter presentations: Exploring homography in camera-projector systems. In *Proceedings of IEEE International Conference on Computer Vision*, 2001.