Stratified Self Calibration from Screw-Transform Manifolds

Russell Manning and Charles Dyer

University of Wisconsin, Madison WI 53715, USA

Abstract. This paper introduces a new, stratified approach for the metric self calibration of a camera with fixed internal parameters. The method works by intersecting *modulus-constraint manifolds*, which are a specific type of screw-transform manifold. Through the addition of a single scalar parameter, a 2-dimensional modulus-constraint manifold can become a 3-dimensional Kruppa-constraint manifold allowing for direct self calibration from disjoint pairs of views. In this way, we demonstrate that screw-transform manifolds represent a single, unified approach to performing both stratified and direct self calibration. This paper also shows how to generate the screw-transform manifold arising from turntable (i.e., pairwise-planar) motion and discusses some important considerations for creating a working algorithm from these ideas.

1 Introduction

Self calibration is the process of determining the internal parameters of a camera directly from views taken by the camera without any a priori knowledge of the scene, the camera, or the camera's motion. Calibration is said to be *metric* or *Euclidean* when it matches "true" calibration (as measured using a standard orthonormal coordinate system) up to an overall scale factor, translation, and rotation. There are two major routes to performing metric self calibration. The first route, which we shall refer to as *direct self calibration*, involves determining calibration directly from a collection of views in a single conceptual step, typically by utilizing the Kruppa constraints and its variations. The second route, which is known as *stratified self calibration*, involves working in stages, first creating a projective reconstruction of the scene and cameras, then upgrading this to an affine reconstruction, and finally upgrading to metric. Each method has strengths and weaknesses: the stratified approach seems generally preferable but some direct approaches can be used in situations where stratified approaches are not possible.

In this paper, we introduce a new algorithm for stratified self calibration based on the recently-introduced concept of screw-transform manifolds [10]. We also discuss how stratified approaches are intimately related to direct approaches; in particular, only one real-valued parameter separates the stratified approach given in this paper from the direct approach given in [10]. In this way, we show that stratified self calibration and direct self calibration are unified by the single concept of screw-transform manifolds. Additionally, we give for the first time an algorithm for generating the screw-transform manifold associated with pure turntable motion (i.e., pairwise-planar motion). We also discuss several issues regarding the effective implementation of the algorithms given in this paper.

The archetypal direct method for metric self calibration, which also happens to be the first metric self-calibration algorithm published, is that due to Faugeras et al. [6] based on the Kruppa constraints. The great strength of this algorithm, besides mathematical simplicity, is its ability to determine metric calibration directly from pairwise fundamental matrices. Any two overlapping views taken by a camera induce a fundamental matrix between the views, and with enough fundamental matrices the Kruppa constraints can be utilized. Thus to perform self calibration it is only necessary to capture a series of pairwise-overlapping views. In contrast, stratified approaches require a series of views that have enough mutual overlap to create an initial projective reconstruction of the entire scene, including locating within the reconstruction the optical centers of all the disparate views. Furthermore, stratified approaches require that such a reconstruction actually be created, which takes time and increases algorithmic complexity and the possibility of failure.

Unfortunately, using the Kruppa constraints has important weaknesses. First, there are certain *critical motion sequences* (see [15]) that do not contain enough information to perform self calibration via the Kruppa constraints but do contain enough information to be successfully handled through a stratified approach. Furthermore, by skipping the affine reconstruction stage, direct approaches never force all camera views to share a common *plane at infinity.* This can lead to a metric calibration that satisfies the pairwise Kruppa constraints for every view but is nevertheless not simultaneously consistent with every view. Stratified approaches ensure that every view shares a common plane at infinity when the projective reconstruction is upgraded to affine. This extra constraint can help disambiguate the effects of noisy data.

More information on stratified approaches to self calibration can be found in [11, 4, 2, 17, 8], among other sources. Numerous other approaches to self calibration also exist (e.g., [16]); however, in this paper we will mostly compare and contrast our approach to the stratified calibration algorithm introduced in [12] which is based on the modulus constraint. We will be especially concerned with differences between the implicit representation provided by the modulus constraint and the explicit representation provided by screw-transform manifolds.

2 Overview of stratified self calibration with the modulus constraint

In this section, we provide an overview of stratified self calibration and introduce terminology and concepts that will be needed later in the paper when discussing the new calibration algorithm. In particular, we will discuss how the modulus constraint can be used for stratified self calibration. A thorough discussion of the modulus constraint can be found in [11, 12].

Our goal is to find the internal calibration of a camera from a series of m views of a static scene. The camera captures the views from various positions and orientations and its internal parameters are assumed to never change. The internal parameters can be represented by an upper-triangular 3×3 matrix $\mathbf{K}$. Each view corresponds to a 3×4 camera matrix

$$\hat{\mathbf{\Pi}}_i = \begin{bmatrix} \hat{\mathbf{H}}_i & \hat{\mathbf{e}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{K}\mathbf{R}_i & \hat{\mathbf{e}}_i \end{bmatrix}$$

where $\hat{\mathbf{H}}_i$ is a 3×3 matrix and $\mathbf{R}_i$ is a rotation matrix. By choosing the appropriate metric coordinate system, we can assume $\hat{\mathbf{e}}_1 = \mathbf{0}$ and $\mathbf{R}_1 = \mathbf{I}$.

Under the stratified self-calibration paradigm, the first step in finding $\mathbf{K}$ and finding $\hat{\mathbf{\Pi}}_i$ is to find the camera matrices in a common projective basis. These initial projective camera matrices will be labeled $\mathbf{\Pi}_i$ and are related to the metric camera matrices by a 4×4 matrix $\mathbf{\Psi}$ representing a transformation of projective basis:

$$\mathbf{\Pi}_i = \begin{bmatrix} \mathbf{H}_i & \mathbf{e}_i \end{bmatrix} = \hat{\mathbf{\Pi}}_i \mathbf{\Psi} \tag{1}$$

We will refer to the initial set of camera matrices in the common projective basis as a *projective reconstruction* of the scene. The usual approach to obtaining this projective reconstruction (e.g., [1]) is to identify feature points in the scene that are visible in more than one camera, then reconstruct the features in a common projective basis (e.g., using fundamental matrices [3]), and then find the projective camera matrices by relating the reconstructed features to their viewed positions. However, using feature points in this way is not strictly necessary so we will not refer to feature points explicitly in discussing self calibration; features are not an implicit part of the projective reconstruction.

We can always choose the projective basis so that $\mathbf{\Pi}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$. Other authors (e.g., [11]) have shown that, under the assumptions so far, $\mathbf{\Psi}$ must have the form

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{K}^{-1} & \mathbf{0} \\ -\hat{\mathbf{a}}^\top & a \end{bmatrix} \tag{2}$$

Different values for the scalar a simply lead to different overall scale factors for the final metric reconstruction. Since metric reconstruction only involves recovering the scene up to a scale factor, we can choose $a = 1$ for convenience.

The second stage in stratified self calibration is to upgrade the projective reconstruction to an affine reconstruction. This is equivalent to finding $\hat{\mathbf{a}}$ in Eq. 2. The final stage is to upgrade the affine reconstruction to metric by finding $\mathbf{K}$. This latter step can be performed in a simple way by solving a linear system [7]; the hard step is finding $\hat{\mathbf{a}}$ during the second stage.

Pollefeys et al. [14] introduced an elegant method for determining $\hat{\mathbf{a}}$ called the *modulus constraint*. Notice that from Eq. 1 and the assumption $a = 1$ we get

$$\begin{aligned} \mathbf{H}_i &= \hat{\mathbf{H}}_i \mathbf{K}^{-1} - \hat{\mathbf{e}}_i \hat{\mathbf{a}}^\top \\ \mathbf{e}_i &= \hat{\mathbf{e}}_i \end{aligned}$$

leading to

$$\mathbf{H}_{1i}^{\infty} := \mathbf{K}\mathbf{R}_i\mathbf{K}^{-1} = \mathbf{H}_i + \mathbf{e}_i\hat{\mathbf{a}}^\top \tag{3}$$

(where the notation $x := y$ defines the symbol x as representing the quantity y). Since the right-hand side of Eq. 3 is conjugate to a rotation matrix, its eigenvalues must all have the same modulus (i.e., absolute value) and this leads to constraints on $\hat{\mathbf{a}}$. Similarly,

$$\mathbf{H}_{ij}^{\infty} := \mathbf{K}\mathbf{R}_j\mathbf{R}_i^{-1}\mathbf{K}^{-1} = \mathbf{H}_{1j}^{\infty}(\mathbf{H}_{1i}^{\infty})^{-1} = (\mathbf{H}_j + \mathbf{e}_j\hat{\mathbf{a}}^\top)(\mathbf{H}_i + \mathbf{e}_i\hat{\mathbf{a}}^\top)^{-1} \tag{4}$$

Here $\mathbf{H}_{ij}^{\infty}$ is conjugate to the rotation matrix $\mathbf{R}_j\mathbf{R}_i^{-1}$ leading to additional constraints on $\hat{\mathbf{a}}$. Let $\mathbf{H}_{ij}^{\infty}(\mathbf{a})$ denote Eqs. 3 and 4, respectively, with $\hat{\mathbf{a}}$ replaced by a generic vector $\mathbf{a} \in \Re^3$. By expanding the characteristic equation for $\mathbf{H}_{ij}^{\infty}(\mathbf{a})$ and imposing the modulus constraint, each (i, j) pair, $i < j$, leads to a fourth-order multivariate polynomial ϕ_{ij} in the components of $\mathbf{a}$ that has the property $\phi_{ij}(\mathbf{a}) = 0$ for all $\mathbf{a}$ that generate legal $\mathbf{H}_{ij}^{\infty}$ matrices (i.e., matrices that meet the modulus constraint). See [13] for the specific form of ϕ_{ij}. Throughout this paper, we will use $\mathbf{H}_{ij}^{\infty}$ for $\mathbf{H}_{ij}^{\infty}(\mathbf{a})$ when context makes the intended meaning clear.

Note that ϕ_{ij} represents a necessary but not sufficient condition on legal $\mathbf{a}$; many solutions to ϕ_{ij} do not satisfy the modulus constraint. However, $\hat{\mathbf{a}}$ has the property $\phi_{ij}(\hat{\mathbf{a}}) = 0$ for *all* pairs (i, j). Thus $\hat{\mathbf{a}}$ is given by

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}\in\Re^3} \sum_{(i,j)} |\phi_{ij}(\mathbf{a})| \tag{5}$$

If m is large enough ($m \geq 4$), Eq. 5 has a single solution except for some special collections of views called *critical surfaces* (e.g., see [5,15]). When $m = 3$ there are a finite number of solutions [12]. Eq. 5 can be solved using nonlinear minimization techniques.

2.1 Modulus-constraint manifolds

Define the *modulus-constraint manifold* for the view pair (i, j) as

$$M_{ij} = \{\mathbf{a} \in \Re^3 : \text{all eigenvalues of } \mathbf{H}_{ij}^{\infty}(\mathbf{a}) \text{ have the same modulus}\} \tag{6}$$

We demonstrate in Section 3.1 that M_{ij} is indeed a (2-dimensional) manifold by providing a (piecewise) continuous mapping from $\Re^2$ to M_{ij}. See Fig. 1 for an idea of what the various M_{ij} look like.

In terms of modulus-constraint manifolds, $\hat{\mathbf{a}}$ is contained in the intersection of all the M_{ij}. Assuming m is large enough, this intersection point is unique:

$$\{\hat{\mathbf{a}}\} = \bigcap_{i<j} M_{ij} \tag{7}$$

Finally, note that M_{ij} is a proper subset of $\{\mathbf{a} \in \Re^3 : \phi_{ij}(\mathbf{a}) = 0\}$ and thus ϕ_{ij} cannot define the modulus-constraint manifold.

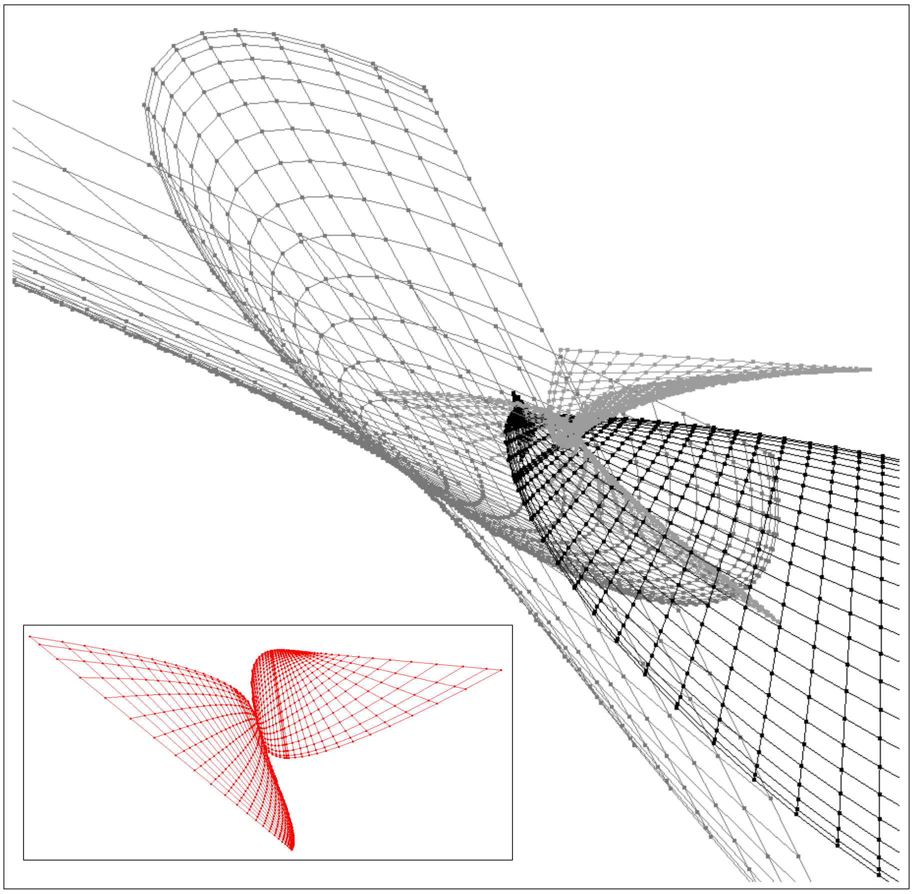

Fig. 1. Three modulus-constraint manifolds sketched in a-space; their mutual intersection point is $\hat{\mathbf{a}}$. The grids demonstrate the underlying coordinates (κ, θ). The inset shows a lone manifold with its single point of discontinuity (see Section 3.2).

3 Screw-transform manifolds in a-space

In this section we show how a modulus-constraint manifold can be viewed as a type of screw-transform manifold. Since screw-transform manifolds can be generated explicitly (from their underlying parameterizations), this represents an alternative method for determining $\hat{\mathbf{a}}$: generate the manifolds M_{ij} explicitly and find their mutual intersection point to get $\hat{\mathbf{a}}$ as in Eq. 7.

Screw-transform manifolds were first described in the context of direct self calibration. They are based on the observation that, when a camera with fixed internal parameters captures views from two different positions and orientations, the transformation that the camera undergoes between views can be decomposed as a screw transformation. A *screw transformation* involves a single rotation around a fixed axis in space followed by a single translation parallel to the axis

ALGORITHM

(1) If $\mathbf{F}_{ij}$ is a cross-product matrix, the motion is pure translational. Projective reconstruction can be directly upgraded to affine allowing $\hat{\mathbf{a}}$ to be determined immediately.

(2) Otherwise, use (κ, θ) to find $\underline{\mathbf{h}}_3$ and $\mathbf{l}_{12} \cong \underline{\mathbf{h}}_1 \times \underline{\mathbf{h}}_2$. If $\det(\mathbf{F}_{ij} + \mathbf{F}_{ij}^\top) = 0$ (or is sufficiently small), the motion is turntable and the algorithm in Fig. 4 is used. Otherwise, the motion is general and the algorithm in Fig. 3 is used.

(3) Use the algorithm in Fig. 5 to find $\mathbf{H}_{ij}^\infty$ from $\underline{\mathbf{h}}_3$ and $\mathbf{l}_{12}$.

(4) Use the algorithm in Fig. 6 to map $\mathbf{H}_{ij}^\infty$ to its corresponding position in a-space.

Fig. 2. Algorithm for mapping (κ, θ) into M_{ij}.

(in fact, the translation and rotation can be done in either order and still yield the same transformation). The axis is called the *screw axis*.

Let $\mathbf{F}_{ij}$ denote the fundamental matrix between views i and j. By picking a world coordinate system based on the screw transformation between views i and j, it is possible to find a legal $\mathbf{H}_{ij}^\infty$ (i.e., one that is conjugate to a rotation matrix) from $\mathbf{F}_{ij}$ by picking two real numbers κ and θ. In this manner, each pair $(\kappa, \theta) \in \Re^2$ can be mapped to an $\mathbf{a} \in \Re^3$ by using Eq. 3 or Eq. 4 (depending on i and j). This is what makes M_{ij} a manifold: it is an image of $\Re^2$ embedded in $\Re^3$ (see Fig. 1). In this context, we refer to $\Re^3$ as *a-space* to emphasize that this is the search space for the desired value of $\mathbf{a}$ denoted $\hat{\mathbf{a}}$. It is in a-space that the modulus-constraint manifolds live.

In the case of general pairwise motion, the mathematics for mapping $\Re^2$ to $\mathbf{H}_{ij}^\infty$ is given in [10] and is also summarized below. An algorithm for mapping $\mathbf{H}_{ij}^\infty$ to its corresponding position in a-space is given here for the first time. The complete case of turntable motion (i.e., pairwise planar motion) is covered here for the first time. The only other case, pure translational motion, does not generate a screw-transform manifold; however, the existence of pure translational motion between any two views allows the projective reconstruction to be immediately upgraded to affine without first determining $\hat{\mathbf{a}}$.

3.1 Algorithms for generating screw-transform manifolds

Screw-transform manifolds for general pairwise motion and for turntable motion are generated in different ways; both algorithms are given in this section. The two algorithms follow the same general approach: given the pairwise fundamental matrix $\mathbf{F}_{ij}$ and a pair of real numbers $(\kappa, \theta) \in \Re^2$, first determine which $\mathbf{H}_{ij}^\infty$ corresponds to the pair and then use either Eq. 3 or Eq. 4 to find the corresponding position in a-space. The image of $\Re^2$ under this mapping is the modulus-constraint manifold M_{ij} for the fundamental matrix $\mathbf{F}_{ij}$.

Recall that $\mathbf{H}_{ij}^\infty = \mathbf{KSK}^{-1}$ for some rotation matrix $\mathbf{S}$. The angle of rotation for $\mathbf{S}$ will correspond to θ. The scalar κ will be used to determine where the

Algorithm

(1) Let $\mathbf{M}$ be any invertible 3×3 matrix such that $\mathbf{F} = [\mathbf{e}]_\times \mathbf{M}$, where $\mathbf{e}$ is the left epipole of $\mathbf{F}$ (i.e., $\mathbf{e}^\top \mathbf{F} = 0$).
(2) Let $\underline{\mathbf{h}}_3 = (\kappa \mathbf{I} - \mathbf{M})^{-1}\mathbf{e}$.
(3) Let $\underline{\mathbf{h}}_1 = (\mathbf{F}^{\mathsf{S}}\mathbf{m}) \times (\mathbf{F}^{\mathsf{S}}\underline{\mathbf{h}}_3)$, where $[\mathbf{m}]_\times = \mathbf{F}^{\mathsf{A}}$.
(4) Find the unique null eigenvector $(\sigma_1, \sigma_2)^\top$ of $\left[\mathbf{F}^{\mathsf{S}}\underline{\mathbf{h}}_1,\ -\mathbf{F}^{\mathsf{A}}\underline{\mathbf{h}}_3\right]$. Note that $(\sigma_1, \sigma_2)^\top = \phi(1/s_1, \gamma/s_3)^\top$, where $s_1\mathbf{h}_1 = \underline{\mathbf{h}}_1$, $s_3\mathbf{h}_3 = \underline{\mathbf{h}}_3$, and ϕ is an unknown scalar determined in step (5).
(5) Solve the overdetermined system $\phi\mathbf{F}^{\mathsf{S}}\underline{\mathbf{h}}_3 = \sigma_1(1 - \cos\theta)(\underline{\mathbf{h}}_1 \times \underline{\mathbf{h}}_3)$ for ϕ.
(6) Let $\mathbf{h}_2 = (\phi\mathbf{m} - \sigma_2 \cos\theta \underline{\mathbf{h}}_3)/(\phi \sin\theta)$.

Fig. 3. Initial steps for generating the modulus-constraint manifold in the case of general motion.

Algorithm

(1) Let $\mathbf{m}$ be given by $[\mathbf{m}]_\times = (\mathbf{F} - \mathbf{F}^\top)/2 = \mathbf{F}^{\mathsf{A}}$.
(2) Let $\mathbf{h}_2 = \mathbf{m}/\sin\theta$.
(3) Let $\mathbf{l}_{12} = -(\mathbf{F}^{\mathsf{S}}\mathbf{m})/((1 - \cos\theta)\sin\theta)$.
(4) Use $\mathbf{F}^{\mathsf{S}}\underline{\mathbf{h}}_1 = 0$ to find $\underline{\mathbf{h}}_1$ (i.e., find the null eigenvector of $\mathbf{F}^{\mathsf{S}}$).
(5) Let $\mathbf{h}_1 = (\|\mathbf{l}_{12}\|/\|\underline{\mathbf{h}}_1 \times \mathbf{h}_2\|)\underline{\mathbf{h}}_1$.
(6) Find $\mathbf{l}_{13}$ from the fact that $(\mathbf{F}^{\mathsf{S}} - (1 - \cos\theta)[\mathbf{h}_1]_\times)$ is $[\mathbf{l}_{13}\ \mathbf{l}_{13}\ \mathbf{l}_{13}]^\top$ up to arbitrary scale factors on the rows.
(7) Let $\underline{\mathbf{h}}_3 = \mathbf{R}(\mathbf{l}_{13}, k)\mathbf{h}_1$ where $\mathbf{R}(\mathbf{l}_{13}, \kappa)$ denotes the rotation matrix with rotation axis $\mathbf{l}_{13}$ and angle of rotation $2\pi\kappa$.

Fig. 4. Initial steps for generating the modulus-constraint manifold in the case of turntable motion.

vanishing point of the screw axis appears in the first view; this vanishing point is the unique eigenvector of $\mathbf{H}_{ij}^\infty$ that has a real eigenvalue. Thus each distinct pair (κ, θ) will lead to a distinct $\mathbf{H}_{ij}^\infty$ and the mapping will be injective.

Since θ is an angle of rotation, θ can be considered a real number between 0 and 1 (just multiply by 2π). Considering angles outside this range would be redundant. This fact is very useful when searching for $\hat{\mathbf{a}}$ in a-space. Similarly, it is discussed in Section 4 how κ can also be considered a real number in the range $[0, 1]$. Thus the modulus-constraint manifold can be considered the image of $[0, 1] \times [0, 1]$ instead of being the image of all of $\Re^2$.

We now introduce some notation and formulas that will be used in the algorithms and proofs of this section. Consider a pair of views for which we want to find the screw-transform manifold. The camera used to capture the views must

Algorithm

(1) Let $\mathbf{M}$ be any invertible 3×3 matrix such that $\mathbf{F} = [\mathbf{e}]_\times \mathbf{M}$, where $\mathbf{e}$ is the left epipole of $\mathbf{F}$ (i.e., $\mathbf{e}^\top \mathbf{F} = 0$).

(2) Let $\xi = \mathbf{e}^\top \mathbf{l}_{12}$.

(3) Solve the following 6×5 system to find the null eigenvector $(\lambda, \lambda\mathbf{a}, 1)^\top$, where $\mathbf{q} = \mathbf{M}^\top \mathbf{l}_{12}$:

$$\begin{bmatrix} \mathbf{M}_{(11)} + \mathbf{M}_{(22)} + \mathbf{M}_{(33)} & \mathbf{e}^\top & -(1 + 2\cos\theta) \\ \mathbf{M}\underline{\mathbf{h}}_3 & \mathbf{e}\underline{\mathbf{h}}_3^\top & -\underline{\mathbf{h}}_3 \\ (\mathbf{l}_{12})_x \mathbf{q}_y - (\mathbf{l}_{12})_y \mathbf{q}_x & (-(\mathbf{l}_{12})_y \xi, (\mathbf{l}_{12})_x \xi, 0) & 0 \\ (\mathbf{l}_{12})_x \mathbf{q}_z - (\mathbf{l}_{12})_z \mathbf{q}_x & (-(\mathbf{l}_{12})_z \xi, 0, (\mathbf{l}_{12})_x \xi) & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \lambda\mathbf{a} \\ 1 \end{bmatrix} = \mathbf{0}$$

(4) Having determined $\mathbf{a} \in \Re^3$ in step (3), find $\mathbf{H}^\infty$ using

$$\mathbf{H}^\infty \cong \mathbf{M} + \mathbf{e}\mathbf{a}^\top \tag{8}$$

Fig. 5. Algorithm for determining $\mathbf{H}^\infty$ from $\mathbf{F}$, θ, $\mathbf{l}_{12}$, and $\underline{\mathbf{h}}_3$.

have fixed internal calibration $\mathbf{K}$ for the manifold to be defined. Assume that the fundamental matrix $\mathbf{F}$ between the pair of views has already been calculated.

There exists a screw transformation of Euclidean space that takes the first camera to the second one. Let θ denote the angle of rotation in the screw transformation and let γ denote the amount of translation parallel to the screw axis. Choose the Euclidean reference frame so that the screw axis is the z-axis and the optical center of the first view is at position $(1,0,0)^\top \in \Re^3$. Let $\mathbf{R}$ denote the tilt of the first camera relative to this frame of reference.

Let $\mathbf{H} = \mathbf{KR}$ and let $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3 \in \Re^3$ denote the column vectors of $\mathbf{H}$, so that $\mathbf{H} = [\, \mathbf{h}_1 \; \mathbf{h}_2 \; \mathbf{h}_3 \,]$. It was shown in [10] that $\mathbf{F}$ has the form:

$$\mathbf{F} = \mathbf{F}^{\mathsf{A}} + \mathbf{F}^{\mathsf{S}} \tag{9}$$

$$\mathbf{F}^{\mathsf{A}} = [\sin\theta \mathbf{h}_2 + \gamma \cos\theta \mathbf{h}_3]_\times \tag{10}$$

$$\begin{aligned}\mathbf{F}^{\mathsf{S}} = \; & \frac{1-\cos\theta}{|\mathbf{H}|} \left[(\mathbf{h}_1 \times \mathbf{h}_3)(\mathbf{h}_1 \times \mathbf{h}_2)^\top + (\mathbf{h}_1 \times \mathbf{h}_2)(\mathbf{h}_1 \times \mathbf{h}_3)^\top \right] + \\ & \frac{\gamma \sin\theta}{|\mathbf{H}|} \left[(\mathbf{h}_1 \times \mathbf{h}_3)(\mathbf{h}_1 \times \mathbf{h}_3)^\top + (\mathbf{h}_2 \times \mathbf{h}_3)(\mathbf{h}_2 \times \mathbf{h}_3)^\top \right]\end{aligned} \tag{11}$$

Observe that $\mathbf{F}^{\mathsf{S}} = (\mathbf{F} + \mathbf{F}^\top)/2$ and $\mathbf{F}^{\mathsf{A}} = (\mathbf{F} - \mathbf{F}^\top)/2$ because $\mathbf{F}^{\mathsf{S}}$ is symmetric and $\mathbf{F}^{\mathsf{A}}$ is antisymmetric, so that these two matrices can be calculated directly from $\mathbf{F}$ without knowing any of the other terms.

We will refer to the $\mathbf{h}_i$ extensively in the algorithms and will also use the notation $\underline{\mathbf{h}}_i$. Finding $\underline{\mathbf{h}}_i$ means finding $\mathbf{h}_i$ up to an unknown scale factor. It is typical in the algorithms to find $\underline{\mathbf{h}}_i$ and then determine the unknown scale factor to get $\mathbf{h}_i$. Finding $\underline{\mathbf{h}}_i$ by itself can be useful because it dictates where $\mathbf{h}_i$ projects onto the image plane; $\underline{\mathbf{h}}_i$ can be thought of as the direction or image of $\mathbf{h}_i$.

ALGORITHM

(1) If $i = 1$, then Eq. 3 can be used to find $\mathbf{a}$. For instance, let $\mathbf{E}_1 = [\mathbf{e}_j\ \mathbf{0}\ \mathbf{0}]$, $\mathbf{E}_2 = [\mathbf{0}\ \mathbf{e}_j\ \mathbf{0}]$, and $\mathbf{E}_3 = [\mathbf{0}\ \mathbf{0}\ \mathbf{e}_j]$, then solve

$$[\, \mathbf{H}^{\infty}_{1j}\ \mathbf{H}_j\ \mathbf{E}_1\ \mathbf{E}_2\ \mathbf{E}_3 \,][\, 1\ \sigma\ \mathbf{a}^x\ \mathbf{a}^y\ \mathbf{a}^z \,]^{\top} = \mathbf{0}.$$

where the 3×3 matrices ($\mathbf{H}^{\infty}_{1j}$, $\mathbf{H}_j$, etc.) are treated as column vectors in $\Re^9$. Since the null eigenvector will be found up to a scale factor, divide by its first component to get the correct $\mathbf{a} = (\mathbf{a}^x, \mathbf{a}^y, \mathbf{a}^z)$. The algorithm is done.

(2) Otherwise $i \neq 1$ and Eq. 4 must be solved for $\mathbf{a}$; the remaining steps of the algorithm are for this task.

(3) Define $\mathbf{q}_i$ and $\mathbf{m}_i$ by $[\, \mathbf{q}_1\ \mathbf{q}_2\ \mathbf{q}_3 \,] = \mathbf{H}^{\infty}_{ij}\mathbf{H}_i$ and $[\, \mathbf{m}_1\ \mathbf{m}_2\ \mathbf{m}_3 \,] = \mathbf{H}_j$. Let $\mathbf{v}_1 = \mathbf{m}_1 \times \mathbf{m}_2$, $\mathbf{v}_2 = \mathbf{q}_1 \times \mathbf{m}_2 + \mathbf{m}_1 \times \mathbf{q}_2$, and $\mathbf{v}_3 = \mathbf{q}_1 \times \mathbf{q}_2$.

(4) Solve $\phi^2\mathbf{v}_1 + \phi\mathbf{v}_2 + \mathbf{v}_3 = \mathbf{0}$ for the scalar ϕ. One closed-form solution is $\phi = -(\mathbf{v}^x_3\mathbf{v}^y_1 - \mathbf{v}^y_3\mathbf{v}^x_1)/(\mathbf{v}^x_2\mathbf{v}^y_1 - \mathbf{v}^y_2\mathbf{v}^x_1)$. A least-squares approach is preferable.

(5) The equation to solve is now $\mathbf{U} + \mathbf{w}\mathbf{a}^{\top} = \mathbf{0}$, where $\mathbf{U} = \mathbf{H}^{\infty}_{ij}\mathbf{H}_i + \phi\mathbf{H}_j$ and $\mathbf{w} = \mathbf{H}^{\infty}_{ij}\mathbf{e}_i + \phi\mathbf{e}_j$. This can be solved as in step (1):

$$[\, \mathbf{U}\ \mathbf{W}_1\ \mathbf{W}_2\ \mathbf{W}_3 \,][\, 1\ \mathbf{a}^x\ \mathbf{a}^y\ \mathbf{a}^z \,]^{\top} = \mathbf{0}.$$

Fig. 6. Algorithm for determining $\mathbf{a}$ from $\mathbf{H}^{\infty}_{ij}$ for view pair (i, j) with $i < j$.

The first task in mapping a pair (κ, θ) to its corresponding position on the modulus-constraint manifold M_{ij} is to determine if the screw transformation between views i and j is a pure translation, a pure rotation, or a general motion. General motion is any case that is not one of the first two, and consists of both a translation and a rotation. If $\mathbf{F}_{ij}$ is a cross-product matrix (i.e., antisymmetric with 0's on the main diagonal), then the motion is pure translation and no screw-transform manifold exists. If $\det(\mathbf{F}^{\mathsf{S}}) = 0$ then the motion is pure rotation (see Theorem 1 in Appendix A). Otherwise the motion is general.

The algorithm for mapping (κ, θ) into M_{ij} is given in Fig. 2. Remember that the values θ and κ in the algorithms are chosen arbitrarily from the domain; every pair leads to a distinct, legal $\mathbf{H}^{\infty}$ and thus a distinct position on the modulus-constraint manifold. The "true" values of θ and κ (those corresponding to $\hat{\mathbf{a}}$) can only be determined by intersecting several different modulus-constraint manifolds to find $\hat{\mathbf{a}}$. If, for instance, θ were known a priori for some pairwise transformation, then only κ would be chosen arbitrarily and the modulus-constraint manifold would be one-dimensional, helping to narrow the search for $\hat{\mathbf{a}}$. If θ could be restricted to a known range (e.g., if it is known that θ is small), then the manifold might be a narrow ribbon, again helping to limit the search space.

3.2 Notes on the algorithms

The algorithms in Fig. 5 and Fig. 3 were discussed in [10]. Observe that in step (2) of Fig. 3, if κ is the real eigenvalue of $\mathbf{M}$ then $\underline{\mathbf{h}}_3$ is not defined, leading to the point of discontinuity noted in Fig. 1. The algorithm of Fig. 6 follows from Eq. 3 and Eq. 4 using straight-forward linear algebra. However, note that Eq. 3 and Eq. 4 only define $\mathbf{H}_{ij}^{\infty}$ up to a scalar; in the case where $i \neq 1$ (steps (2) through (5)), Eq. 4 is first written $\mathbf{H}_{ij}^{\infty}(\mathbf{H}_i + \mathbf{e}_i\mathbf{a}^\top) = -\phi(\mathbf{H}_j + \mathbf{e}_j\mathbf{a}^\top)^{-1}$ and then ϕ is determined before finding $\mathbf{a}$.

In the algorithm of Fig. 4, step (2) follows immediately from Eq. 10 because $\gamma = 0$ (for turntable motion). Step (3) requires the knowledge that $|\mathbf{H}| := \det(\mathbf{H}) = \mathbf{h}_1 \cdot \mathbf{h}_2 \times \mathbf{h}_3 = \mathbf{h}_1 \times \mathbf{h}_2 \cdot \mathbf{h}_3$. The line $\mathbf{l}_{12}$ through $\underline{\mathbf{h}}_1$ and $\underline{\mathbf{h}}_2$ (as seen on the image plane) is $\underline{\mathbf{h}}_1 \times \underline{\mathbf{h}}_2$; scale does not matter unless this vector is used to find $\mathbf{h}_1$ or $\mathbf{h}_2$, as it is in step (5). Step (4) follows from Eq. 11 with $\gamma = 0$ and the knowledge that $\mathbf{F}^{\mathsf{S}}$ is rank 2. Step (6) is the most interesting of the algorithm: Let $\mathbf{Q} = \mathbf{F}^{\mathsf{S}} - (1 - \cos\theta)[\mathbf{h}_1]_\times$ and observe that $\mathbf{Q}\mathbf{h}_1 = \mathbf{0}$ (from step (4)) and $\mathbf{Q}\mathbf{h}_3 = \mathbf{0}$ (from Eq. 11). Thus $\mathbf{Q}$ annihilates any linear combination of $\mathbf{h}_1$ and $\mathbf{h}_3$, giving $\mathbf{Q}$ the form $[\,\mathbf{l}_{13}\;\mathbf{l}_{13}\;\mathbf{l}_{13}\,]^\top$ up to arbitrary scale factors on the rows.

4 Creating a working self-calibration algorithm

For the experiments of Section 5, we used the voting algorithm described in [10] to find the intersection point $\hat{\mathbf{a}}$ of the modulus-constraint manifolds M_{ij}, although other algorithms could conceivably be used.

It is important to realize that the equation $\underline{\mathbf{h}}_3 = (\kappa\mathbf{I} - \mathbf{M})^{-1}\mathbf{e}$ used to find $\underline{\mathbf{h}}_3$ in step (1) of Fig. 3 can represent an ill-conditioned system. This means that, under the right conditions, small changes in κ will lead to large changes in $\underline{\mathbf{h}}_3$. We now discuss how to condition the system.

Note that $\underline{\mathbf{h}}_3$ is the image of the vanishing point of the screw axis. If one considers the viewing sphere around a camera's optical center, then the possible locations of $\underline{\mathbf{h}}_3$ form a one-dimensional manifold on the surface of the sphere. As κ goes towards infinity, $\underline{\mathbf{h}}_3$ approaches $\mathbf{e}$ from one direction along this manifold ($\mathbf{e}$ is also on the manifold) and as κ goes towards negative infinity, $\underline{\mathbf{h}}_3$ approaches $\mathbf{e}$ from the other direction. Once κ gets large enough (e.g., $|\kappa| > 10$), $\underline{\mathbf{h}}_3$ is very close to $\mathbf{e}$ and stops changing in any meaningful way. Thus we can assume $\kappa \in [-\mu, \mu]$ for some fixed, sufficiently-large μ. This means we can treat κ as being a real number in $[0, 1]$, which then gets mapped into $[-\mu, \mu]$. In the turntable motion algorithm of Fig. 4, κ is used in step (7) and is already assumed to be in the range $[0, 1]$.

The observation that κ has a finite range provides a way to condition the equation for $\underline{\mathbf{h}}_3$. Before beginning the search for $\hat{\mathbf{a}}$, establish a map from $[0, 1]$ to $[-\mu, \mu]$ that produces $\underline{\mathbf{h}}_3$ at approximately regularly-spaced intervals along the one-dimensional manifold on the viewing sphere. The preconditioning map in our implementation sends 40 equally-spaced "guide" numbers in [0,1] to 40 numbers in $[-\mu, \mu]$ that yield evenly-spaced $\underline{\mathbf{h}}_3$ in $\Re^3$. The remaining members of $[0, 1]$ are mapped by linearly interpolating the image of the nearest two guides.

A second consideration is that, since a-space depends on the choice of initial projective reconstruction, it is possible for a-space to be so skewed that all the modulus-constraint manifolds are approximately parallel, thus making it impossible to determine their mutual intersection point $\hat{\mathbf{a}}$. This is an issue for any

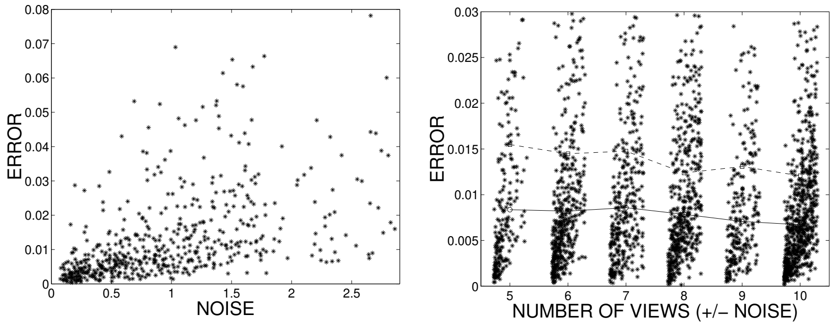

Fig. 7. Graphs from synthetic-data experiments.

stratified calibration algorithm, but using screw-transform manifolds provides a way out of the problem.

Because screw-transform manifolds are explicitly defined rather than implicitly, it is possible to generate positions on each manifold for the entire feasible range of θ and κ values. Once each manifold is approximately sketched out, a-space can be stretched using principal-component analysis to introduce greater separation between the manifolds. When using the voting algorithm, a-space needs to be renormalized in this manner after each zoom-in step, using only those manifold points that lie in the current, reduced search region.

5 Experimental results

There is an inherent difficulty in presenting self-calibration experiments. The difficulty arises because the accuracy of the self-calibration algorithm is highly dependent on the performance of a variety of external modules. External modules are required for feature extraction, feature matching or tracking, fundamental matrix calculation, creating an initial projective reconstruction, and finding dense, pairwise image correspondences to produce detailed metric reconstructions.

Our goal in presenting experiments is to demonstrate general trends (in the case of experiments with synthetic data) and to demonstrate the feasibility of our algorithm (in the case of experiments with real cameras and views).

For determining fundamental matrices, we use the basic normalized algorithm described by Hartley combined with a RANSAC technique. There are more sophisticated algorithms for determining fundamental matrices and presumably these would improve self calibration. For creating the initial projective reconstruction, we use a simple exhaustive search to find a “good” projective reconstruction (i.e., one that is not flat and elongated). We have not attempted to recreate the sophisticated algorithm of Beardsley et al. [1] that allows for extended projective scene reconstructions nor have we recreated the excellent dense-correspondence algorithm of Koch [9].

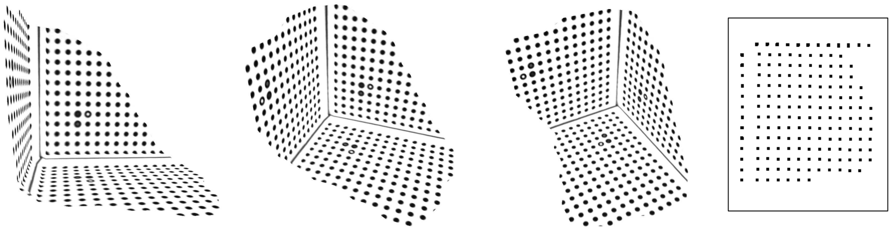

Fig. 8. Texture-mapped reconstructions of a calibration grid. On the right is an orthographic overhead view of the reconstructed feature points only.

5.1 Synthetic-view experiments

We have chosen to demonstrate two trends using synthetic-data experiments. First is the relationship between noise in feature locations and error in calibration; the graph on the left side of Fig. 7 shows the results. Each point on the graph represents a single synthetic data trial. Each trial consisted of a calibration-grid-like object (with 108 features) positioned at the origin and 6 cameras located randomly in the first octant viewing the object. For each trial, a new internal calibration was generated randomly using realistic parameters. Noise was added to the projected feature points on each camera's image plane with a uniform distribution. The maximum radius of the uniform noise distribution was varied randomly between trials and is shown on the x-axis of the graph. For an error metric, we used

$$\textsc{error}(\mathbf{K}, \mathbf{M}) = \textsc{frob}(\mathbf{K}/\textsc{frob}(\mathbf{K}) - \mathbf{M}/\textsc{frob}(\mathbf{M}))$$

where $\textsc{frob}(\mathbf{M})$ is the Frobenius norm of matrix $\mathbf{M}$. The results show a general linear trend, with error approaching 0 as noise approaches 0.

The second experiment investigated how increasing the number of views could reduce calibration error. The general procedure used was the same as in the first experiment, and the results are shown on the right-hand side of Fig. 7. In the graph, each vertical cluster of trials corresponds to a particular number of views; the cluster around n corresponds to n views. The amount of noise added during each trial is indicated by positioning of the trial to the left or right of the integer number of views. Superimposed on the graph is a line connecting the mean error for each cluster. This line demonstrates a general downward trend showing that error is gradually reduced as the number of views is increased.

5.2 Real-view experiments

We provide two experiments that used real camera views. The first used a standard "calibration box" data set. No calibration information available from the box (e.g., known distance between points) was used in the experiment. The dot centers were extracted automatically and then matched automatically between views using special-colored dots for reference. Thus the calibration box was used

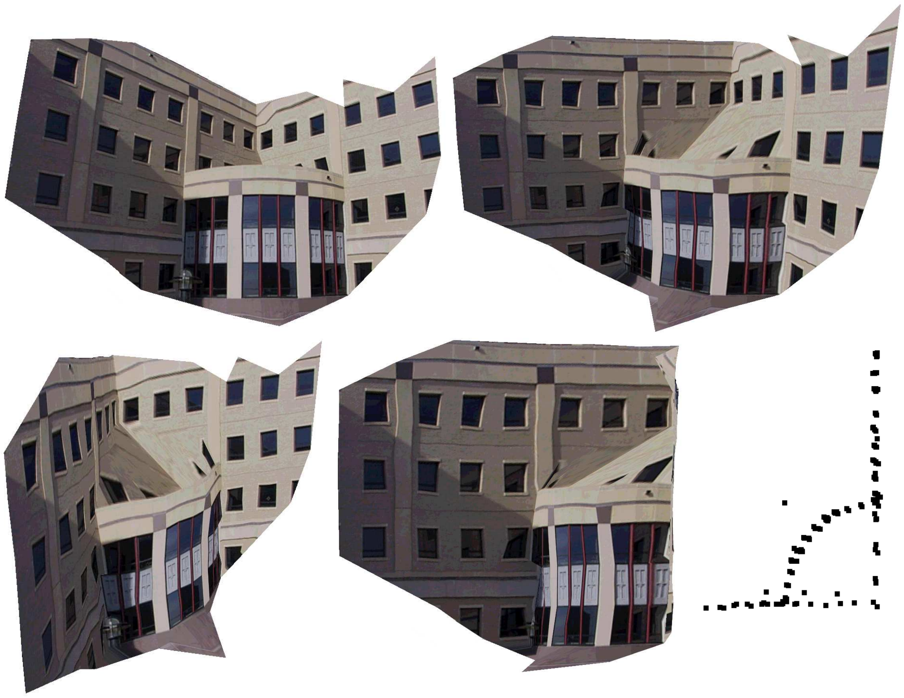

Fig. 9. Reconstruction of a building.

solely to obtain a set of highly-accurate feature correspondences between views. Self calibration was performed using six views and results are shown in Fig. 8. The reconstructed feature positions are near perfect: features on the same plane in the original scene are on the same plane in the reconstructed scene, features are equidistant in the reconstruction, and the three faces are orthogonal (see the right-hand side of Fig. 8 for an overhead view showing features only).

The purpose of using such exact data was to demonstrate that the underlying theory and models of the algorithm (e.g., pinhole-perspective projection) really do apply to actual cameras and that self calibration and reconstruction can be highly successful with real views provided the data is accurate.

The second experiment shows the reconstruction of a building. In the region under consideration, the building has two orthogonal walls and a quarter-circle annex nestled between the walls. Seven views were used in the reconstruction and features were chosen by hand, mostly at the corners of windows. About 150 features overall were used. Note that hand-picked features are only accurate to about 1 or 2 pixels at best (the original views were 1024×768 pixels in size).

The lower-right-hand corner of Fig. 9 shows an overhead view of the reconstructed scene as features only; note the orthogonal walls and circular annex. A texture-mapped reconstruction was created using Delauny triangulation of one view; this is shown in the rest of Fig. 9. The texture is highly accurate when seen from near the original source view but is of course inaccurate in regions of the building that have not been correctly reconstructed (e.g., the roof of the circular annex, which was not visible from the ground).

6 Discussion

In this paper, we have shown how to generate the modulus-constraint manifold in a-space using the mathematics of screw-transform manifolds and demonstrated the use of these ideas in a new stratified self-calibration algorithm. In both the case of general motion and turntable motion, a (κ, θ) pair is used to find $\mathbf{h}_1$, $\mathbf{h}_2$, and $\underline{\mathbf{h}}_3$, which are then used to locate a position in a-space. However, by using a third parameter γ along with κ and θ we can find $\mathbf{h}_3$ from $\underline{\mathbf{h}}_3$, and thus we can find $[\, \mathbf{h}_1 \;\, \mathbf{h}_2 \;\, \mathbf{h}_3 \,] = \mathbf{H} = \mathbf{KR}$. Since $\mathbf{HH}^\top = \mathbf{KK}^\top$, finding $\mathbf{H}$ would place direct constraints on the internal calibration matrix $\mathbf{K}$, and this is the principle behind the direct self-calibration method of [10].

Thus by using two parameters κ and θ we can generate the two-dimensional modulus-constraint manifold in three-dimensional a-space, and by using a third parameter γ we can generate a three-dimensional Kruppa-constraint manifold in five-dimensional K-space. These two manifolds are intimately related, but the first technique allows for stratified self calibration and the second for direct self calibration.

A Discerning turntable motion

If the amount γ of translation parallel to the screw axis is 0, then the pairwise camera motion is equivalent to a stationary camera viewing a rotating turntable. Alternatively, this kind of motion can be seen as translation and rotation of the camera parallel to a fixed plane in space (e.g., a camera on a tripod which is kept at constant height and repositioned on a level surface).

There is a simple test to determine directly from the pairwise fundamental matrix $\mathbf{F}$ whether or not $\gamma = 0$:

Theorem 1. *When θ is not a multiple of π, $\gamma = 0$ if and only if* $\det(\mathbf{F} + \mathbf{F}^\top) = 0$.

Proof. Note that the theorem involves the determinant of $\mathbf{F}^{\mathsf{S}}$. Call the two cameras involved camera A and camera B. Let $\mathbf{e}_A$ and $\mathbf{e}_B$ denote the epipole in camera A and camera B, respectively. Clearly if $\gamma = 0$ then $\det(\mathbf{F}^{\mathsf{S}}) = 0$ because $\mathbf{h}_1$ is a null eigenvector of $\mathbf{F}^{\mathsf{S}}$ (examine Eq. 11). Now assume $\det(\mathbf{F}^{\mathsf{S}}) = 0$ but $\gamma \neq 0$. In the logic below we will be using a fixed-camera, rising-turntable formulation [10] with the optical center at the origin of $\Re^3$. Also, for vectors $\mathbf{f}, \mathbf{g} \in \Re^3$, $< \mathbf{f}, \mathbf{g} >$ will denote both the space spanned by $\mathbf{f}$ and $\mathbf{g}$ and the line on the image plane induced by this space. Let $\mathbf{u} \in \Re^3$ be a scene point whose projection into camera B, denoted by $\mathbf{u}_B$, is in the null space of $\mathbf{F}^{\mathsf{S}}$ (i.e., $\mathbf{F}^{\mathsf{S}}\mathbf{u}_B = \mathbf{0}$). Using Eq. 11, $\mathbf{F}^{\mathsf{S}}\mathbf{h}_3 \cong \mathbf{h}_1 \times \mathbf{h}_3 \neq \mathbf{0}$ so $\mathbf{h}_3$ corresponds to a different position in view B than $\mathbf{u}_B$. Since the epipolar line for $\mathbf{u}_A$ is represented by both $< \mathbf{u}_A, \mathbf{e}_A >$ and by $\mathbf{u}_B^\top \mathbf{F} = \mathbf{u}_B^\top \mathbf{F}^{\mathsf{S}} + \mathbf{u}_B^\top \mathbf{F}^{\mathsf{A}} = \mathbf{u}_B^\top \mathbf{F}^{\mathsf{A}} = (\mathbf{m} \times \mathbf{u}_B)^\top$, we conclude $\mathbf{m}$, $\mathbf{u}_B$, $\mathbf{u}_A$, and $\mathbf{e}_A$ are all coplanar. Let $\mathbf{q} \in \Re^3$ be an arbitrary scene point that projects to the line $< \mathbf{u}_B, \mathbf{e}_A >$ in view A. So $\mathbf{q}_A$ is a linear combination of $\mathbf{u}_B$ and $\mathbf{e}_A$ and thus $\mathbf{Fq}_A \cong \mathbf{Fu}_B \cong \mathbf{m} \times \mathbf{u}_B$, implying $\mathbf{q}_B$ lies in the plane $< \mathbf{m}, \mathbf{u}_B >$ which is also the plane $< \mathbf{u}_A, \mathbf{e}_A >$. In other words, points visible on the line $< \mathbf{u}_A, \mathbf{e}_A >$ in view A are also on that line in view B (after the screw transformation). If the planes $< \mathbf{u}_A, \mathbf{e}_A >$ and $< \mathbf{h}_1, \mathbf{h}_3 >$ are identical then $\mathbf{e}_A$ lies on the rotation axis $< \mathbf{h}_1, \mathbf{h}_3 >$, which can only happen if θ is a multiple of π. Since we assume this is not the case, $< \mathbf{u}_A, \mathbf{e}_A >$ and $< \mathbf{h}_1, \mathbf{h}_3 >$ represent distinct lines in view A which intersect at a point $\mathbf{v}_A$. For some scale factor k, $\mathbf{v} = k\mathbf{v}_A$ is a point on the rotation axis in space. $\mathbf{v}_B$ is the projection of $\mathbf{v}$ after the screw motion; in this case, the screw motion translates $\mathbf{v}$ by γ along the screw axis, so under the fixed-camera formulation $\mathbf{v}_B$ and $\mathbf{v}_A$ are at different positions on the axis line $< \mathbf{h}_1, \mathbf{h}_3 >$. But $\mathbf{v}_A$ is on the line $< \mathbf{u}_A, \mathbf{e}_A >$ and so $\mathbf{v}_B$ must also be on this line, leading to the conclusion that $\mathbf{v}_B = \mathbf{v}_A$, a contradiction. $\square$

References

1. P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model aquisition from extended image sequence. In *Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge*, pages 683–695, 1996.
2. F. Devernay and O. D. Faugeras. From projective to euclidean reconstruction. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 264–269, 1996.
3. O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In *Proc. Second European Conf. on Computer Vision*, pages 563–578, 1992.
4. O. D. Faugeras. Stratification of 3-dimensional vision: Projective, affine, and metric representations. *Journal of the Optical Society of America*, 12(3):465–484, 1995.
5. O. D. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images.* The MIT Press, Cambridge, Massachusetts, 2001.
6. O. D. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self-calibration: theory and experiments. In *Proc. European Conf. on Computer Vision*, pages 321–334, 1992.
7. R. Hartley. Self-calibration from multiple views with a rotating camera. In *Proc. Third European Conf. on Computer Vision*, pages 471–478, 1994.
8. R. Hartley and A. Zisserman. *Multiple View Geometry.* Cambridge University Press, New York, 2000.
9. R. Koch. *Automatische Oberflaechenmodellierung starrer dreidimensionaler Objekte aus stereoskopischen Rundum-Ansichten.* PhD thesis, Hannover, 1996.
10. R. Manning and C. Dyer. Metric self calibration from screw-transform manifolds. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 590–597, 2001.
11. M. Pollefeys. *Self-Calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences.* PhD thesis, Katholieke Universiteit Leuven, Belgium, 1999.
12. M. Pollefeys and L. Van Gool. A stratified approach to metric self-calibration. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 407–412, 1997.
13. M. Pollefeys and L. Van Gool. A stratified approach to metric self-calibration with the modulus constraint. Technical Report 9702, K. U. Leuven – ESAT-MI2, 1997.
14. M. Pollefeys, L. Van Gool, and A. Oosterlinck. The modulus constraint: A new constraint for self-calibration. In *Proc. Int. Conf. on Pattern Recognition*, pages 349–353, 1996.
15. P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 1100–1105, 1997.
16. W. Triggs. Autocalibration and the absolute quadric. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 609–614, 1997.
17. A. Zisserman, P. Beardsley, and I. Reid. Metric calibration of a stereo rig. In *IEEE Workshop on Representation of Visual Scenes, Boston*, pages 93–100, 1995.