Manufacturing Multiple View Constraints

David N. R. M^{c}Kinnon and Brian C. Lovell
{mckinnon,lovell}@itee.uq.edu.au
Intelligent Real-Time Imaging and Sensing (IRIS) Group, EMI
School of Information Technology and Electrical Engineering
University of Queensland
St. Lucia QLD 4072, Australia

Abstract

In this paper we present an algorithm for the generation of the multiple view constraints for arbitrary configurations of cameras and image features correspondences. Multiple view constraints are an important commodity in computer vision since they facilitate in determining camera locations using only the correspondences between common features observed in sets of uncalibrated images. We show that by a series of counting arguments and a systematic application of the principles of antisymmetric algebra it is possible to generate arbitrary multiple view constraints in a completely automated fashion. The algorithm has already been utilized to discover new sets of multiple view constraints for surfaces.

1. Introduction

Structure From Motion (SFM) is the process of calculating the structure of a scene observed by the motion of an uncalibrated camera/s simultaneous with the egomotion of the camera/s and their intrinsic calibration properties. Calculation of multiple view (a.k.a. multiview) constraints is a key component of SFM and is mandatory in order for a 3D/4D reconstruction to be achieved without apriori knowledge of the scene, camera's motion or calibration.

A precise understanding of the antisymmetric algebra underlying the multiview constraints is necessary in order for their utilization. Typically the implementor of SFM software would reference the exact algebra for these correspondences from resources such as [4, 1]. The most common multiview constraints are the 2-view (Fundamental Matrix) and 3-view (Trifocal Tensor) these utilize correspondences between sets of common points and/or lines observed in all the images. An example of the trifocal configuration for a point observed in 3-views is shown in Figure 1.

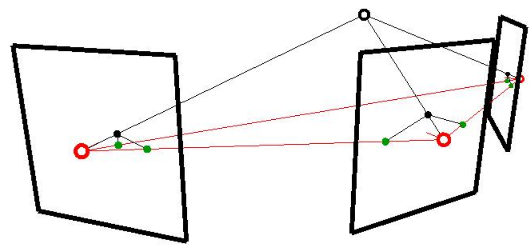

Figure 1. Trifocal Tensor point transfer.

In this paper we present an algorithm to determine the precise nature of multiview constraints for arbitrary combinations of cameras and feature correspondences. This is not only useful from a practical viewpoint but also from a theoretical one seeing as new multiview constraints can be generated in some instances by changing the inputs to the process. This approach to generating multiview constraints has been of central importance in the discovery and utilization of a new set of multiview constraints for degree-2 dual surfaces [6].

The development of this approach to determining multiview constraints relies upon the principles of tensor algebra in the style of [12] utilizing the concept of the tensor tableaux introduced in [7]. A rudimentary introduction to some of these concepts is presented in the proceeding section.

2 Tensor Basics

Tensors are a generalization of the concept of vectors and matrices. In this sense vectors and matrices are 1 and 2-dimensional instances of a tensor. Tensors are composed entirely from vector spaces. These vector spaces can be combined using a range of well defined operators resulting in differently structured tensors.

2.1 Vector Spaces

We will limit our study of the geometry herein to projective vector space $\mathbb{P}^n$. An element of an n-dimensional projective vector space in the tensor notation is denoted as $\mathbf{x}^{{}_m A_i^s} \in \mathbb{P}^n$. The symbol ${}_m A_i^s$ is called an indeterminant and identifies several important properties of the vector space. Firstly in order to better understand the notation we must rewrite $\mathbf{x}^A$ in the standard vector form. This is achieved by listing the elements of the vector space using the indeterminant as the variables of the expression. In this manner the symbol $(\mathbf{x})$ that adjoins the indeterminant is merely cosmetic. For example a tensor and the equivalent vector space can be defined as,

$$\mathbf{x}^{{}_m A_i^s} \equiv [{}_m A_0^s, {}_m A_1^s, \ldots, {}_m A_n^s]^\top \tag{1}$$

where m identifies the multilinearity of the indeterminant, s depicts the degree (or step) of the indeterminant. The last element describing the indeterminant is i, we most commonly refer to i as the *index* of the indeterminant. The index reflects a position within the vector space described by the indeterminant. We stress that the labeling of indexes for a given indeterminant is arbitrary but must remain consistent. The standard indexing is $i \in \{0 \ldots n\}$ for an n-dimensional projective vector space (lexicographic).

Indeterminants of a regular vector (vertical) space $(\mathbb{P}^n)$ are called *contravariant* and indeterminants of a dual (horizontal) vector space ${}^*\mathbb{P}^n$ are called *covariant*. The conventions of linear algebra refer to contravariant vector spaces as simply *vectors* and covariant vector spaces as *covectors*. The notation for a dual vector (covector) space is analogous to that for a regular vector space,

$$\mathbf{x}_{{}^*_m A_i^s} \equiv [{}^*_m A_0^s, {}^*_m A_1^s, \ldots, {}^*_m A_n^s] \tag{2}$$

the only difference is that the vector is transposed. In the interests of compactness and clarity often we will abandon the entire set of labels for an indeterminant via an initial set of assignments.

2.2 Tensor Products

The basic tools used to construct the algebraic/geometric entities in the tensor notation are called operators. There are three different types of operators that we use and for each operator we will maintain two differing representations. We refer to these different representations as the tensor form and the equivalent vector form (Table 1). In Table 1 the symbols $\nu_n^d = \binom{d+n}{d} - 1$, $\eta_n^k = \binom{n+1}{k} - 1$ and $\pi_n^d = \prod_{i=1}^d (n_i + 1) - 1$.

The two different forms of the tensor are representative of the fact that we can always rewrite a tensor expression as an ordered vector of it's unique coefficients. Writing a tensor as a vector of coefficients abandons any symmetry present in the tensor, resulting in a less fruitful representation for symbolic derivations since it limits the way in which a tensor expression can be contracted. The advantages of the vector representation of a tensor expression arise from a reduction in the redundancy created by the (anti)symmetry of the elements within a tensor resulting in a more efficient representation for mappings between vector spaces.

2.3 Tensor Tableaux

Tensor tableaux provide a tool that may be used for the description of tensor expressions. Tensor tableaux facilitate study of the precise composition of a tensor expression that may also be translated directly into an algorithm to compute a tensor expression from composite parts. In the following examples $\mathbf{x}^A, \mathbf{x}^B \in \mathbb{P}^2$.

The basic structure of the tensor tableaux is determined from the tensor expression itself. As a first example we present the tableaux for a Segre product (a.k.a. outer product) $\mathbf{x}^A \mathbf{y}^B \equiv \mathbf{z}^{AB}$ resulting in.

A	B	AB
0	0	00
0	1	01
0	2	02
1	0	10
1	1	11
1	2	12
2	0	20
2	1	21
2	2	22

We can see that the columns on the left of the tableaux are filled by the indeterminants of the expression that we wish to formulate and the column on the right is the result of the expression. The rule for building a minimal tableaux given an expression is to first write the result of the expression in the right column, indexing only the unique non-zero terms. Columns to the left of the result include the singular indeterminants (or composite terms) that compose each row of the result. Moving to another example for antisymmetric operations, $\mathbf{x}^{[A}\mathbf{y}^{B]} \equiv \mathbf{z}^{[AB]}$ results in the following tableaux.

A	B	AB
0	1	01
0	2	-02
1	2	12

In this example we see that the columns to the left of the result are the elements of a 2-step antisymmetric sequence in $\mathbb{P}^2$. The signs in the front of the result

Operator	Symbol	Tensor Form	Vector Form
Segre	-	$\mathbf{x}^{A_i \ldots B_j}$	$\mathbf{x}^{\alpha^d} \in \mathbb{P}^{\pi_n^d}$ where $\mathbf{x}^{A_i} \in \mathbb{P}^{n_i}$
Antisymmetric (Step-k)	$[\ldots]$	$\mathbf{x}^{[A_i \ldots B_j]}$	$\mathbf{x}^{\alpha^{[k]}} \in \mathbb{P}^{\eta_n^k}$
Symmteric (Degree-d)	$(\ldots)$	$\mathbf{x}^{(A_i \ldots A_j)}$	$\mathbf{x}^{\alpha^{(d)}} \in \mathbb{P}^{\nu_n^d}$

Table 1. Tensor Operators

indeterminants are derived according to the rules for antisymmetrization given in [7].

From a computational perspective the advantage of using the tableaux formulation is that the structure of complex sequences of tensor operations can be predetermined and reduced into a minimal sequence of multiplications and additions with simple array indexing. The sequence of terms displayed in each row of the tableaux are indexed such that they may be used as pointer offsets into arrays to calculate tensor expressions on a computer.

3 Multiple View Constraints

Multiview constraints can be utilized as a means to determine a projective estimate of the cameras location entirely from feature correspondences between a set of images. Due to this fact the utilization of multiview constraints forms the basis for structure recovery in SFM applications. Multiview constraints used in conjunction with robust statistics are critical in identifying and handling incorrectly tracked features in SFM applications [10, 9, 11, 4].

In the proceeding sections we present the theory relating to the multiview constraints for a set of views. Firstly, we introduce the concept of the Joint Image Grassmannian tensor [12]. Following this we outline an algorithm to calculate arbitrary degree-d multiview constraints in m-images.

3.1 The Joint Image Grassmannian

The multiview constraints for a given configuration of cameras and scene features (in general position) can be formulated via an antisymmetrization of the joint image projection (JIP) matrix derived from the reconstruction equations [12, 4]. This method of generating multiview constraints is consistent with viewing the coefficients of the constraints as the Grassmann coordinates of a particular configuration of cameras [12].

The step-$(n+1)$ antisymmetrization of independent vector spaces $\mathbf{x}^{i\beta} \in \mathbb{P}^n$ is $\mathbf{x}^{[0\beta} \cdots \mathbf{x}^{n\beta]} = \mathbf{0}$. By definition we can also state that a step-$(k+1)$ antisymmetrization of a n-dimensional projective vector space forms a k-dimensional projective subspace for an abstract projective vector space $\mathbb{P}^n$ [2]. This manner of forming subspaces allows us to determine Grassmann tensors characterizing the span of projective vector spaces that are invariant (up to scale) to changes in the projective basis.

Applying this concept to the problem of determining the multiview constraints for a given set of cameras, we find that it is possible to form a Grassmann tensor from a selection of independent row vectors from the JIP matrix. This special Grassmann tensor is referred to as the Joint Image Grassmannian (JIG) tensor in the multiple view geometry literature [12],

$$\mathbf{I}^{[A \ldots B]} \equiv \mathbf{P}^{[A}_{[a_0} \cdots \mathbf{P}^{B]}_{a_3]} \tag{3}$$

where $\mathbf{x}^a \in \mathbb{P}^3$ and $\mathbf{x}^A, \mathbf{x}^B \in \mathbb{P}^2$, resulting a 3-dimensional projective subspace spanning $\mathbb{P}^3$. The selection of the image indeterminants $A \ldots B$ from the rows of the JIP matrix determines which images the resulting multiview constraint will represent.

The choice of rows for linear features obeys the simple rule that for an image to be included in the multiview constraint, it must be represented by at least one row, and less than 3 rows. This leads to well known set of matching tensors for points (Table 2) and also explains why there is at most 4-view multiview constraints for linear features in $\mathbb{P}^3$. In order to make the expressions for the multiview constraints in Table 2 succinct, we assign $\mathbf{x}^A, \mathbf{x}^B, \mathbf{x}^C, \mathbf{x}^D \in \mathbb{P}^2$ to be coordinates in images 1 to 4. The number of **DOF** in the multiview constraints for m-views is given as follows [12],

Views	Constraint
2	$\mathbf{I}^{[A_1 A_2 B_1 B_2]} \mathbf{x}^{A_0} \mathbf{x}^{B_0} = 0$
3	$\mathbf{I}^{[A_1 A_2 B_1 C_1]} \mathbf{x}^{A_0} \mathbf{x}^{B_0} \mathbf{x}^{C_0} = \mathbf{0}_{[B_2 C_2]}$
4	$\mathbf{I}^{[A_1 B_1 C_1 D_1]} \mathbf{x}^{A_0} \mathbf{x}^{B_0} \mathbf{x}^{C_0} \mathbf{x}^{D_0} = \mathbf{0}_{[A_2 B_2 C_0 D_0]}$

Table 2. Linear Multiview Constraints for Points

$$\mathbf{DOF}^m_{mc} = 11m - 15 \tag{4}$$

since each camera has $(3 \times 4 - 1 =)11$ **DOF** modulo the $(4 \times 4 - 1 =)15$ **DOF** for an arbitrary projective transform in $\mathbb{P}^3$. In the next section we will expand upon these concepts in order to derive an algorithm for manufacturing generalized multiview constraints.

3.2 Manufacturing Multiview Constraints

In order to utilize multiview constraints to solve for the relative orientation between a set of cameras, it is necessary to be able to reformulate the joint image feature vector associated with these cameras into the appropriate set of multiview constraints. The most general approach for solving for the coefficients of a multiview tensor is to reshape it's coefficients into a vector $\mathbf{x}^{\alpha}$ and form the multiview constraints derived from the joint image features into a matrix $\mathbf{A}^{\beta}_{\alpha}$ that contracts against the coefficients of the multiview tensor,

$$\mathbf{A}^{\beta}_{\alpha}\mathbf{x}^{\alpha} = \mathbf{0}^{\beta} \tag{5}$$

this is always possible.

We now proceed by making some general remarks about the dimensionality and combinatorics of multiview constraints, including the extension to embedded features of higher degree. This is necessary in order to develop an algorithm for the construction of the constraint matrix $\mathbf{A}^{\beta}_{\alpha}$. Firstly, the total number of coefficients composing a degree-d matching tensor over m images is,

$$\Lambda^{m,d}_{\mathrm{mt}} \equiv \prod_{i=1}^{m} \binom{\nu_2^d + 1}{\gamma_i} - 1 \qquad \text{where } \gamma_i \in \{\gamma_1, \dots \gamma_m\} \tag{6}$$

where each γ_i is equal to the number of rows chosen from image i's projection matrix. This implies that the vector of coefficients can be defined as $\mathbf{x}^{\alpha} \in \mathbb{P}^{\Lambda^{m,d}_{\mathrm{mt}}}$, this is a homogeneous vector since one of the overall coefficients of the multiview tensor will always be lost to scaling. By packing the elements in the vector in the same sequence as they are specified symbolically in the JIG tensor expression we can arrive at a lexicographic ordering for the vector.

The dimension β is determined by the number of solutions for a particular multiview constraint given a particular combination of image features. We represent the combination of image features as the set $\zeta_i \in \{\zeta_1, \dots, \zeta_m\}$ where again m is the number of images involved in the multiview constraint. The elements of this set are the $\mathbf{DOF}_i$ of the various image features (in $\mathbb{P}^2$) involved in the multiview constraint, these can be referenced from [7]. The result is an expression for the total number of solutions for a particular combination of image features,

$$\mathbf{DOF}_{\mathrm{if}} \equiv \prod_{i=1}^{m} \binom{\nu_2^d + 1}{\zeta_i - \gamma_i} \tag{7}$$

and consequentially $\mathbf{x}^{\beta} \in \mathbb{R}^{\mathbf{DOF}_{\mathrm{if}}}$. The fact that $\zeta_i - \gamma_i$ can never be negative in the binomial equation is coincident with the fact that no multiview constraint relationship is possible unless the $\mathbf{DOF}_i \geq \gamma_i$ for each image i included in multiview tensor. If we are only interested in the independent solutions to multiview constraints then we can make a substantial reduction in the size of $\mathbf{DOF}_{\mathrm{if}}$ by using only the affine part of each image feature,

$$\overline{\mathbf{DOF}}_{\mathrm{if}} \equiv \prod_{i=1}^{m} \binom{\nu_2^d}{\zeta_i - \gamma_i} \tag{8}$$

the resulting constraint matrix $\mathbf{A}^{\overline{\beta}}_{\alpha}$ will contract with the tensor's coefficients $\mathbf{x}^{\alpha}$ leaving just the independent solutions in the associated zero vector $\mathbf{0}^{\overline{\beta}}$.

In practise this reduction in the number of solutions is easy to achieve due to the fact that dependant solutions correspond to entries in the zero vector $\mathbf{0}^{\beta}$ involving one of the projective scaling coefficients from the (embedded) image feature in $\mathbb{P}^{\nu_2^d}$. By normal convention in the computer vision literature this scaling coefficient is at the end of the vector and canonically scaled to 1 for an affine representation. Therefore by indexing one short of the complete length of each indeterminant composing the zero vector of solutions, we will be left with $\mathbf{A}^{\overline{\beta}}_{\alpha}$.

An optimization is available when determining the constraints corresponding the $\Lambda^{m,d}_{\mathrm{mt}}$ columns in each row of the constraint matrix. In cases where the number of solutions $\mathbf{DOF}_{\mathrm{if}} > 1$, there will be numerous zero entries throughout the rows of the constraint matrix $\mathbf{A}^{\beta}_{\alpha}$. The number of non-zero entries in each row is precisely,

$$\Upsilon^{m,d}_{\mathrm{mt}} \equiv \prod_{i=1}^{m} [(\nu_2^d + 1) - (\zeta_i - \gamma_i)] \tag{9}$$

where $\Upsilon^{m,d}_{\mathrm{mt}} \leq \Lambda^{m,d}_{\mathrm{mt}}$. This equation accounts for the fact that when $(\zeta_i = \gamma_i)$ the indeterminants corresponding to rows of the i^{TH} image's camera matrix in the JIG tensor (3) can be dualized resulting in the interaction between the coefficients of the multiview tensor and the image feature for that image being simplified to a standard vector contraction (this is illustrated in the examples below).

One last observation is in regard to the **DOF** of a combination composed of a multiview tensor and a set of image features contracting against it. We will refer to this as the **DOF** of the multiview constraint,

$$\mathbf{DOF}_{\mathrm{mc}} = \prod_{i=1}^{m} (\zeta_i - \gamma_i + 1) \tag{10}$$

this equation reflects the **DOF** provided by one (singular) set of the image features (ζ_i) in correspondence with a matching tensor (γ_i). The effective measure of the $\mathbf{DOF}_{\rm mc}$ may reduce as further sets of image features (ζ_i) are included in the total set of constraints $\mathbf{A}^{\beta}_{\alpha}$ used to solve for the multiview tensor. This is the case for the linear quadrifocal tensor (as was shown in [3, 8]) and is also the case for other higher degree embedded multiview tensors.

It is now possible to consolidate this information regarding a particular multiview constraint combination into a precise algorithm to formulate the constraint matrix $\mathbf{A}^{\beta}_{\alpha}$ (see Algorithm 1). This algorithm will only ever need to be run once in order to generate a map (tensor tableaux) that transforms a given joint image feature vector into it's corresponding multiview constraint $\mathbf{A}^{\beta}_{\alpha}$.

Algorithm 1: Manufacturing Multiview Constraint Tableaux

Input : The number of images m, the degree-d, the **DOF** of the image features $\zeta_i \in \{\zeta_1, \ldots, \zeta_m\}$ and the number of rows used to generate the multiview tensor $\gamma_i \in \{\gamma_1, \ldots \gamma_m\}$

Output: A tensor tableaux corresponding to the construction of the constraint matrix $\mathbf{A}^{\beta}_{\alpha}$ ($[\mathbf{DOF}_{\rm if} \times \Lambda^{m,d}_{\rm mt}]$)

```
begin
    Determine Υ_mt^{m,d} (9) and DOF_if (7)
    for i ← 1 to DOF_if do
        for j ← 1 to Υ_mt^{m,d} do
            1. Determine the true index (j' ⟵ j)
            2. Evaluate the sequence of m image
            feature coefficients from the joint
            image feature vector corresponding to
            A_{α_j'}^{β_i} by eliminating the
            indeterminants associated with 0^{β_i} and
            x^{α_j'} from the total set available, this
            simplifies in the case ζ_i = γ_i.
            3. Evaluate the sign of A_{α_j'}^{β_i}
end
```

4 Examples

We now present several examples of the application of Algorithm 1 to a selection of different multiview constraints. These examples have been picked to best illustrate the range of problem types to which the algorithm is applicable. In light of the depiction in equation (5) of the coefficients of the matching tensor ($\mathbf{x}^{\alpha}$) being contravariant and the function of the image features in the constraint matrix ($\mathbf{A}^{\beta}_{\alpha}$) being covariant we will utilize the '$*$' expression in front of the image feature's indeterminants in the tensor tableaux.

4.1 The Fundamental Matrix

The first example of the application of Algorithm 1, is in determining the multiple view constraints for the Fundament Matrix (2-view) assuming a point-point correspondence between the images. From Table 2 we can state the JIG expression for this combination as $\mathbf{I}^{[A_1A_2B_1B_2]}\mathbf{x}^{A_0}\mathbf{x}^{B_0} = 0$. This form of JIG expression assumes a selection of rows $\gamma_i \in \{2,2\}$ from the JIP. This selection of rows corresponds with the $\Lambda^{2,1}_{\rm mt} = 9$ according to (6) and since the image features are both points ($\zeta_i \in \{2,2\}$) $\mathbf{DOF}_{\rm if} = 1$ (8).

This is a special case of the algorithm since $\zeta_i = \gamma_i\ \forall i$, this means that the indeterminants from both images associated with the matching tensor (A & B) can both be dualized resulting in one covariant indeterminant for each image that contracts precisely with the image feature's indeterminants ($^*A$ & $^*B$). The corresponding tensor tableaux for this constraint is given as follows.

AB	$^*A^*B$
00	00
01	01
02	02
10	10
11	11
12	12
20	20
21	21
22	22

4.2 The Trifocal Tensor

As a further example of the application of Algorithm 1, we demonstrate it's utilization in determining the independent multiview constraints for the Trifocal Tensor (3-view) assuming a point-point-point correspondence between the images (see Figure 1). From Table 2 we can state the JIG expression for this combination as $\mathbf{I}^{[A_1A_2B_1C_1]}\mathbf{x}^{A_0}\mathbf{x}^{B_0}\mathbf{x}^{C_0} = \mathbf{0}_{[B_2C_2]}$. This form of JIG expression assumes a selection of rows $\gamma_i \in \{2,1,1\}$ from the JIP - this isn't the only valid combination of rows - 2 rows could also be attributed to either the second or third image.

This selection of rows corresponds with $\Lambda^{3,1}_{\rm mt} = 27$ coefficients according to (6), since all the image features are points ($\zeta_i \in \{2,2,2\}$) there exists $\overline{\mathbf{DOF}}_{\rm if} = 4$ solutions (8). In this case just the first image's indeterminants can dualized and second and third image's indeterminants are alternating. The tensor tablueaux corresponding to the $\Upsilon^{3,1}_{\rm mt} = 12$ rows for the first of the constraints is as follows.

ABC	$^*A^*B^*C$	B_2C_2
022	011	00
021	-012	00
012	-021	00
011	022	00
122	211	00
121	-212	00
112	-221	00
111	222	00
222	211	00
221	-212	00
212	-221	00
211	222	00

4.3 The Dual Quadric Fundamental Matrix

The 2-view multiview constraint for dual quadrics was first introduced in [5], the concepts associated with degree-2 symmetric embedding of the projection matrix are discussed in [7]. In this case the dimension of the image feature space is $\nu_2^2+1=6$ and the dimension of the scene feature space is $\nu_3^2+1=10$.

The rank of the 2-view JIP matrix for dual quadrics is only 9 (instead of the full 10) [5]. The 2-view multiview constraint is composed of a selection of 9 rows of the available 12 from the degree-2 JIP matrix resulting (for example) $\gamma_i \in \{5,4\}$ therefore $\Lambda_{\text{mt}}^{2,2}=90$. The image features in this case are the dual apparent contours of the quadric $\zeta_i \in \{5,5\}$ thus $\overline{\mathbf{DOF}}_{\text{if}}=5$ (8). In this case just the first image's indeterminants can be dualized and second image's indeterminants are alternating. The first 6 of $\Upsilon_{\text{mt}}^{2,2}=30$ rows for the first of these constraints is as follows.

AB	$^*A^*B$	B_5
0 10	05	0
0 11	-04	0
0 12	03	0
0 13	-02	0
0 14	01	0
1 10	15	0

5 Discussion

In this paper we have presented an algorithm for the generation of the multiple view constraints corresponding with arbitrary configurations of image features. We showed that via an application of the principles of anti-symmetric algebra it is possible to treat the formation of constraints in an entirely general fashion.

This algorithm can be incorporated into a toolkit for multiple view geometry and utilized to generate any manner of multiview constraint. The application of this algorithm to new projection operators (and combinations of image-to-scene feature correspondences) can be used to derive novel configurations of multiview constraints. It has already been used successfully in the generation of the novel multiview constraints presented in [6].

References

[1] O. D. Faugeras, Q. T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications.* MIT Press, Cambridge, Massachusetts, 2001.

[2] J. Harris. *Algebraic Geometry, A first course.* Springer Verlag, first edition, 1992.

[3] R. I. Hartley. Computation of the quadrifocal tensor. 1406:20–35, 1998.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[5] F. Kahl and A. Heyden. Using conic correspondences in two images to estimate the epipolar geometry. *Int. Conf. on Computer Vision*, 1998.

[6] D. N. McKinnon and B. C. Lovell. Towards closed form solutions to the multiview constraints of curves and surfaces. *DICTA03*, pages 519–528, 2003.

[7] D. N. McKinnon and B. C. Lovell. Tensor algebra: A combinatorial approach to the projective geometry of figures. *IWCIA04*, pages 558–568, 2004.

[8] A. Shashua and L. Wolf. On the structure and properties of the quadrifocal tensor. In *ECCV (1)*, pages 710–724, 2000.

[9] P. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–605, 1997.

[10] P. H. S. Torr. *Motion segmentation and outlier detection.* PhD thesis, 1995.

[11] P. H. S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[12] B. Triggs. The geometry of projective reconstruction i: Matching constraints and the joint image. *Int. Conf. on Computer Vision*, pages 338–343, 1995.