

# Towards Closed Form Solutions to the Multiview Constraints of Curves and Surfaces

David N. McKinnon and Brian C. Lovell

Intelligent Real-Time Imaging and Sensing (IRIS) Group,  
School of Information Technology and Electrical Engineering,  
University of Queensland, St. Lucia QLD 4066, Australia,  
mckinnon@itee.uq.edu.au

**Abstract.** In this paper we present the theoretical setting for the closed form solutions to the multiview constraints of curves and surfaces observed by the motion of a camera in a scene. We find that knowledge of a curve or surface's structure can provide matching constraints and thus closed form solutions for the position of a camera in a scene over many images. This is advantageous for the process of Structure From Motion (SFM) as it may negate some of the need for non-linear optimisation (Bundle-Adjustment) of an initial linear solution to the motion of the camera and structure of the scene.

## 1 Introduction

The literature on the multiview constraints of curves and surfaces is rather sparse. To date the only cases that have received attention are the constraints for degree-2 curves (planar) and degree-2 surfaces.

This paper will present some theoretical work on the multiview constraints of curves and surfaces. There will be very little discussion of practical issues and no specific numerical algorithms given. However, we will place the novel matching constraints on a similar footing to those of the existing matching constraints which have well known methods for their solution [7].

This study of the multiview constraints of curves and surfaces shows that while the constraints for arbitrary curved features can be derived, the constraints can contain many hundreds or thousands of free variables. We show that the overwhelming majority of these variables are redundant, however it may still pose a significant task to enforce these constraints algebraically in a practical setting where the data is affected by noise.

### 1.1 Notation and Basics

Firstly, we wish to develop the notation that will be used for the rest of the paper. This notation is adapted from [13] with the usage of the symmetric operators which we will use to derive representations for curves and surfaces.

We are motivated to develop a system of symbolic algebra that allows us to express geometric concepts such as varieties and the operations upon them [1]. To this end we can use the language of vector spaces, determinants and different representations of the

symmetric group to define various geometric objects that we wish to study. The entire presentation of the geometry herein will be limited to projective vector space  $\mathbb{P}^n$ .

An element of an  $n$ -dimensional projective vector space in the tensor notation is denoted as  $\mathbf{x}^{mA_i^s} \in \mathbb{P}^n$ . The symbol  ${}_m A_i^s$  is called an indeterminate and identifies several important properties of the vector space. To better understand the notation we may rewrite it in the standard vector form. Rewriting the symbolic notation can be achieved by listing the elements of the vector space using the indeterminate as the variables of the expression. In this manner the symbol that adjoins the indeterminate is merely cosmetic, for example

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

where  $m$  identifies the multilinearity of the indeterminate,  $s$  depicts the degree (or step) of the indeterminate, we show in the next section that there are several different types of degree that we will be concerned with. The last element specifying the indeterminate is  $i$ , this a *choice* of the positioning of the elements in the vector. We stress that this is merely a choice as to how the elements in a vector are to be labelled. The standard labelling is just  $0 \dots n$  for an  $n$ -dimensional projective vector space. Due to space constraints we are omitting a third representation of the tensor algebra which is the tableaux form. The tableaux form for the tensor algebra is the most effective representation for computational purposes and will be a feature of future work.

Indeterminants of a regular vector space ( $\mathbb{P}^n$ ) are called *contravariant* and indeterminants of a dual vector space ( ${}^*\mathbb{P}^n$ ) are called *covariant*. The notation for a dual vector space is similar to that for the regular vector space,

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

tensor contraction is achieved via a dot product of elements for a regular and dual vector space,

$$\mathbf{x}_m^* A_i^s \mathbf{x}^{mA_i^s} \equiv [{}_m^* A_0^s, {}_m^* A_1^s, \dots, {}_m^* A_n^s] [{}_m A_0^s, {}_m A_1^s, \dots, {}_m A_n^s]^\top = 0 \tag{3}$$

since our vector spaces are projective the contraction results in the scalar 0. In the interests of compactness and clarity often we will abandon the entire set of labels for an indeterminate via an initial set of assignments. If this is the case assume that  $i$  is any arbitrary scalar between 0 and  $n$  and  $s, m = 1$ . If an indeterminate is used in a covariant expression then the  $*$  maybe omitted.

## 1.2 Operators

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 will use in this paper Table 1, the symbols  $\nu_n^d = \binom{d+n}{d} - 1$  and  $\eta_n^k = \binom{n+1}{k} - 1$ .

The two different forms of the tensor are representative of the fact that we can always rewrite any tensor expression as the vector of its coefficients. We call this alternative to the tensor form the vector form. The vector form is simply a vector space of the unique coefficients in any tensor expression. Writing the tensor as a vector of

Operator	Symbol	Tensor Form	Vector Form	Conditions
Segre	-	$\mathbf{x}^{A_i \dots B_j}$	$\mathbf{x}^{\alpha^k} \in \mathbb{P}^{n_A} \times \dots \times \mathbb{P}^{n_B}$	-
Anti-Symmetric (Step- $k$ )	[...]	$\mathbf{x}^{[A_i \dots B_j]}$	$\mathbf{x}^{\alpha^{[k]}} \in \mathbb{P}^{n_k}$	$i < j$ and, $A \neq \dots \neq B$
Symmetric (Degree- $d$ )	(...)	$\mathbf{x}^{(A_i \dots A_j)}$	$\mathbf{x}^{\alpha^{(d)}} \in \mathbb{P}^{n_d}$	$i \leq j$

**Table 1.** Tensor Operators

coefficients abandons the symmetry of the tensor form so this results in a less fruitful representation for symbolic derivations but reduces the redundancy so the result is a more effective representation for mappings between vector spaces.

### 1.3 Geometric Algebra

The application of the tensor operators given in Table 1 to vector spaces gives us a means represent the geometry of various features we encounter in computer vision as the embedding of a coefficient ring into some other vector space. Table 2 summarises the representation and the DOF for linear features in the projective plane ( $[A_0, A_1, A_2] \in \mathbb{P}^2$ ).

Hyperplane	$\mathbb{P}^2$	$^*\mathbb{P}^2$	DOF <sub><i>i</i></sub>	Embedding
Points	$\mathbf{x}^{A_0}$	$\mathbf{x}^{A_0} \rightarrow \epsilon_{A_0 A_1 A_2} \mathbf{x}^{A_0} = \mathbf{x}_{[A_1 A_2]}$	2	$\mathbb{P}^2$
Lines	$\mathbf{x}^{[A_0 A_1]}$	$\mathbf{x}^{[A_0 A_1]} \rightarrow \epsilon_{A_0 A_1 A_2} \mathbf{x}^{A_0 A_1} = \mathbf{x}_{A_2}$	1	$\mathbb{P}^2$

**Table 2.** Linear features and their duals in  $\mathbb{P}^2$

Similarly, Table 3 summarises the representation and the DOF for linear features in projective space ( $[a_0, a_1, a_2, a_3] \in \mathbb{P}^3$ ).

Hyperplane	$\mathbb{P}^3$	$^*\mathbb{P}^3$	DOF <sub><i>s</i></sub>	Embedding
Points	$\mathbf{x}^{a_0}$	$\mathbf{x}^{a_0} \rightarrow \epsilon_{a_0 a_1 a_2 a_3} \mathbf{x}^{a_0} = x_{[a_1 a_2 a_3]}$	3	$\mathbb{P}^3$
Lines	$\mathbf{x}^{[a_0 a_1]}$	$\mathbf{x}^{[a_0 a_1]} \rightarrow \epsilon_{a_0 a_1 a_2 a_3} \mathbf{x}^{a_0 a_1} = x_{[a_2 a_3]}$	2	$\mathbb{P}^5$
Planes	$\mathbf{x}^{[a_0 a_1 a_2]}$	$\mathbf{x}^{[a_0 a_1 a_2]} \rightarrow \epsilon_{a_0 a_1 a_2 a_3} \mathbf{x}^{a_0 a_1 a_2} = x_{a_3}$	1	$\mathbb{P}^3$

**Table 3.** Linear features and their duals in  $\mathbb{P}^3$

These tables also demonstrate the process of dualization for linear feature types via the dualization mapping ( $\rightarrow$ ). The antisymmetrization operator should be considered as a determinantal method to generate the algebra for linear features, by performing an alternating tensor contraction ( $\epsilon_{\alpha_0 \dots \alpha_n}$ ) over the space to which the operator is applied ( $\mathbf{x}^\alpha \in \mathbb{P}^n$ ) [4].

The other fundamental feature type is the hypersurface, which we will construct from the symmetric operator (...) as demonstrated in Table 4. Hypersurfaces embedded

in  $\mathbb{P}^3$  depict the equation of a surface and those embedded in  $\mathbb{P}^2$  depict the equation of a planar curve. In the Algebraic-Geometry literature hypersurfaces are referred to as the Degree- $d$  Veronese Embedding of a vector space  $\mathbf{x}^\alpha \in \mathbb{P}^n$  [6]. This results in a degree- $d$  surface satisfying the equation  $\mathbf{S}_{a^{(d)}} \mathbf{x}^{a^{(d)}} = 0$ .

Hypersurface	Regular	Dual	DOF	Embedding
$\mathbb{P}^2$	$\mathbf{x}_{(A\dots A)}$	$\mathbf{x}_{(A\dots A)} \rightarrow \mathbf{x}^{(A\dots A)}$	$\nu_2^d$	$\mathbb{P}^{\nu_2^d}$
$\mathbb{P}^3$	$\mathbf{x}_{(a\dots a)}$	$\mathbf{x}_{(a\dots a)} \rightarrow \mathbf{x}^{(a\dots a)}$	$\nu_3^d$	$\mathbb{P}^{\nu_3^d}$

**Table 4.** Degree- $d$  hypersurfaces and their duals in  $\mathbb{P}^2$  &  $\mathbb{P}^3$

Dualization of a hypersurface is performed as the tensor adjoint of the symmetric tensor form representing the hypersurface. This can be achieved by evaluating all the cofactors of the tensor. If the tensor is singular then the corresponding surface is degenerate which is to say that it contains a ruling. Surfaces that contain a ruling are not irreducible and thus the dual does not exist. From this point on we will always consider the hypersurfaces to be irreducible and thus the dual must exist.

The final feature type of interest is the curve embedded in  $\mathbb{P}^3$ . Typically a curve is thought of as the intersection of two or more surfaces, in which case it would seem logical to depict the curve as the intersection of several surfaces in embedded  $\mathbb{P}^3$ . This however is not suitable for the purposes of reconstruction as the coefficient ring for the curve will contain terms from several different surfaces, which would be hard to constrain in a practical setting. So for this purpose we represent the equation of a curve embedded in  $\mathbb{P}^3$  as the degree- $d$  embedding of the Plucker line  $\mathbf{x}^\omega \in \mathbb{P}^5$  (Table 3). This results in a degree- $d$  curve satisfying the equation  $\mathbf{C}_{\omega^{(d)}} \mathbf{x}^{\omega^{(d)}} = 0$  we will refer to this equation as the *Chow Polynomial* of the curve. Due to the redundancy of the Plucker equation for a line the DOF of the Chow Polynomial is given as  $\xi_5^d = \binom{d+5}{d} - \binom{d+3}{d-2} - 1$  [5].

## 2 Linear Multiview Geometry

In this section we seek to review the multiview geometry of linear features. By linear we envisage features of degree 1, namely points, lines and planes in  $\mathbb{P}^2$  and  $\mathbb{P}^3$  (Table 2 & 3). This review is by no means exhaustive (see [7]) however it does serve to develop some of the mechanisms we use to specify the multiview geometry of curves and surfaces in subsequent sections of this paper.

The two main problems we look to address are the triangulation of linear features observed by two or more cameras and the camera resectioning problem. Resectioning is the process of determining the cameras position through an arbitrary movement/s, from the observed correspondences between features segmented in the images and their corresponding location in the scene. It is also possible to resection the camera's position and orientation directly from the matching constraint tensors. We are more interested in this method since it derives a solution via a direct closed form linear computation.

### 2.1 Triangulation

**Linear Projection Operators** Generally it may be stated that the projection of a feature  $\mathbf{x}^\beta$  embedded in  $\mathbb{P}^3$  to its location in image  $m$  ( $\mathbb{P}^2$ ) may be found up to an arbitrary scale factor  $\lambda_m$  as  $\lambda_m \mathbf{x}^{m\alpha} = \mathbf{P}_\beta^{m\alpha} \mathbf{x}^\beta$ . Since all our geometry is projective we will usually assume equality up to scale as a regular artifact of the vector spaces and denote the relationship as  $\mathbf{x}^{m\alpha} \sim \mathbf{P}_\beta^{m\alpha} \mathbf{x}^\beta$ . The range of projection operators for linear features is given in Table 5, these operators assume that  $\mathbf{x}^\beta$  does not intersect the camera center  $\mathbf{e}^\beta$  where  $\mathbf{P}_\beta^{m\alpha} \mathbf{e}^\beta = 0$ , except for the last operator (Line-to-Point) which explicitly assumes that the line passes through the camera center and intersects the image plane in a single point.

Type	$\mathbb{P}^3$	$^* \mathbb{P}^3$
Point-to-Point	$\mathbf{x}^A \sim \mathbf{P}_a^A \mathbf{x}^a$	-
Line-to-Line	$\mathbf{x}^{[A_0 A_1]} \sim \mathbf{P}_{[a_0 a_1]}^{[A_0 A_1]} \mathbf{x}^{[a_0 a_1]}$	$\mathbf{x}_A \sim \mathbf{P}_A^{[a_2 a_3]} \mathbf{x}_{[a_2 a_3]}$
Line-to-Point	$\mathbf{x}^A \sim \mathbf{P}_{[a_0 a_1]}^A \mathbf{x}^{[a_0 a_1]}$	$\mathbf{x}_{[A_0 A_1]} \sim \mathbf{P}_{[A_0 A_1]}^{[a_2 a_3]} \mathbf{x}_{[a_2 a_3]}$

**Table 5.** Projection operators for linear features

**Reconstruction Equations** The reconstruction equations provide a form to represent the triangulation of a scene feature observed in  $m$  images,

$$\begin{pmatrix} \mathbf{P}_a^{1A} & x^{1A} & 0 & \dots & 0 \\ \mathbf{P}_a^{2A} & 0 & x^{2A} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{P}_a^{mA} & 0 & 0 & \dots & x^{mA} \end{pmatrix} \begin{pmatrix} \mathbf{x}^a \\ -\lambda_1 \\ -\lambda_2 \\ \vdots \\ -\lambda_m \end{pmatrix} = \mathbf{0} \tag{4}$$

where the resulting nullvector of these equations presents a solution for the scene feature and the scale factors  $\lambda_m$ . The stack of camera matrices on the left hand side of (4) is referred to as the *joint image projection matrix* ( $\mathbf{P}_a^\gamma \equiv \mathbf{P}_a^{1A_2A \dots mA}$ ) and can be thought of as a vector of camera matrices that projects a common feature from the scene ( $\mathbf{x}^a$ ) to its *joint image* feature location ( $\mathbf{x}^\gamma \equiv \mathbf{x}^{1A_2A \dots mA}$ ).

The reconstruction equations have,

$$\left( \sum_m (\mathbf{DOF}_i^m + 1) - (\mathbf{DOF}_s + 1) \right) (\mathbf{DOF}_s + 1) - m + 1 \tag{5}$$

DOF, where  $\mathbf{DOF}_i^m$  and  $\mathbf{DOF}_s$  denote the DOF of the  $m^{th}$  image feature and scene features respectively. Furthermore the reconstruction equations are rank- $(\mathbf{DOF}_s + m)$ , which implies that any  $(\mathbf{DOF}_s + m) - 1$  minor of the reconstruction equation must vanish.

## 2.2 Multiview Constraints

Multiview constraints provide a multilinear relationship between projections of scene features observed in two or more images. We show here the general method of determining the matching constraint for a set of views by an antisymmetrization of elements from the reconstruction equations (4).

**The Joint Image Grassmannian** The approach to building multiview constraints stems from the representation of a subspace in the *Grassmann* or anti-symmetric algebra. Here we wish to find a  $d_s$ -dimensional subspace for the scene (where the scene is embedded in  $\mathbb{P}^{d_w}$ ), from the joint image projection matrix. This is achieved by antisymmetrizing over  $d_s + 1$  of the joint image's scene indeterminants, with corresponding unique choices of any  $d_s + 1$  of the images' indeterminants.

$$\mathbf{I}^{\gamma_0 \dots \gamma_{d_s}} \equiv \frac{1}{(d_s + 1)!} \mathbf{P}_{d_0}^{\gamma_0} \dots \mathbf{P}_{d_s}^{\gamma_{d_s}} \epsilon^{d_0 \dots d_s} \equiv \mathbf{P}_{[d_0}^{\gamma_0} \dots \mathbf{P}_{d_s]}^{\gamma_{d_s}} \quad (6)$$

Equation (6) is known as the *Joint Image Grassmannian*. The selection of the image indeterminants  $\gamma_0 \dots \gamma_{d_s}$  from the rows of the joint image projection matrix determines which images the resulting matching constraint will represent. The choice of rows obey the simple rules that for an image to be included in the matching constraint, it must be represented by at least one row, and less than  $d_i + 1$  rows (where the feature in the image plane is embedded in  $\mathbb{P}^{d_i}$ ). This leads to well known sets of matching tensors (Table 6) and also explains why there is at most 4-view matching constraints for points and lines.

Views	Constraint
2	$\mathbf{I}^{[1 A_{11} A_{22} A_{12} A_2 \mathbf{x}^1 A_0 \mathbf{x}^2 A_0]} = 0$
3	$\mathbf{I}^{[1 A_{11} A_{22} A_{13} A_1 \mathbf{x}^1 A_0 \mathbf{x}^2 A_0 \mathbf{x}^3 A_0]} = \mathbf{0}_{[2 A_{23} A_2]}$
4	$\mathbf{I}^{[1 A_{12} A_{13} A_{14} A_1 \mathbf{x}^1 A_0 \mathbf{x}^2 A_0 \mathbf{x}^3 A_0 \mathbf{x}^4 A_0]} = \mathbf{0}_{[1 A_{22} A_{23} A_{04} A_0]}$

Table 6. Linear Matching Constraints for Points

There are many variations of the atypical matching constraints given in Table 6, see [13, 7]. Closed-form solutions to the matching constraints of linear features are well known [7].

## 3 Multiview Geometry of Surfaces

In this section we present some novel theoretical work on the multiview constraints of surfaces. The general formulation for the triangulation problem follows the abstract setting given by [9], however the extension into to the anti-symmetric tensor algebra and the eventual matching constraints represents a new setting for theoretical framework.

### 3.1 Triangulation

As in the previous section on the linear matching our first step in solving the triangulation problem is addressing the nature of projection operators for degree- $d$  hypersurfaces. Since the camera is observing a surface the projection into the image results in the curve formed by the intersection of the surfaces tangent cone with the camera center and the image plane. This curve is known as the *apparent contour* ( $\mathbf{c}^{A^{(d)}}$ ) and the curve formed by the intersection of the tangent cone with the surface is called the *contour generator*. The contour generator sweeps out the surface as the perspective of the camera changes.

We also know that [9] the degree of the dual surface in the scene will be the common degree of the dual curves observed in the image planes. Algebraically the tangent cone is denoted as  $\mathbf{S}^{a^{(d)}}$  which is the dual surface thus the only projection operator of interest is for the projection of degree- $d$  dual hypersurfaces. The projection is given as  $\mathbf{c}^{A^{(d)}} \sim \mathbf{P}_{a^{(d)}}^{A^{(d)}} \mathbf{S}^{a^{(d)}}$ . We can also state that these projection matrices are the  $d$ -fold symmetric powers of the dual point projection matrix and the dimensions of tensor are  $[(\nu_2^d + 1) \times (\nu_3^d + 1)]$  respectively.

Since we are concerned with finding the the equation of the surface generating the apparent contour in the image. We must take the intersection of the dual hypersurfaces tangent cone with the image plane. This leads us to the equivalent set of *dual* reconstruction equations for degree  $d$  hypersurfaces,

$$\begin{pmatrix} \mathbf{P}_{a^{(d)}}^{1A^{(d)}} & \mathbf{c}^{1A^{(d)}} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{P}_{a^{(d)}}^{mA^{(d)}} & \mathbf{0} & \dots & \mathbf{c}^{mA^{(d)}} \end{pmatrix} \begin{pmatrix} \mathbf{S}^{a^{(d)}} \\ -\lambda_1 \\ \vdots \\ -\lambda_m \end{pmatrix} = \mathbf{0} \quad (7)$$

again the resulting nullvector of these equations presents a solution for the dual surface ( $\mathbf{S}^{a^{(d)}}$ ) in the scene and the scale factors  $\lambda_m$ .

We also know that the minimum number of image dual hypersurfaces required to reconstruct the corresponding scene hypersurface is given as the lower bound of,

$$\nu_3^d + 1 \geq m \geq \frac{(d^2 + 6d + 11)}{3(d + 3)} \quad (8)$$

[9] the lower bound  $m$  must be rounded up to the closest integer value. The upper bound is the limit on the number of images for the resulting matching constraint. The **DOF** of these reconstruction equations and their rank are analogous to those stated for (4).

### 3.2 Multiview Constraints

Firstly, we should note that several degree-2 matching constraints have been cited by [11, 8, 3] using an equivalent dual formulation. We extend these results to the degree- $d$  formulation.

**Degree-2** Before we address the general formulation for the degree- $d$  matching constraints for dual hypersurfaces, we will tread gently by outlining the concepts for degree-2.

An application of (8) suggests the presence of degree-2 matching constraints for 2 through to 10 image projections. Again, the object in building the matching constraints is select  $\nu_3^2 + 1$  unique rows from the joint image projection matrix to make up the matching constraints. The corresponding matching constraints are given in Table 7.

Views	Constraint
2	$I[{}_1A_1^{(2)} {}_1A_2^{(2)} {}_1A_3^{(2)} {}_1A_4^{(2)} {}_1A_5^{(2)} {}_2A_1^{(2)} {}_2A_2^{(2)} {}_2A_3^{(2)} {}_2A_4^{(2)} {}_2A_5^{(2)} x_1A_0^{(2)} x_2A_0^{(2)}] = 0$
3	$I[{}_1A_1^{(2)} \dots {}_1A_5^{(2)} {}_2A_1^{(2)} {}_2A_2^{(2)} {}_2A_3^{(2)} {}_2A_4^{(2)} {}_3A_1^{(2)} x_1A_0^{(2)} x_2A_0^{(2)} x_3A_0^{(2)}] = \mathbf{0}_{[{}_2A_5^{(2)} {}_3A_2^{(2)} {}_3A_3^{(2)} {}_3A_4^{(2)}]}$
4	$I[{}_1A_1^{(2)} {}_1A_2^{(2)} {}_1A_3^{(2)} {}_1A_4^{(2)} {}_1A_5^{(2)} {}_2A_1^{(2)} {}_2A_2^{(2)} {}_2A_3^{(2)} {}_3A_1^{(2)} {}_4A_1^{(2)} x_1A_0^{(2)} x_2A_0^{(2)} x_3A_0^{(2)} x_4A_0^{(2)}] = \mathbf{0}_{[\dots]}$
5	$I[{}_1A_1^{(2)} {}_2A_1^{(2)} {}_3A_1^{(2)} {}_4A_1^{(2)} {}_4A_2^{(2)} {}_5A_1^{(2)} {}_5A_2^{(2)} {}_5A_3^{(2)} {}_5A_4^{(2)} {}_5A_5^{(2)} x_1A_0^{(2)} x_2A_0^{(2)} \dots x^5A_0^{(2)}] = \mathbf{0}_{[\dots]}$
6	$I[{}_1A_1^{(2)} {}_2A_1^{(2)} {}_3A_1^{(2)} {}_4A_1^{(2)} {}_5A_1^{(2)} {}_6A_1^{(2)} {}_6A_2^{(2)} {}_6A_3^{(2)} {}_6A_4^{(2)} {}_6A_5^{(2)} x_1A_0^{(2)} x_2A_0^{(2)} \dots x^6A_0^{(2)}] = \mathbf{0}_{[\dots]}$
7	$I[{}_1A_1^{(2)} {}_2A_1^{(2)} {}_3A_1^{(2)} {}_4A_1^{(2)} {}_5A_1^{(2)} {}_6A_1^{(2)} {}_7A_1^{(2)} {}_7A_2^{(2)} {}_7A_3^{(2)} {}_7A_4^{(2)} x_1A_0^{(2)} x_2A_0^{(2)} \dots x^7A_0^{(2)}] = \mathbf{0}_{[\dots]}$
8	$I[{}_1A_1^{(2)} {}_2A_1^{(2)} {}_3A_1^{(2)} {}_4A_1^{(2)} {}_5A_1^{(2)} {}_6A_1^{(2)} {}_7A_1^{(2)} {}_8A_1^{(2)} {}_8A_2^{(2)} {}_8A_3^{(2)} x_1A_0^{(2)} x_2A_0^{(2)} \dots x^8A_0^{(2)}] = \mathbf{0}_{[\dots]}$
9	$I[{}_1A_1^{(2)} {}_2A_1^{(2)} {}_3A_1^{(2)} {}_4A_1^{(2)} {}_5A_1^{(2)} {}_6A_1^{(2)} {}_7A_1^{(2)} {}_8A_1^{(2)} {}_9A_1^{(2)} {}_9A_2^{(2)} x_1A_0^{(2)} x_2A_0^{(2)} \dots x^9A_0^{(2)}] = \mathbf{0}_{[\dots]}$
10	$I[{}_1A_1^{(2)} {}_2A_1^{(2)} {}_3A_1^{(2)} {}_4A_1^{(2)} {}_5A_1^{(2)} {}_6A_1^{(2)} {}_7A_1^{(2)} {}_8A_1^{(2)} {}_9A_1^{(2)} {}_{10}A_1^{(2)} x_1A_0^{(2)} x_2A_0^{(2)} \dots x^{10}A_0^{(2)}] = \mathbf{0}_{[\dots]}$

**Table 7.** Degree-2 Matching Constraint Tensors for Surfaces

In initial experimentation we have found that any combination of rows that meets the aforementioned requirements for defining a Grassmann subspace is adequate to construct the matching tensor. The most pertinent factor in selecting a number of rows to form the matching constraints, is minimizing the size of the actual matching tensor. The selection of  $k$  rows from an image space of size  $n$  will result in the size of associated dimension of the matching tensor being  $\binom{n+1}{k}$ , so naturally values close to either  $n$  or 1 will yield smaller matching constraints.

**Degree- $n$**  Finally, we can now see that an application of equation (8) will give the upper and lower bounds for the degree- $d$  multi-view constraints and an application of equation (7) will generate the reconstruction equations for the problem. Any selection of rows from the reconstruction equations meeting the aforementioned criteria of a valid subspace, will be adequate to reconstruct the degree- $d$  matching constraints.

#### 4 Multiview Geometry of Curves

In this section we will focus initially upon the triangulation problem, or more specifically how the Chow polynomial may be solved resulting in a method to determine a space curve from multiple image correspondences [9]. We will then extend the framework for matching constraints to case of space curves.



#### 4.1 Triangulation

Determining a space curve from image correspondences can be achieved by first making the simple observation that the equation for the resulting Chow polynomial is equivalent up to a scale factor to the expression for the space curve when projected into the image,

$$\mathbf{x}^{A^{(d)}} \simeq \mathbf{P}_{\omega^{(d)}}^{A^{(d)}} \mathbf{x}^{\omega^{(d)}} \quad (9)$$

where the projection operator use here is the degree- $d$  symmetrization of the (Line-to-Point) operator given in Table 5. We see that the relationship given above will result in  $\nu_2^d$  independant equations for each image curve which contribute to solving the Chow polynomial, there are a total of  $\nu_5^d$  monomial coefficients but only  $\xi_5^d$  DOF. This means that solving for a degree- $d$  space curve requires image points from,

$$\xi_5^d + 1 \geq m \geq \frac{d^3 + 5d^2 + 8d + 4}{6d} \quad (10)$$

images, where no more than  $\nu_2^d$  image points from any one image curve may be used in the reconstruction.

#### 4.2 Multiview Constraints

In this section we will sketch the general picture for the multiview constraints of space curves. We follow exactly the path taken in the preceeding analysis for the multiview constraints of surfaces.

**Degree-2** By application of equation (10) we see that for degree-2 space curves the multiview constraints extend from 4 to 20 views. The 4 view constraint is particularly appealing as it contracts exactly to a single scalar (0 in the ideal case). For degree-3 curves the matching constraints exist for 6 through to 50 views.

The general program that has been outlined for the calculation of the matching constraints in the linear and surface cases can be used here in the case of curves to realise the degree- $d$  matching constraints.

### 5 Discussion and Future Work

The multiview constraints for surfaces have been simulated in MATLAB in the noise free case however we are still currently working on an implementation of the multiview constraints for curves. Future work will also consider the efficacy of the matching constraints in the presence noise. The major difficulty in testing the constraints in a noisy setting is translating a noise model into a real valued implicit functions coefficients. Progress in this direction is well advanced will be a feature of future work.

In the numerical simulations we found that any surfaces of degree greater than 3 had such large matching constraints associated with them that computation was very expensive. In practice we don't expect that higher degree curves and surfaces will be

particular useful for reconstruction purposes. This is because  $4^{TH}$  order polynomials have a significantly higher complexity than  $3^{RD}$  and  $2^{ND}$  order polynomials, as well as an increase in the size of the corresponding vector spaces.

There are many effective algorithms to fit explicit/parametric cubic curves to  $2D$  point and derivative data [10], a  $3^{RD}$  order polynomial fitted in a piecewise fashion is capable of capturing the complexity of a standard edge string obtained from an edge filter. Algorithm 9.10 from [10] is a good way to set the complexity as a parameter that can be traded off against the accuracy of the segmentation. This method of segmentation of curves from the images will form the basis for a practical assessment of the reconstruction and multiview constraints of curves. The process of implicitization [2, 12] is required to convert the explicit polynomials from the segmentation into implicit polynomials.

Future work will involve testing of the constraints which have been presented in the presence of noise to determine how well this formulation can cope with erroneous segmentations and discriminate against mismatches. Also the problem of extracting the camera matrices from the matching tensors will need to be analysed in greater detail.

## References

1. David Cox, John Little, and Donal O'Shea. *Varieties, Ideals and Algorithms*. Springer-Verlag, 1992.
2. David Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry*. Springer, 1998.
3. G. Cross. *Surface Reconstruction from Image Sequences : Texture and Apparent Contour Constraints*. PhD thesis, University of Oxford, Trinity College, 2000.
4. W. Greub. *Multilinear Algebra*. Springer-Verlag, 1978.
5. C. J. Harris and M. Stephens. A combined corner and edge detector. *4th Alvey Vision Conference*, pages 147–151, 1988.
6. Joe Harris. *Algebraic Geometry, A First course*. Springer Verlag, first edition, 1992.
7. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
8. F. Kahl and A. Heyden. Using conic correspondences in two images to estimate the epipolar geometry. *Int. Conf. on Computer Vision*, 1998.
9. J. Y. Kaminski, M. Fryers, A. Shashua, and M. Teicher. Multiple view geometry of non-planar algebraic curves. *Int. Conf. on Computer Vision*, 2001.
10. L. Piegl and W. Tiller. *The NURBS Book*. Springer, 1995.
11. L. Quan. Conic reconstruction and correspondence from two views. *Transactions on Pattern Analysis and Machine Intelligence*, 18(2), 1996.
12. Thomas W. Sederberg and Falai Chen. Implicitization using moving curves and surfaces. *Computer Graphics*, 29(Annual Conference Series):301–308, 1995.
13. B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. *Int. Conf. on Computer Vision*, pages 338–343, 1995.