Solving the Recognition Problem for Six Lines Using the Dixon Resultant ⋆

Robert H. Lewis[1]

Dept. of Mathematics, Fordham University, Bronx, NY 10458 USA

Peter F. Stiller[2]

Dept. of Mathematics, Texas A&M University, College Station, TX 77843-3368

Abstract

The "Six-Line Problem" arises in computer vision and in the automated analysis of images. Given a three-dimensional object, one extracts geometric features (for example six lines) and then, via techniques from algebraic geometry and geometric invariant theory, produces a set of three-dimensional invariants that represents that feature set. Suppose that later an object is encountered in an image. (For example a photograph taken by a camera modeled by standard perspective projection, i.e. a "pinhole" camera.) Suppose further that six lines are extracted from the object appearing in the image. The problem is to decide if the object in the image is the original 3D object. To answer this question two-dimensional invariants are computed from the lines in the image. One can show that conditions for geometric consistency between the three-dimensional object features and the two dimensional image features can be expressed as a set of polynomial equations in the combined set of two and three dimensional invariants. The object in the image is geometrically consistent with the original object if the set of equations has a solution. One well known method to attack such sets of equations is with *resultants*. Unfortunately, the size and complexity of this problem made it appear overwhelming until recently. This paper will describe a solution obtained using our own variant of the Cayley-Dixon-Kapur-Saxena-Yang resultant. There is reason to suspect that the resultant technique we employ here may solve other complex polynomial systems.

Preprint submitted to Elsevier Preprint *21 February 2001*

1 Introduction

The Recognition Problem for Six Lines (Six Line Problem) arises in computer vision and in the automated recognition of three-dimensional objects. From an object, six lines are extracted, and from those six lines, nine three-dimensional ("3D") invariants are computed as a kind of signature. Later, a two-dimensional "snapshot" of some possibly different object is obtained from an arbitrary perspective, and from this snapshot six lines are extracted leading to the computation of four two-dimensional ("2D") invariants. The question is, is the snapshot a picture of the original object, i.e. a perspective projection of the original six lines? We desire a method that can rapidly and reliably decide if a given set of 2D data represents the same 3D object, or at least that a given 2D set can *not* represent that object.

Using algebraic geometry, Stiller [5] showed that there should be a single equation relating the nine 3D invariants to the four 2D invariants. He reduced the problem to a system of 4 equations in 16 variables involving three additional variables (actually four, but one may be set to 1). The resulting four polynomial equations $d_i = 0, i = 1, \dots, 4$ in the three new variables are quartic and involve the $9 + 4 = 13$ invariants as parameters in the coefficients. The image is consistent with the original object if and only if the four equations have a solution in the three variables (subject to a mild nondegeneracy constraint).[3] Note that we do not need to know what the values of the three auxiliary variables actually are, only that a solution exists. Image recognition questions of this general type, but for points, were considered by L. Quan [4] and Stiller [6].

The solution of systems of polynomial equations is important in many fields of applied mathematics. One of the classic methods of solving such systems is with *resultants*. In general a resultant is a single polynomial derived from a system of polynomial equations that encapsulates the solution (common zeroes) of the system. The *Sylvester Determi-*

* Expanded version of talks presented to the Maui IMACS meeting, July 1997, and the Prague IMACS meeting, August 1998.

[1] Partially supported by the Office of Naval Research

[2] Partially supported by the Air Force Office of Scientific Research

[3] We do not assume homogeneity. Thus, we expect $n + 1$ equations in n variables to, in general, not be solvable. The resultant places a constraint on the 13 coefficient parameters that characterizes solvability.

nant is the best known method of computing a resultant. However, it is not a realistic tool for solving equations of more than one variable. Other methods exist, which usually compute not the resultant itself but rather a multiple of it, containing *extraneous factors.* The standard Macaulay resultant yields no information for our problem since both the numerator and denominator determinants are identically zero. Another resultant method is that of Dixon (generalizing Cayley), recently extended by Kapur, Saxena, and Yang [2]. The authors of that paper show that their method must work if a certain condition holds. The condition is rather strong, and in our case it is not satisfied. Yet we are able to make the method work anyway. This suggests to us that more theoretical work should be done on the Dixon-Kapur-Saxena-Yang approach, and that probably our approach here will succeed for many problems of interest.

2 The Basic Geometric Approach

The moduli space of equivalence classes of (semi-stable) six-tuples of lines in $\mathbb{P}^3$, projective 3-space, under the action of projective transformations (the matrix group PGL_4, 4×4 matrices modulo scalars) is a rational variety of dimension 9. We can thus expect to find 9 functions of the parameters defining the lines which are invariant, in the sense that they provide coordinates on a Zariski open set of the moduli space. We explain briefly how this is done. It is sufficient to work in a Zariski open subset of the set of 6-tuples of lines, so we will not hesitate to impose various general position assumptions that will become apparent below.

Let $\ell_1, \ell_2, \ell_3, \ell_4, \ell_5$, and ℓ_6 be six lines in space. We assume ℓ_1, ℓ_2, and ℓ_3 are mutually skew (our first general position assumption). Without loss of generality, we can complexify and work in complex projective space $\mathbb{P}^3$. Since lines in $\mathbb{P}^3$ are parameterized by the 4 dimensional (complex) Grassmannian, $G(2,4)$, of two-planes through the origin in (complex) four-space, an ordered six-tuple $(\ell_1, \ldots, \ell_6)$ of lines can be viewed as a point in the 24 dimensional manifold $\widehat{X} = G(2,4) \times \cdots \times G(2,4)$. The group PGL_4 of projective linear transformations acts on $\mathbb{P}^3$ sending lines to lines and hence acts on $\widehat{X}$ sending a 6-tuple of lines to another 6-tuple. We are interested in the quotient $X = \widehat{X}/PGL_4$ of $\widehat{X}$ by this action. Since PGL_4 is 15 dimensional, we expect X to have dimension 9. For various technical reasons (in fact to get a good quotient space) we must limit ourselves to an open dense subset, in fact a Zariski open subset, $\widehat{U}$ of $\widehat{X}$, and construct the quotient

$U = \widehat{U}/PGL_4$. For example, the requirement that ℓ_1, ℓ_2, and ℓ_3 be mutually skew is one of the conditions defining $\widehat{U}$.

Now lines in projective space correspond to planes through the origin in 4-space, and two skew lines correspond to two planes that intersect only in the origin. We can therefore move ℓ_1 to the z, w-plane and ℓ_2 to the x, y-plane by a 4 by 4 invertible matrix. In this position, ℓ_1 corresponds to the z-axis in space and ℓ_2 corresponds to a line at infinity that meets both the x and y axes. Specifically the points $(0:0:1:0)$ and $(0:0:0:1)$ will be on ℓ_1 and likewise, ℓ_2 will contain the points $(1:0:0:0)$ and $(0:1:0:0)$.

Having moved ℓ_1 and ℓ_2 to the above "canonical" positions, the 4×4 invertible matrices that fix these two lines have the form:

$$M = \begin{pmatrix} aa & b & \vdots & 0 & 0 \\ c & d & \vdots & 0 & 0 \\ \dots & & & & \\ 0 & 0 & \vdots & e & f \\ 0 & 0 & \vdots & g & h \end{pmatrix}, \tag{1}$$

with $ad - bc \neq 0$ and $eh - gh \neq 0$.

Now ℓ_3 is assumed to be skew to both ℓ_1 and ℓ_2. Suppose $(m_1 : n_1 : r_1 : s_1)$ and $(m_2 : n_2 : r_2 : s_2)$ are two distinct points on ℓ_3, which is then the line $\alpha(m_1 : n_1 : r_1 : s_1) + \beta(m_2 : n_2 : r_2 : s_2) = (\alpha m_1 + \beta m_2 : \alpha n_1 + \beta n_2 : \alpha r_1 + \beta r_2 : \alpha s_1 + \beta s_2)$ as $(\alpha : \beta)$ runs through all points in $\mathbb{P}^1$. If ℓ_3 were to meet ℓ_1, we would have $\alpha m_1 + \beta m_2 = 0$ and $\alpha n_1 + \beta n_2 = 0$ for some non-trivial (α, β). This can happen if and only if det $\left(\begin{smallmatrix} m_1 & m_2 \\ n_1 & n_2 \end{smallmatrix}\right) = 0$. Thus ℓ_3 being skew to ℓ_1 means det $\left(\begin{smallmatrix} m_1 & m_2 \\ n_1 & n_2 \end{smallmatrix}\right) \neq 0$. Likewise ℓ_3 skew to ℓ_2 means det $\left(\begin{smallmatrix} r_1 & r_2 \\ s_1 & s_2 \end{smallmatrix}\right) \neq 0$.

We can choose $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) = \left(\begin{smallmatrix} m_1 & m_2 \\ n_1 & n_2 \end{smallmatrix}\right)^{-1}$ and $\left(\begin{smallmatrix} e & f \\ g & h \end{smallmatrix}\right) = \left(\begin{smallmatrix} r_1 & r_2 \\ s_1 & s_2 \end{smallmatrix}\right)^{-1}$ so that the 4×4 matrix (1) above moves ℓ_3 to the line through $(1:0:1:0)$ and $(0:1:0:1)$ without moving ℓ_1 or ℓ_2.

The set of 4×4 matrices fixing ℓ_1, ℓ_2, and ℓ_3 consists of all matrices of the form

$$\begin{pmatrix} \begin{matrix} a & b \\ c & d \end{matrix} & \vdots & \bigcirc \\ \cdots & \cdots & \cdots \\ \bigcirc & \vdots & \begin{matrix} a & b \\ c & d \end{matrix} \end{pmatrix},$$

where $ad - bc \neq 0$. In other words, we are reduced to finding invariants for an action of PGL_2 on the remaining three lines.

Assume now that ℓ_4 is skew to ℓ_1 and goes through the points $(\tilde{m}_1 : \tilde{n}_1 : \tilde{r}_1 : \tilde{s}_1)$ and $(\tilde{m}_2 : \tilde{n}_2 : \tilde{r}_2 : \tilde{s}_2)$. Our group, PGL_2, which fixes ℓ_1, ℓ_2, ℓ_3, will act on ℓ_4 as follows:

$$\begin{pmatrix} \begin{matrix} a & b \\ c & d \end{matrix} & \vdots & \bigcirc \\ \cdots & \cdots & \cdots \\ \bigcirc & \vdots & \begin{matrix} a & b \\ c & d \end{matrix} \end{pmatrix} \begin{pmatrix} \tilde{\mathrm{m}}_1 & \tilde{\mathrm{m}}_2 \\ \tilde{\mathrm{n}}_1 & \tilde{\mathrm{m}}_2 \\ \cdots & \cdots \\ \tilde{\mathrm{r}}_1 & \tilde{\mathrm{r}}_2 \\ \tilde{\mathrm{s}}_1 & \tilde{\mathrm{s}}_2 \end{pmatrix}, \quad \det \begin{pmatrix} a & b \\ c & d \end{pmatrix} \neq 0,$$

where we will have $\det \begin{pmatrix} \tilde{m}_1 & \tilde{m}_2 \\ \tilde{n}_1 & \tilde{n}_2 \end{pmatrix} \neq 0$ (because ℓ_4 is skew to ℓ_1). Here the line is represented by a 4×2 matrix whose columns are the homogeneous coordinates of two points on the line. Now without loss of generality, we can assume $\begin{pmatrix} \tilde{m}_1 & \tilde{m}_2 \\ \tilde{n}_1 & \tilde{n}_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

The action yields

$$\begin{pmatrix} a\ b \\ c\ d \\ \dots \\ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \tilde{r}_1 & \tilde{r}_2 \\ \tilde{s}_1 & \tilde{s}_2 \end{pmatrix} \end{pmatrix}$$

which is a new line ℓ going through the two points given by the columns of this 4×2 matrix.

Choosing two different points on ℓ amounts to postmultiplying by an arbitrary invertible 2 by 2 matrix. We can choose $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)^{-1}$ for this purpose. This means that ℓ can be given by

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \dots & \dots \\ & N \end{pmatrix}$$

where N is the 2×2 matrix

$$N = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \tilde{r}_1 & \tilde{r}_2 \\ \tilde{s}_1 & \tilde{s}_2 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} .$$

In other words, the orbit of ℓ is just the orbit of $N = \left(\begin{smallmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{smallmatrix}\right)$ under conjugation.

The orbits with N a scalar matrix, $N = \left(\begin{smallmatrix} \tau & 0 \\ 0 & \tau \end{smallmatrix}\right)$, are just points, i.e. they are fixed points of the action. The nature of the orbits with N not scalar depends on the Jordan form of N. The possibilities are:

Case 1: $\left(\begin{smallmatrix} \lambda & 1 \\ 0 & \lambda \end{smallmatrix}\right)$. Here the orbit is two dimensional since the matrices which fix $\left(\begin{smallmatrix} \lambda & 1 \\ 0 & \lambda \end{smallmatrix}\right)$ under conjugation $\Big($i.e. commute with $\left(\begin{smallmatrix} \lambda & 1 \\ 0 & \lambda \end{smallmatrix}\right)\Big)$ are of the form $\left(\begin{smallmatrix} a & b \\ 0 & a \end{smallmatrix}\right)$ $a \neq 0$.

Case 2: $\left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right)$ with $\lambda_1 \neq \lambda_2$. Here the orbit is two dimensional since the matrices which fix $\left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right)$ under conjugation are of the form $\left(\begin{smallmatrix} a & 0 \\ 0 & d \end{smallmatrix}\right)$.

We will assume that ℓ_4 is in case 2, which is the generic case. In other words, we will assume that $\left(\begin{smallmatrix} \tilde{r}_1 & \tilde{r}_2 \\ \tilde{s}_1 & \tilde{s}_2 \end{smallmatrix}\right)$ has distinct (unequal) eigenvalues. Thus we can move ℓ_4 to either the line through $(1:0:\lambda_1:0)$ and $(0:1:0:\lambda_2)$ or the line through $(1:0:\lambda_2:0)$ and $(0:1:0:\lambda_1)$. This ambiguity arises because Jordan form in this case isn't unique! It can be either $\left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right)$ or $\left(\begin{smallmatrix} \lambda_2 & 0 \\ 0 & \lambda_1 \end{smallmatrix}\right)$. Now fix ℓ_4 to be one of these two lines. (It doesn't matter which. Moreover we will never in practice need to make a choice between the two.)

The transformations that fix ℓ_1, ℓ_2, ℓ_3 and ℓ_4 take the form

$$\begin{pmatrix} \begin{matrix} a & 0 \\ 0 & d \end{matrix} & \vdots & \bigcirc \\ \cdots & \cdots & \cdots \\ \bigcirc & \vdots & \begin{matrix} a & 0 \\ 0 & d \end{matrix} \end{pmatrix}$$

modulo scalar matrices. Thus we have essentially reduced the group to $\mathbb{C}^* \times \mathbb{C}^*/\mathbb{C}^* \cong \mathbb{C}^*$ where the $\mathbb{C}^*$ in the quotient is embedded diagonally in $\mathbb{C}^* \times \mathbb{C}^*$. We say "essentially", because there is still a $\mathbb{Z}_2$-action lurking that switches $\left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right)$ with $\left(\begin{smallmatrix} \lambda_2 & 0 \\ 0 & \lambda_1 \end{smallmatrix}\right)$. This is accounted for below.

If we assume, in addition to ℓ_4, that ℓ_5 and ℓ_6 are skew to ℓ_1, then we can reinterpret our problem as one of finding invariants for the action of PGL_2 on the three-fold product of 2×2 matrices by conjugation in each factor; specifically

$$(N_4, N_5, N_6) \longrightarrow (AN_4A^{-1}, AN_5A^{-1}, AN_6A^{-1})$$

for A an invertible 2×2 matrix representing an element of PGL_2. Here ℓ_i, $i = 4, 5, 6$ is the line passing through the points in $\mathbb{P}^3$ which are the columns of the 4×2 matrix

$$\begin{pmatrix} 1\ 0 \\ 0\ 1 \\ \cdots \\ N_i \end{pmatrix}.$$

A known set of invariants are the traces of $N_1, N_2, N_3, N_1^2, N_2^2, N_3^2, N_1N_2, N_1N_3, N_2N_3$ and $N_1N_2N_3$ which have one relation among them. We take a different approach. Since we have assumed that N_4 has distinct eigenvalues, we can find an A which conjugates N_4 to either $\left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right)$ or $\left(\begin{smallmatrix} \lambda_2 & 0 \\ 0 & \lambda_1 \end{smallmatrix}\right)$.

Consider the following subgroup G of PGL_2:

$$G = \left\{ \begin{pmatrix} a\ 0 \\ 0\ d \end{pmatrix} a \neq 0, d \neq 0 \quad \text{or} \quad \begin{pmatrix} 0\ b \\ c\ 0 \end{pmatrix} b \neq 0, c \neq 0 \right\} \text{ mod scalars}$$

Action by G leaves N_4 in diagonal (Jordan) form. Thus we can reduce our action to one of G acting on $(\mathbb{C} \times \mathbb{C} - \Delta) \times N_5 \times N_6$ where Δ is the diagonal in $\mathbb{C} \times \mathbb{C}$ and where we identify (λ_1, λ_2), $\lambda_1 \neq \lambda_2$, in $\mathbb{C} \times \mathbb{C} - \Delta$ with $\left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right)$.

We now try to move ℓ_5 to a canonical position using just the $\mathbb{C}^*$ action of $\left(\begin{smallmatrix} a & 0 \\ 0 & d \end{smallmatrix}\right)$ mod scalars. (This doesn't depend on our choice for the position of ℓ_4.) If we assume ℓ_5 is skew to ℓ_1 so that it can be taken to go through the points $(1 : 0 : n_{11} : n_{21})$ and $(0 : 1 : n_{12} : n_{22})$, then the group acts via

$$\left(\begin{array}{cc:cc} a & 0 & & \\ 0 & d & \multicolumn{2}{c}{\bigcirc} \\ \hdashline \multicolumn{2}{c:}{\bigcirc} & a & 0 \\ & & 0 & d \end{array}\right) \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ n_{11} & n_{12} \\ n_{21} & n_{22} \end{pmatrix} = \begin{pmatrix} a & 0 \\ 0 & d \\ an_{11} & an_{12} \\ dn_{21} & dn_{22} \end{pmatrix}$$

which is the same line as

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ n_{11} & \frac{a}{d}n_{12} \\ \frac{d}{a}n_{21} & n_{22} \end{pmatrix}$$

We will assume that ℓ_5 is sufficiently generic so that $n_{12} \neq 0$ and $n_{21} \neq 0$. We can then normalize $\left(\begin{smallmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{smallmatrix}\right)$ so that $n_{12} = n_{21} = g \neq 0$, by choosing $\frac{d}{a} = \sqrt{\frac{n_{12}}{n_{21}}}$.

Note that

$$\begin{pmatrix} 0 & b \\ c & 0 \end{pmatrix} \begin{pmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{pmatrix} \begin{pmatrix} 0 & b \\ c & 0 \end{pmatrix}^{-1} = \begin{pmatrix} n_{22} & \frac{b}{c}n_{21} \\ \frac{c}{b}n_{12} & n_{11} \end{pmatrix}$$

Thus if we normalize N_5 to $\left(\begin{smallmatrix} n_{11} & g \\ g & n_{22} \end{smallmatrix}\right)$, $g \neq 0$, then the elements in the subgroup G which preserve our "normal form", namely that N_4 be diagonal and that N_5 have equal off-diagonal elements (non-zero), form a subgroup H:

$$H = \left\{ \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} a \neq 0 \right\} \cup \left\{ \begin{pmatrix} a & 0 \\ 0 & -a \end{pmatrix} a \neq 0 \right\} \cup \left\{ \begin{pmatrix} 0 & a \\ a & 0 \end{pmatrix} a \neq 0 \right\} \cup \left\{ \begin{pmatrix} 0 & -a \\ a & 0 \end{pmatrix} a \neq 0 \right\}$$

mod scalars. Clearly $H < PGL_2$ is a finite group isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.

We are therefore reduced to the action of this finite group H on $U = (\mathbb{C} \times \mathbb{C} - \Delta) \times (\mathbb{C}^2 \times \mathbb{C}^*) \times \mathbb{C}^4 \subset \mathbb{C}^9$ with coordinates $(\lambda_1, \lambda_2, n_{11}, n_{22}, g, p_{11}, p_{12}, p_{21}, p_{22})$ where ℓ_6 is assumed skew to ℓ_1 so it can be represented by

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}.$$

Note that $U \subset \mathbb{C}^9$ is defined by $g \neq 0$ and $\lambda_1 \neq \lambda_2$, i.e. by $g(\lambda_1 - \lambda_2) \neq 0$. Thus U is an affine variety with coordinate ring

$$R = \mathbb{C}\left[\lambda_1, \lambda_2, \frac{1}{\lambda_1 - \lambda_2}, n_{11}, n_{22}, g, \frac{1}{g}, p_{11}, p_{12}, p_{21}, p_{22}\right]$$

and function field

$$F = \mathbb{C}(\lambda_1, \lambda_{,2}, n_{11}, n_{22}, g, p_{11}, p_{12}, p_{21}, p_{22}).$$

The desired quotient variety U/H is affine with coordinate ring given by the invariants R^H and function field given by the fixed field F^H. One can show that this variety is rational, i.e. F^H is a field of rational functions in nine algebraically independent quantities – the desired invariants.

To generate the desired equations one works with the 9 "invariants" λ_1, λ_2, n_{11}, n_{22}, g, p_{11}, p_{12},p_{21}, and p_{22} (modulo the action of $H \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$). In the plane one will have 4 standard invariants q_1, q_2, q_3, q_4 which are rational expressions in the coefficients of the 6 lines $a_i x + b_i y + c_i = 0$ viewed in $\mathbb{P}^2$ as the points $(a_i : b_i : c_i)$ $i = 1, \ldots, 6$. These are $q_1 = \frac{q_{5,0}}{q_{5,2}}, q_2 = \frac{q_{5,1}}{q_{5,2}}, q_3 = \frac{q_{6,0}}{q_{6,2}}, q_4 = \frac{q_{6,1}}{q_{6,2}}$ in the notation of [6].

Now one can use the above invariants, and the description of the relationship between 6 lines in 3D and 6 lines in 2D as a correspondence (in the sense of algebraic geometry), to produce a system of 4 equations in $17 = 9 + 4 + 4$ variables, nine 3D invariants, four 2D invariants, and four variables which represent an invertible 2×2 matrix

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \qquad a_{11}a_{22} - a_{21}a_{12} \neq 0$$

acting by conjugation as above on

$$\left(\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \begin{pmatrix} n_{11} & g \\ g & n_{22} \end{pmatrix}, \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} \right).$$

The key result is that it can be shown that these 3D configurations for fixed $\lambda_1, \lambda_2, \ldots, p_{22}$ and variable a_{ij} sweep out a Zariski open set of the 3-dimension set of all possible 2D equivalence classes obtainable by all possible perspective projections. The resulting four equations appear in the appendix. Note they are linear in the 2D invariants, quadratic in the 3D invariants and homogeneous quartic in the a_{ij}. By eliminating the a_{ij} one arrives at the desired object/image equation. This is the problem we take up. One complication is that the system always has degenerate solutions a_{ij} where $a_{11}a_{22} - a_{12}a_{21} = 0$. This is what causes the classical Macaulay resultant to fail.

The reader may wonder about the fact that $\lambda_1, \lambda_2, \ldots, p_{22}$ aren't quite invariant and that a $\mathbb{Z}_2 \times \mathbb{Z}_2$ action still lurks. This causes no serious problem. In fact a test of the final single resultant equation relating $\lambda_1, \ldots, p_{22}$, $q_1, \ldots, q_4$ shows it to be invariant under this action. For simplicity we stick with these "not quite invariant" invariants.

3 The Basic Computational Approach

We thus have:

- Nine 3D parameters: $\lambda_1, \lambda_2, n_{11}, n_{22}, g, p_{11}, p_{12}, p_{21}, p_{22}$.
- Four 2D parameters: q_1, q_2, q_3, q_4.
- Three (initially four) conversion variables: $(a_{11} = 1), a_{12}, a_{21}, a_{22}$.
- Four quartic equations (see appendix) in the variables a_{ij} and the 13 parameters.

The four equations have the useful property that q_i appears only in equation i, and only with degree 1. It is therefore quite easy to solve for each q_i in terms of the other variables. While this is an unnatural thing to do from the standpoint of the Six Line Problem, we will exploit it later to check answers.

The Cayley-Dixon method to eliminate the three variables a_{ij} may be summarized as follows (see [2] for details):

- Adjoin three new auxiliary variables, r, s, t.
- Create the Dixon matrix, DM. Then compute the Dixon polynomial

$$dm = \frac{Det(DM)}{(r - a_{12})(s - a_{21})(t - a_{22})}$$

- If desired, we may work with a certain "fixed object," i.e. a set of numerical 3D invariants. Stiller provided an algorithm for creating such test cases of 3D (and corresponding 2D) data sets. The data are integers or rational numbers. We may then substitute into dm some or all of the nine 3D numerical values. This reduces the size and complexity of dm.

- Create the second Dixon matrix by extracting coefficients from dm in a certain way. These coefficients are polynomials in the four 2D parameters $q_i, i = 1, \ldots, 4$ and those 3D parameters that remain from the previous step. It is a 105×105 matrix.
- The determinant of this second matrix is the classical Dixon Resultant. If there is a common solution of the original system of four equations, then this determinant must be 0. Ideally that provides an equation that must be satisfied by the parameters. However, in our case (and in many others) it is identically 0.

But that is not the end of the story. The Kapur-Saxena-Yang ("KSY") method continues:

- Extract the non-zero rows and columns from the second matrix. This leaves a 51×56 matrix. Call this *the third matrix.*
- If a certain condition holds on the third matrix, compute the determinant of *any* maximal rank submatrix. These polynomials must vanish if the original system has a solution.

In other words, these necessarily nonzero polynomials, any of which we will call ksy, play the role of the classical Dixon Resultant. We will have more to say about the "certain condition" in section five.

4 Phase One of the Computation

Unless some of the numerical 3D parameters from a "real" object are substituted into dm before the creation of the second Dixon matrix, the polynomials ksy will be hopelessly large for any existing computer system. In the first phase of the project, we substituted rational values for all nine 3D parameters, thus reducing the goal to computing a resultant for that one object – a polynomial in the four 2D invariants q_1, q_2, q_3, q_4.

- Input the numerical (rational or integral) data for the nine 3D parameters. For example $\lambda_1, \lambda_2, \ldots, p_{22} = 3, 4, 2, 3, 2, 3, 1, 2, 1/2$.
- Compute ksy, a polynomial in the four parameters q_1, q_2, q_3, q_4.
- To determine if a set of 2D data q_1, q_2, q_3, q_4 "matches" the 3D object, substitute the four numerical values into ksy and see if the result is 0. If it is not, the 2D set cannot be a perspective projection of the 3D object.

An important simplification results by reconsidering what is meant by "the result is 0." Recall that the coefficients of the polynomial ksy are rational numbers. Since we seek solutions of $ksy = 0$, we can clear out denominators and assume that all coefficients are integers. Rather than work over the ring of integers, we can save enormously in both time and space if we choose a moderately large prime number p at random and reduce all the equations modulo this prime. We are then working over the field $\mathbb{Z}_p$, and it is sufficient to test a candidate set of 2D parameters s_1, s_2, s_3, s_4 by reducing them modulo p and checking $ksy(s_1, s_2, s_3, s_4) = 0$ in $\mathbb{Z}_p$. The resulting algorithm is probabalistic, with an enormously high probability of success. An incorrect set of parameters will not pass the test unless $ksy(s_1, s_2, s_3, s_4)$ is a multiple of p, which is extremely unlikely. A correct set of parameters will pass it unless one of the parameters is a fraction with denominator a multiple of p. The probability of a mistaken judgment can be further reduced by simply doing the algorithm twice with two different primes.

Lewis wrote programs to create the third matrix and compute ksy in his computer algebra system *Fermat* [3]. One method is to compute the product of the pivot elements that come up as one normalizes (say, into the Hermite form) the third matrix. One can learn the rank of this matrix very easily by plugging in integers at random for the four q_i parameters and computing a matrix normal form. The matrix has rank 26. Therefore, ksy is the product of 26 terms that will appear on the main diagonal as the matrix is normalized. Depending on the algorithm, these terms may not be all polynomials. Nevertheless, the product of all 26 will be a polynomial.

The row and column reductions went well, up to the 17th row/column. Beyond that the complexity of the computation becomes overwhelming. However, it is not necessary to continue the normalization algorithm. Recall that we have reason to think that any maximal rank submatrix will do. By substituting random integers for three of the parameters q_i it is easy to discover a 26×26 maximal rank submatrix. ksy is just the determinant of this *fourth matrix*. Since all its entries are (4 variable) polynomials, the determinant algorithm in *Fermat* (there are several) which works by recursive LaGrange interpolation is suitable. It completed in three hours and produced a ksy with around 500,000 terms. (All times in this paper are for a 233mhz Macintosh with 604e chip.) It had degree 26 or 25 in each of the four q_i. As an ASCII file, this ksy occupied a file of 3.5 megabytes.

To evaluate ksy at 4 numerical values took about 2 seconds, so this is feasible in real time. Extensive testing with 2D data sets, valid and invalid, verified the correctness of ksy. This was all done using the prime 44449. Using 41999 produced essentially the same results.

Wishing to look more closely at ksy, we returned to the idea of computing it by row reductions on the third matrix, over the field $\mathbb{Q}$, rather than $\mathbb{Z}_p$. The first nine diagonal pivot elements were enlightening:

$$q_4 - q_2,\ q_4 - q_2,\ q_4 - q_2,\ q_2(q_4 + q_2),\ q_2(q_4 + q_2),\ (q_3 + q_1)(q_4 - q_2),$$

$$(q_3 - q_1)(q_4 - q_2),\ (q_3 - 2q_1)(q_4 - q_2),\ (q_3 - 1/2q_1 + 1/6)(q_4 - q_2)$$

This suggests, but does not prove, that ksy has many simple factors. After much testing Lewis verified that

$$q_4(q_4 - q_2)^4(q_3 - q_1)^4 q_2^2(q_3 - 1/2q_1 + 1/6) \tag{2}$$

is a factor. One of the *Fermat* determinant algorithms can take advantage of a known factor. It then computed the rest of ksy (the other factor) in only 25 minutes, down from the original three hours. This "reduced ksy" has 100,000 terms and occupies only 670K of disk space. Numerical tests show that the actual resultant is indeed a factor of the reduced ksy.

Even more extraneous factors can be removed from the reduced ksy. First, since the resultant must be irreducible, we may divide out all the contents of ksy. Secondly, with different maximal rank submatrices, simple variations of (1) divide their determinants (and this remains true for different choices of 3D invariants, not just the values used here, $3, 4, 2, 3, 2, 3, 1, 2, 1/2$). Thus, it is not hard to compute another reduced determinant ksy', and the true resultant should be a factor of $GCD(ksy, ksy')$. We have therefore the following algorithm:

$res := ksy$;
REPEAT
  Compute new reduced ksy using a new maximal rank submatrix;
  $ksy := ksy$ / all contents(ksy);
  $res := GCD(res, ksy)$
UNTIL DONE

After five repetitions of this loop, the polynomial *res* contained only three hundred terms! It was small enough to be factored with standard algorithms. The factor that vanishes on a known 2D data set is:

$$sixline = \quad q_1^2 q_4^2 - 2q_1 q_4^2 + 8q_4^2 + 6q_2 q_3 q_4 + 12q_1 q_3 q_4 - 60q_3 q_4 - 2q_1 q_2 q_4 + 2q_1^2 q_4 +$$

$$28q_1 q_4 - q_2^2 q_3^2 + 8q_3^2 - 2q_2^2 q_3 - 14q_1 q_2 q_3 + 60q_2 q_3 - 16q_1 q_3 - 8q_2^2 - 28q_1 q_2 + 8q_1^2$$

This was all done over a finite field, $\mathbb{Z}_p$. But the coefficients above are suggestively small integers. Indeed, *this is the actual resultant over $\mathbb{Q}$, not just over $\mathbb{Z}_p$.* That is easy to prove: recall that each q_i may be solved for in $d_i = 0$, then just substitute into *sixline* each q_i with its formula in terms of the other variables. The expression evaluates to 0. It is as if we had set out to use the Chinese Remainder Theorem to find the resultant over $\mathbb{Q}$, and discovered that one prime was enough.

In summary, the polynomial *sixline* provides the solution to the problem for the given particular 3D data set. If any set of 2D invariants be presented in the future, plug them into *sixline*. If the result is not 0, then they do not represent a perspective projection of the original object.

Now, our entire method, which we know has worked because *sixline* is verifiably correct, is based on the Kapur-Saxena-Yang idea of computing the determinant of a maximal rank submatrix. In [2] they show that the resultant must be a factor of any such determinant, provided that a certain condition holds. This (sufficient) condition is that some column in the 105×105 second matrix be linearly independent of all the others. However, in our case the condition fails! Yet the method works anyway.

It may be asked why it was necessary to produce the polynomial *sixline* at all. Instead, one could simply take a candidate set of 2D invariants and plug them into the third matrix, whose rank is known to be 26. If the rank drops, which is surely a simple thing to check, then the determinant of every maximal rank submatrix must vanish on that 2D set.

To answer, there are several reasons why the derivation of the polynomial *sixline* is very desirable:

• It is not clear that the g.c.d. of all the maximal rank submatrices is exactly the resultant. If it is not, there may be spurious zeros.

• The 2D invariants $\{q_1, q_2, q_3, q_4\}$ will probably be obtained by extracting and measuring lines on photographs. It is necessary to match the six 2D lines with the six lines on the original 3D object. This will probably require testing all $6! = 720$ possible permutations. The time saved in plugging the $\{q_i\}$ into *sixline* versus finding the rank of the third matrix may not be significant, but it will be multiplied by 720.

• We have been assuming that the 2D invariants are known exactly, but if they come from measurements, there may be errors. Error analysis is much easier if it is based on the polynomial *sixline*.

• In the next sections we generalize our method to produce a completely symbolic version of *sixline*; i.e., we forgo plugging in numerical values for the nine 3D parameters, and all 13 variables appear in the resultant.

• It is possible to consider recognition of n lines by similar methods. However 6 is the minimum for the problem to be meaningful; sets of 5 or fewer lines cannot be distinguished in this manner.

5 Phase Two

In Phase One we substituted numerical values for all parameters except q_1, q_2, q_3, q_4. Lewis then redid the computations keeping various other subsets of the parameters, such as the four p_{ij}, the set $\{\lambda_1, \lambda_2, g, n_{11}, n_{22}\}$, and various combinations of the preceding with some of the q_i. In this way we learned the degree of the resultant in all of its parameters. Each degree is either 1 or 2. We learned also that if we order the parameters so that the four q_i have highest precedence, the leading term is $f(\lambda_1, \lambda_2, p_{11}, p_{22}, p_{12}, p_{21})q_1^2q_4^2$, for some polynomial f in the indicated parameters only.

6 Phase Three

The work done in Phase One constitutes a viable solution to the Six-Line Problem, given the 3D data of an object. But we want to compute the complete resultant for all objects, in all 13 parameters.

Grosshans, Gleason, Williams, and Hirsch [1] were the first to compute this polynomial res, using invariant theory and experimenting with lots of numerical cases, observing various dependencies among the variables and exploiting various symmetries in the equation. They found a res with 239 terms. The final answer is quartic in the 3D invariants and quartic in the 2D invariants, yielding total degree 8. An alternative approach by Stiller and Ma used interpolation by generating a large number of "matching" object image pairs and exploiting the degree bounds predicted by Lewis. How do we know this polynomial is correct? Recall that each q_i occurs only in equation $d_i = 0$ and can be solved for, yielding a rational expression, for example

$$\begin{aligned}
q_1 = (&g\ \lambda_2\ a_{22}^3 - g\ \lambda_1\ a_{12}\ a_{21}\ a_{22}^2 - n_{22}\ \lambda_2\ a_{21}\ a_{22}^2 + n_{11}\ \lambda_2\ a_{21}\ a_{22}^2 + 2\ g\ \lambda_2\ a_{12}\ a_{22}^2 - \\
&g\ \lambda_1\ a_{12}\ a_{22}^2 - n_{11}\ n_{22}\ \lambda_2\ a_{22}^2 + n_{11}\ \lambda_2\ a_{22}^2 + g^2\ \lambda_2\ a_{22}^2 + n_{22}\ \lambda_1\ a_{12}\ a_{21}^2\ a_{22} - n_{11}\ \lambda_1\ a_{12}\ a_{21}^2\ a_{22} - \\
&g\ \lambda_2\ a_{21}^2\ a_{22} - g\ \lambda_1\ a_{12}^2\ a_{21}\ a_{22} + n_{11}\ n_{22}\ \lambda_2\ a_{12}\ a_{21}\ a_{22} - 2\ n_{22}\ \lambda_2\ a_{12}\ a_{21}\ a_{22} + \\
&n_{11}\ \lambda_2\ a_{12}\ a_{21}\ a_{22} - g^2\ \lambda_2\ a_{12}\ a_{21}\ a_{22} + n_{11}\ n_{22}\ \lambda_1\ a_{12}\ a_{21}\ a_{22} + n_{22}\ \lambda_1\ a_{12}\ a_{21}\ a_{22} - \\
&2\ n_{11}\ \lambda_1\ a_{12}\ a_{21}\ a_{22} - g^2\ \lambda_1\ a_{12}\ a_{21}\ a_{22} - g\ \lambda_2\ a_{21}\ a_{22} + g\ \lambda_2\ a_{12}^2\ a_{22} - g\ \lambda_1\ a_{12}^2\ a_{22} - \\
&n_{11}\ n_{22}\ \lambda_2\ a_{12}\ a_{22} + n_{11}\ \lambda_2\ a_{12}\ a_{22} + g^2\ \lambda_2\ a_{12}\ a_{22} + n_{11}\ n_{22}\ \lambda_1\ a_{12}\ a_{22} - n_{11}\ \lambda_1\ a_{12}\ a_{22} - \\
&g^2\ \lambda_1\ a_{12}\ a_{22} + g\ \lambda_1\ a_{12}\ a_{21}^3 - n_{11}\ n_{22}\ \lambda_1\ a_{12}^2\ a_{21}^2 + n_{22}\ \lambda_1\ a_{12}^2\ a_{21}^2 + g^2\ \lambda_1\ a_{12}^2\ a_{21}^2 - g\ \lambda_2\ a_{12}\ a_{21}^2 + \\
&2\ g\ \lambda_1\ a_{12}\ a_{21}^2 + n_{11}\ n_{22}\ \lambda_2\ a_{12}^2\ a_{21} - n_{22}\ \lambda_2\ a_{12}^2\ a_{21} - g^2\ \lambda_2\ a_{12}^2\ a_{21} - n_{11}\ n_{22}\ \lambda_1\ a_{12}^2\ a_{21} + \\
&n_{22}\ \lambda_1\ a_{12}^2\ a_{21} + g^2\ \lambda_1\ a_{12}^2\ a_{21} - g\ \lambda_2\ a_{12}\ a_{21} + g\ \lambda_1\ a_{12}\ a_{21})\ /\ (g\ \lambda_2\ a_{12}\ a_{21}\ a_{22}^2 - \\
&g\ \lambda_1\ a_{12}\ a_{21}\ a_{22}^2 - n_{22}\ \lambda_2\ a_{21}\ a_{22}^2 + n_{22}\ \lambda_1\ a_{21}\ a_{22}^2 + g\ \lambda_1\ \lambda_2\ a_{12}\ a_{22}^2 - g\ \lambda_1\ a_{12}\ a_{22}^2 - \\
&n_{22}\ \lambda_1\ \lambda_2\ a_{22}^2 + n_{22}\ \lambda_1\ a_{22}^2 + n_{11}\ \lambda_2\ a_{12}\ a_{21}^2\ a_{22} - n_{11}\ \lambda_1\ a_{12}\ a_{21}^2\ a_{22} - g\ \lambda_2\ a_{21}^2\ a_{22} + \\
&g\ \lambda_1\ a_{21}^2\ a_{22} - g\ \lambda_1\ \lambda_2\ a_{12}^2\ a_{21}\ a_{22} + 2\ g\ \lambda_2\ a_{12}^2\ a_{21}\ a_{22} - g\ \lambda_1\ a_{12}^2\ a_{21}\ a_{22} + n_{22}\ \lambda_1\ \lambda_2\ a_{12}\ a_{21}\ a_{22} + \\
&n_{11}\ \lambda_1\ \lambda_2\ a_{12}\ a_{21}\ a_{22} - 2\ n_{22}\ \lambda_2\ a_{12}\ a_{21}\ a_{22} + n_{11}\ \lambda_2\ a_{12}\ a_{21}\ a_{22} + n_{22}\ \lambda_1\ a_{12}\ a_{21}\ a_{22} - \\
&2\ n_{11}\ \lambda_1\ a_{12}\ a_{21}\ a_{22} - g\ \lambda_1\ \lambda_2\ a_{21}\ a_{22} - g\ \lambda_2\ a_{21}\ a_{22} + 2\ g\ \lambda_1\ a_{21}\ a_{22} + g\ \lambda_1\ \lambda_2\ a_{12}^2\ a_{22} - \\
&g\ \lambda_1\ a_{12}^2\ a_{22} - n_{22}\ \lambda_1\ \lambda_2\ a_{12}\ a_{22} + n_{11}\ \lambda_1\ \lambda_2\ a_{12}\ a_{22} + n_{22}\ \lambda_1\ a_{12}\ a_{22} - n_{11}\ \lambda_1\ a_{12}\ a_{22} - \\
&g\ \lambda_1\ \lambda_2\ a_{22} + g\ \lambda_1\ a_{22} - n_{11}\ \lambda_1\ \lambda_2\ a_{12}^2\ a_{21}^2 + n_{11}\ \lambda_2\ a_{12}^2\ a_{21}^2 + g\ \lambda_1\ \lambda_2\ a_{12}\ a_{21}^2 - g\ \lambda_2\ a_{12}\ a_{21}^2 - \\
&g\ \lambda_1\ \lambda_2\ a_{12}^3\ a_{21} + g\ \lambda_2\ a_{12}^3\ a_{21} + n_{22}\ \lambda_1\ \lambda_2\ a_{12}^2\ a_{21} - n_{11}\ \lambda_1\ \lambda_2\ a_{12}^2\ a_{21} - n_{22}\ \lambda_2\ a_{12}^2\ a_{21} + \\
&n_{11}\ \lambda_2\ a_{12}^2\ a_{21} + g\ \lambda_1\ \lambda_2\ a_{12}\ a_{21} - g\ \lambda_2\ a_{12}\ a_{21})
\end{aligned}$$

Lewis simply substituted for each q_i its expression as above into res and checked that the result is identically (symbolically) 0. (200 meg of RAM, 11 minutes, using *Fermat*. No other computer algebra system that we are aware of could do this computation.)

We felt strongly that the Dixon-KSY method ought to work as well to compute res. But recall that even after plugging in integers for 9 of the 13 parameters, the KSY method produced a 500,000 term answer, almost all of which was spurious factors. Brute force is therefore rejected. Several ideas led eventually to the solution.

The first idea, due to George Nakos, is as follows. Instead of applying the KSY method to four equations $\{d_i = 0\}$ to eliminate the three variables $\{a_{12}, a_{21}, a_{22}\}$, do it in stages:

- Apply KSY to $\{d_1, d_2, d_3\}$ eliminating 2 variables, obtaining a polynomial y_1 that still has a_{12}.
- Apply KSY to $\{d_2, d_3, d_4\}$ eliminating 2 variables, obtaining a polynomial y_2 that still has a_{12}.
- Apply KSY to $\{y_1, y_2\}$ to get the final res.

However, it's not that easy. Each y_i would have had many millions of terms, making the third step hopeless. Lewis applied two fairly standard ideas to reduce the size of each y_i.

- Interpolation: Plug in authentic 3D values for some of the parameters. Run the above three steps with enough such sets of values, then construct res with standard interpolation techniques.
- Quotient Ring: We know that the final answer res is of low degree in each parameter; for example, it is degree 2 in g and degree 1 in n_{11}. Analogously to working over $\mathbb{Z}_p$ instead of $\mathbb{Z}$, we could work modulo a cubic polynomial in g and a quadratic polynomial in n_{11}. This eliminates high degree (in g and n_{11}) intermediate results while, ideally, not changing the final answer. *Fermat* allows one to work easily and efficiently over such fields.

While either technique alone might have sufficed, we decided to use both. There is a problem, however, with the second technique, the well known *leading coefficient problem.* Suppose R is a polynomial ring, say $R = F[a, b, c, \ldots]$. Let $I \subset R$ be an ideal such that R/I is a field. We wish to compute in $(R/I)[x, y, z, \ldots]$ instead of $R[x, y, z, \ldots]$. When working over such a quotient field, algorithms such as polynomial g. c. d. dispense with leading coefficients involving the field variables $a, b, c, \ldots$. The leading coefficients are divided through to produce "pseudo-monic" polynomials. This makes reconstruction of the actual answer in $R[x, y, z, \dots]$ problematic. But due to the work accomplished in Phase Two, we know that the leading term relative to the q_i is $f(\lambda_1, \lambda_2, p_{11}, p_{22}, p_{12}, p_{21})q_1^2 q_4^2$, for some polynomial f in the indicated parameters. Therefore, by choosing to mod out by g and n_{11} we avoid this problem. (We could mod out by n_{22} in addition, but that greatly slows down the computations in *Fermat*.)

In summary, we chose to work modulo $g^3 - 3$ and $n_{11}^2 - 7$, and over the prime $p = 17041$. $\mathbb{Z}_p[g, n_{11}] \ / < g^3 - 3, n_{11}^2 - 7 >$ is a field. We interpolated for λ_1, λ_2 and n_{22}. We know from Phase Two that the answer is of degree 1 in each of the latter parameters, so we need to run the three steps eight times.

However, it still doesn't work. y_1 and y_2 each have about 300,000 terms and, worse yet, are of high degree (≥ 30) in a_{12}. That makes the third step unworkable. The problem is solved by recalling from Phase One the idea of dividing out by the contents. Compute y_1 (12 minutes). Then compute all its contents and divide out by them (123 minutes; 132 meg RAM). The result has only 90 terms! Repeat for y_2. Then do the third step (about one minute). This produces a preliminary answer with a set of values plugged in for λ_1, λ_2 and n_{22}. We then repeat seven times and interpolate for the final answer. In doing so, one final problem arises. Because the contents were divided out often, the eight preliminary answers may be missing leading numerical coefficients – another incarnation of the leading coefficient problem. Especially likely is that one or more needs to be multiplied by -1. Since we know that the final answer is a polynomial with integer coefficients, it is easy to experiment and compute the right answer.

7 Conclusion

Elimination in stages using the Cayley-Dixon-Kapur-Saxena-Yang method succeeded for two reasons:

(1) The final answer is of low degree in most of its variables (in fact, all of them).
(2) At each stage, polynomials are produced that are multiples of the resultant, with huge spurious factors. *But the resultant is the only factor involving all the variables.* It can therefore be found by dividing out all the contents.

Unless there is something very special about the equations that came up in this problem, it is reasonable to conjecture that our successive elimination method with KSY may be applicable to other large polynomial systems.

8 Appendix. The Four Equations

- Nine 3D parameters: $\lambda_1, \lambda_2, n_{11}, n_{22}, g, p_{11}, p_{12}, p_{21}, p_{22}$.
- Four 2D parameters: q_1, q_2, q_3, q_4.
- Four conversion variables (later we set $a_{11} = 1$): $a_{11}, a_{12}, a_{21}, a_{22}$.
- Four equations d_1, d_2, d_3, d_4 in the 3 variables a_{ij} and the 13 parameters. Note that q_i appears only in d_i and only with exponent 1.

$$d_1 = (g\, a_{11}^2 + g\, a_{11}\, a_{21} + n_{22}\, a_{11}\, a_{12} + a_{11}\, a_{22}\, n_{22} - n_{11}\, a_{11}\, a_{12} - a_{12}\, a_{21}\, n_{11} - g\, a_{12}^2 - g\, a_{12}\, a_{22})\,(-\lambda_2\, a_{12}\, a_{21} - \lambda_2\, a_{21}\, a_{22} + \lambda_1\, a_{11}\, a_{22} + \lambda_1\, a_{21}\, a_{22} - \lambda_1\, \lambda_2\, a_{11}\, a_{22} + \lambda_1\, \lambda_2\, a_{12}\, a_{21})\, q_1 - (-g\, a_{11}\, a_{21} - g\, a_{21}^2 - n_{22}\, a_{12}\, a_{21} - n_{22}\, a_{21}\, a_{22} + n_{11}\, a_{11}\, a_{22} + n_{11}\, a_{21}\, a_{22} + g\, a_{12}\, a_{22} + g\, a_{22}^2 - a_{11}\, a_{22}\, n_{11}\, n_{22} + a_{11}\, a_{22}\, g^2 + a_{12}\, a_{21}\, n_{11}\, n_{22} - a_{12}\, a_{21}\, g^2)\,(\lambda_2\, a_{11}\, a_{12} + a_{11}\, a_{22}\, \lambda_2 - \lambda_1\, a_{11}\, a_{12} - a_{12}\, a_{21}\, \lambda_1),$$

$$d_2 = (g\, a_{11}^2 + g\, a_{11}\, a_{21} + n_{22}\, a_{11}\, a_{12} + a_{11}\, a_{22}\, n_{22} - n_{11}\, a_{11}\, a_{12} - a_{12}\, a_{21}\, n_{11} - g\, a_{12}^2 - g\, a_{12}\, a_{22})\,(a_{11}\, a_{22} - a_{12}\, a_{21} - a_{11}\, a_{22}\, \lambda_2 + a_{12}\, a_{21}\, \lambda_1)\, q_2 - (a_{11}\, a_{22} - a_{12}\, a_{21} - g\, a_{11}\, a_{21} - a_{11}\, a_{22}\, n_{22} + a_{12}\, a_{21}\, n_{11} + g\, a_{12}\, a_{22})\,(\lambda_2\, a_{11}\, a_{12} + a_{11}\, a_{22}\, \lambda_2 - \lambda_1\, a_{11}\, a_{12} - a_{12}\, a_{21}\, \lambda_1),$$

$$d_3 = (p_{12}\, a_{11}^2 + p_{12}\, a_{11}\, a_{21} + p_{22}\, a_{11}\, a_{12} + a_{11}\, a_{22}\, p_{22} - p_{11}\, a_{11}\, a_{12} - a_{12}\, a_{21}\, p_{11} - p_{21}\, a_{12}^2 - p_{21}\, a_{12}\, a_{22})\,(-\lambda_2\, a_{12}\, a_{21} - \lambda_2\, a_{21}\, a_{22} + \lambda_1\, a_{11}\, a_{22} + \lambda_1\, a_{21}\, a_{22} - \lambda_1\, \lambda_2\, a_{11}\, a_{22} + \lambda_1\, \lambda_2\, a_{12}\, a_{21})\, q_3 - (-p_{12}\, a_{11}\, a_{21} - p_{12}\, a_{21}^2 - p_{22}\, a_{12}\, a_{21} - p_{22}\, a_{21}\, a_{22} + p_{11}\, a_{11}\, a_{22} + p_{11}\, a_{21}\, a22 + p_{21}\, a_{12}\, a_{22} + p_{21}\, a_{22}^2 - a_{11}\, a_{22}\, p_{11}\, p_{22} + a_{11}\, a_{22}\, p_{12}\, p_{21} + a_{12}\, a_{21}\, p_{11}\, p_{22} - a_{12}\, a_{21}\, p_{12}\, p_{21})\,(\lambda_2\, a_{11}\, a_{12} + a_{11}\, a_{22}\, \lambda_2 - \lambda_1\, a_{11}\, a_{12} - a_{12}\, a_{21}\, \lambda_1),$$

$$d_4 = (p_{12}\, a_{11}^2 + p_{12}\, a_{11}\, a_{21} + p_{22}\, a_{11}\, a_{12} + a_{11}\, a_{22}\, p_{22} - p_{11}\, a_{11}\, a_{12} - a_{12}\, a_{21}\, p_{11} - p_{21}\, a_{12}^2 - p_{21}\, a_{12}\, a_{22})\,(a_{11}\, a_{22} - a_{12}\, a_{21} - a_{11}\, a_{22}\, \lambda_2 + a_{12}\, a_{21}\, \lambda_1)\, q_4 - (a_{11}\, a_{22} - a_{12}\, a_{21} - p_{12}\, a_{11}\, a_{21} - a_{11}\, a_{22}\, p_{22} + a_{12}\, a_{21}\, p_{11} + p_{21}\, a_{12}\, a_{22})\,(\lambda_2\, a_{11}\, a_{12} + a_{11}\, a_{22}\, \lambda_2 - \lambda_1\, a_{11}\, a_{12} - a_{12}\, a_{21}\, \lambda_1).$$

Acknowledgements

We wish to thank Robert M. Williams (NAWCAD, Patuxent River, Maryland) for championing this problem and being a steady source of support, George Nakos (U. S. Naval Academy) for teaching us the Dixon Resultant, and Michael Hirsch (University of Delaware) for helping on early stages of the project. We also wish to acknowledge Williams, Ron Gleeson (College of New Jersey), and Frank Grosshans (Westchester of Pennsylvania) for their many contributions to this problem and their work in first solving the system in the appendix.

References

[1] F. Grosshans, R. Gleason, R. Williams, and M. Hirsch. personal communication.

[2] D. Kapur, T. Saxena, and L. Yang, Algebraic and Geometric Reasoning Using Dixon Resultants, in: *Proc. of the International Symposium on Symbolic and Algebraic Computation* (A.C.M. Press, New York, 1994).

[3] Robert H. Lewis, Computer Algebra System *Fermat*. http://www.bway.net/˜lewis/, http://www.fordham.edu/lewis/

[4] L. Quan, Computation of the Invariants of a Point Set in P^3 from its Projections in P^2, in: Neil L. White, ed., *Invariant Methods in Discrete and Computational Geometry*, (1994) 223–244.

[5] Peter Stiller, Symbolic Computation of Object/Image Equations, in: *Proc. of the International Symposium on Symbolic and Algebraic Computation* (A.C.M. Press, New York, 1997) 359–364.

[6] Peter Stiller, Charles Asmuth, and Charles Wan, Single-View Recognition – the Perspective Case, *Proceedings SPIE International Conference, Vision Geometry V*, Vol. 2826, Denver, CO; 8/96, pp. 226–235 (1996).