The Cubic Rational Polynomial Camera Model

Richard I. Hartley	Tushar Saxena
G.E. Corporate R & D	CMA Consulting
1 Research Circle	1400 Balltown Rd
Niskayuna, NY 12309	Niskayuna, NY 12301

June 21, 2001

Abstract

This paper describes an implementation of the Cubic Rational Polynomial Camera model developed as part of the FOCUS project. FOCUS ([1]) is on ongoing "shared vision" IR&D project jointly sponsored by Lockheed Martin Missiles and Space (LMMSS/Sunnyvale) and General Electric CR&D. A cubic camera has the advantage that all cameras, such as projective, affine and the linear pushbroom, which map the image points as rational polynomial functions (of degree no greater than 3) of the coordinates of a world point, can be treated as special cases of the cubic camera. This paper demonstrates that the cubic camera can very effectively model even those cameras which express the image points as complicated functions of world coordinates, such as radicals. In particular, it is empirically demonstrated that a SAR sensor is very accurately approximated by a cubic camera, but not by any linear camera model.

The paper also outlines an algorithm for estimating the parameters of the cubic camera, given a set of image to world correspondences. The non-linear nature of this camera can make parameter estimation a very unstable process. The slightest noise in the coefficients of the nonlinear terms can lead to a completely unrealistic model of the camera. This paper discusses some refinements such as avoiding degeneracies, data normalization, and regularization which are necessary for accurate estimation of the cubic camera parameters and minimization of noise in the coefficients of the higher degree terms.

1 Introduction

A basic requirement of the FOCUS ([1]) project is to be able to compute camera models and do model building using complex and general camera models. To this purpose, a Cubic Rational Polynomial camera model has been developed in FOCUS to aid in these tasks. Lockheed Martin and GE initiated the FOCUS project in January 1996 using GE's TargetJr as the MSE/IU platform. Funding for FOCUS at GE CR&D is provided by the Lockheed Martin Corporation. We acknowledge that this work would not have been possible without Governrment sponsored camera modelling and IU technology during the past decade.

A cubic camera models the coordinates of the image point as ratios of third degree polynomials in the coordinates of the world point. Given a set of image-world correspondences, the objective of the cubic camera estimation problem is to determine the set of coefficients (a total of 80) of the polynomials in the cubic camera model such that the error with which the camera maps the world points to the image points in this set of correspondences, is minimized. In this paper, we outline an algorithm which solves the cubic camera estimation problem by applying a least squares minimization to make an initial *guess* of the camera model, and then iteratively refining that guess and minimizing the error using a method based on Levenberg-Marquardt algorithm.

Due to the existence of non-linear terms in the camera model, even a small noise in the coefficients of the higher degree terms can lead to a large amount of error. A related problem is that of extrapolation. Since the solution of the camera model is not unique, there may exist models which produce a small error in the given set of correspondences, but assign such values to the coefficients of the higher degree terms which produce a completely unrealistic mapping of points outside the given set of correspondences. This leads to complications while extrapolating the model to points outside the given correspondence set. To overcome this such problems which are unique to non-linear camera models, we use the techniques of Data Normalization and Regularization. Specifically, we demonstrate that constraining the coefficients of the nonlinear terms to be as small as possible, generates a more realistic camera model which extrapolates better on the points outside the data set.

It is easy to see that all linear cameras such as the affine, perspective, and linear pushbroom cameras, can be viewed as special cases of the cubic camera. However, it is not so straightforward to use the cubic camera estimation for this special cases. Some of the problems encountered are the same as before, namely those concerning the unstability of higher degree terms leading to *over-parametrization*, which basically estimates a non-linear approximation to a completely linear camera. We demonstrate how the techniques such as regularization can also be exploited to overcome this problem, and get a more linear approximation in these special cases.

Finally, we conjecture that rational polynomial cameras indeed provide a very accurate approximation of even the *non-polynomial* cameras. In particular, we consider the SAR sensor, which models the image point as complicated radical functions of the coordinates of the world point. We provide empirical evidence which demonstrates that the cubic camera provides an extremely accurate approximation of this sensor, despite the fact that SAR is not a rational polynomial camera. The cubic-approximation of SAR is compared to the perspective and linear pushbroom approximations of the same. Our empirical results show that the cubic-approximation performs at least four orders of magnitude better.

2 The Cubic Camera Model

The cubic Rational Polynomial (RP) camera provides an abstraction of many types of camera models. The essential aspect of a camera model is the manner in which it maps points in space to points in an image. In the case of the RP camera, this mapping can be expressed in terms of rational polynomial functions of the world-coordinates of the object point.

Thus, the mapping defined by an RP camera is of the form

$$u = N_u(\mathbf{x})/D_u(\mathbf{x})v = N_v(\mathbf{x})/D_v(\mathbf{x}) \tag{1}$$

where $\mathbf{x} = (x, y, z, t)^\top$ is the homogeneous coordinate of a 3D point, $(u, v)^\top$ is the corresponding image point, and N_u, D_u, N_v and D_v are homogeneous polynomials of degree n.

A general homogeneous polynomial of degree n in r variables contains $\binom{n+r-1}{n} = \frac{(n+r-1)!}{n!\,(r-1)!}$ terms. In the particular case of polynomials in the coordinates of $\mathbf{x}$ we have $r = 4$ and so the number of terms of degree n is $(n+1)(n+2)(n+3)/6$.

We will consider most particularly the case in which $n = 3$ and refer to this as the Cubic camera. Each of the polynomials $N_u(\mathbf{x})$, $D_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D_v(\mathbf{x})$ has 20 terms, and hence may be parametrized by 20 coefficients. This amounts to a total of 80 parameters in all. It may be noted that in some descriptions of the Cubic camera, each of the coordinates x, y and z as well as the image coordinates u and v is subject to a scaling and offset, which adds an extra 10 parameters. However, these extra transformations may be incorporated into the rational cubic polynomial mappings, and are hence non-essential. They will be ignored in this exposition.

The polynomials N_u, D_u, N_v and D_v are homogeneous polynomials in the coordinates x, y, z and t of the 3D points. This means that each of the terms has the same degree, in this case 3. This is done so that the mapping is not dependent on the particular representation of the point $\mathbf{x}$ as a homogeneous vector. It is possible to dehomogenize the polynomials by setting $t = 1$. In this case the terms of the polynomials will have different degrees, and we can talk of constant, linear, quadratic, cubic terms. Whenever we talk of the degree of a term of a polynomial, or of the corresponding coefficient, it is this dehomogenized degree that will be meant.

3 Special Cases of the Cubic Camera

Many of the common cameras may be considered as special cases of the Cubic camera.

Projective Camera. The projective camera is defined by a mapping $(wu, wv, w)^\top = P\mathbf{x}$ where P is a 3×4 matrix, u and v are the image coordinates, and w is an unknown scale. This may also be written as

$$\begin{aligned} u &= \frac{\mathbf{p_1}^\top \mathbf{x}}{\mathbf{p_3}^\top \mathbf{x}} \\ v &= \frac{\mathbf{p_2}^\top \mathbf{x}}{\mathbf{p_3}^\top \mathbf{x}} \ . \end{aligned}$$

Thus, we see that this is a special case of the RP camera in which $N_u(\mathbf{x})$, $D_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D_v(\mathbf{x})$ are linear functions and $D_u = D_v$.

Linear Pushbroom Camera. The linear pushbroom camera described in [5] is an example of an RP camera. The linear pushbroom camera is an approximation of the camera model represented by a SPOT satellite pushbroom sensor. The defining equation is $(u, wv, w)^\top = P(x, y, z, 1)^\top$ where as before, P is a 3×4 matrix, u and v are the image coordinates, and w is an unknown scale. In terms of a homogeneous object point $\mathbf{x} = (x, y, z, t)^\top$, this may be written as

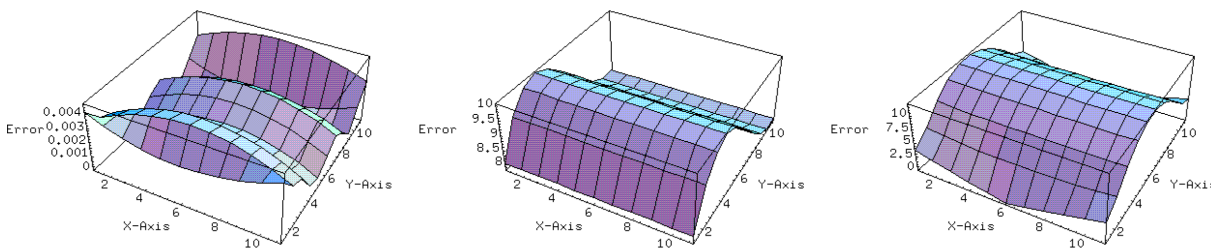
$$\begin{aligned} u &= \frac{\mathbf{p_1}^\top \mathbf{x}}{t} \\ v &= \frac{\mathbf{p_2}^\top \mathbf{x}}{\mathbf{p_3}^\top \mathbf{x}} \end{aligned}$$

where $\mathbf{x} = (x, y, z, 1)^\top$. In this case, it is equivalent to an RP camera with $N_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D_v(\mathbf{x})$ linear functions, and $D_u(\mathbf{x}) = t$.

Affine Camera. The affine camera is a special case of the projective camera in which the camera matrix has a special form in which the last row is $(0, 0, 0, 1)$. This may be modelled as a Cubic camera for which

$$\begin{aligned} u &= \frac{\mathbf{p_1}^\top \mathbf{x}}{t} \\ v &= \frac{\mathbf{p_2}^\top \mathbf{x}}{t} \end{aligned}$$

SAR images. SAR sensors may be approximated with the Cubic camera with excellent accuracy. This was demonstrated by testing the Cubic model against some synthetic correspondence data constructed as follows. Consider a SAR sensor moving in the x axial direction at an altitude of 3000m above a nominal ground plane and imaging a section of the ground at distances between 5000 and 7000 metres to the side of the flight path. Points were chosen over a 2000m $\times$ 2000m swath of ground at altitudes between -500m and 500m, and



Cubic model LP model Perspective model

Figure 1: *Fitting error for synthetic SAR data using Cubic, Perspective and Linear Pushbroom models. The graphs show the error for points in the plane* $z = 0$*, but for fitting, data points at altitdues between -500m and 500m were used. The average error for Perspective and LP sensors was approximately 6 pixels, with a maximum of 10 pixels. The error achieved with the Cubic camera was only 0.02 pixels.*

their corresponding image coordinates were computed, assuming a 1m pixel, thus creating a 2000×2000 pixel image. Thus, the u coordinate in the image of a point $\mathbf{x} = (x, y, z)^\top$ in space was equal to the x coordinate of the point, and the v coordinate was equal to the radial distance of the point from the line of flight. In symbols

$$
\begin{aligned}
u &= x \\
v &= \sqrt{y^2 + (z - 3000)^2}
\end{aligned}
$$

This data was then fitted with a Cubic camera, a Perspective camera and a Linear Pushbroom camera model and the residual reprojection error was recorded. The results are shown in Fig 1.

4 Solving for the Cubic Camera

We now consider the basic photogrammetry problem of parameter estimation for the Cubic camera. We assume given a set of image to world correspondences $\mathbf{u}_i \to \mathbf{x}_i$. The task is to compute the parameters of the Cubic RP camera, namely the coefficients of the polynomials $N_u(\mathbf{x})$, $D_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D_v(\mathbf{x})$. Two methods will be used to do this.

1. A linear method based on linear least-squares minimization. This method

is based on the DLT method ([7]) for estimating the parameters of a projective camera.

2. An iterative method based using the Levenberg-Marquardt method ([6]). The linear method was used to provide an initial estimate of camera parameters, which is refined by iteration. This method was implemented using a general-purpose camera solving program Carmen ([4]). Little more will be said in this report concerning the iterative method.

4.1 Linear Estimation of the Cubic Camera Model.

From the equations

$$\begin{aligned} u &= N_u(\mathbf{x})/D_u(\mathbf{x}) \\ v &= N_v(\mathbf{x})/D_v(\mathbf{x}) \end{aligned}$$

defining the cubic camera model, one may obtain by cross multiplication a pair of equations

$$\begin{aligned} uD_u(\mathbf{x}) - N_u(\mathbf{x}) &= 0 \\ vD_v(\mathbf{x}) - N_v(\mathbf{x}) &= 0 \ . \end{aligned} \tag{2}$$

Although these equations are non-linear in $\mathbf{x}$, they are linear in the coefficients of the polynomials. Since each such correspondence gives a pair of equations, and there are a total of 80 unknown parameters, a total of at least 40 correspondences are required to solve for the polynomial coefficients. With more than 40 points one has an over-determined system of equations which will be solved by least-squares techniques. The total set of equations are of the form $A\mathbf{p} = \mathbf{0}$, where $\mathbf{p}$ is the set of parameters. We are not interested in the trivial solution $\mathbf{p} = \mathbf{0}$. Since the polynomials are homogeneous, their quotient $N_u(\mathbf{x})/D_u(\mathbf{x})$ (and the same thing for v) is independent of scale. We find the parameter vector $\mathbf{p}$ that minimizes $||A\mathbf{p}||$ subject to $||\mathbf{p}|| = 1$. The solution is the singular vector corresponding to the smallest singular value of A ([2]).

This is the barest outline of the method. More will be said later about important implementation details and refinements to this algorithm.

4.2 Degeneracy of the Cubic Model

We would like to be able to treat cameras such as the projective and linear pushbroom cameras as special cases of the Cubic camera and use the same parametrization method for all. Care must be taken in doing this, however because of over-parametrization of the camera model. Consider for instance a set of world to image correspondences $\mathbf{u}_i \rightarrow \mathbf{x}_i$ corresponding to a projective camera. In the absence of noise, there will exist linear polynomials $N_u(\mathbf{x})$, $N_v(\mathbf{x})$ and $D(\mathbf{x})$ such that $u_i = N_u(\mathbf{x}_i)/D(\mathbf{x}_i)$ and $v_i = N_v(\mathbf{x}_i)/D(\mathbf{x}_i)$. Unfortunately, these are not the only polynomials which give rise to the correct image points. In particular, one may multiply numerator and denominator by an arbitrary polynomial and obtain the same mapping. In symbols the value of

$$\mathbf{u}_i = \frac{A(\mathbf{x}_i)N_u(\mathbf{x}_i)}{A(\mathbf{x}_i)D(\mathbf{x}_i)}$$

is constant for all polynomials $A(\mathbf{x}_i)$. Since $N_u(\mathbf{x}_i)$ and $D(\mathbf{x}_i)$ have degree 1 for a perspective camera, $A(\mathbf{x}_i)$ may be an arbitrary degree 2 homogeneous polynomial. Such a polynomial has $C_2^5 = 10$ degrees of freedom. Since the numerator and denominator of $\mathbf{v}_i = N_v(\mathbf{x}_i)/D(\mathbf{x}_i)$ may independently be multiplied by a polynomial $B(\mathbf{x})$, there exists a 20-parameter family of cubic polynomials defining a projective camera mapping. In this case, the matrix A in the set of equations $\mathtt{A}\mathbf{p} = 0$ will have diminished rank. In fact A has 80 columns, but its rank will be at most 60 because of the 20-parameter family of solutions. The solution of $A\mathbf{p} = \mathbf{0}$ will not be well defined, and there is no reason to expect the linear solution to be selected, if one is chosen arbitrarily.

In the presence of a degree of noise in the measurements of image points, or 3D points, the matched points $\mathbf{u}_i \rightarrow \mathbf{x}_i$ will not correspond precisely with a true perspective model. The cubic model will attempt to correct for this by the introduction of spurious higher-order terms. This will cause the model to match the data more precisely on the measured data. However, it can lead to large errors in other parts of the scene, far from measured control points. In brief, in the presence of instabilities of this nature, one can not extrapolate reliably beyond the measured data. This point will be illustrated later on in this paper.

4.3 Data Normalization

It has been pointed out by several authors, for instance the present authors ([3]) that prenormalization of the input data is essential for obtaining a good result from linear algorithms of this kind which do not minmize geometrically meaningful quantity. Before running the linear algorithm to compute the camera parameters, it is absolutely essential to normalize the data. The general method involves three steps.

1. Choose transforms $T_\mathbf{u}$ and $T_\mathbf{x}$ of the image and object points such that $\mathbf{u}_i \mapsto \mathbf{u}'_i = T_\mathbf{u}\mathbf{u}_i$ and $\mathbf{x}_i \mapsto \mathbf{x}'_i = T_\mathbf{x}\mathbf{x}_i$.

2. Solve to find a parametrized Cubic RP camera model, represented by a map P' (in general, non-linear), that provides a best possible solution to the set of equations $\mathbf{u}'_i = P'\mathbf{x}'_i$. This is done using the linear algorithm described above.

3. Replace P' by $P = T_\mathbf{u}P'T_\mathbf{x}^{-1}$. This mapping will satisfy $P\mathbf{x}_i = T_\mathbf{u}^{-1}P'T_\mathbf{x}\mathbf{x}_i = T_\mathbf{u}^{-1}P'\mathbf{x}'_i = T_\mathbf{u}^{-1}\mathbf{u}'_i = \mathbf{u}_i$ as required. Note that composition of mappings

and not matrix multiplication is implied by this juxtaposition $T_{\mathbf{u}}P'T_{\mathbf{x}}^{-1}$, since P' is not linear.

The recommended normalizing transforms $T_{\mathbf{u}}$ and $T_{\mathbf{x}}$ are both of the same type : translation of the data to place its centroid at the origin, and scaling so that the average point is a distance $\sqrt{2}$ from the origin in the case of $T_{\mathbf{u}}$, which is a 2D transformation, and $\sqrt{3}$ in the case of $T_{\mathbf{x}}$, which is a 3D transformation. The reasons for this choice of scaling are given in [3]. The main purpose of data normalization is to improve the conditioning of the problem. To see why this would otherwise be a problem, consider a 3D object point with coordinates $(x, y, z, t)^\top = (500, 500, 500, 1)^\top$ in some coordinate system mapping to an image point $(u, v)^\top = (500, 500)$. In writing the set of equations (2), the entry corresponding to the term x^3 of N_u will be 500^4, whereas the entry corresponding to term t^3 of D_u will be 1. This wide range of entries in matrix A means that A will be poorly conditioned, and the solution very unstable in the presence of noise. The normalization transformations are designed to give each entry in A an equal weight.

In doing this, it is important that if P' is a cubic RP mapping, then so is $P = T_{\mathbf{u}}^{-1}P'T_{\mathbf{x}}$. This will be true for any linear transformation $T_{\mathbf{x}}$, but interestingly enough, not for any $T_{\mathbf{u}}$. This is easily seen as follows. First, suppose that $P'_u(\mathbf{x}) = N'_u(\mathbf{x})/D'_u(\mathbf{x})$ where both N'_u and D'_u have degree n. (For the Cubic camera, $n = 3$.) Then, $P'_u T_{\mathbf{x}}(\mathbf{x}) = N'_u(T_{\mathbf{x}}(\mathbf{x}))/D'_u(T_{\mathbf{x}}(\mathbf{x}))$, and both numerator and denominator are degree n polynomials in $\mathbf{x}$, since $T_{\mathbf{x}}(\mathbf{x})$ is linear.

On the other hand, consider $TP'(\mathbf{x})$ where T is any affine transformation. The u coordinate in the image is given by $u = T(u', v') = \alpha u' + \beta v' + \gamma$. The v coordinate is expressed similarly. In this case, we have

$$TP'(\mathbf{x}) = \alpha \frac{N'_u(\mathbf{x})}{D'_u(\mathbf{x})} + \beta \frac{N'_v(\mathbf{x})}{D'_v(\mathbf{x})} + \gamma$$

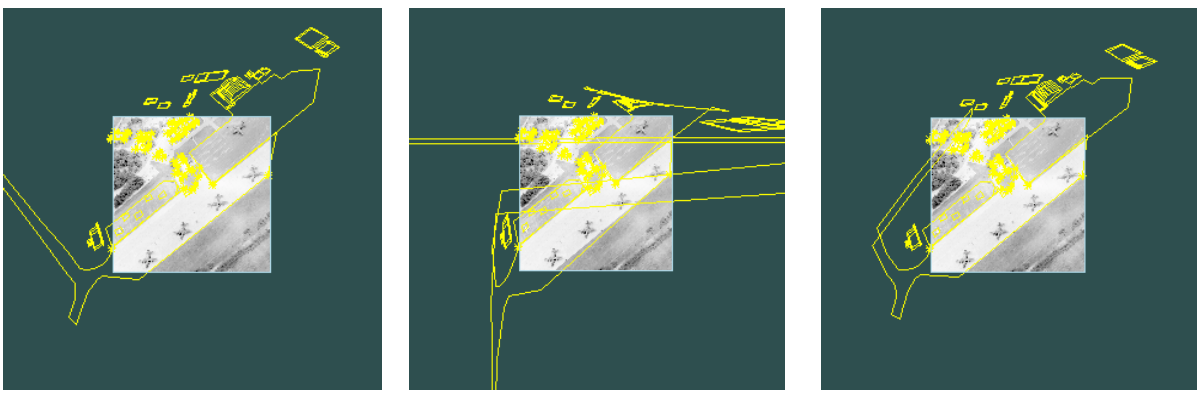
$$= \frac{\alpha N'_u(\mathbf{x})D'_v(\mathbf{x}) + \beta N'_v(\mathbf{x})D'_u(\mathbf{x}) + \gamma D'_u(\mathbf{x})D'_v(\mathbf{x})}{D'_u(\mathbf{x})D'_v(\mathbf{x})}$$

Thus, the degree of the RP transformation is increased. There are two evident exceptions to this :

1. $D'_u = D'_v$. This is the case for a projective camera.
2. $\beta = 0$. This is the case in which the transformation is a simple scaling and translation. This is the recommended sort of transformation.

In this latter case, one has

$$TP'(\mathbf{x}) = \frac{\alpha N'_u(\mathbf{x}) + \gamma D'_u(\mathbf{x})D'_v(\mathbf{x})}{D'_u(\mathbf{x})} \tag{3}$$



Perspective model   Cubic model : $k = 0.1$ Cubic model : $k = 10.0$

Figure 2: *This shows the result of camera resectioning using the Cubic camera model. Hand-picked correspondences between a site model and an image are used to compute the camera model. The site model is then projected into the image using the computed camera model. The three examples show a projective camera, and two parametrized cubic cameras computed with different settings of* k*, the constraint weight for high-order coefficients. In all cases, the site model is well aligned with the image within the image area. For the cubic camera, the agreement of the site model outside of the area where control points are chosen is not so good, though for larger value of* k*, the site model is projected reasonably well.*

4.4 Computing Composition of Mappings

The composition of $T_{\mathbf{u}}^{-1}P'(\mathbf{x})$ is easily computed using (3). The computation of $P'T_{\mathbf{x}}$ is a little more complicated, however. This is a standard sort of algebraic manipulation problem, but complicated if one does not do it the right way. A simple implementation is possible using tensors, as described now.

A cubic homogeneous polnomial $N(\mathbf{x})$ may be written in terms of a symmetric tensor N_{ijk} defined such that if $\mathbf{x}$ has components x^i, then $N(\mathbf{x}) = N_{ijk}x^i x^j x^k$. Here, the superscripts represent indices, not powers, and a repeated index in the upper and lower positions implies summation. This may be more familiar in the degree 2 case, where a quadratic form may be written as $\mathbf{x}^\top A\mathbf{x} = A_{ij}x^i x^j$.

Now, if we apply a linear transformation such that $x^i = T^i_p x'^p$, then it follows that $N(\mathbf{x}) = N_{ijk}x^i x^j x^k = N_{ijk}T^i_p T^j_q T^k_r x'^p x'^q x'^r = N'_{pqr}x'^p x'^q x'^r$ where N'_{pqr} is defined by

$$N'_{pqr} = N_{ijk}T^i_p T^j_q T^k_r \tag{4}$$

This is the composition rule required to compute the composition $PT_{\mathbf{x}}$.

4.5 Regularization

It was shown in a previous section that in cases where a Cubic camera is well approximated by a projective camera, or some other linear camera, the computation of the camera model may be unstable. A way that we have found useful for dealing with this problem is regularization. In this method, a constraint is put on quadratic and cubic terms in N_u, N_v, D_u and D_v constraining them to be close to zero. This constraint is weighted by a parameter k, where high values of k provide a strong constraint on the values of the higher order terms. The requirements that these terms be small is balanced against the requirements imposed by the data.

This has two effects :

1. Low degree polynomials are favoured over high-degree polynomials. This has the effect of resolving the ambiguity in the set of solutions. In the presence of only a low degree of noise, for instance, a perspective cameras will be modelled with linear (or near linear) rational polynomials.

2. It is possible to solve for the camera parameters with fewer than the full number (40 in the Cubic camera case) of point correspondences. This is useful when it is difficult to find such a large number of control points.

Figure 2 shows the effect of different values of k.

References

[1] Eamon B. Barrett, Paul M. Payton, and Joseph L. Mundy. Focus : A shared vision technology transfer project. In *Proc. DARPA Image Understanding Workshop*, 1997.

[2] O. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint.* The MIT Press, Cambridge, MA, 1993.

[3] R. I. Hartley. A linear method for reconstruction from lies and points. In *Proc. International Conference on Computer Vision*, pages 882 – 887, 1995.

[4] Richard I. Hartley. An object-oriented approach to scene reconstruction. In *Proc. IEEE International Conference on Systems Man and Cybernetics, Peking*, pages 2475 – 2480, October 1996.

[5] Richard I. Hartley and Rajiv Gupta. Linear pushbroom cameras. In *Computer Vision - ECCV '94, Volume I, LNCS-Series Vol. 800, Springer-Verlag*, pages 555–566, May 1994.

[6] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1988.

[7] I.E. Sutherland. Three dimensional data input by tablet. *Proceedings of IEEE*, Vol. 62, No. 4:453–461, April 1974.