Short on camera geometry and camera calibration

Maria Magnusson, maria.magnusson@liu.se
Computer Vision Laboratory, Department of Electrical Engineering,
Linköping University, Sweden

Report No: LiTH-ISY-R-3070

November, 2010

Contents

1 Introduction

We will present the basic theory for the camera geometry. Our goal is camera calibration and the tools necessary for this. We start with homogeneous matrices that can be used to describe geometric transformations in a simple manner. Then we consider the pinhole camera model, the simplified camera model that we will show how to calibrate.

A camera matrix describes the mapping from the 3D world to a camera image. The camera matrix can be determined through a number of corresponding points measured in the world and the image. We also demonstrate the common special case of camera calibration when it can be assumed that the world is flat. Then, a plane in the world is transformed to the image plane. Such a plane-to-plane mapping is called a homography.

Finally, we discuss some useful mathematical tools needed for camera calibration. We show that the solution we present for the determination of the camera matrix is equivalent to a least-squares solution. We also show how to solve a homogeneous system of equations using SVD (singular value decomposition).

The content of this text is based largely on [1], [4], [6] and [7]. It is also meant to be used as a supplement and aid during reading of [6] and [7].

2 Camera calibration

2.1 Geometrical transformations with homogeneous matrices

Geometrical transformations are commonly used in related topics such as 3D computer graphics. The idea is to use homogeneous coordinates to describe affine transformations.

A point in the 3D world coordinate system can be described by $(X, Y, Z, 1)^T$. It can be transformed to a new point $(X_1, Y_1, Z_1, 1)^T$ using

$$(X_1, Y_1, Z_1, 1)^T = M \cdot (X, Y, Z, 1)^T, \tag{1}$$

where M is a 4×4-matrix.

Alternatively, a point in the 3D world coordinate system $(X, Y, Z, 1)^T$ can be transformed to a point $(U, V, W, 1)^T$ in a different coordinate system, such as a camera coordinate system, using

$$(U, V, W, 1)^T = M \cdot (X, Y, Z, 1)^T. \tag{2}$$

Note that we have drawn a horizontal dashed line in the homogeneous matrices below. The fourth row is actually quite uninteresting, since it is always $[0, 0, 0, 1]$

and it is only needed during multiplication of homogeneous matrices. Therefore, in the following sections, a 3×4-matrix is used instead of a 4×4-matrix to describe how the camera is related to the 3D world.

2.1.1 Scaling

The scaling matrix is given by

$$S(s_a, s_b, s_c) = \begin{pmatrix} s_a & 0 & 0 & 0 \\ 0 & s_b & 0 & 0 \\ 0 & 0 & s_c & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

A point in the 3D world, $(X, Y, Z, 1)^T$, is transformed to $(s_aX, s_bY, s_cZ, 1)^T$ by the S-matrix according to

$$\begin{pmatrix} s_aX \\ s_bY \\ s_cZ \\ 1 \end{pmatrix} = \begin{pmatrix} s_a & 0 & 0 & 0 \\ 0 & s_b & 0 & 0 \\ 0 & 0 & s_c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{4}$$

2.1.2 Translation

The translation matrix is given by

$$T(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ \hdashline 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5}$$

A point in the 3D world, $(X, Y, Z, 1)^T$, is transformed to $(X+t_x, Y+t_y, Z+t_z, 1)^T$ by the T-matrix according to

$$\begin{pmatrix} X + t_x \\ Y + t_y \\ Z + t_z \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{6}$$

2.1.3 Rotation

Here we suppose that the rotation is measured counterclockwise around the actual axis. Rotation an angle θ around the X-axis is given by

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ \hdotsfor{4} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{7}$$

rotation an angle θ around the Y-axis is given by

$$R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ \hdotsfor{4} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

and rotation an angle θ around the Z-axis is given by

$$R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hdotsfor{4} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{9}$$

2.1.4 Skewing

Finally, we should also mention skewing. Skewing is a linear change of coordinates based on one coordinate. Skewing can transform a square to a parallelogram. The matrix below shows a skewing in the X-direction which depends on the Y-coordinate.

$$\begin{pmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hdotsfor{4} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{10}$$

A general skewing is described by

$$\begin{pmatrix} 1 & a & b & 0 \\ c & 1 & d & 0 \\ e & f & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{11}$$

2.2 The pinhole camera model

Figure 1 shows a simple model of a camera, the so-called pinhole camera model. It works decently for a regular camera. However, it must be modified and supplemented if one is dealing with thick lenses, such as microscope or wide-angle lenses. There is a connection between the coordinate systems in the figure and this

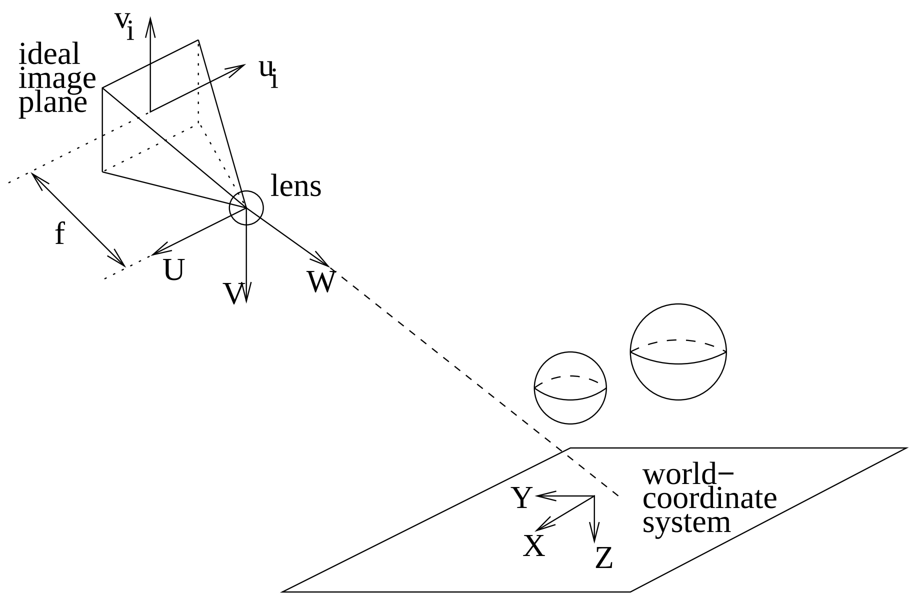

Figure 1: The pinhole camera model, real geometry.

connection can be modeled using a matrix. Certainly, in reality the image plane is located behind the lens. However, it is easier to see the connection between the coordinate systems if the image plane is reflected so that it is located in front of the lens. This reflected geometry is shown in Figure 2. The following formula gives the connection between the coordinate systems,

$$W(u_i/f, v_i/f, 1)^T = (U, V, W)^T = [R\ t] \cdot (X, Y, Z, 1)^T, \tag{12}$$

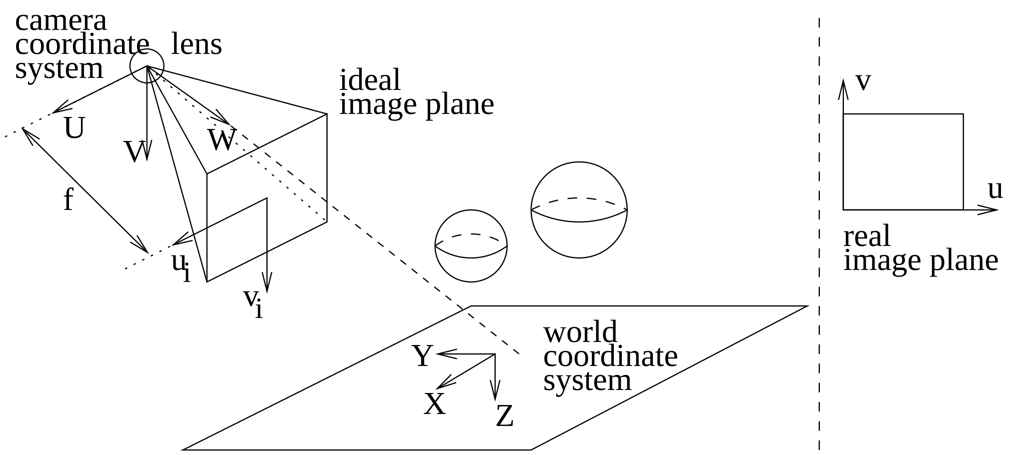

Figure 2: The pinhole camera model, calculation friendly geometry.

where $(X, Y, Z)^T$ are the world coordinates, $(U, V, W)^T$ are the camera coordinates and $(u_i, v_i)^T$ are the ideal image coordinates. The matrix $[R\ t]$ is modelling rotation and translation and can be written

$$[R\ t] = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}. \tag{13}$$

The first part of Equation (12) shows one way to describe the perspective transformation. See Figure 3. We wish to project the point (U, V, W) on the ideal image plane (u_i, v_i), which is located at a distance of f from the lens. Uniform triangles give

$$\begin{cases} \frac{v_i}{f} = \frac{V}{W} \\ \frac{u_i}{f} = \frac{U}{W} \end{cases} \tag{14}$$

which is exactly what is written in the first part of Equation (12). The real im-

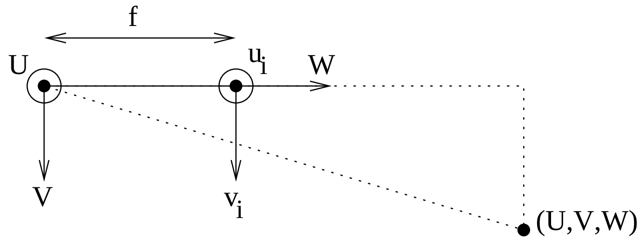

Figure 3: Perspective transformation.

age coordinates $(u, v)^T$ is normally different from the ideal image coordinates $(u_i, v_i)^T$. Also, we should mention the normalized image coordinates which are simply $(u_n, v_n)^T = (u_i/f, v_i/f)^T$. The following equations show the relationships,

$$(u, v, 1)^T = A \cdot (u_i/f, v_i/f, 1)^T = A \cdot (u_n, v_n, 1)^T \tag{15}$$

and

$$A = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} f & \gamma & u_0 \\ 0 & kf & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{16}$$

where $(u_0, v_0)^T$ are the real image coordinates corresponding to the intersection of the optical axis with the ideal image plane, $(u_i, v_i)^T = (0, 0)^T$, where α and β are the scale factors for the u- and v-axes of the image, and where γ describes the skew of the two image axes. An angular skew measurement is given by

$$\xi = \arctan(\gamma/\beta). \tag{17}$$

Equation (12) and (15) now gives

$$s(u, v, 1)^T = A[R\ t] \cdot (X, Y, Z, 1)^T, \tag{18}$$

where we have replaced W with s to obtain a more general relation. The reason is that not only $A[R\ t]$ transforms $(X, Y, Z)^T$ to $(u, v)^T$, but also $k \cdot A[R\ t]$ transforms $(X, Y, Z)^T$ to $(u, v)^T$, where k is a constant.

In the following, we will show how do determine $C = A[R\ t]$ through a calibration procedure. When C is determined, there are methods to sort out A and $[R\ t]$ individually. An alternative is to use Zhang's method, [6] and [7], which gives A and $[R\ t]$ directly.

2.3 Calibration of a camera in the 3D world

See Figure 2. A point $(X, Y, Z, 1)^T$ in the world is transformed to the real image plane $(u, v, 1)^T$ through the C-matrix,

$$s(u, v, 1)^T = C \cdot (X, Y, Z, 1)^T. \tag{19}$$

If we determine a number of corresponding points in the world (X_i, Y_i, Z_i) and in the image (u_i, v_i), where $1 \leq i \leq N$, the matrix C can be determined up to a scale factor because of the variable s in Equation (19). We can therefore choose to set the last element $C_{34} = 1$, which gives

$$C = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & 1 \end{pmatrix}. \tag{20}$$

Let

$$c = (C_{11}, C_{12}, C_{13}, C_{14}, C_{21}, C_{22}, C_{23}, C_{24}, C_{31}, C_{32}, C_{33})^T . \tag{21}$$

Using the measured corresponding points in the world and in the image, the following equation system is obtained,

$$D \cdot c =$$

$$\begin{pmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -u_2X_2 & -u_2Y_2 & -u_2Z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -v_NX_N & -v_NY_N & -v_NZ_N \end{pmatrix} \cdot \begin{pmatrix} C_{11} \\ C_{12} \\ C_{13} \\ \vdots \\ C_{33} \end{pmatrix}$$

$$= \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ \vdots \\ v_N \end{pmatrix} = f. \tag{22}$$

As you can see, at least six (or rather $5\ 1/2$) points (X_i, Y_i, Z_i) in the world are needed to determine C. However, the solution will obviously be more certain the more points we measure. We now rewrite the system of equations as follows

$$\begin{aligned} D \cdot c &= f \\ D^T D \cdot c &= D^T f \\ c &= (D^T D)^{-1} D^T f \\ c &= D^\dagger f, \end{aligned} \tag{23}$$

where $D^\dagger$ is the so-called pseudo-inverse of D. Note that $D^T D$ becomes a square matrix. Provided that $D^T D$ is invertible and that the number of rows in D is greater than or equal to the number of columns, Equation (23) is valid. It can be shown that Equation (23) is equivalent to both maximum likelihood minimization and least squares adjustment, see Section 3.1.

After this calibration, where the matrix C has been determined, we will be able to predict how a point in the world $(X, Y, Z)^T$ is transformed to a point in the image $(u, v)^T$. Note, however, that we cannot say that a point $(u, v)^T$ in the image corresponds to a certain point in the world. Instead, a point $(u, v)^T$ in the image corresponds to a *line* in the world.

If we have more knowledge of the points in the world, it might be possible to accurately determine the correspondence between image point and world point.

Such a case is when all points are located in a flat world, a plane. This will be shown in the next section.

Another possibility is to use stereo, i.e. using two calibrated cameras. The interesting point is identified in the images of both cameras, which provides two straight lines in the world. The intersection between those two lines gives the exact location of the point in the world.

2.4 Calibration of a camera and a flat world, a homography

See the Figure below. We want to calibrate a camera against a flat world, $z = 0$,

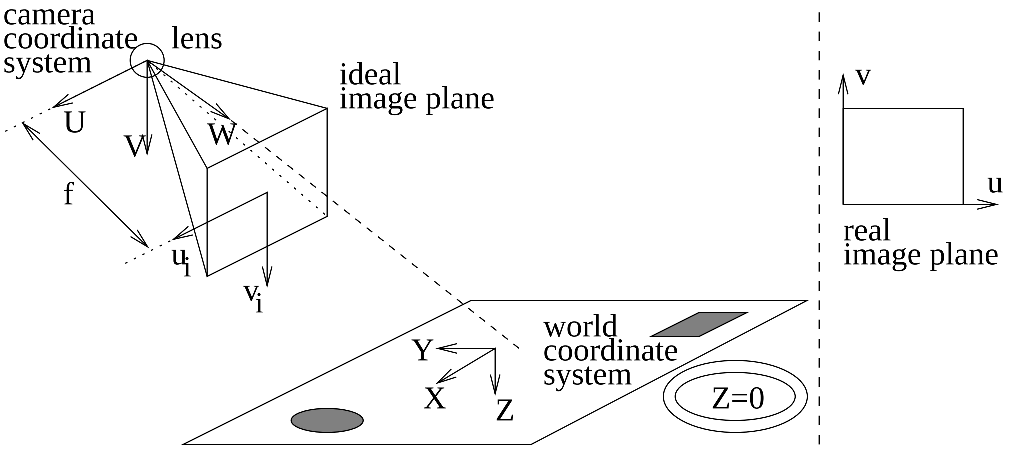

Figure 4: The pinhole camera model with a flat world, a homography, $Z = 0$

by determining the relationship between the image coordinates $(u, v)^T$ and the world coordinates $(X, Y, Z = 0)^T$. We recall Equation (19) and set $Z = 0$, which gives

$$s(u, v, 1)^T = C \cdot (X, Y, 1)^T. \tag{24}$$

If we determine a number of corresponding points in the world (X_i, Y_i) and in the image (u_i, v_i), where $1 \leq i \leq N$, the matrix C can be determined up to a scale factor because of the variable s in Equation (24). We can therefore choose to set the last element $C_{33} = 1$, which gives

$$C = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & 1 \end{pmatrix}. \tag{25}$$

Let

$$c = (C_{11}, C_{12}, C_{13}, C_{21}, C_{22}, C_{23}, C_{31}, C_{32})^T . \tag{26}$$

Using the measured corresponding points in the world and in the image, the following system of equations is obtained,

$$D \cdot c =$$

$$\begin{pmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -v_1X_1 & -v_1Y_1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -u_2X_2 & -u_2Y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & X_N & Y_N & 1 & -v_NX_N & -v_NY_N \end{pmatrix} \cdot \begin{pmatrix} C_{11} \\ C_{12} \\ C_{13} \\ \vdots \\ C_{32} \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ \vdots \\ v_N \end{pmatrix} = f. \tag{27}$$

As you can see, at least four points (X_i, Y_i) in the world are needed to determine C. However, the solution will obviously be more certain the more points we measure. Just as before, in Equation (23), the solution becomes

$$c = D^{\dagger} f. \tag{28}$$

After this calibration, where the matrix C has been determined, we will be able to predict how a point in the world $(X, Y, Z = 0)^T$ is transformed to a point in the image $(u, v)^T$. As opposed to the general case in section 2.3, here we also know that a point $(u, v)^T$ in the image corresponds to a certain point in the world.

3 Some mathematical tools

3.1 Least squares solution with the pseudo-inverse method

Consider the systems of equations in (22) and (27). These systems of equations are over-determined in the general case. Suppose that we have measured the points (X_i, Y_i, Z_i) (or (X_i, Y_i)) and (u_i, v_i), for $1 \leq i \leq N$. Then

$$f = Dc + E, \tag{29}$$

where E are the errors caused by inaccuracies in the measurements. Let us now minimize the square of the sum of the errors, i.e.

$$\begin{aligned} E^T E &= (f - Dc)^T (f - Dc) \\ &= (f^T - (Dc)^T)(f - Dc) = (f^T - c^T D^T)(f - Dc) \\ &= f^T f - c^T D^T f - f^T Dc + c^T D^T Dc \\ &= f^T f - 2c^T D^T f + c^T D^T Dc. \end{aligned} \tag{30}$$

The minimum will be located where the derivatives depending on c are zero, i.e.

$$0 = -2D^T f + 2D^T Dc, \tag{31}$$

which gives

$$c = (D^T D)^{-1} D^T f = D^\dagger f, \tag{32}$$

where $D^\dagger$ is called the pseudo-inverse of D. The pseudo-inverse is received in MATLAB using:

```
D_ps_inv = pinv(D);
```

3.2 Solution of a homogeneous system of equations using SVD

Here we will show a general method for solving a homogeneous system of equations,

$$Xb = 0 \tag{33}$$

The method uses SVD which means *singular value decomposition*. The method is, for example, used in [6] and [7]. It can be shown that an arbitrary matrix X can be written

$$X = USV^T, \tag{34}$$

see for example [2], [3] or [5]. The matrices U and V^T are orthonormal, i.e. $U \cdot U^T = E$ and $V \cdot V^T = E$, where E is the identity matrix. For orthonormal matrices, the row and column vectors are orthogonal. S is a diagonal matrix.

Suppose that X is a 4×3-matrix. Then

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ u_{41} & u_{42} & u_{43} & u_{44} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{pmatrix}.$$

We now rewrite Equation (33) as

$$\begin{aligned} X \cdot b &= 0 \\ USV^T b &= 0 \\ U^T USV^T b &= U^T 0 = 0 \\ SV^T b &= 0 \end{aligned}$$

Set $b = \mu v_3 = \mu(v_{13}, v_{23}, v_{33})^T$. This gives

$$\begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{pmatrix} \mu \begin{pmatrix} v_{13} \\ v_{23} \\ v_{33} \end{pmatrix} = \mu \begin{pmatrix} 0 \\ 0 \\ \sigma_3 \end{pmatrix},$$

where μ is an arbitrary constant and the last equal sign is given by the orthonormality of the rows and columns in V. In the general case, σ_3 is replaced with σ_n and according to the definition, $\sigma_i > \sigma_{i+1}$. The smaller σ_n, the better solution, and $\sigma_n = 0$ solves $Xb = 0$ perfectly. Consequently, the solution to Equation (33) is

$$b = \mu v_n = \mu(v_{1n}, v_{2n}, ..., v_{nn})^T. \tag{35}$$

The corresponding MATLAB code becomes:

```
[U,S,V] = svd(X)
b = V(:,n);
```

References

[1] Dana H. Ballard and Christopher M. Brown, *Computer Vision*, Prentice-Hall Inc., ISBN 0-13-165316-4, 1982

[2] Åke Björck, *Numerical Methods for Least Squares Problems*, SIAM, ISBN 0-89871-360-9, 1996
http://www.mai.liu.se/ akbjo/LSPbook.html

[3] Germund Dahlquist and Åke Björck, *Numerical Mathematics in Scientific Computation*
http://www.mai.liu.se/ akbjo/NMbook.html

[4] Per-Erik Forssén, *Personal communication*, 2004-2005

[5] Michael T. Heath, *Scientific Computing*, McGraw-Hill, ISBN 0-07-112229-X, 2002

[6] Zhengyou Zhang, *A Flexible New Technique for Camera Calibration*, Technical Report, MSR-TR-98-71, 1998, Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, 1998
http://research.microsoft.com/˜ zhang

[7] Zhengyou Zhang, *A Flexible New Technique for Camera Calibration*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, No. 11, Nov. 2000