Slide 1

Camera Calibration

The purpose of camera calibration is to determine the intrinsic camera parameters (c_0, r_0), f, s_x, s_y, skew parameter ($s = \cot\alpha$), and the lens distortion (radial distortion coefficient k_1). Skew parameter defines the angle between the c and r axes of a pixel. For most CCD camera, we have rectangular pixel. The angle is 90. The skew parameter is therefore zero. k_1 is often very small and can be assumed to be zero. The intrinsic camera matrix is

$$W = \begin{pmatrix} fs_x & \cot\alpha & c_0 \\ 0 & \frac{fs_y}{\sin\alpha} & r_0 \\ 0 & 0 & 1 \end{pmatrix}$$

If we assume $\alpha = 90$, then we have

Slide 2

$$W = \begin{pmatrix} fs_x & 0 & c_0 \\ 0 & fs_y & r_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Slide 3

Camera Calibration Methods

- Conventional method: uses 2D and 3D data to compute camera parameters. may work with single view of image.
- camera self-calibration: only need image data for camera calibration but need image data from multiple views. Problem is often less constrained than the conventional method.

Slide 4

Conventional Camera Calibration

Determine the intrinsic as well as exterior camera parameters using 2D image features and the corresponding 3D features. The image features can be points, lines, or curves or their combination.

- Determine the projection matrix P
- Derive camera parameters from P

$$P = \begin{pmatrix} s_x f r_1 + c_0 r_3 & s_x f t_x + c_0 t_z \\ s_y f r_2 + r_0 r_3 & s_y f t_y + r_0 t_z \\ r_3 & t_z \end{pmatrix}$$

Slide 5

Compute P: Linear Method using 2D/3D Points

Given image points $m_i^{2\times 1} = (c_i, r_i)^t$ and the corresponding 3D points $M_i^{3\times 1} = (x_i\ y_i\ z_i)^t$, where i=1, ..., N, we want to compute P. Let P be represented as

$$P = \begin{pmatrix} p_1^t & p_{14} \\ p_2^t & p_{24} \\ p_3^t & p_{34} \end{pmatrix} \tag{1}$$

where p_i, i=1,2,3 are 3×1 vectors and p_{i4}, i=1,2,3, are scalers.

Slide 6

Compute P: Linear Method (cont'd)

Then for each pair of 2D-3D points, we have

$$\begin{aligned} M_i^t p_1 + p_{14} - c_i M_i^t p_3 - c_i p_{34} &= 0 \\ M_i^t p_2 + p_{24} - r_i M_i^t p_3 - r_i p_{34} &= 0 \end{aligned}$$

For N points, we can setup a system of linear equations

$$AV = 0$$

where A is a $2N \times 12$ matrix depending only on the 3-D and 2-D coordinates of the reference points, and V is a 12×1 vector $(p_1^t\ p_{14}\ p_2^t\ p_{24}\ p_3^t\ p_{34})^t$.

Slide 7

$$A = \begin{pmatrix} M_1^t & 1 & \vec{0} & 0 & -c_1 M_1^t & -c_1 \\ \vec{0} & 0 & M_1^t & 1 & -r_1 M_1^t & -r_1 \\ & & \vdots & & & \\ M_N^t & 1 & \vec{0} & 0 & -c_N M_N^t & -c_N \\ \vec{0} & 0 & M_N^t & 1 & -r_N M_N^t & -r_N \end{pmatrix}$$

where $\vec{0}^{1\times 3} = [0\ 0\ 0]$

Slide 8

Compute P: Linear Method (cont'd)

For the linear method to work, we need $N >= 6$ and the N points cannot be coplanar points.

Slide 9

Rank of A

In general, the rank of A is 11 (for 12 unknowns), which means the solution is up to a scale factor. But due to effect of noise and locational errors, A may be full rank, which may make the solution (corresponding to the eigenvector of the smallest eigen value) unique.

The rank of A may also change for certain special configurations of input 3D points, for example collinear points, coplanar points, etc. The issue is of practical relevance.

Slide 10

Rank of A

Rank of A is a function of the input points configurations (see section 3.4.1.3 of Faugeras). If 3D points are coplanar, then rank(A) < 11 (in fact, it equals 8), which means there is an infinite number of solutions. Faugeras proves that 1) in general Rank(A)=11; 2) for coplanar points (N $\geq$4), rank(A)=8.

How about the rank of A if points are located a sphere or on the planes that are orthogonal or parallel to each other ?

Slide 11

Linear Solution 1

Minimize

$$||AV||^2$$

to solve for V. Can we solve for V ? Solution to $AV = 0$ is not unique (up to a scale factor).

If rank(A)=11, then V is the null vector of A, multiplied by a scaler. The null vector A can be obtained by performing SVD on A, yielding

$$A^{m\times n} = U^{m\times m} D^{m\times n} (S^t)^{n\times n}$$

The null vector A is the last column of S matrix. Note V is solved only up to a scale factor. The scale factor can be recovered using the fact that $||p_3||^2 = 1$. Let the null vector of A be V'. The scale factor $\alpha = \sqrt{\frac{1}{V'^2(9)+V'^2(10)+V'^2(11)}}$. Hence, $V = \alpha V'$.

Alternatively, we can also solve V by minimizing $||AV||^2$, which

Slide 12

yields $(A^tA)V = 0$ or $(A^tA)V = \lambda V$, where $\lambda = 0$. As a result, solution to V is the eign vector of matrix (A^tA) corresponding to zero eigen value.

Slide 13

Linear Solution 2

Let $A = [B \;\; b]$, where

$$B = \begin{pmatrix} M_1^t & 1 & \vec{0} & 0 & -c_1 M_1^t \\ \vec{0} & 0 & M_1^t & 1 & -r_1 M_1^t \\ & \vdots & & & \\ M_N^t & 1 & \vec{0} & 0 & -c_N M_N^t \\ \vec{0} & 0 & M_N^t & 1 & -r_N M_N^t \end{pmatrix}$$

$$b = (-c_1 \; -r_1 \; \ldots \; -c_N \; -r_N)^t$$

Slide 14

$$V = p_{34} \begin{pmatrix} Y \\ 1 \end{pmatrix}$$

$$Y = (p_1^t \; p_{14} \; p_2^t \; p_{24} \; p_3^t)^t / p_{34}$$

Then, $AV = p_{34}(BY + b)$. Since p_{34} is a constant, minimizing $||AV||^2$ is the same as minimizing $||BY + b||^2$, whose solution is $Y = -(B^t B)^{-1} B^t b$. The rank of matrix B must be eleven.

Slide 15

Linear Solution 2 (cont'd)

The solution to Y is up to a scale factor p_{34}. To recover the scale factor, we can use the fact that $|p_3| = 1$. The scale factor p_{34} can be recovered as $p_{34} = \frac{1}{\sqrt{Y^2(9)+Y^2(10)+Y^2(11)}}$, where $Y(9)$, $Y(10)$, and $Y(11)$ are the last 3 elements of Y. The final projection matrix (vector) is therefore equal to

$$V = \begin{pmatrix} p_{34} Y \\ p_{34} \end{pmatrix}$$

Slide 16

Linear Solution 3

Imposing the orthonormal constraint, $R^t = R^{-1}$, i.e., minimize

$$||AV||^2$$

subject to $R^t = R^{-1}$. Solution to this problem is non-linear !.

Slide 17

Linear Solution 3 (cont'd)

To yield a linear solution, we can impose one of the normal constraints, i.e., $||p_3||^2 = 1$, then the problem is converted to a constrained linear least-squares problem. That is, minimize $||AV||^2$ subject to $||p_3||^2 = 1$.

$$\epsilon^2 = ||AV||^2 + \lambda(||p_3||^2 - 1) \tag{2}$$

Slide 18

Decomposing A into two matrices B and C, and V into Y and Z

$$A = (B \;\; C)$$

$$V = \begin{pmatrix} Y \\ Z \end{pmatrix}$$

$$B^{2N\times 9} = \begin{pmatrix} M_1^t & 1 & \vec{0} & 0 & -c_1 \\ \vec{0} & 0 & M_1^t & 1 & -r_1 \\ & \vdots & & & \\ M_N^t & 1 & \vec{0} & 0 & -c_N \\ \vec{0} & 0 & M_N^t & 1 & -r_N \end{pmatrix} \qquad C^{2N\times 3} = \begin{pmatrix} -c_1 M_1^t \\ -r_1 M_1^t \\ \vdots \\ -c_N M_N^t \\ -r_N M_N^t \end{pmatrix}$$

Slide 19

$$Y = \begin{pmatrix} p_1 \\ p_{14} \\ p_2 \\ p_{24} \\ p_{34} \end{pmatrix} \qquad Z = p_3$$

Slide 20

Then equation 2 can be rewritten as

$$\epsilon^2 = ||BY + CZ||^2 + \lambda(||Z||^2 - 1)$$

Slide 21

Taking partial derivatives of ϵ^2 with respect to Y and Z and setting them to zeros yield

$$Y = -(B^tB)^{-1}B^tCZ$$
$$C^t(I - B(B^tB)^{-1}B^t)CZ = \lambda Z$$

Apparently, the solution to Z is the eigenvector of matrix $C^t(I - B(B^tB)^{-1}B^t)C$. Given Z, we can then obtain solution to Y.

Slide 22

Substituting Y into $||BY + CZ||^2$ leads to

$$\begin{aligned}
& ||BY + CZ||^2 \\
= & || - B(B^tB)^{-1}B^tCZ + CZ||^2 \\
= & ||(I - B(B^tB)^{-1}B^t)CZ||^2 \\
= & Z^tC^t(I - B(B^tB)^{-1}B^t)^t(I - B(B^tB)^{-1}B^t)CZ \\
= & Z^tC^t(I - B(B^tB)^{-1}B^t)CZ \\
= & Z^t\lambda Z \\
= & \lambda
\end{aligned}$$

This proves that solution to Z corresponds to the eigen vector of the smallest positive eigenvalue of matrix $C^t(I - B(B^tB)^{-1}B^t)C$.

Note $(I - B(B^tB)^{-1}B^t)(I - B(B^tB)^{-1}B^t) = (I - B(B^tB)^{-1}B^t)$

Slide 23

Other Linear Techniques

- Another new method at http://www.ecse.rpi.edu/homepages/qji/CV/new_method.pdf
- Zhang Zhengyou's method at http://research.microsoft.com/∼zhang/calib/

Slide 24

Robust Linear Method with RANSAC

The linear LSQ method is sensitive to image errors and outliers. One solution is to use a robust method. The most commonly used robust method in CV is the RANSAC (Random Sample Consensus) method. It works as follows

- Step 1: Randomly pick a subset of K points from N (K>6) pixels in the image and compute the projection matrix P using the selected points.
- Step 2: For each of the remaining pixels in the image, compute its projection error using the P computed from step 1. If it is within a threshold distance, increment a counter of the number of points (the "inliers") that agree with the hypothesized P.
- Step 3: Repeat Steps 1 and 2 for a sufficient number of times [a],

[a] the exact number of times is determined by the required probability that one of subset does not contain the outliers

Slide 25

and then select the subset of points corresponding to the P with the largest count.

- Step 4: Using the subset of points selected in Step 3 plus all of the other inlier points which contributed to the count, recompute the best P for all of these points.

Slide 26

Compute P: Non-linear Method

Let the 3D points be $M_i = (x_i, y_i, z_i)^t$ and the corresponding image points be $m_i = (c_i, r_i)^t$ for $i = 1, 2, \ldots, N$.

The criterion function to minimize is

$$\epsilon^2 = \sum_{i=1}^{N} \left(\frac{M_i^t p_1 + p_{14}}{M_i^t p_3 + p_{34}} - c_i\right)^2 + \left(\frac{M_i^t p_2 + p_{24}}{M_i^t p_3 + p_{34}} - r_i\right)^2 \qquad (3)$$

subject to $||p_3||^2 = 1$ and $(p_1 \wedge p_3) * (p_2 \wedge p_3) = 0$.

Note other criterion function such as the Sampson error (section 3.2.6 of Hartley's book) function. Sampson represents the first order approximation to the geometric error in equation 3.

Slide 27

Compute P: Non-linear Method (cont'd)

Another way of solving this problem is to perform minimization directly with respect to the intrinsic and extrinsic parameters.

Slide 28

Standard Non-Linear Solution

Let

$$\begin{aligned}
\Theta &= (c_0\ r_0\ f\ s_x\ s_y\ \omega\ \phi\ \kappa\ t_x\ t_y\ t_z)^t \\
g(\Theta, M_i) &= \frac{M_i^t p_1 + p_{14}}{M_i^t p_3 + p_{34}} \\
f(\Theta, M_i) &= \frac{M_i^2 p_2 + p_{24}}{M_i^t p_3 + p_{34}}
\end{aligned}$$

Then the problem can be stated as follows :

Find Θ by minimizing

$$\epsilon^2 = \sum_{i=1}^{N} [g(M_i, \Theta) - c_i]^2 + [f(M_i, \Theta) - r_i]^2$$

Methods to solve for non-linear optimization include Newton method, Gauss-Newton, and Levenberg-Marquardt method. See

Slide 29

chapter 3 of Forsyth and Ponce's book on how these methods work. Refer to appendix 4 of Hartley's book for additional iterative estimation methods. Non-linear methods all need good initial estimates to correctly converge. Implement one of the non-linear method using Matlab. It could improve the results a lot.

Slide 30

Linear Method v.s Non-linear Method

- Linear method is simple but less accurate and less robust
- Linear method does not require initial estimate
- Non-linear method is more accurate and robust but complex and require good initial estimates

The common approach in CV is two steps:

- Use a linear method to obtain initial estimates of the camera parameters.
- Refine the initial estimates using an non-linear method.

Slide 31

Data Normalization

Hartley introduces a data normalization technique to improve estimation accuracy. Details of the normalization technique may be found on section 3.4.4 of Hartley's book. This normalization should precede all estimation that involves image data.

A brief discussion of this normalization procedure can also be found at page 156 of Trucco's (textbook) book.

Slide 32

Compute Camera Parameters from P

$$P = \begin{pmatrix} s_x f r_1 + c_0 r_3 & s_x f t_x + c_0 t_z \\ s_y f r_2 + r_0 r_3 & s_y f t_y + r_0 t_z \\ r_3 & t_z \end{pmatrix}$$

$$\begin{aligned} r_3 &= p_3 & t_z &= p_{34} \\ c_0 &= p_1^t p_3 & r_0 &= p_2^t p_3 \\ s_x f &= \pm\sqrt{p_1^t p_1 - c_0^2} = \pm||p_1 \wedge p_3|| \\ s_y f &= \pm\sqrt{p_2^t p_2 - r_0^2} = \pm||p_2 \wedge p_3|| \\ t_x &= (p_{14} - c_0 t_z)/(s_x f) \\ t_y &= (p_{24} - r_0 t_z)/(s_y f) \end{aligned}$$

Slide 33

$$
\begin{aligned}
r_1 &= (p_1 - c_0 r_3)/(s_x f) \\
r_2 &= (p_2 - r_0 r_3)/(s_y f)
\end{aligned}
$$

Slide 34

Compute Camera Parameters from P (cont'd)

Alternatively, we can compute W algebraically from P. Since $P = WM = W[R \;\; T] = [WR \;\; WT]$, let P_3 be the first 3×3 submatrix of P, the $P_3 = WR$.

Hence,

$$K = P_3 P_3^t = WW^t$$

Given K, from equation $K = WW^t$, we can obtain W via Choleski factorization. Since K is symmetric, from SVD, we have:

$$K = UDU^t$$

where D is a diagonal matrix and U an orthonormal matrix, whose columns are the eigenvectors of K. Since D is diagonal, we may write $D = EE^t$, where

Slide 35

E is another diagonal matrix. As a result,

$$K = HH^t$$

where $H = UE$. Since H is not upper triangular yet, we can perform a

QR decomposition on H, this yields to $H = QR$, where Q is an upper triangular matrix and R is just a rotation matrix. W equals to Q.

Given W, T can be recovered as $T = W^{-1}P_4$, where P_4 is the last column of P, and $R = W^{-1}P_3$.

Slide 36

Approximate solution to imposing orthonormality

Let $\hat{R}$ be the estimated rotation matrix. It is not orthonormal. We can find an orthonormal matrix R that is closest to $\hat{R}$ via SVD. Let $\hat{R} = UDV^t$, find another matrix E that is closest to D and that satisfies $E^{-1} = E^t$. If we take E be a diagonal matrix, then we have $E = I$, the identity matrix. As a result, we have a new estimate of R, which is $\hat{\hat{R}} = UIV^t$. $\hat{\hat{R}}$ is the orthonormal matrix that is closest to $\hat{R}$.

Slide 37

Image Center using Vanishing Points

Let L_i, i=1,2, ..., N be parallel lines in 3D, l_i be the corresponding image lines. Due to perspective projection, the lines L_i appear to meet in a point, called *vanishing point*, defined as the common intersection of all the image lines i_i. Given the orientation of the lines be $N = (n_x, n_y, n_z)^t$ relative to the camera frame, then the coordinates of the vanishing point in the image frame are $(\frac{n_x}{n_z}, \frac{n_y}{n_z})$.

Let T be the triangle on the image plane defined by the three vanishing points of three mutually orthogonal sets of parallel lines in space. The image center is the orthocenter [a] of T.

[a] it is defined as the intersections of the three altitudes.

Slide 38

Camera Calibration using Lines and Conics

Besides using points, we can also perform camera calibration using correspondences between 2D/3D lines and 2D/3D conics. Furthermore, we can also extend the point-based camera calibration to estimate the lens distortion coefficient k_1. These can be topics for the final project.

Slide 39

Camera Calibration with Weak Perspective Projection

For weak perspective projection, we have

$$\begin{pmatrix} \bar{c} \\ \bar{r} \end{pmatrix} = M_{2\times 3} \begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{pmatrix}$$

Given $(\bar{c}_i, \bar{r}_i)$ and $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$, the goal is to solve for matrix M. A minimum of 3 points are enough to uniquely solve for the matrix M. And, more importantly, these points can be coplanar points.

Slide 40

Camera Calibration with Weak Perspective Projection

Given M and the parameterization for M introduced in the previous chapter, we have

$$\begin{aligned} \frac{fs_x}{\bar{z}_c} &= |m_1| \\ \frac{fs_y}{\bar{z}_c} &= |m_2| \end{aligned}$$

where m_1 and m_2 are the first row and the second row of the M matrix.

Then,

$$\begin{aligned} r_1 &= \frac{m_1}{|m_1|} \\ r_2 &= \frac{m_2}{|m_2|} \\ r_3 &= r_1 \times r_2 \end{aligned}$$

Slide 41

With weak perspective projection, we need a minimum of 3 points. But we can only solve the above parameters, i.e., the fs_x, fs_y, and the orientation of the object frame relative to the camera frame.

Slide 42

Camera Calibration with Lens Distortion

We present an approach for simultaneous linear estimation of the camera parameters and the lens distortion, based on the *division* lens distortion model proposed by Fitzgibbon [a]. According to the *divisional* model, we have

$$\begin{pmatrix} \hat{c} - c_0 \\ \hat{r} - r_0 \end{pmatrix} = (1 + ks^2) \begin{pmatrix} c - c_0 \\ r - r_0 \end{pmatrix}$$

where $s^2 = (\hat{c} - c_0)^2 + (\hat{r} - r_0)^2$. This is an approximation to the

[a] the paper appears in CVPR 01, pages 125-132

Slide 43

camera true distortion model. Hence,

$$\frac{\lambda}{1 + ks^2} \begin{pmatrix} \hat{c} - c_0 \\ \hat{r} - r_0 \\ 0 \end{pmatrix} = P \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} - \lambda \begin{pmatrix} c_0 \\ r_0 \\ 1 \end{pmatrix}$$

After solving for $\lambda = p_3(x\ y\ z\ 1)^t$ and with some algebraic simplifications yield

$$(D_1 + kD_2)V = 0$$

where V=$(p_1\ p_2\ p_3)^t$, p_i is the ith row of matrix P, and

$$D_1 = \begin{pmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -\hat{c}x & -\hat{c}y & -\hat{c}z & -c \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -\hat{r}x & -\hat{r}y & -\hat{r}z & -r \end{pmatrix}$$

Slide 44

$$D_2 = \begin{pmatrix} xs^2 & ys^2 & zs^2 & s^2 & 0 & 0 & 0 & 0 & -c_0xs^2 & -c_0ys^2 & -c_0 \\ 0 & 0 & 0 & 0 & xs^2 & ys^2 & zs^2 & s^2 & -r_0xs^2 & -r_0ys^2 & -r_0 \end{pmatrix}$$

The above equation can be solved as a polynomial eigen value problem. MATLAB function *polyeig* can be used to obtain the solution for both k and V. To use *ployeig* function, the matrices on the left side of above equations must be square matrices. To achieve this, multiple both sides of the above equation by $(D_1 + kD_2)^t$ yields the following, which can be solved for using *polyeig*

$$(D_1^tD_1 + k(D_1^tD_2 + D_2^tD_1) + k^2D_2^tD_2)V = 0^b$$

The solution, however, assumes the knowledge of the image center. We can fix it as the center of the image. Study shows that the precise location of the distortion center does not strongly affect the

[b] when k is small, D_1 is close to the A matrix.

Slide 45

correction (see Ref. 9 of Fitz's paper).

Alternatively, we can perform alternation, i.e., assume image center as the principal point, then use the above to compute k and the internal camera parameters. Then, substitute the computed center back to recompute k and the camera parameters. This process repeats until it converges, i.e., when the change in the estimated parameters is small.

The procedure can be summarized as follows:

1. Assume principal center is at image center. This allows to compute $\hat{c} - c_0, \hat{r} - r_0$, and $s^2 = (\hat{c} - c_0)^2 + (\hat{r} - r_0)^2$ for each point.
2. Use Polyeig to solve for k and matrix P
3. Obtain the intrinsic camera parameters from P
4. Repeat steps 2) and 3) with the new principal center until the

Slide 46

change in the computed intrinsic parameters is less than a pre-defined threshold.

Slide 47

Degeneracy with Camera Calibration

Degeneracy occurs when the solution to the projection matrix is not unique due to special spatial point configurations.

see sections 3.2.3 and 3.3.3 of Forsyth's book.

Slide 48

Camera Self Calibration

Self camera calibration refers to determining the interior camera parameters of a camera by using only image data obtained from different view points.

Slide 49

Methods for Camera Self-Calibration

- General camera movement (involving both rotation and translation)
- Only rotational movement
- Only translational movement

Slide 50

Camera Self-Calibration With Only Rotation

Let's assume that we select one camera frame as the reference frame and the object frame coincide with the reference frame. Let the image generated by the reference image be represented with subscript 0.

$$\lambda_0 \begin{pmatrix} c_0 \\ r_0 \\ 1 \end{pmatrix} = \mathbf{W M_0} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{W}[\mathbf{I} \quad \mathbf{0}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{W} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{4}$$

If we rotate the camera frame from the reference by a rotation

Slide 51

matrix R_i, we have

$$\lambda_i \begin{pmatrix} c_i \\ r_i \\ 1 \end{pmatrix} = \mathbf{W M_i} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{W}[\mathbf{R_i} \ \mathbf{0}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{W R_i} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{5}$$

Denote $\lambda_i = \frac{\lambda_0}{\lambda_i}$, substituting 4 to 5 to remove $(X, Y, Z)^t$ yields

$$\begin{pmatrix} c_i \\ r_i \\ 1 \end{pmatrix} = \lambda_i \mathbf{W R_i W}^{-1} \begin{pmatrix} c_0 \\ r_0 \\ 1 \end{pmatrix} \tag{6}$$

Slide 52

Let $\mathbf{B_i} = \mathbf{W R_i W^{-1}} = \begin{pmatrix} B_{i_{11}} & B_{i_{12}} & B_{i_{13}} \\ B_{i_{21}} & B_{i_{22}} & B_{i_{23}} \\ B_{i_{31}} & B_{i_{32}} & B_{i_{33}} \end{pmatrix}$, we have

$$\begin{pmatrix} c_i \\ r_i \\ 1 \end{pmatrix} = \lambda_i \begin{pmatrix} B_{i_{11}} & B_{i_{12}} & B_{i_{13}} \\ B_{i_{21}} & B_{i_{22}} & B_{i_{23}} \\ B_{i_{31}} & B_{i_{32}} & B_{i_{33}} \end{pmatrix} \begin{pmatrix} c_0 \\ r_0 \\ 1 \end{pmatrix} \tag{7}$$

This leads to three equations

$$\begin{aligned} c_i &= \lambda_i (B_{i_{11}} c_0 + B_{i_{12}} r_0 + B_{i_{13}}) \\ r_i &= \lambda_i (B_{i_{21}} c_0 + B_{i_{22}} r_0 + B_{i_{23}}) \\ 1 &= \lambda_i (B_{i_{31}} c_0 + B_{i_{32}} r_0 + B_{i_{33}}) \end{aligned} \tag{8}$$

Since $\lambda_i = 1/(B_{i_{31}} c_0 + B_{i_{32}} r_0 + B_{i_{33}})$, substituting λ_i to the above

Slide 53

equations yields

$$\begin{aligned} c_i(B_{i_{31}}c_0 + B_{i_{32}}r_0 + B_{i_{33}}) &= (B_{i_{11}}c_0 + B_{i_{12}}r_0 + B_{i_{13}}) \\ r_i(B_{i_{31}}c_0 + B_{i_{32}}r_0 + B_{i_{33}}) &= (B_{i_{21}}c_0 + B_{i_{22}}r_0 + B_{i_{23}}) \end{aligned}$$

Dividing the both sides of the above 2 equations by $B_{i_{33}}$ and they can be rewritten in matrix format

$$\begin{pmatrix} -c_0 & -r_0 & -1 & 0 & 0 & 0 & c_ic_0 & c_ir_0 \\ 0 & 0 & 0 & -c_0 & -r_0 & -1 & r_ic_0 & r_ir_0 \end{pmatrix} \mathbf{b_i} = \begin{pmatrix} -c_i \\ -r_i \end{pmatrix} \tag{9}$$

where $\mathbf{b_i} = (B_{i_{11}}\ B_{i_{12}}\ B_{i_{13}}\ B_{i_{21}}\ B_{i_{22}}\ B_{i_{23}}\ B_{i_{31}}\ B_{i_{32}})^T / B_{i_{33}}$. If we know at least 4 points in two images (such as $i = 0, 1$), we can solve for $\mathbf{b_i}$ up to a scale factor. The scale factor can be solved using the fact that the determinant of $\mathbf{WR_iW^{-1}}$ is unit.

If R_i is known, then we can solve for W using the equation $B_iW = WR_i$. In this case, one rotation, i.e., a total of two images,

Slide 54

is enough to solve for W.

To solve for W with a unknown R_i. From $\mathbf{B_i} = \mathbf{WR_iW^{-1}}$, we have

$$\mathbf{R_i} = \mathbf{W^{-1}B_iW} \qquad \mathbf{R_i^{-T}} = \mathbf{W^TB_i^{-1}W^{-T}}$$

Since $\mathbf{R} = \mathbf{R^{-T}}$, therefore we have

$$(\mathbf{WW^T})\mathbf{B_i^{-T}} = \mathbf{B_i}(\mathbf{WW^T}) \tag{10}$$

Assume $\mathbf{C} = \mathbf{WW^T}$, we have

$$\begin{aligned} \mathbf{C} &= \begin{pmatrix} s_xf & 0 & c_0 \\ 0 & s_yf & r_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_xf & 0 & 0 \\ 0 & s_yf & 0 \\ c_0 & r_0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} s_x^2f^2 + c_0^2 & c_0r_0 & c_0 \\ c_0r_0 & s_y^2f^2 + r_0^2 & r_0 \\ c_0 & r_0 & 1 \end{pmatrix} \end{aligned} \tag{11}$$

Slide 55

equation 10 can be rewritten

$$\mathbf{B_iCB_i^T} - \mathbf{C} = 0 \tag{12}$$

Since $\mathbf{C}$ is symmetric, Eq. 12 provides only six equations for 9 unknowns in C. To solve for C, it is necessary to use two B_i, i.e., two rotations, which leads to three images. Given two or more B_i, C can be solved using equation 12 up to a scale factor. The scale factor can subsequently be resolved using the fact that the last element of W is 1.

Given C, from equation $C = WW^t$, we can obtain W via Choleski factorization. Since C is symmetric, from SVD, we have:

$$C = UDU^t$$

where D is a diagonal matrix and U an orthonormal matrix, whose columns are the eigenvectors of C. Since D is diagonal, we may

Slide 56

write $D = EE^t$, where E is another diagonal matrix. As a result,

$$C = VV^t$$

where $V = UE$. Since V is not upper triangular yet, we can perform a QR decomposition on V, this yields to $V = WR$, where W is an upper triangular matrix and R is just a rotation matrix. W is what we want to compute. We can also prove that the solution is unique. The solution requires C be positive definite.

Slide 57

Camera Self-Calibration With Only Translation

Like for the previous case, the camera frame coincides with the object frame. We then translate the camera frame by $\mathbf{T}_i$. For the image points in the reference frame and the newly translated frame, we have

$$\lambda_0 \begin{pmatrix} c_0 \\ r_0 \\ 1 \end{pmatrix} = \mathbf{W}[\mathbf{I}\ \mathbf{T_0}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{W} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{13}$$

$$\lambda_i \begin{pmatrix} c_i \\ r_i \\ 1 \end{pmatrix} = \mathbf{W}[\mathbf{I}\ \mathbf{T_i}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{W} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{W}\mathbf{T_i} \tag{14}$$

Slide 58

From the above equations, we have

$$\lambda_i \begin{pmatrix} c_i \\ r_i \\ 1 \end{pmatrix} = \lambda_0 \begin{pmatrix} c_0 \\ r_0 \\ 1 \end{pmatrix} + \mathbf{W}\mathbf{T_i} \tag{15}$$

Equation 15 can be rewritten

$$\lambda_i \begin{pmatrix} c_i \\ r_i \\ 1 \end{pmatrix} = \lambda_0 \begin{pmatrix} c_0 \\ r_0 \\ 1 \end{pmatrix} + \begin{pmatrix} s_x f & 0 & c_0 \\ 0 & s_y f & r_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} t_{i_x} \\ t_{i_y} \\ t_{i_z} \end{pmatrix} \tag{16}$$

then, we get three equations, assuming T is known [a]

$$\begin{aligned} \lambda_i c_i &= \lambda_0 c_0 + (s_x f t_{i_x} + c_0 t_{i_z}) \\ \lambda_i r_i &= \lambda_0 r_0 + (s_y f t_{i_y} + r_0 t_{i_z}) \end{aligned}$$

[a] there is no linear solution if T is unknown

Slide 59

$$\lambda_i = \lambda_0 + t_{i_z} \tag{17}$$

Substituting $\lambda_i = \lambda_0 + t_{i_z}$ to the above two equations yields

$$\begin{aligned} \lambda_0 c_i + t_{i_z} c_i = \lambda_0 c_i + s_x f t_{i_x} + c_0 t_{i_z} \\ \lambda_0 r_i + t_{i_z} r_i = \lambda_0 r_i + s_y f t_{i_y} + r_0 t_{i_z} \end{aligned} \tag{18}$$

Equation 15 can be rewritten

$$\begin{pmatrix} t_{i_x} & 0 & t_{i_z} & 0 & c_0 - c_i \\ 0 & t_{i_y} & 0 & t_{i_z} & r_0 - r_i \end{pmatrix} \mathbf{W}' = \begin{pmatrix} t_{i_z} c_i \\ t_{i_z} r_i \end{pmatrix} \tag{19}$$

where $\mathbf{W}' = (s_x f \quad s_y f \quad c_0 \quad r_0 \quad \lambda_0)^T$. If we know at least 3 points in three images (producted by two translations), we can solve **W'**, then we can get the solution to **W**. Note the matrix is rank deficient (rank=3) if only using one translation, no matter how many points are used.

Slide 60

Pose Estimation

The goal of pose estimation is to estimate the relative orientation and position between the object frame and the camera frame, i.e., determining R and T or extrinsic camera parameters. The camera is usually assumes to be calibrated already.

Pose estimation is an area that has many applications including HCI, robotics, photogtametry, etc..

Slide 61

Linear Pose Estimation

For pose estimation, it is necessary to know 3D features and their 2D image projections. They are then used to solve for R and T.

Assume W is known, then from the projection equation

$$\lambda W^{-1} \begin{pmatrix} c \\ r \\ 1 \end{pmatrix} = [R, T] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Given more than 6 sets of 2D/3D points, we can solve for R and T linearly in the similar fashion to that of linear camera calibration. The solution, however, does not impose the constraint that R be orthnormal, i.e., $R^{-1} = R^t$.

We can perform a postprocessing of the estimated R to find

Slide 62

another $\hat{R}$, that is closest to R and orthnormal. See previous pages for details.

Alternatively, we can still have a linear solution if we impose one constraint, i.e., $||r_3|| = 1$ during optimization.

Slide 63

Non-linear Pose Estimation (cont'd)

Alternatively, we can set it up as a non-linear optimization problem, with the constraint of $R^{-1} = R^t$ imposed.

Slide 64

Pose Estimation Under Weak Perspective Projection

For weak perspective projection, we have

$$\begin{pmatrix} \bar{c} \\ \bar{r} \end{pmatrix} = M_{2\times 3} \begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{pmatrix}$$

Given $(\bar{c}_i, \bar{r}_i)$ and $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$, the goal is to solve for matrix M. A minimum of 3 points are enough to uniquely solve for the matrix M.

Slide 65

Pose Estimation Under Weak Perspective Projection

Given M and the parameterization for M introduced in the previous chapter, we have

$$\begin{aligned} \frac{f s_x}{\bar{s}} &= |m_1| \\ \frac{f s_y}{\bar{s}} &= |m_2| \end{aligned}$$

where m_1 and m_2 are the first row and the second row of the M matrix.

Then,

$$\begin{aligned} r_1 &= \frac{m_1}{|m_1|} \\ r_2 &= \frac{m_2}{|m_2|} \\ r_3 &= r_1 \times r_2 \end{aligned}$$

Slide 66

With weak perspective projection, we need a minimum of 3 points. But we can only solve the above parameters, i.e., the scale factor that represents the distance to the camera and the orientation of the object frame relative to the camera frame.