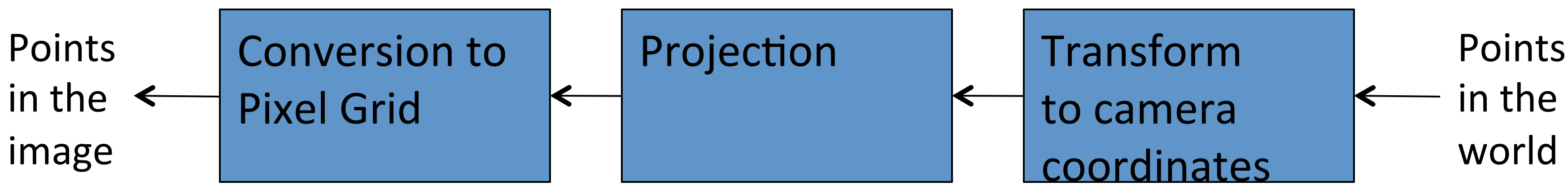
The Projection "Chain"



Projection Geometry: Standard Camera Coordinates

- By convention, we place the image in front of the optical center
  - typically we approximate by saying it lies one focal distance from the center
  - in reality this can't be true for a finite size chip!
- Optical axis is z axis pointing outward
- X axis is parallel to the scanlines (rows) pointing to the right!
- By the right hand rule, the Y axis must point downward
- Note this corresponds with indexing an image from the upper left to the lower right, where the X coordinate is the column index and the Y coordinate is the row index.

Intrinsic Parameters

Intrinsic Parameters describe the conversion from unit focal length metric to pixel coordinates (and the reverse)

$x_{mm} = -(x_{pix} - o_x)\, s_x$ --> $-1/s_x\, x_{mm} + o_x = x_{pix}$

$y_{mm} = -(y_{pix} - o_y)\, s_y$ --> $-1/s_y\, y_{mm} + o_y = y_{pix}$

or

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix}_{pix} = \begin{pmatrix} -1/s_x & 0 & o_x \\ 0 & -1/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}_{mm} = K_{\text{int}}\, p$$

It is common to combine scale and focal length together as the are both scaling factors; note projection is unitless in this case!

Putting it All Together

Now, using the idea of *homogeneous transforms*, we can write:

$$p' = \begin{pmatrix} R & & & T \\ 0 & 0 & 0 & 1 \end{pmatrix} p$$

R and T both require 3 parameters. These correspond to the 6 *extrinsic parameters* needed for camera calibration

Then we can write

q = Π p' for some projection model Π

Finally, we can write

u = K q for intrinsic parameters K

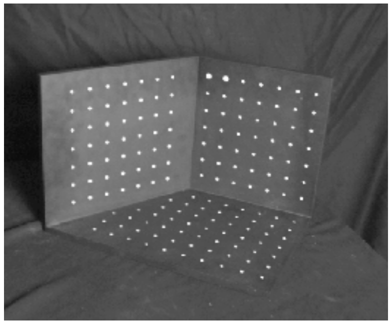
Camera parameters

- Summary:
  - points expressed in external frame
  - points are converted to canonical camera coordinates
  - points are projected
  - points are converted to pixel units

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{projection model} \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

point in pixel coords. | point in metric image coords. | point in cam. coords. | point in world coords.

Calibration using a 3D target



R.Y. Tsai, *An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision.*
Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, pp. 364-374, 1986

Camera Calibration

Calibration = the computation of the camera intrinsic and extrinsic parameters

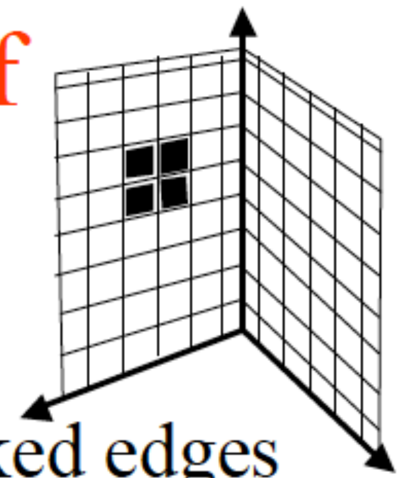
- General strategy:
  - view calibration object
  - identify image points
  - obtain camera matrix by minimizing error
  - obtain intrinsic parameters from camera matrix
- Most modern systems employ the multi-plane method
  - avoids knowing absolute coordinates of calibration points

- Error minimization:
  - Linear least squares
    - easy problem numerically
    - solution can be rather bad
  - Minimize image distance
    - more difficult numerical problem
    - solution usually rather good, but can be hard to find
      - start with linear least squares
  - Numerical scaling is an issue

Calibration Procedure

- Calibration target: 2 planes at right angle with checkerboard patterns (Tsai grid)
  - We know positions of pattern corners only with respect to a coordinate system of the target
  - We position camera in front of target and find images of corners
  - We obtain equations that describe imaging and contain internal parameters of camera
    - As a side benefit, we find position and orientation of camera with respect to target (camera ***pose***)

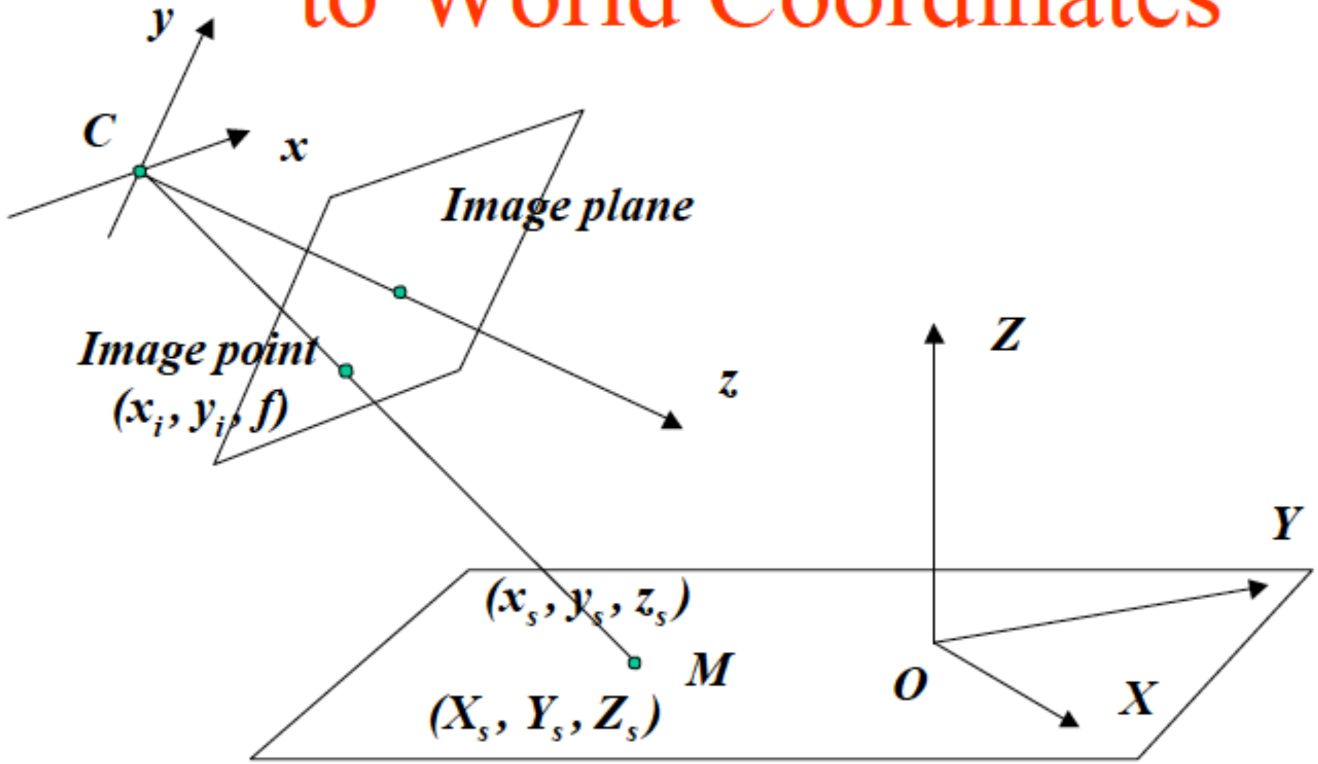
Image Processing of Image of Target



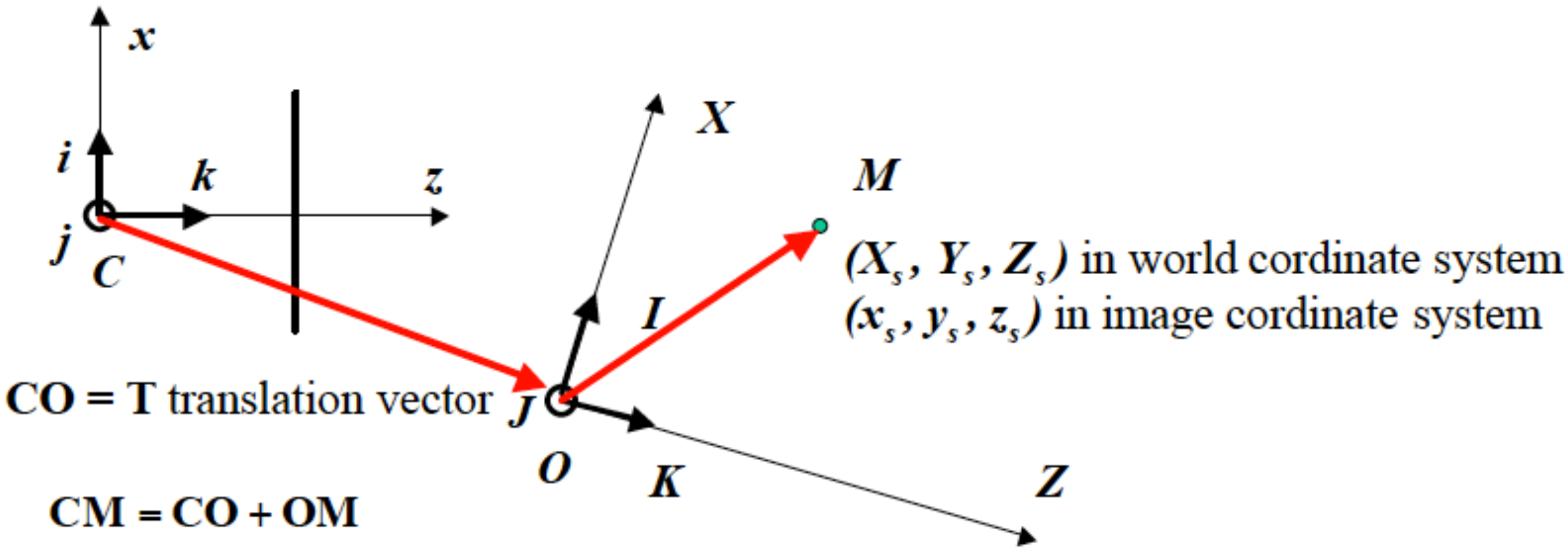
- Canny edge detection
- Straight line fitting to detected linked edges
- Intersecting the lines to obtain the image corners
- Matching image corners and 3D target checkerboard corners
  - By counting if whole target is visible in image
- We get pairs (image point)--(world point)

$$(x_i, y_i) \rightarrow (X_i, Y_i, Z_i)$$

From Camera Coordinates to World Coordinates



From Camera Coordinates to World Coordinates 2



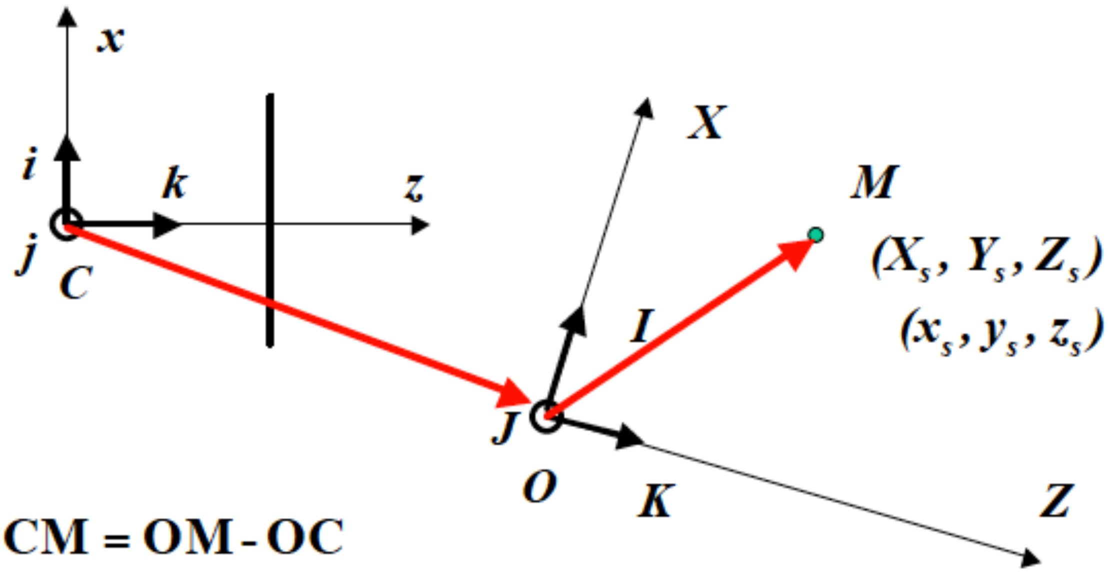
CM = CO + OM

$$x_S\mathbf{i} + y_S\mathbf{j} + z_S\mathbf{k} = T_x\mathbf{i} + T_y\mathbf{j} + T_z\mathbf{k} + X_S\mathbf{I} + Y_S\mathbf{J} + Z_S\mathbf{K}$$

$$x_S = T_x + X_S\mathbf{I.i} + Y_S\mathbf{J.i} + Z_S\mathbf{K.i}$$

$$\begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} \mathbf{I.i} & \mathbf{J.i} & \mathbf{K.i} \\ \mathbf{I.j} & \mathbf{J.j} & \mathbf{K.j} \\ \mathbf{I.k} & \mathbf{J.k} & \mathbf{K.k} \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ Z_S \end{bmatrix}$$

From Camera Coordinates to World Coordinates 3



$$\mathbf{CM} = \mathbf{OM} - \mathbf{OC}$$

$$x_S\mathbf{i} + y_S\mathbf{j} + z_S\mathbf{k} = (X_S - X_C)\mathbf{I} + (Y_S - Y_C)\mathbf{J} + (Z_S - Z_C)\mathbf{K}$$

$$x_S = (X_S - X_C)\mathbf{I.i} + (Y_S - Y_C)\mathbf{J.i} + (Z_S - Z_C)\mathbf{K.i}$$

$$\mathbf{x}_{\mathbf{cam}} = \mathbf{R}(\mathbf{X} - \tilde{\mathbf{C}}) \qquad (\mathbf{T} = -\mathbf{R}\tilde{\mathbf{C}})$$

$\tilde{\mathrm{C}}$ is vector OC expressed in world coordinate system

Homogeneous Coordinates 2

- Here we use $-\mathbf{R}\widetilde{\mathbf{C}}$ instead of $\mathbf{T}$

$$\begin{bmatrix} x_S \\ y_S \\ z_S \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\widetilde{\mathbf{C}} \\ \mathbf{0}_3^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ Z_S \\ 1 \end{bmatrix}$$

Linear Transformation from World Coordinates to Pixels

- Combine camera projection and coordinate transformation matrices into a single matrix **P**

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{K}\left[\mathbf{I}_3 \;\middle|\; \mathbf{0}_3\right] \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_S \\ y_S \\ z_S \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0}_3^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ Z_S \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{K}\left[\mathbf{I}_3 \;\middle|\; \mathbf{0}_3\right] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0}_3^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ Z_S \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_S \\ Y_S \\ Z_S \\ 1 \end{bmatrix} \qquad \mathbf{x} = \mathbf{P}\,\mathbf{X}$$

Properties of Matrix **P**

- Further simplification of **P**:

$$\mathbf{x} = \mathbf{K}\left[\mathbf{I}_3 \quad | \quad \mathbf{0}_3\right]\begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0}_3^T & 1 \end{bmatrix}\mathbf{X}$$

$$\left[\mathbf{I}_3 \quad | \quad \mathbf{0}_3\right]\begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0}_3^T & 1 \end{bmatrix} = \left[\mathbf{R} \quad -\mathbf{R}\tilde{\mathbf{C}}\right] = \mathbf{R}\left[\mathbf{I}_3 \quad | \quad -\tilde{\mathbf{C}}\right]$$

$$\mathbf{x} = \mathbf{K}\,\mathbf{R}\left[\mathbf{I}_3 \quad | \quad -\tilde{\mathbf{C}}\right]\mathbf{X}$$

$$\boxed{\mathbf{P} = \mathbf{K}\,\mathbf{R}\left[\mathbf{I}_3 \quad | \quad -\tilde{\mathbf{C}}\right]}$$

- **P** has 11 degrees of freedom:
  - 5 from triangular calibration matrix **K**, 3 from **R** and 3 from $\tilde{\mathbf{C}}$
- **P** is a fairly general 3 x 4 matrix
  - left 3x3 submatrix **KR** is non-singular

Calibration

- 1. Estimate matrix **P** using scene points and their images
- 2. Estimate the interior parameters and the exterior parameters

$$\mathbf{P} = \mathbf{K}\,\mathbf{R}\left[\mathbf{I}_3 \;\;|\;\; -\tilde{\mathbf{C}}\right]$$

  - Left 3x3 submatrix of **P** is product of upper-triangular matrix and orthogonal matrix

Finding Camera Translation

- Find homogeneous coordinates of C in the scene
- $\mathbf{C}$ is the null vector of matrix $\mathbf{P}$
  - $\mathbf{P}\,\mathbf{C} = 0$:

$$\mathbf{P} = \mathbf{K}\,\mathbf{R}\left[\mathbf{I}_3 \quad | \quad -\tilde{\mathbf{C}}\right]$$

$$\begin{bmatrix} 1 & 0 & 0 & -X_c \\ 0 & 1 & 0 & -Y_c \\ 0 & 0 & 1 & -Z_c \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Find null vector $\mathbf{C}$ of $\mathbf{P}$ using SVD
  - $\mathbf{C}$ is the unit singular vector of $\mathbf{P}$ corresponding to the smallest singular value (the last column of $\mathbf{V}$, where $\mathbf{P} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$ is the SVD of $\mathbf{P}$)

Finding Camera Orientation and Internal Parameters

- Left 3x3 submatrix **M** of **P** is of form **M=K R**
  - **K** is an upper triangular matrix
  - **R** is an orthogonal matrix
- Any non-singular square matrix **M** can be decomposed into the product of an upper-triangular matrix **K** and an orthogonal matrix **R** using the RQ factorization
  - Similar to QR factorization but order of 2 matrices is reversed

RQ Factorization of $\mathbf{M}$

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}, \quad \mathbf{R}_y = \begin{bmatrix} c' & 0 & s' \\ 0 & 1 & 0 \\ -s' & 0 & c' \end{bmatrix}, \quad \mathbf{R}_z = \begin{bmatrix} c'' & -s'' & 0 \\ s'' & c'' & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Compute $c = -\dfrac{m_{33}}{(m_{32}^2 + m_{33}^2)^{1/2}}, \quad s = \dfrac{m_{32}}{(m_{32}^2 + m_{33}^2)^{1/2}}$
- Multiply $\mathbf{M}$ by $\mathbf{R}_x$. The resulting term at (3,2) is zero because of the values selected for c and s
- Multiply the resulting matrix by $\mathbf{R}_y$, after selecting c' and s' so that the resulting term at position (3,1) is set to zero
- Multiply the resulting matrix by $\mathbf{R}_z$, after selecting c'' and s'' so that the resulting term at position (2,1) is set to zero

$$\mathbf{M}\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z = \mathbf{K} \Rightarrow \mathbf{M} = \mathbf{K}\mathbf{R}_z^T\mathbf{R}_y^T\mathbf{R}_x^T = \mathbf{K}\mathbf{R}$$

Computing Matrix **P**

- Use corresponding image and scene points
  - 3D points $\mathbf{X_i}$ in world coordinate system
  - Images $\mathbf{x_i}$ of $\mathbf{X_i}$ in image
- Write $\mathbf{x_i} = \mathbf{P}\ \mathbf{X_i}$ for all i
- Similar problem to finding projectivity matrix **H** (i.e. homography)

Computation of P

- $\mathbf{x_i} = \mathbf{P\,X_i}$ involves homogeneous coordinates, thus $\mathbf{x_i}$ and $\mathbf{P\,X_i}$ just have to be proportional: $\mathbf{x}_i \times \mathbf{P\,X}_i = 0$
- Let $\mathbf{p}_1^T$, $\mathbf{p}_2^T$, $\mathbf{p}_3^T$ be the 3 row vectors of $\mathbf{P}$

$$\mathbf{P\,X}_i = \begin{bmatrix} \mathbf{p}_1^T\mathbf{X}_i \\ \mathbf{p}_2^T\mathbf{X}_i \\ \mathbf{p}_3^T\mathbf{X}_i \end{bmatrix} \qquad \mathbf{x}_i \times \mathbf{P\,X}_i = \begin{bmatrix} v'_i\,\mathbf{p}_3^T\mathbf{X}_i - w'_i\,\mathbf{p}_2^T\mathbf{X}_i \\ w'_i\,\mathbf{p}_1^T\mathbf{X}_i - u'_i\,\mathbf{p}_3^T\mathbf{X}_i \\ u'_i\,\mathbf{p}_2^T\mathbf{X}_i - v'_i\,\mathbf{p}_1^T\mathbf{X}_i \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \mathbf{0}_4^T & -w'_i\,\mathbf{X}_i^T & v'_i\,\mathbf{X}_i^T \\ w'_i\,\mathbf{X}_i^T & \mathbf{0}_4^T & -u'_i\,\mathbf{X}_i^T \\ -v'_i\,\mathbf{X}_i^T & u'_i\,\mathbf{X}_i^T & \mathbf{0}_4^T \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = 0 \qquad \begin{bmatrix} \mathrm{p}_1 \\ \mathrm{p}_2 \\ \mathrm{p}_3 \end{bmatrix} \text{is a } 12 \times 1 \text{ vector}$$

Computation of P

- Third row can be obtained from sum of u'_i times first row - v'_i times second row

$$\begin{bmatrix} \mathbf{0}_4^T & -w'_i \mathbf{X}_i^T & v'_i \mathbf{X}_i^T \\ w'_i \mathbf{X}_i^T & \mathbf{0}_4^T & -u'_i \mathbf{X}_i^T \\ -v'_i \mathbf{X}_i^T & u'_i \mathbf{X}_i^T & \mathbf{0}_4^T \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = 0$$

- So we get 2 independent equations in 11 unknowns (ignoring scale)
- With 6 point correspondences, we get enough equations to compute matrix **P**

$$\mathbf{A}\,\mathbf{p} = 0$$

Solving $\mathbf{A}\ \mathbf{p} = 0$

- Linear system $\mathbf{A}\ \mathbf{p} = 0$
- When possible, have at least 5 times as many equations as unknowns (28 points)
- Minimize $\|\ \mathbf{A}\ \mathbf{p}\ \|$ with the constraint $\|\ \mathbf{p}\ \| = 1$
  - P is the unit singular vector of **A** corresponding to the smallest singular value (the last column of **V**, where $\mathbf{A} = \mathbf{U}\ \mathbf{D}\ \mathbf{V}^T$ is the SVD of **A**)
- Called Direct Linear Transformation (DLT)

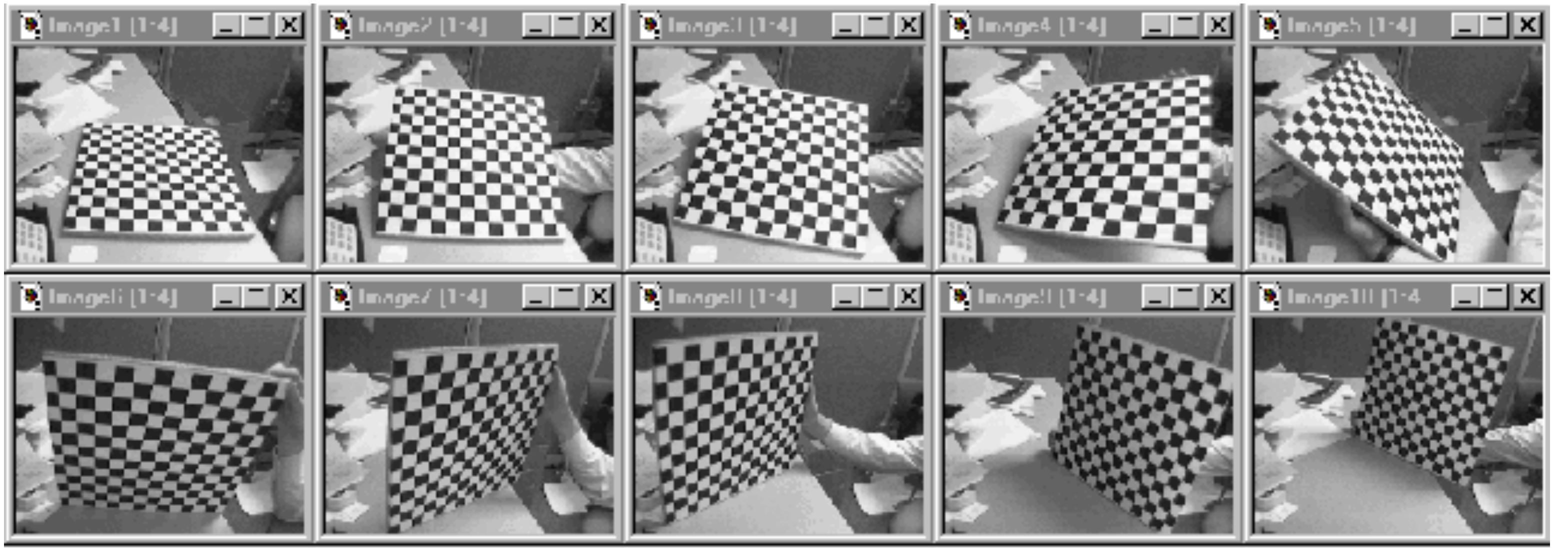
Improving **P** Solution with Nonlinear Minimization

- Find **p** using DLT
- Use as initialization for nonlinear minimization of $\sum_{\mathbf{i}} d(\mathbf{x_i}, \mathbf{P}\mathbf{X_i})^2$
  - Use Levenberg-Marquardt iterative minimization

Multi-planar calibration

Z. Zhang. *A flexible new technique for camera calibration.*
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 11, pp. 1330-1334, 2000

Alternative: multi-plane calibration



Images courtesy Jean-Yves Bouguet, Intel Corp.

Advantage

- Only requires a plane
- Don't have to know positions/orientations
- Good code available online!
  - Intel's OpenCV library: http://www.intel.com/research/mrl/research/opencv/
  - Matlab version by Jean-Yves Bouget: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
  - Zhengyou Zhang's web site: http://research.microsoft.com/~zhang/Calib/

Parameters

- (same model as before - just different notations)

Projection equation: $$s\widetilde{\mathbf{m}} = \mathbf{A}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\widetilde{\mathtt{M}}$$

Intrinsic parameters:

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

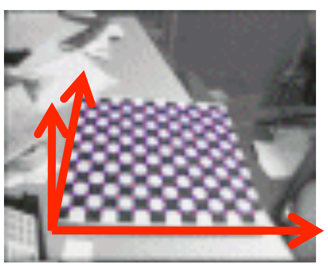
Planar homography

- Assume world coordinate system is such that Z=0 on the calibration pattern

- Projected pattern point:

$$s\tilde{\mathbf{m}} = \mathbf{A}\left[\mathbf{R} \quad \mathbf{t}\right]\tilde{\mathtt{M}}$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{t}\right]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$



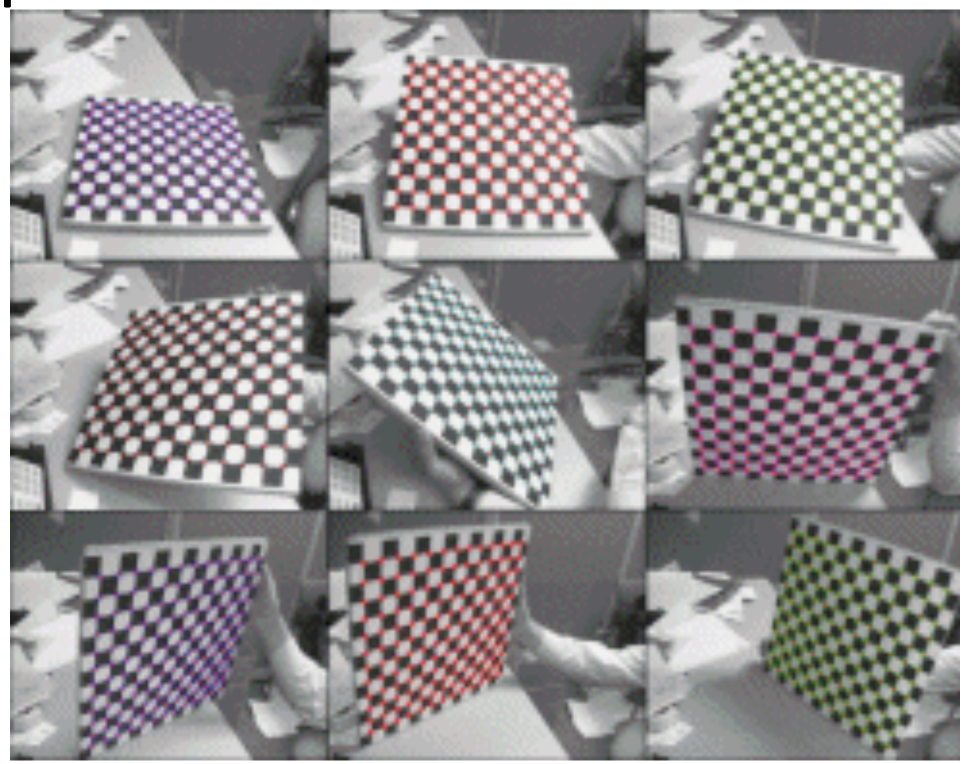
$$= \mathbf{A}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}\right]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}.$$

Pattern and its image are related by homography:

$$s\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathtt{M}} \quad \text{with} \quad \mathbf{H} = \mathbf{A}\left[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}\right]$$

Homography H

- Compute H for each image using known correspondences



Constraints on internal parameters

- Noting

$$\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix}$$

- And using the equation

$$\begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix} = \lambda\mathbf{A}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}$$

- One derives two equations on internal parameters only:

$$\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2 = 0$$
$$\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_1 = \mathbf{h}_2^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2$$

Computing Intrinsics

- Rotation Matrix is orthogonal....

$$r_i^T r_j = 0$$
$$r_i^T r_i = r_j^T r_j$$

- Write the homography in terms of its columns...

$$\begin{aligned} h_1 &= sAr_1 \\ h_2 &= sAr_2 \\ h_3 &= sAt \end{aligned}$$

Computing Intrinsics

- Derive t

$$
\begin{aligned}
h_1 &= sAr_1 \\
\frac{1}{s}A^{-1}h_1 &= r_1 \\
\frac{1}{s}A^{-1}h_2 &= r_2
\end{aligned}
$$

$$
\begin{aligned}
r_1^T r_2 &= 0 \\
h_1^T A^{-T} A^{-1} h_2 &= 0
\end{aligned}
$$

$$
\begin{aligned}
r_1^T r_1 &= r_2^T r_2 \\
h_1^T A^{-T} A^{-1} h_1 &= h_2^T A^{-T} A^{-1} h_2
\end{aligned}
$$

Closed-Form Solution

$$\text{Let } B = A^{-T}A^{-1} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta))}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta))}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

- Notice B is symmetric, 6 parameters can be written as a vector b.
- From the two constraints, we have $h_1^T B h_2 = v_{12} b$

$$\begin{bmatrix} v_{ij}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0;$$

- Stack up n of these for n images and build a 2n*6 system.
- Solve with SVD.
- Intrinsic parameters “fall-out” of the result easily using algebra

Solve for extrinsic parameters

- Using equation (A is known)

$$\begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix} = \lambda \mathbf{A} \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}$$

- One obtains

$$\lambda = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\| = 1/\|\mathbf{A}^{-1}\mathbf{h}_2\|$$

- Issue: due to noise r1 and r2 may not be orthogonal

$$\begin{aligned}\mathbf{r}_1 &= \lambda \mathbf{A}^{-1}\mathbf{h}_1 \\ \mathbf{r}_2 &= \lambda \mathbf{A}^{-1}\mathbf{h}_2 \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \lambda \mathbf{A}^{-1}\mathbf{h}_3\end{aligned}$$

(-> board)

Non-linear Refinement

- Closed-form solution minimized algebraic distance.
- Since full-perspective is a non-linear model
  - Can in tanger

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||m_{ij} - \hat{m}(A, R_k, T_k, M_j)||^2$$

  - Use maximum likelihood inference for our estimated parameters.

Notes

- Improvement (like for previous method)
  - use non-linear optimization to refine the results
  - compute radial distortion

- Degenerate configurations ?
  - Pure translation
  - Parallel planes