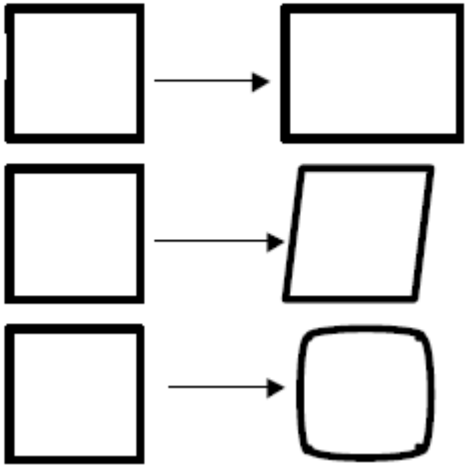
Camera Calibration

CAP5416 – Computer Vision

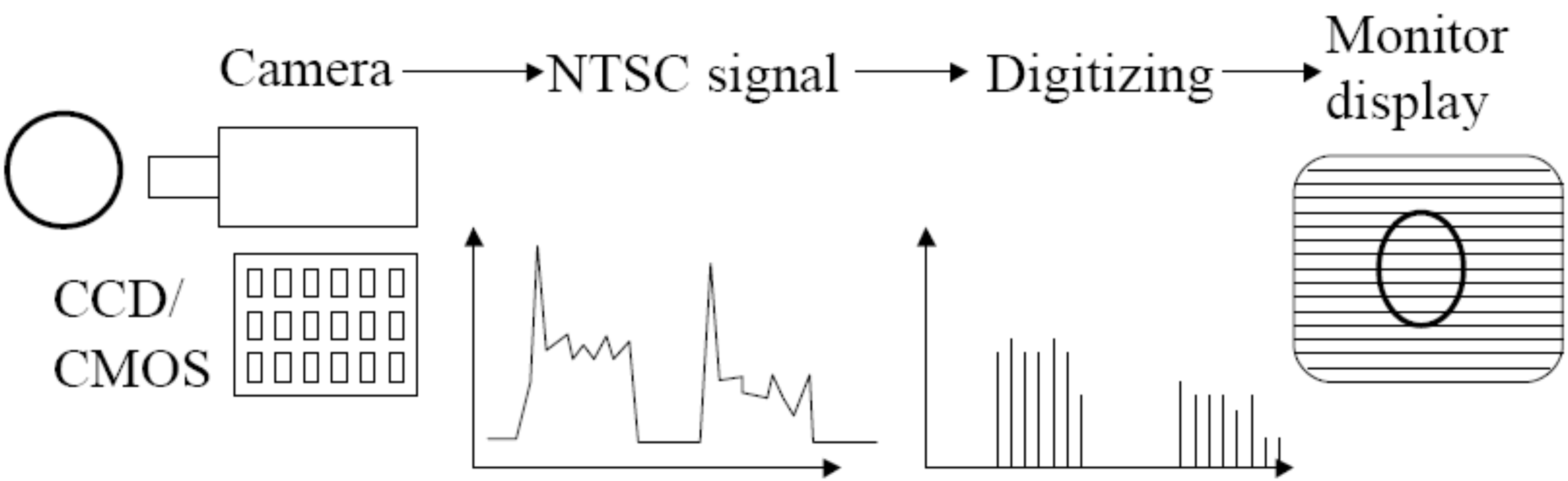
Camera Calibration

- Finding camera's internal parameters that effect the imaging process.
  1. Position of the image center on the image.
  2. Focal length
  3. Scaling factors for row and column pixels.
  4. Lens distortion.



Row and Column Pixels Scaling

1. Camera pixels are not necessarily squares
2. Analog output and digitization.



Camera Calibration

Each camera can be considered as a function: a function that takes each 3D point to a point in 2D image plane.

(X, Y, Z) ------ > (x, y)

Camera calibration is about finding (or approximating) this function.

Difficulty of the problem depends on the assumed form of this function, e.g., perspective model, radial distortion of the lens.

Camera Calibration

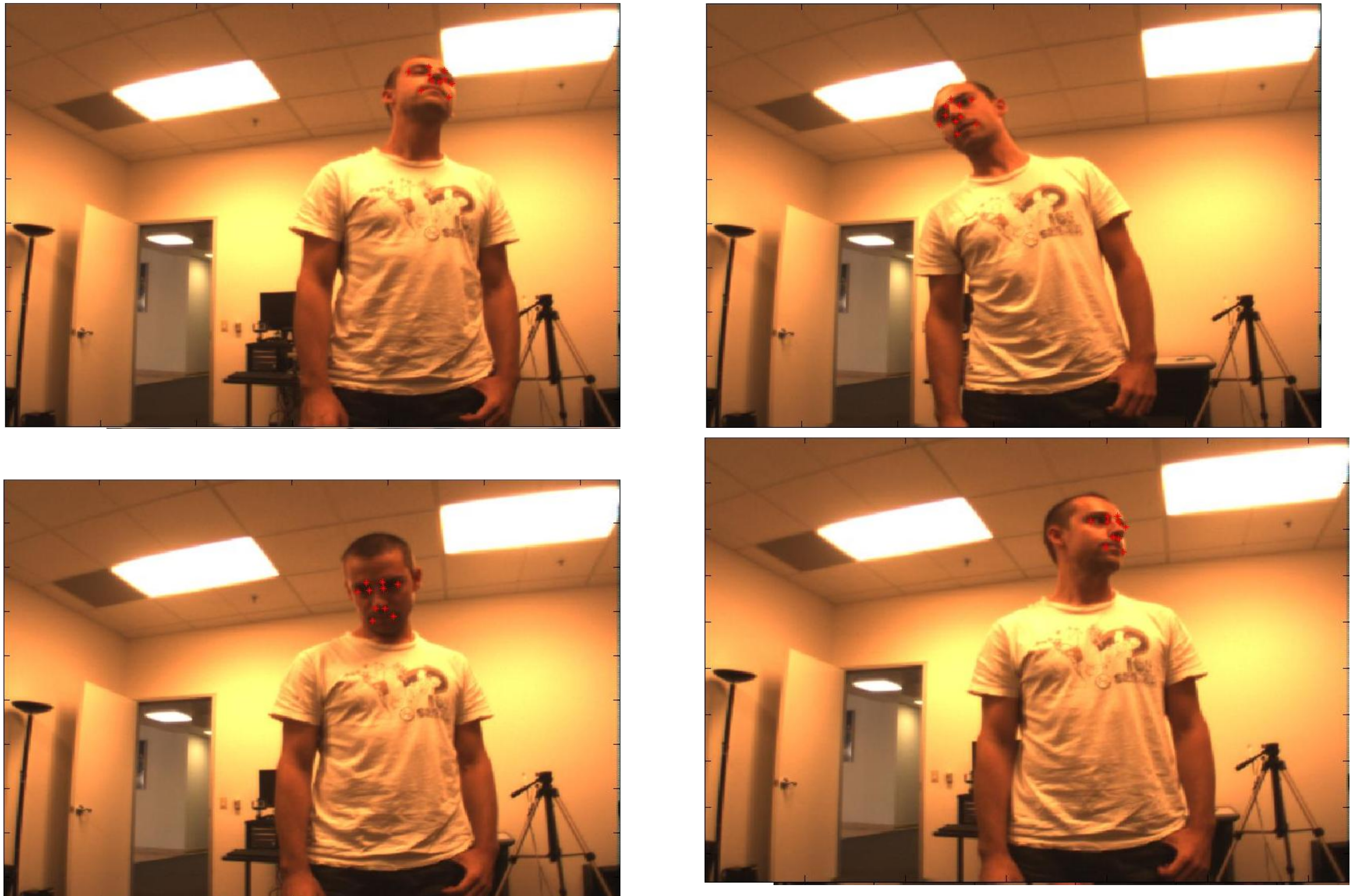
Good camera calibration is needed when we want to reconstruct the 3D geometry from images, with applications to :

- Robotics, human-robot interaction
- Robot navigation
- Scene reconstruction

Example: Locating feature points in 3D

Suppose the camera is calibrated and the 3D positions of the feature pts. are known for the first frame. Also given: 2D tracker

Example: Locating feature pts. in 3D using calibrated camera



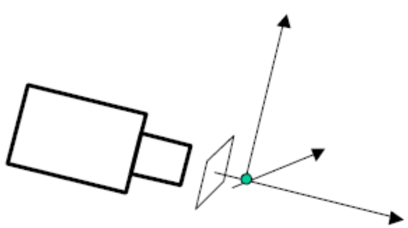
Camera Calibration

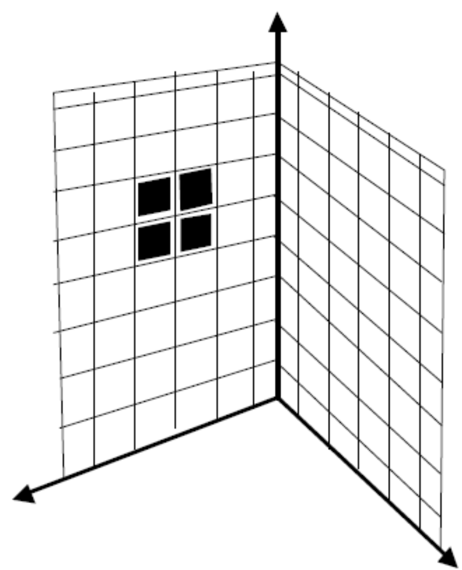
- Hundreds of published papers on this topic.
- We will stick with linear algebraic methods here
- Can be used as initialization for iterative non-linear method.
- Vanishing points and projective geometry

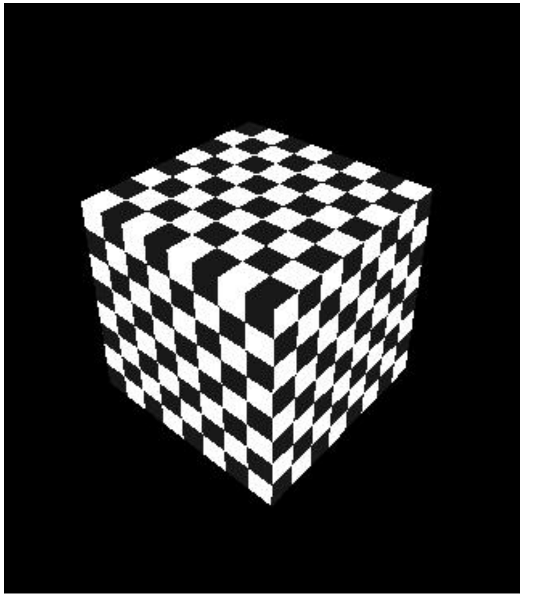
Calibration Procedure

Calibration: finding the function (defined by the camera) that maps 3D points to 2D image plane.

First step : Obtain pairs of corresponding 3D and 2D points. (X_1, Y_1, Z_1) (x_1, y_1), (X_2, Y_2, Z_2) (x_2, y_2), ….







Calibration Procedure

Calibration Target: Two perpendicular planes with chessboard pattern.

1. We know the 3D positions of the corners with respect to a coordinates system defined on the target.
2. Place a camera in front of the target and we can locate the corresponding corners on the image. This gives us the correspondences.
3. Recover the equation that describes imaging projection and camera's internal parameters. At the same time, also recover the relative orientation between the camera and the target (pose).

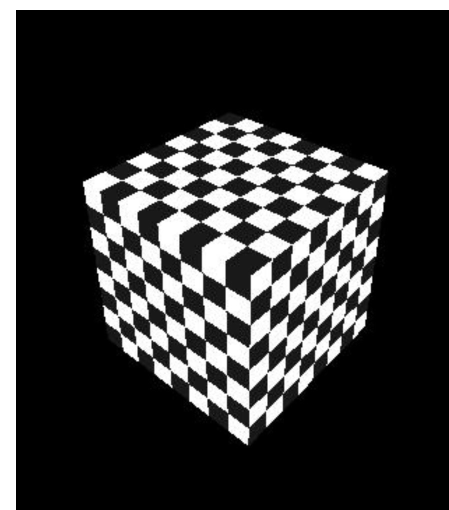
Finding Corners

1. Corner detector
2. Canny Edge detector plus fitting lines to the detected edges. Find the intersections of the lines.
3. Manual input.

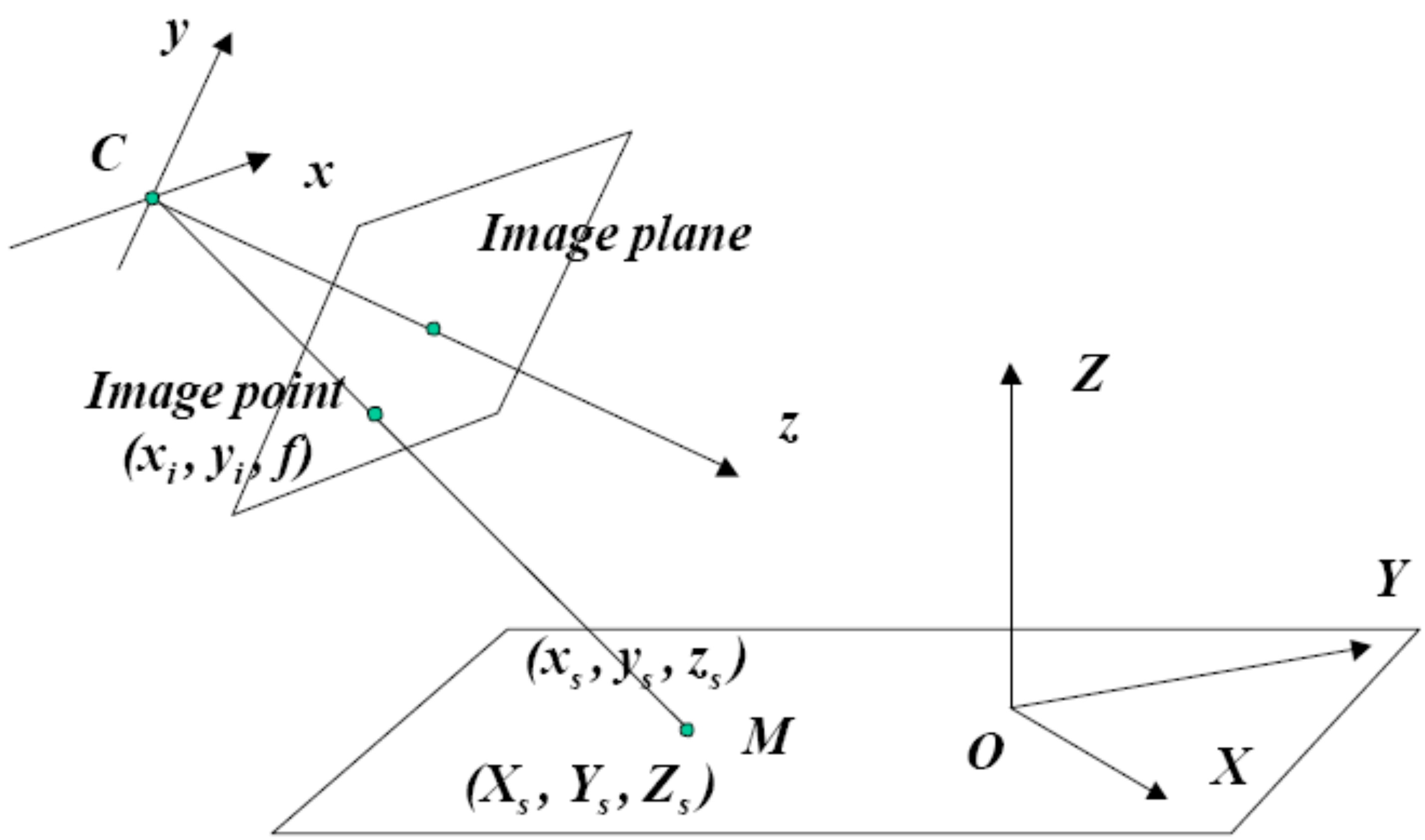
Matching 3D and 2D points (we know the number of corners) by counting. This gives corresponding pairs

(world point) < --- > (image point)

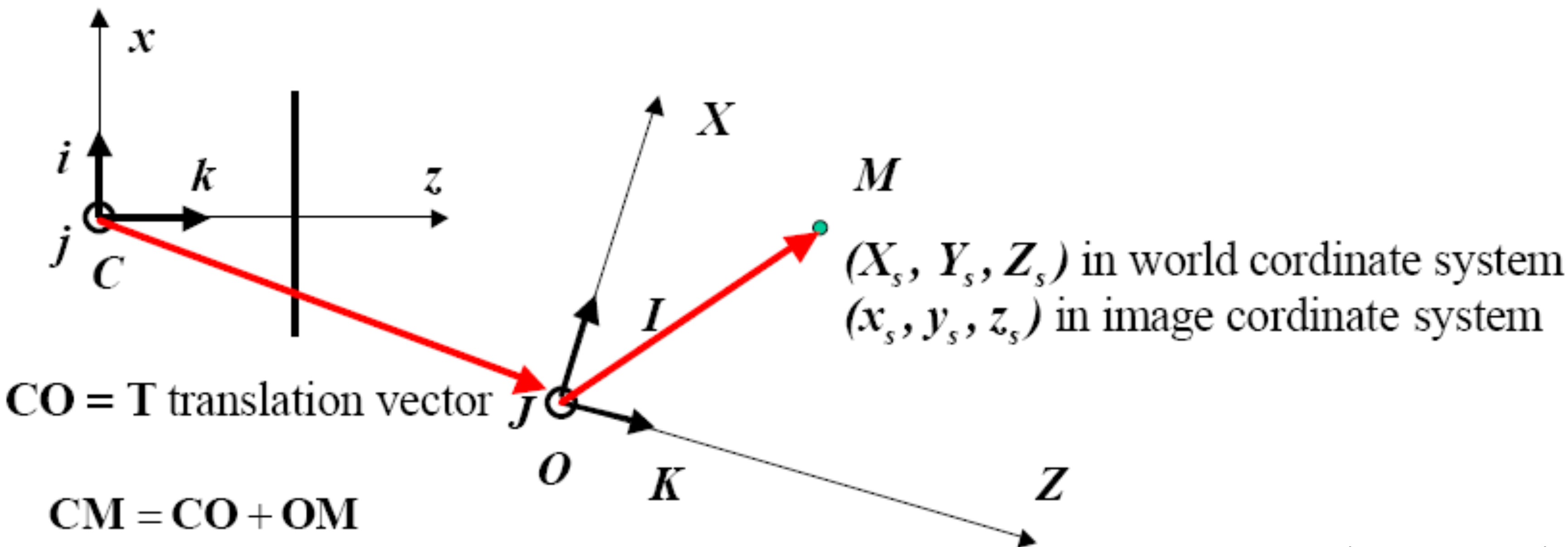
(X_1, Y_1, Z_1) (x_1, y_1),



World Coordinates and Camera Coordinates



Camera Frame to World Frame



$$\mathbf{CM} = \mathbf{CO} + \mathbf{OM}$$

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

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

Rotation matrix

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

Take inner product on both sides of the first equation with the unit vector **i.**

Homogeneous Coordinates

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

$$\begin{bmatrix} x_S \\ y_S \\ z_S \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I.i} & \mathbf{J.i} & \mathbf{K.i} & T_x \\ \mathbf{I.j} & \mathbf{J.j} & \mathbf{K.j} & T_y \\ \mathbf{I.k} & \mathbf{J.k} & \mathbf{K.k} & T_z \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ Z_S \\ 1 \end{bmatrix}$$

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

Let $\tilde{C} = -\mathbf{R}^\mathbf{t}\,\mathbf{T}$

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

Perspective Projection

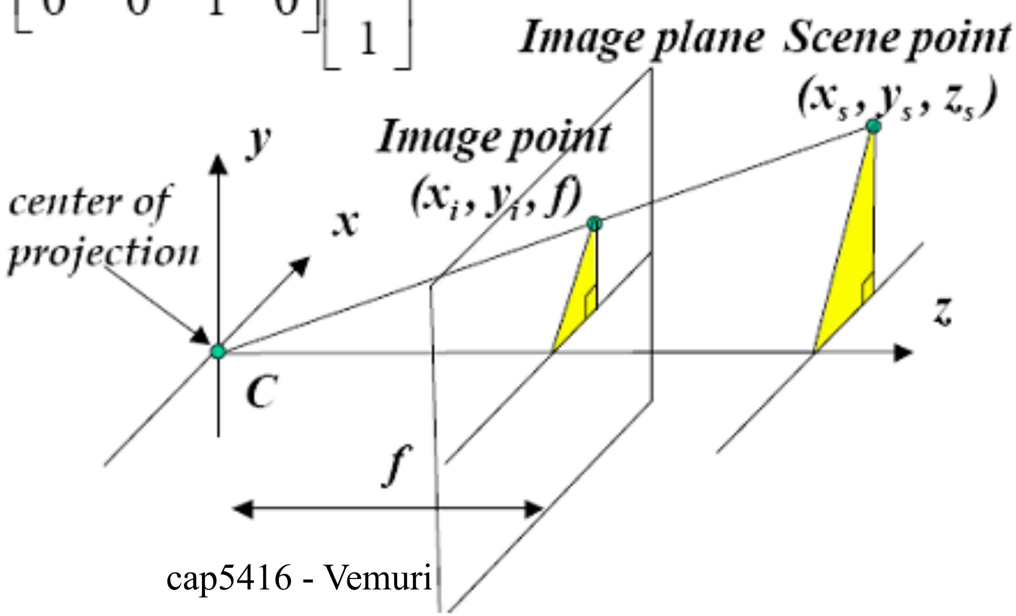
Use Homogeneous coordinates, the perspective projection becomes linear.

$$x_i = f\frac{x_s}{z_s}$$

$$y_i = f\frac{y_s}{z_s}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

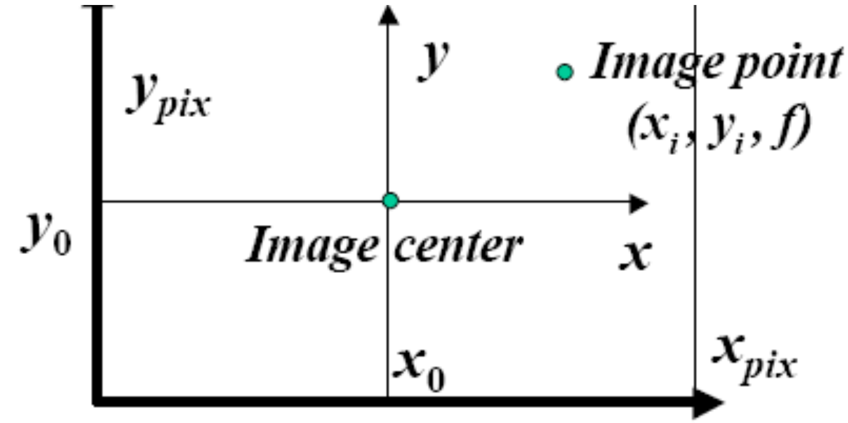
$$x_i = u/w, \quad y_i = v/w$$



Pixel Coordinates

Transformation uses:

- image center (x_0, y_0)
- scaling factors k_x and k_y



$$x_i = f\frac{x_s}{z_s} \qquad x_{pix} = k_x x_i + x_0 = f\,k_x\frac{x_s + z_s x_0}{z_s}$$

$$y_i = f\frac{y_s}{z_s} \qquad y_{pix} = k_y y_i + y_0 = f\,k_y\frac{y_s + z_s y_0}{z_s}$$

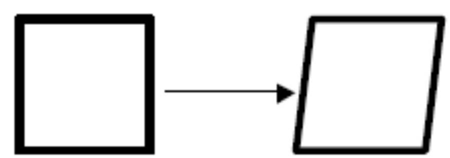
$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \text{ with } \begin{matrix} \alpha_x = f\,k_x \\ \alpha_y = f\,k_y \end{matrix} \quad \text{then} \quad \begin{matrix} x_{pix} = u'/w' \\ y_{pix} = v'/w' \end{matrix}$$

Calibration Matrix

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad \text{with} \quad \begin{matrix} \alpha_x = f\,k_x \\ \alpha_y = -f\,k_y \end{matrix} \qquad \begin{matrix} x_{pix} = u'/w' \\ y_{pix} = v'/w' \end{matrix}$$

$$\begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \mathbf{K}\left[\mathbf{I}_3 \quad | \quad \mathbf{0}_3\right]$$

- α_x and α_y "focal lengths" in pixels
- x_0 and y_0 coordinates of image center in pixels
- Added parameter s is skew parameter



- **K** is called *calibration matrix.* **Five degrees of freedom.**
  - **K** is a 3x3 upper triangular matrix

Putting Everything Together

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

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

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

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

Calibration

1. Estimate matrix **P** using scene points and their images

2. Estimate the intrinsic and extrinsic parameters

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

Left 3x3 submatrix is the product of an upper triangular matrix and an orthogonal matrix.

Computing the 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

Computing the Matrix P

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

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

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

Computing the Matrix P

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

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

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

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

Computing the Matrix P

- 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 $\mathbf{A} = \mathbf{U}\ \mathbf{D}\ \mathbf{V}^{\mathbf{T}}$ is the SVD of **A**)

Computing the translation component

- Find homogeneous coordinates of C in the scene
- **C** is the null vector of matrix **P**
  - **P C** = 0:

$$\mathbf{P} = \mathbf{K}\,\mathbf{R}\left[\mathbf{I}_3 \quad | \quad -\tilde{\mathbf{C}}\right] \qquad \begin{bmatrix} 1 & 0 & 0 & X_c \\ 0 & 1 & 0 & Y_c \\ 0 & 0 & 1 & Z_c \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

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

Computing the Matrix P

We now know **P** and we can get the translation **C.**

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

Then let M = KR be the upper left (3,3) submatrix of P. Find the QR decomposition of M to get K and R !! Note that Q is upper triangular =K

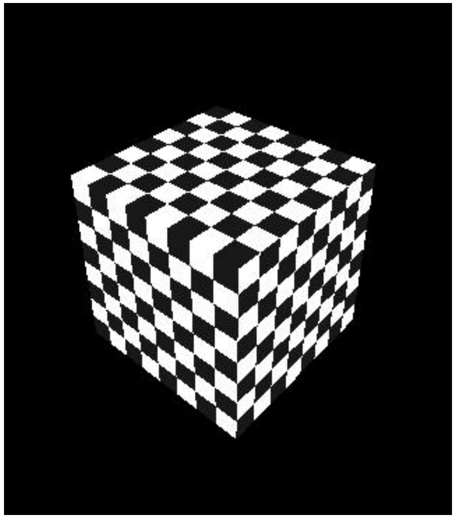
Further Improvement

Use as initialization for nonlinear minimization of $\sum_{\mathbf{i}} d(\mathbf{x}_{\mathbf{i}}, \mathbf{P}\mathbf{X}_{\mathbf{i}})^2$

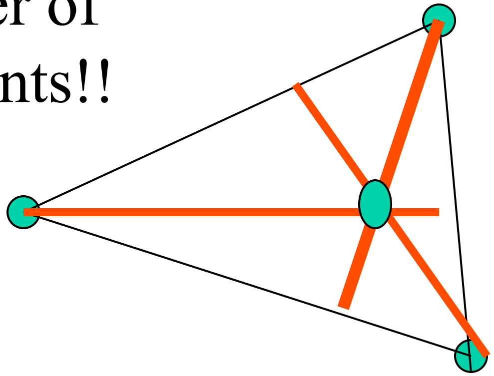
Can use a gradient based method e.g. Newton's method

Vanishing Points and Image Center

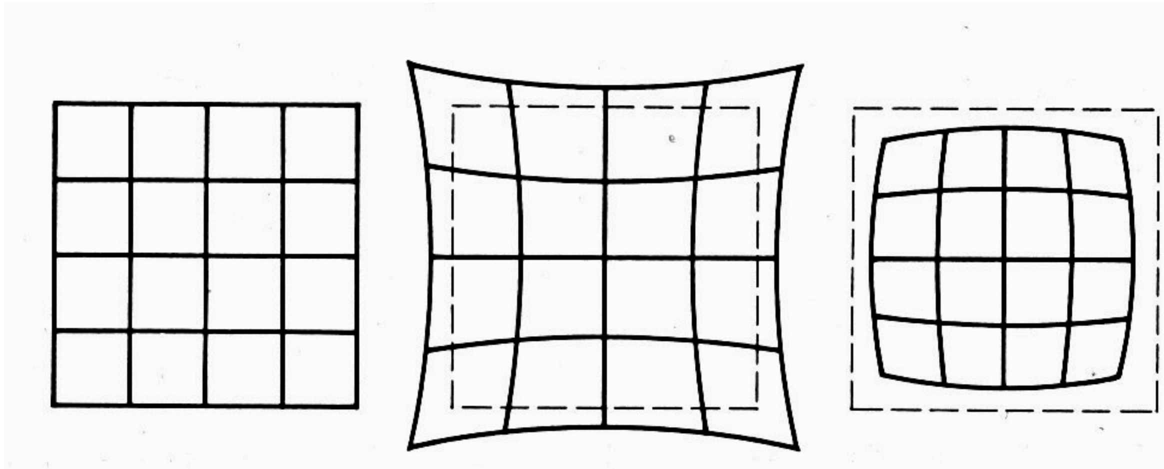
Parallel lines in 3D have a vanishing point on the 2D image plane. Point of intersection of a line parallel to these lines (and passing through the camera center) with the image plane is the vanishing point. It depends only on the direction of these lines & not the position.



Take three bundles of mutually perpendicular lines in 3D and compute the three vanishing points. The image center is the orthocenter of the triangle formed by the 3 vanishing points!!



Lens Distortion (Read the ref. for details)

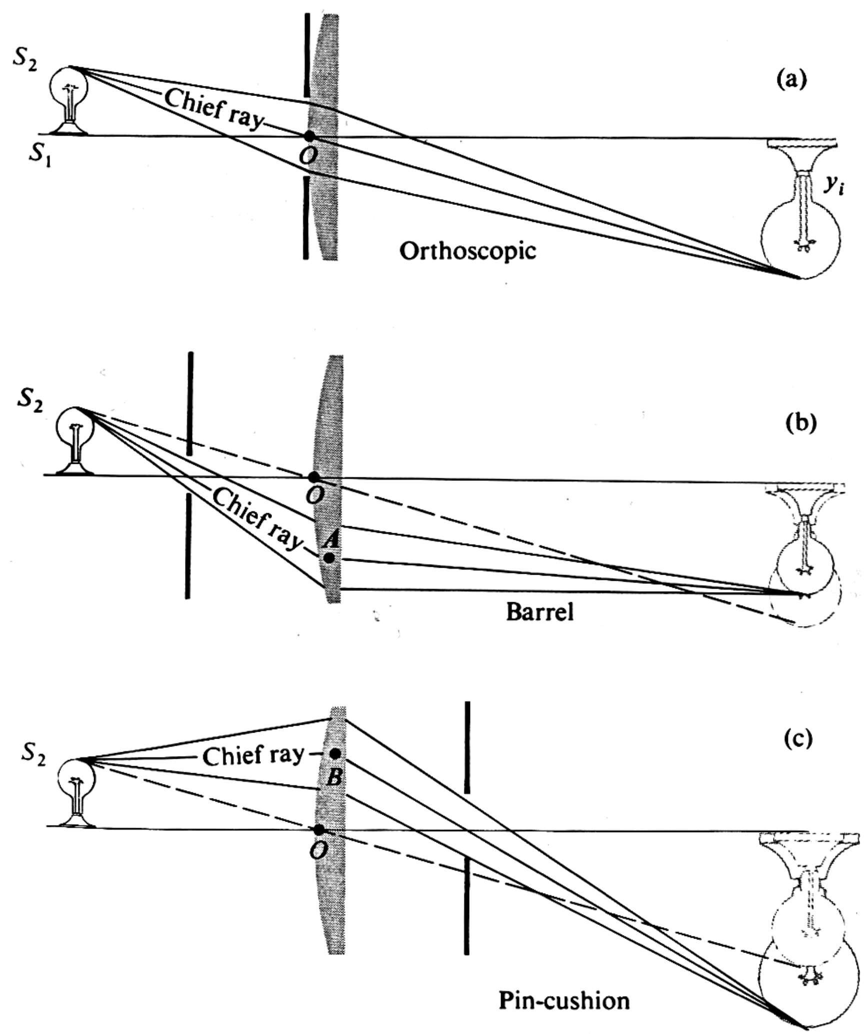


No distortion Pin cushion Barrel

- Radial distortion of the image
  - Caused by imperfect lenses
  - Deviations are most noticeable for rays that pass through the edge of the lens

Correcting radial distortion

from Helmut Dersch



S_2
S_1
Chief ray
O
(a)
y_i
Orthoscopic
S_2
Chief ray
O
A
(b)
Barrel
S_2
Chief ray
B
O
(c)
Pin-cushion

Modeling distortion

Project $(\widehat{x}, \widehat{y}, \widehat{z})$ to "normalized" image coordinates

$$x'_n = \widehat{x}/\widehat{z}$$
$$y'_n = \widehat{y}/\widehat{z}$$

Apply radial distortion

$$r^2 = {x'_n}^2 + {y'_n}^2$$
$$x'_d = x'_n(1 + \kappa_1 r^2 + \kappa_2 r^4)$$
$$y'_d = y'_n(1 + \kappa_1 r^2 + \kappa_2 r^4)$$

Apply focal length translate image center

$$x' = f x'_d + x_c$$
$$y' = f y'_d + y_c$$

- To model lens distortion
  - Use above projection operation instead of standard projection matrix multiplication

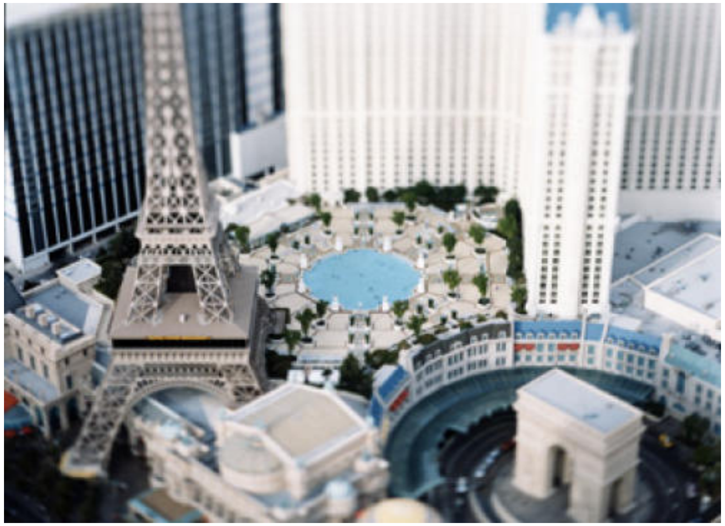
Other types of projection

- Lots of intriguing variants…
- (I'll just mention a few fun ones)

360 degree field of view…

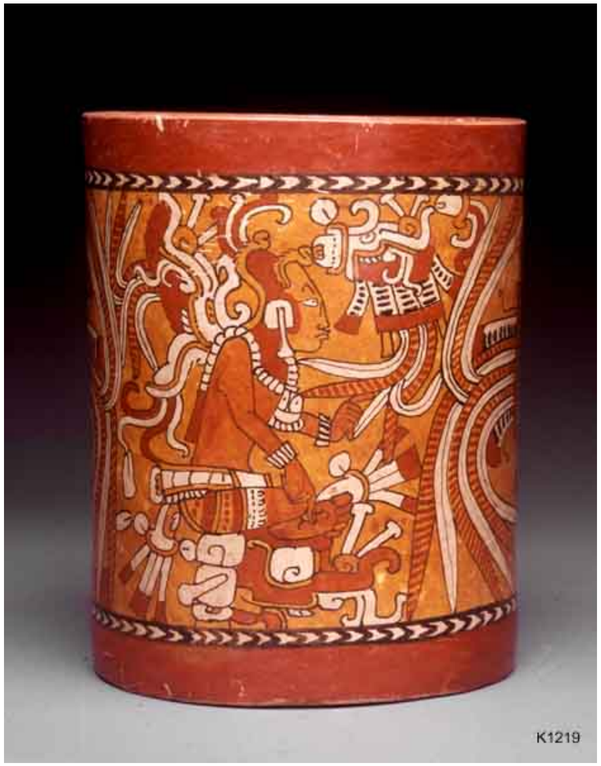
- Basic approach
  - Take a photo of a parabolic mirror with an orthographic lens (Nayar)
  - Or buy a lens from a variety of omnicam manufacturers…
    - See http://www.cis.upenn.edu/~kostas/omni.html

http://www.northlight-images.co.uk/article_pages/tilt_and_shift_ts-e.html



Tilt-shift images from Olivo Barbieri
and Photoshop imitations

Rotating sensor (or object)

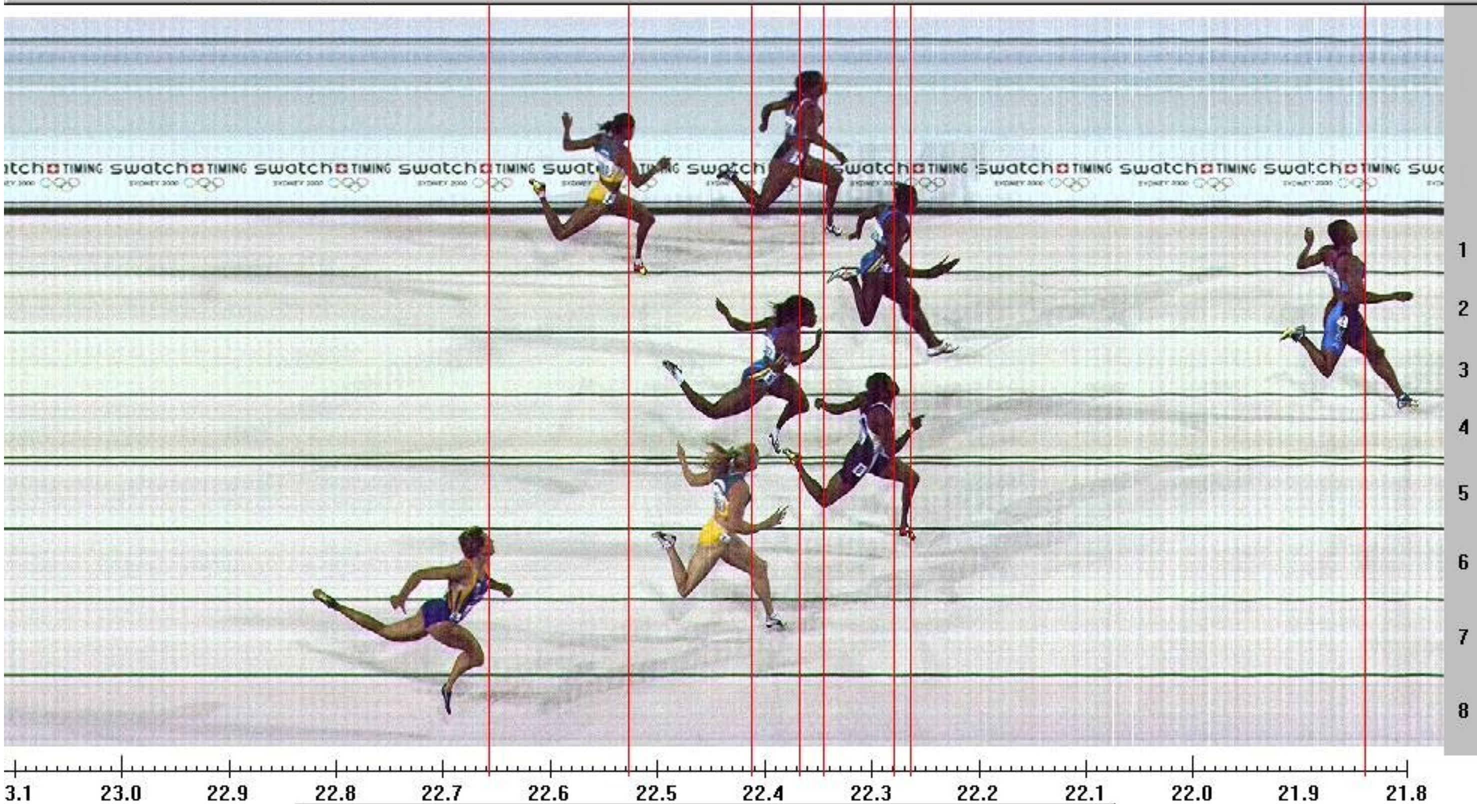


Rollout Photographs © Justin Kerr

http://research.famsi.org/kerrmaya.html

Also known as "cyclographs", "peripheral images"

The 2000 Sydney Olympic Games - 200m Women Final



Results Wind : + 0.7 m/s

Rank	La	Bib Nu			Time	R_time
1.	4	3357	Jones Marion	USA	21.84	0.174
2.	3	1174	Davis-Thompson Pauline	BAH	22.27	0.185
3.	6	3058	Jayasinghe Susanthika	SRI	22.28	0.207
4.	1	2291	McDonald Beverly	JAM	22.35	0.151
5.	5	1178	Ferguson Debbie	BAH	22.37	0.196
6.	7	1111	Gainsford-Taylo Melinda	AUS	22.42	0.178
7.	2	1110	Freeman Cathy	AUS	22.53	0.235
8.	8	3239	Pintusevych Zhanna	UKR	22.66	0.190

Start: 28. 9.2000 19:57:19.033 @414
Print: 28. 9.2000 20:00:54 @417

Scan'O'Vision Color
Race ID: W200FI00

TM © SOCOG 1996