3D Reconstruction of Scene Structure and Camera Motion

Gang Qian
Arizona State University

Reference: Multiple View Geometry in Computer Vision, by *Hartley and Zisserman*
Some slides modified from Marc Pollefeys at UNC
http://www.cs.unc.edu/Research/vision/comp256/

Three questions:

(i) **Correspondence geometry:** Given an image point x in the first image, how does this constrain the position of the corresponding point x' in the second image?

(ii) **Camera geometry (motion):** Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, i=1,…,n, what are the cameras P and P' for the two views?

(iii) **Scene geometry (structure):** Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras P, P', what is the position of (their pre-image) X in space?

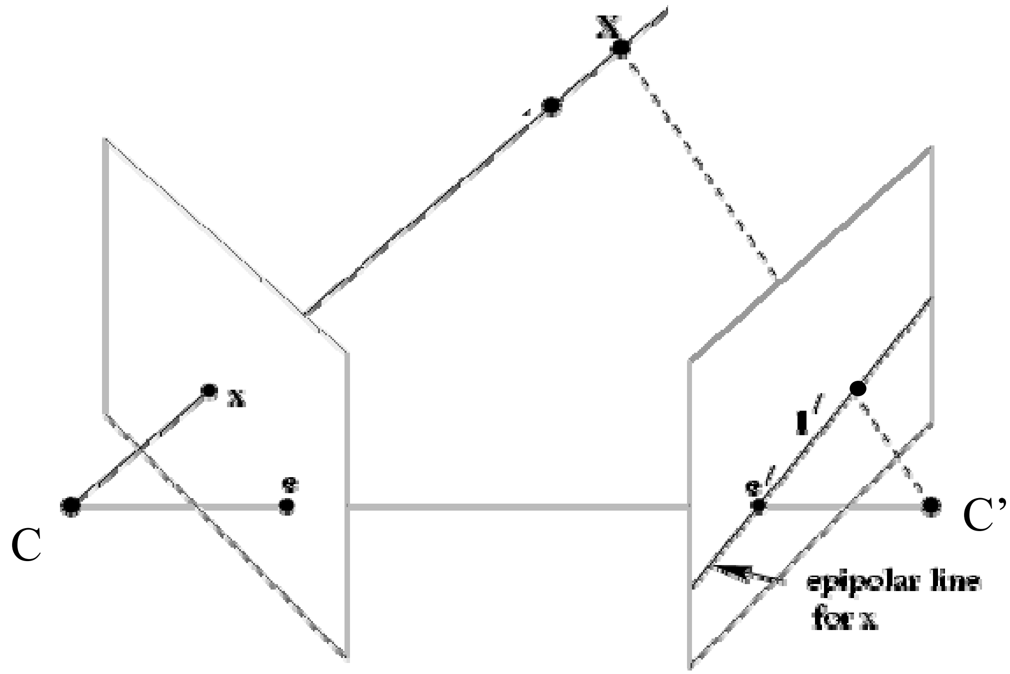
(ii) and (iii) form the *structure from motion* (SfM) problem

Various scenarios

- SfM using calibrated camera(s)
  - A moving camera
  - A camera network
- SfM using an affine camera
  - No need to calibrate the camera / partially calibrated e.g. rectangular pixels and known principal point
  - Camera translation can only be recovered partially
- SfM through a projective reconstruction
  - Obtain a projective reconstruction first
  - Upgrade the projective reconstruction to a metric reconstruction
- SfM with direct estimation of unknown camera parameters

Metric Motion and Structure Recovery Using Calibrated Cameras Through Essential Matrix

Recap: the fundamental matrix F and epipolar geometry



$$P = K[I|0], \quad P' = K'[R|t]$$

$$F = K'^{-T} [t]_\times R K^{-1}$$

$$x'^T F x = 0$$

$$x'^T K'^{-T} [t]_\times R K^{-1} x = 0$$

Consider the canonical image planes of both cameras, $x = Kx_0$, and $x' = K'x_0'$

$$x'^T K'^{-T} [t]_\times R K^{-1} x = x_0'^T [t]_\times R x_0 = 0$$

The essential matrix: Fundamental matrix for calibrated cameras

$$\underbrace{x'^T K'^{-T} [t]_\times R K^{-1}}_{\text{Fundamental matrix}} x = x_0'^T \underbrace{[t]_\times R}_{\text{Essential matrix}} x_0 = 0$$

$x=Kx_0$, and $x'=K'x'_0$

$$x'^T_0 E x_0 = 0 \qquad E = K'^T F K \qquad E = [t]_\times R = R[R^T t]_\times$$

E: 5 d.o.f. (3 for R; 2 for t up to scale)

> Theorem: E is an essential matrix if and only if its two singular values are equal and third is zero.

$$E = U \mathrm{diag}(1,1,0) V^T$$

Four possible reconstructions from E

Theorem: For a given essential matrix $\mathbf{E=U diag(1,1,0) V^T}$, and the first camera matrix $P=[I|0]$, there are four possible choices for the second camera matrix $P'=[R|t]$, namely:

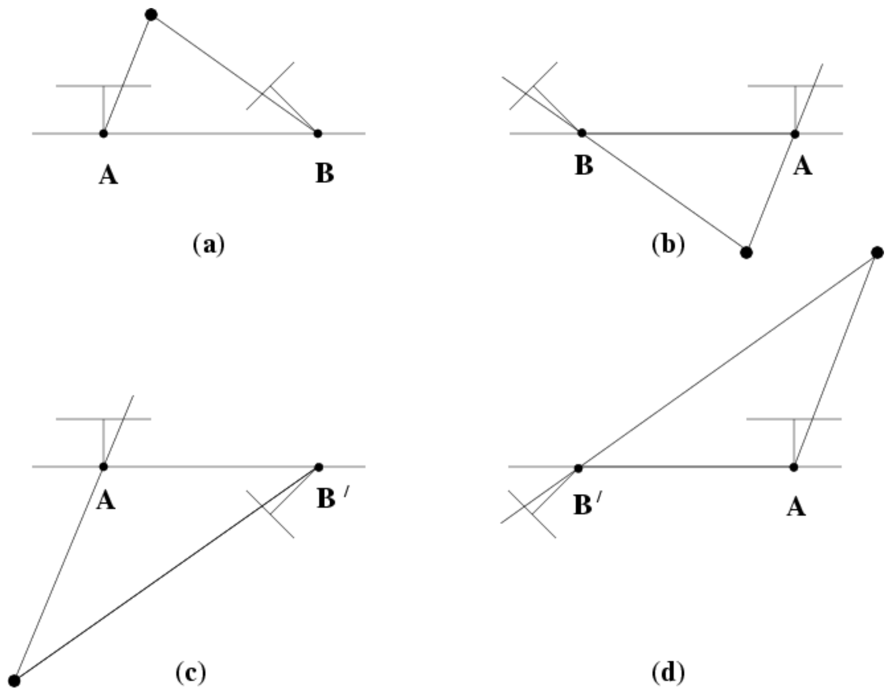
$P'=[UWV^T|+u_3]$, or

$[UWV^T|-u_3]$, or

$[UW^TV^T|+u_3]$, or

$[UW^TV^T|-u_3]$

where $W=\begin{bmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}$



(only one solution is valid where points are in front of both cameras)

Camera Motion Estimation

- **Camera translation**

$$C = -R^T t, \text{ since } t = -RC$$

C is the camera center position in the world system.

- **Recovery of rotation angles**

Given rotation angle, $\Psi = (\psi_x, \psi_y, \psi_z)$,

$$R(\Psi) = \begin{bmatrix} n_1^2 + (1-n_1^2)\eta & n_1 n_2 (1-\eta) + n_3\zeta & n_1 n_3 (1-\eta) - n_2\zeta \\ n_1 n_2 (1-\eta) - n_3\zeta & n_2^2 + (1-n_2^2)\eta & n_2 n_3 (1-\eta) + n_1\zeta \\ n_1 n_3 (1-\eta) + n_2\zeta & n_2 n_3 (1-\eta) - n_1\zeta & n_3^2 + (1-n_3^2)\eta \end{bmatrix}$$

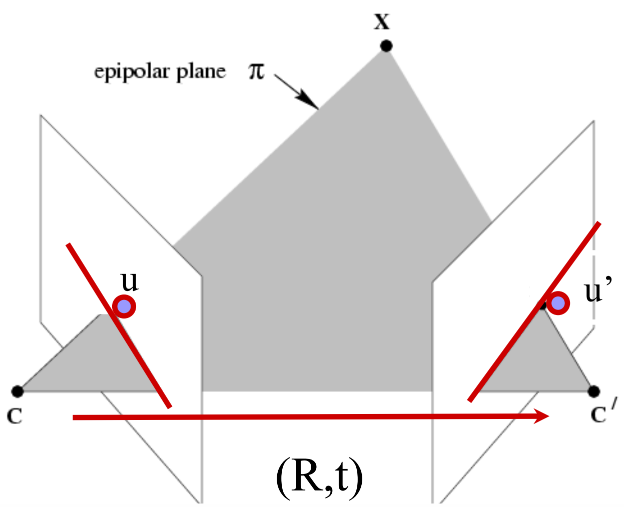
where $(n_1, n_2, n_3) = \Psi / |\Psi|$ and $\eta = \cos|\Psi|, \zeta = \sin|\Psi|$

Given, rotation matrix R(Ψ), Ψ can be computed as follows:

$$\Psi = \frac{\phi}{2 \sin \phi}(r_{23} - r_{32}, r_{31} - r_{13}, r_{12} - r_{21})^T,$$

where $\phi = \cos^{-1}(0.5\ \text{trace}(R(\Psi)) - 0.5))$

Structure Recovery Using Triangulation

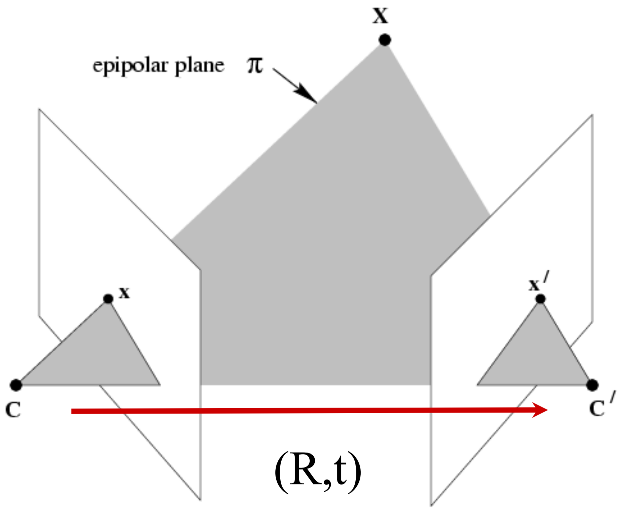


- Due to feature matching error, u and u' might not on the same epipolar plane.
- We need to find an epipolar plane such that the epipolar distances from u and u' to λ and λ' are minimized.
- Epipolar planes is a family of one parameter (a pencil).
- An epipolar plane can be specified by one parameter.

1. Parametrize the pencil of epipolar lines in the first image by a parameter t. Thus an epipolar line in the first image may be written as $\boldsymbol{\lambda}(t)$.
2. Using the fundamental matrix F, compute the corresponding epipolar line $\boldsymbol{\lambda}'(t)$ in the second image.
3. Express the distance function $d(\boldsymbol{u}, \boldsymbol{\lambda}(t))^2 + d(\boldsymbol{u}', \boldsymbol{\lambda}'(t))^2$ explicitly as a function of t.
4. Find the value of t that minimizes this function.

"In this way, the problem is reduced to that of finding the minimum of a function of a single variable, ***t***. It will be seen that for a suitable parameterization of the pencil of epipolar lines the distance function is a rational polynomial function of ***t***. Using techniques of elementary calculus, the minimization problem reduces to finding the real roots of a polynomial of degree 6." – R. Hartley and P. Sturm, "Triangulation", *Computer Vision and Image Understanding*, 68(2):146-157, 1997.

Structure Recovery Using Triangulation – a suboptimal method



Assume that P=K[I|0], P'=K'[R|t]. Given feature correspondences and 3D camera movement (R,t), the depth of features can be recovered using triangulation. Namely, we need to minimize the reprojection error:

$$z = \arg\min_z \left\| x' - K'[R \mid t] \begin{bmatrix} zK^{-1}x \\ 1 \end{bmatrix} \right\|^2$$

It can be shown that the solution to the above optimization is given by

$$z^* = \frac{(t_x r_z - t_z r_x)(t_z u - t_x) + (t_y r_z - t_z r_y)(t_z v - t_y)}{(t_x r_z - t_z r_x)(r_z u - r_x) + (t_y r_z - t_z r_y)(r_z v - r_y)}$$

where $(u,v,1)^T = K'^{-1}x'$, $(t_x,t_y,t_z)^T = -t$, $(r_x,r_y,r_z)^T = RK^{-1}x$

with structure recovery scaled with respect to the translational magnitude.

More than 2 views?
– Sequential structure from motion

- Initialize structure and motion from two views
- For each additional view
  - Determine pose
  - Refine and extend structure for new points

- Determine correspondences robustly by jointly estimating matches and epipolar geometry

Limitations of the method

- Needs camera calibration
- Need large baseline compared to the scene structure

Metric Motion and Structure Recovery Using Calibrated Cameras Through Non-linear Filtering

A.Azarbayejani and A. Pentland, ‘‘Recursive estimation of motion, structure, and focal length,’’
B.IEEE Trans. Pattern Anal. Mach. Intell. **17**, 562–575 (1995).
G. Qian and R. Chellappa, "Structure From Motion Using Sequential Monte Carlo Methods,"
International Journal of Computer Vision, vol. 59, pp 5-31, August 2004

Challenges

- Nonlinear dynamic system
- Observation noises, large baseline vs. small baseline
- Feature occlusion
- Inherent ambiguities

- Solutions: non-linear filtering
  - Extended Kalman filter
  - Sequential Monte Carlo (Particle filter)

Sequential Monte Carlo

- A set of simulation-based methods for sequentially computing the posterior densities
- Appeared as a number of variants
  - Condensation
  - Particle filters
  - Bootstrap filters
  - Monte Carlo filters
  - Sequential importance sampling (SIS)

Dynamic Systems

- Assumptions

$$\begin{cases} x_t = Q_t(x_{t-1}) + v \\ y_t = F_t(x_t) + n \end{cases}, \quad \begin{cases} x_t \sim q_t(x_t \mid x_{t-1}) \\ y_t \sim f_t(y_t \mid x_t) \end{cases}, \quad p(x_0)$$

- State and observation sequences

$$x_{0:t} = (x_0, x_1, \cdots, x_t)$$

$$y_{1:t} = (y_1, y_2, \cdots, y_t)$$

- Goal: Recursively approximate the *a posteriori* density $\pi(x_{0:t}) = p(x_{0:t} \mid y_{0:t})$ and the expectation of any integrable function f_t

$$I(f_t) = \int p(x_{0:t} \mid y_{1:t}) f_t(x_{0:t}) dx_{0:t}$$

Dynamic Systems (cont'd)

- When $\pi(x_{0:t})$ is high-dimensional, Monte Carlo integration is needed to evaluate $I(f_t)$.

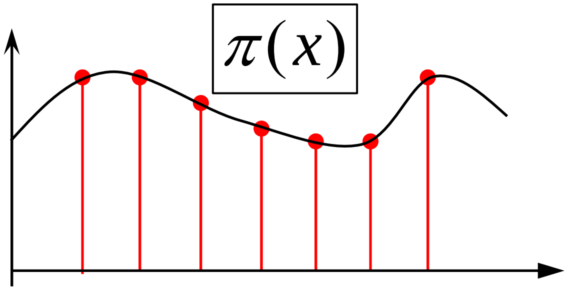
$$I(f_t) = \int p(x_{0:t} \mid y_{1:t}) f_t(x_{0:t}) dx_{0:t}$$

- SLLN:

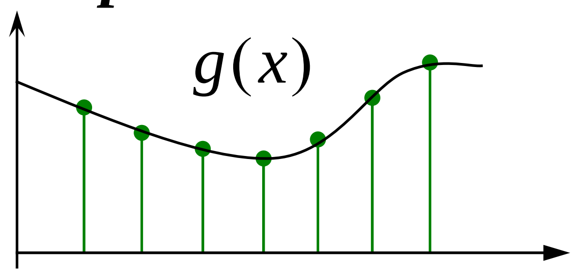
$$\frac{1}{N}\sum_{j=1}^{N} f(x_{0:t}^{(j)}) \underset{N\to\infty}{\overset{a.s.}{\longrightarrow}} I(f_t)$$

Importance Sampling

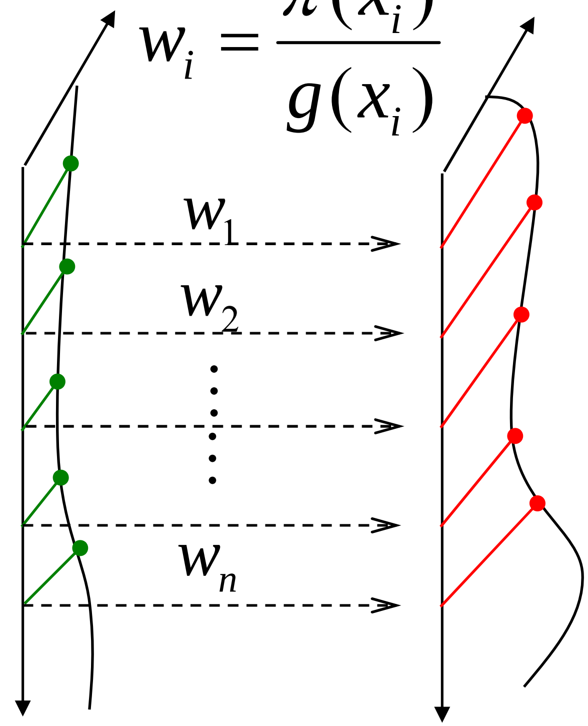
Target distribution



Proposal distribution



$$w_i = \frac{\pi(x_i)}{g(x_i)}$$



$$\lim_{N\to\infty} \frac{\sum_{i=1}^{N} f(x_i) w_i}{\sum_{i=1}^{N} w_i} = \int f(x)\pi(x)dx$$

x_i's are properly weighted by w_i's w.r.t. $\pi(x)$.

Sequential Importance Sampling with Resampling

$$\left\{x_{0:t}^{(j)}, w_t^{(j)}\right\} \sim \pi_t(x_{0:t}) \xrightarrow{SISR} \left\{x_{0:t+1}^{(j)}, w_{t+1}^{(j)}\right\} \sim \pi_{t+1}(x_{0:t+1})$$

SISR steps: for $j=1,2,\ldots,m$

(A) Draw $x_{t+1}^{(j)}$ from $g_{t+1}(x_{t+1} \mid x_{0:t}^{(j)})$. Attach $x_{t+1}^{(j)}$ to form $x_{0:t+1}^{(j)} = \left(x_{0:t}^{(j)}, x_{t+1}^{(j)}\right)$

(B) Compute the incremental weight u_{t+1} by

$$u_{t+1}^{(j)} = \frac{\pi_{t+1}\left(x_{0:t+1}^{(j)}\right)}{\pi_t\left(x_{0:t}^{(j)}\right) g_{t+1}\left(x_{t+1}^{(j)} \middle| x_{0:t}^{(j)}\right)}$$

and update weight via

$$w_{t+1}^{(j)} = u_{t+1}^{(j)} w_t^{(j)}$$

(C) Redraw samples according to their weights

SISR For Dynamic Systems

- Chain rule for target density

$$\pi_t(x_{0:t}) \propto p(x_{0:t}, y_{1:t}) = \pi(x_0) \cdot \prod_{k=1}^{t} q(x_k | x_{k-1}) f(y_k | x_k)$$

$$\pi_{t+1}(x_{0:t+1}) \propto \pi_t(x_{0:t}) q(x_{t+1} | x_t) f(y_{t+1} | x_{t+1})$$

- Let the instantaneous proposal density be the state transition density.

$$g_{t+1}(x_t | x_{0:t}) = q(x_{t+1} | x_t)$$

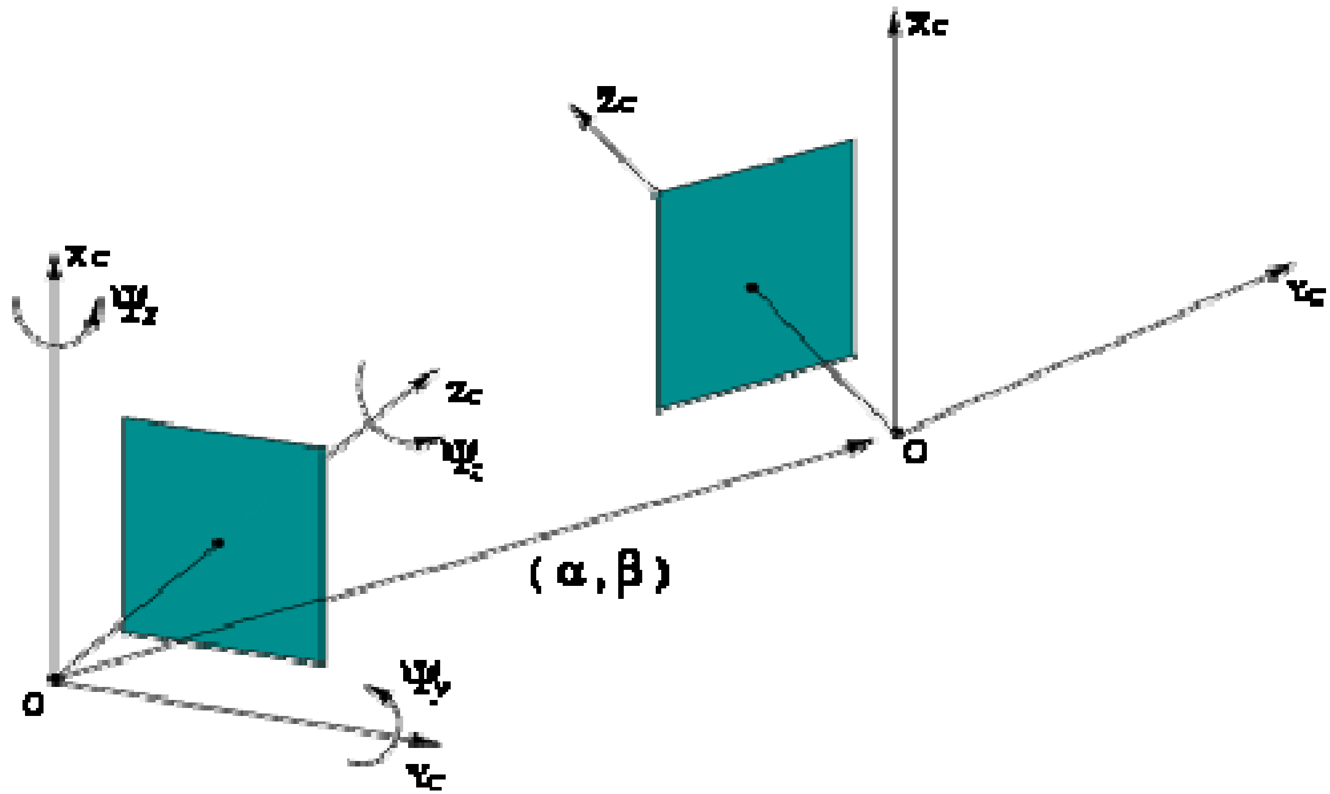
- The incremental weight equals the likelihood function.

$$u_{t+1} = \frac{\pi_{t+1}(x_{0:t+1})}{\pi_t(x_{0:t}) g_{t+1}(x_{t+1} | x_{0:t})}$$

$$\propto \frac{\pi_t(x_{0:t}) q(x_{t+1} | x_t) f(y_{t+1} | x_{t+1})}{q(x_{t+1} | x_t)}$$

$$= f(y_{t+1} | x_{t+1})$$

3-D Motion Parameterization



$$m_t = (\Psi_x, \Psi_y, \Psi_z, \alpha, \beta)$$

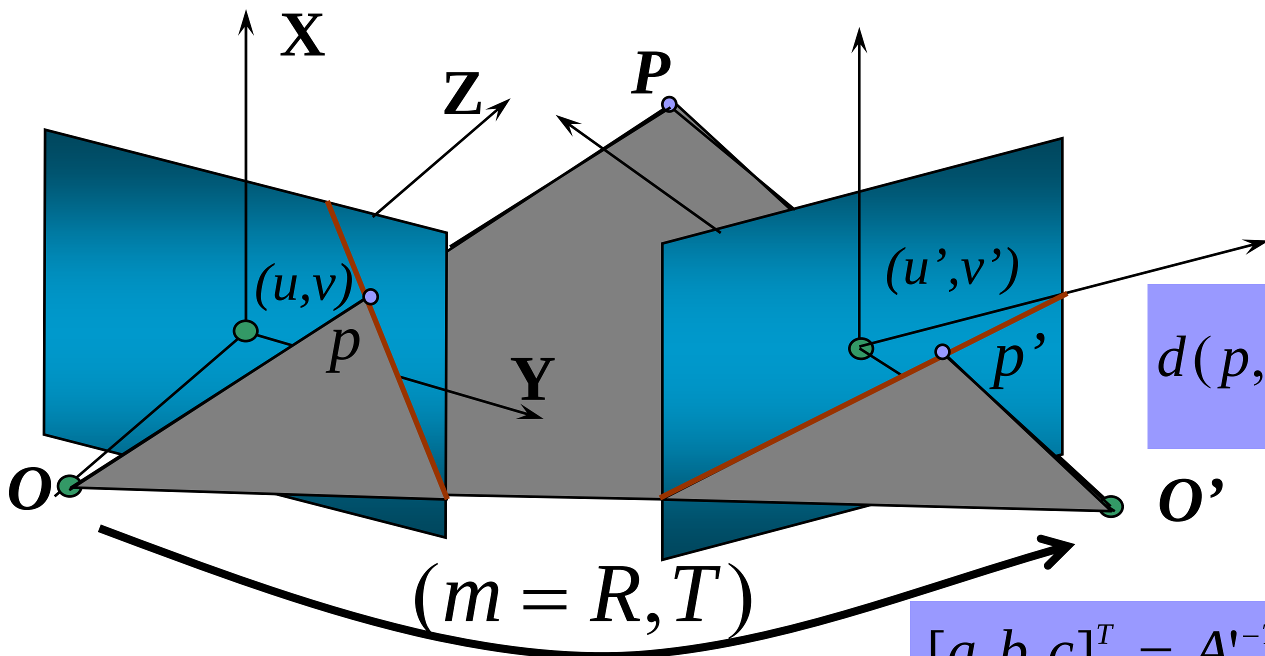
$$T(\alpha, \beta) = [\sin(\alpha)\cos(\beta), \sin(\alpha)\sin(\beta), \cos(\alpha)]$$

Implementation Issues

- Instantaneous proposal distribution

$$g_{t+1}(x_t | x_{0:t}) = q(x_{t+1} | x_t) \quad \Longrightarrow \quad x_{t+1} = x_t + n_x$$

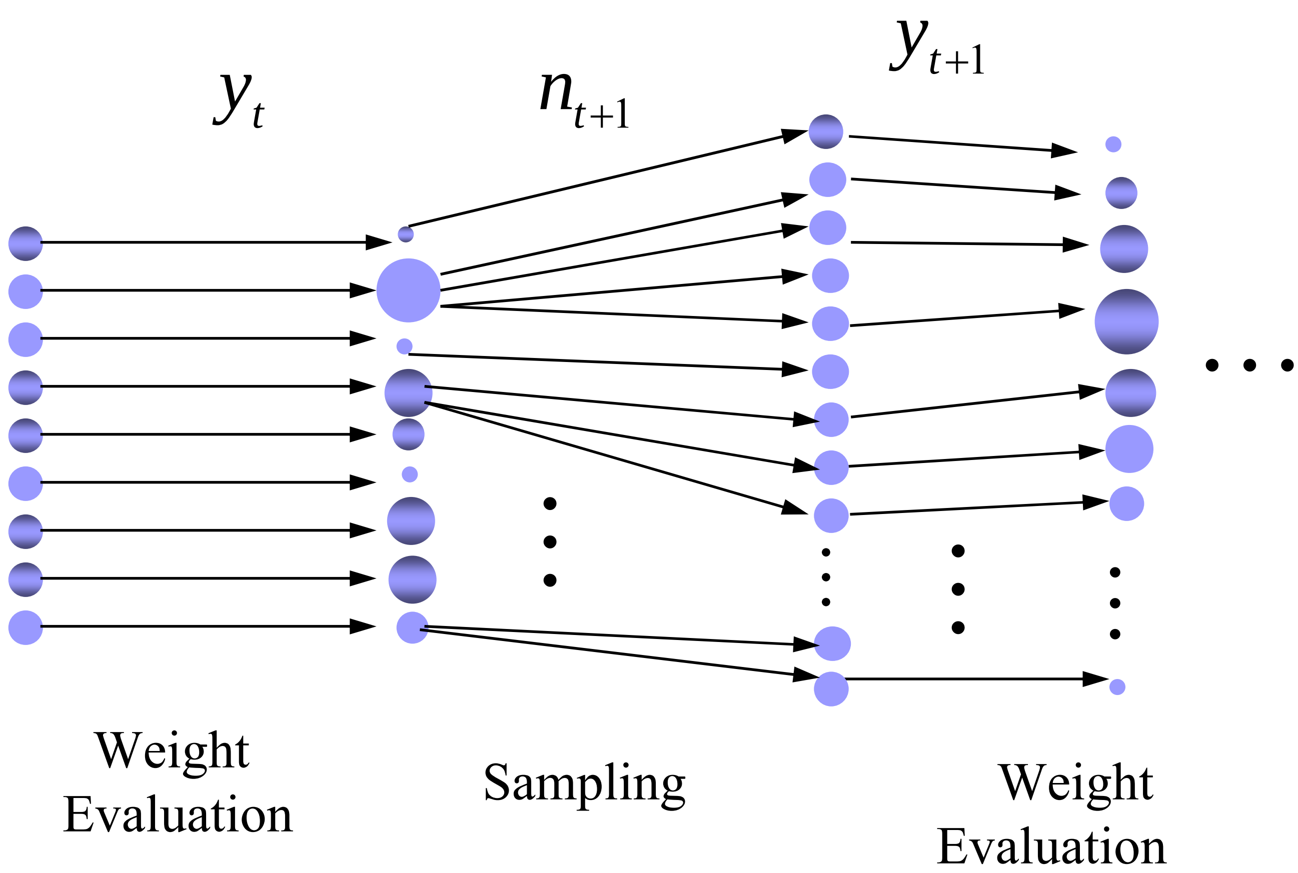
- Sample weight evaluation via the epiploar distances



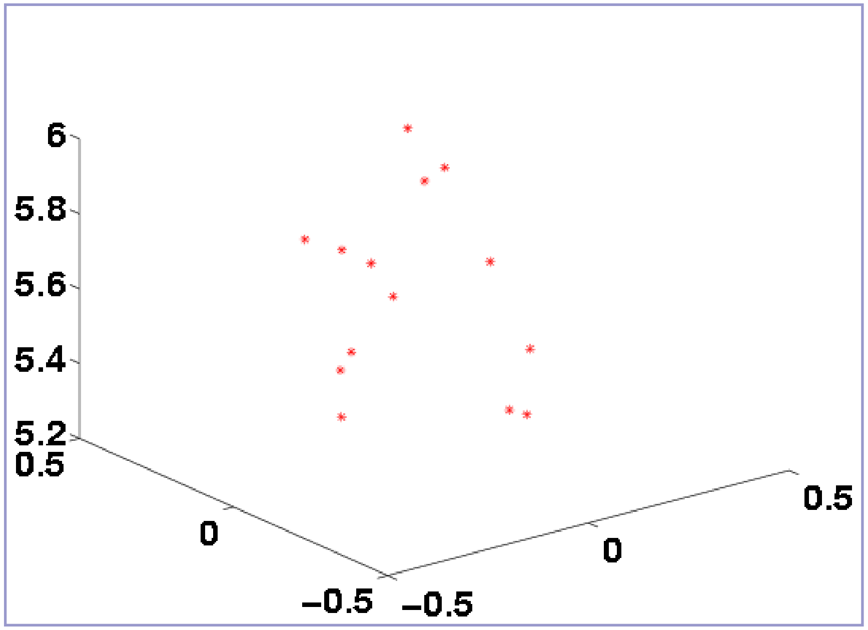
$$d(p,p',m) = \frac{au'+bv'+c}{\sqrt{a^2+b^2}}$$

$$[a,b,c]^T = A'^{-T}[-RT]_\times RA^{-1}[u,v,1]^T$$

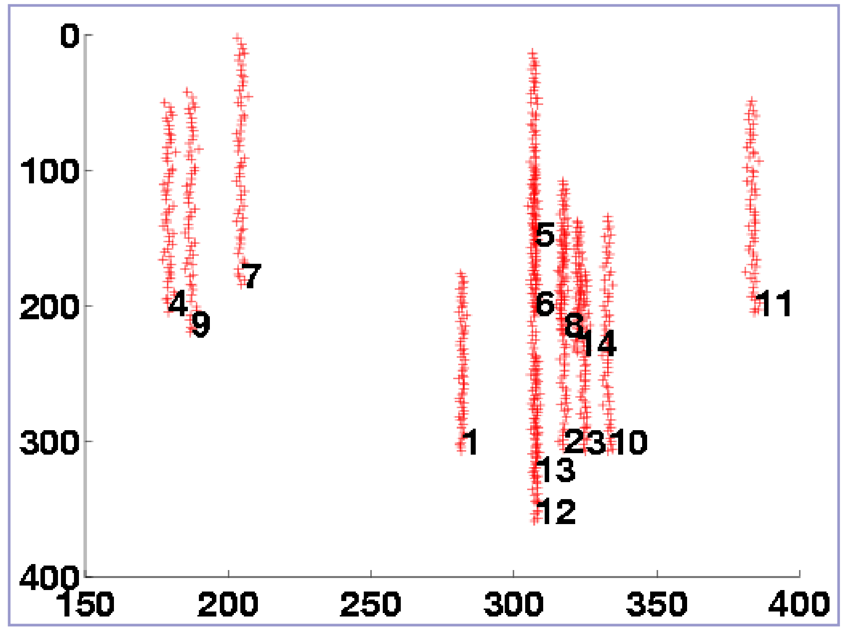
SMC for 3-D Motion Estimation



Ambiguous Sequences

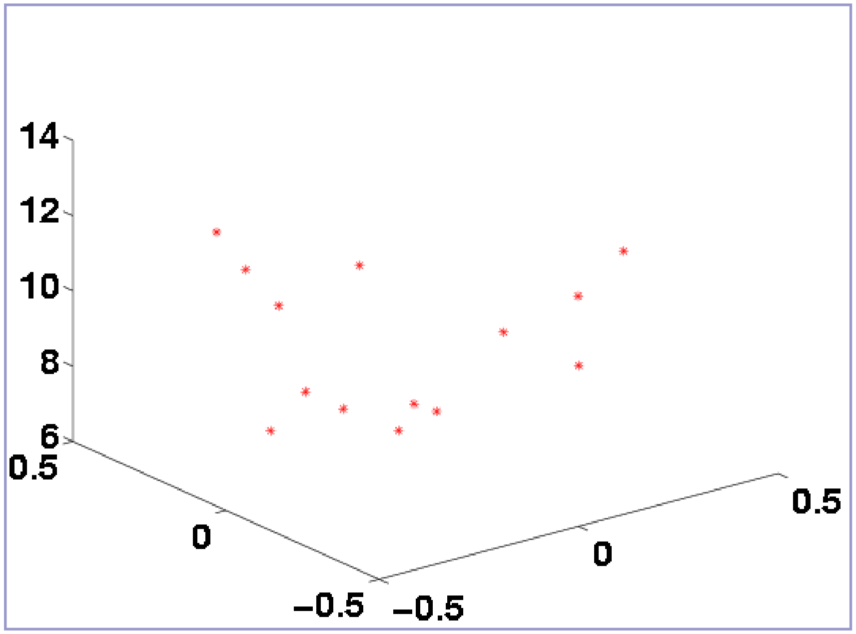


3-D points

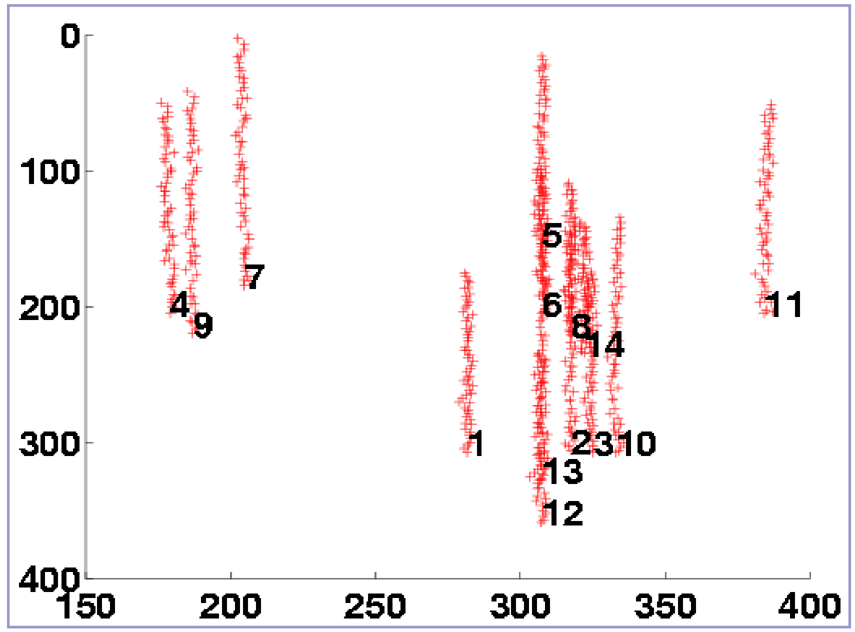


2-D trajectories

Ambiguous Sequences



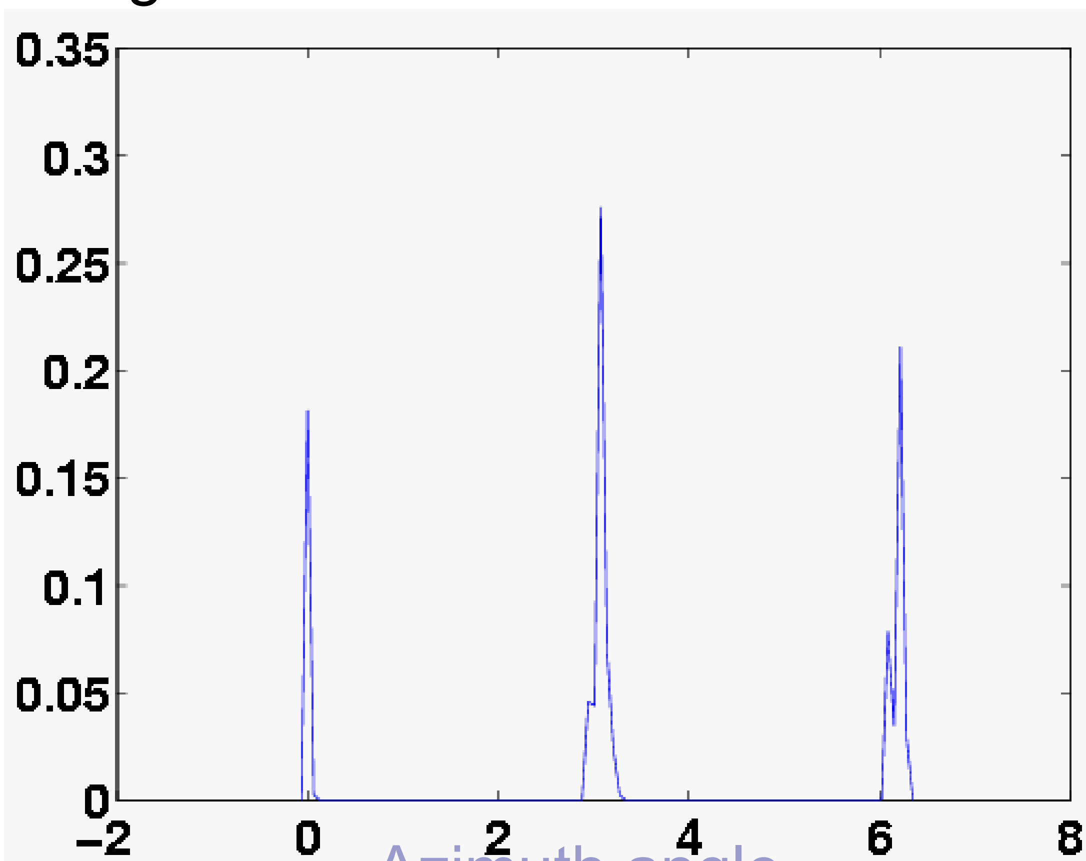
False shape solution



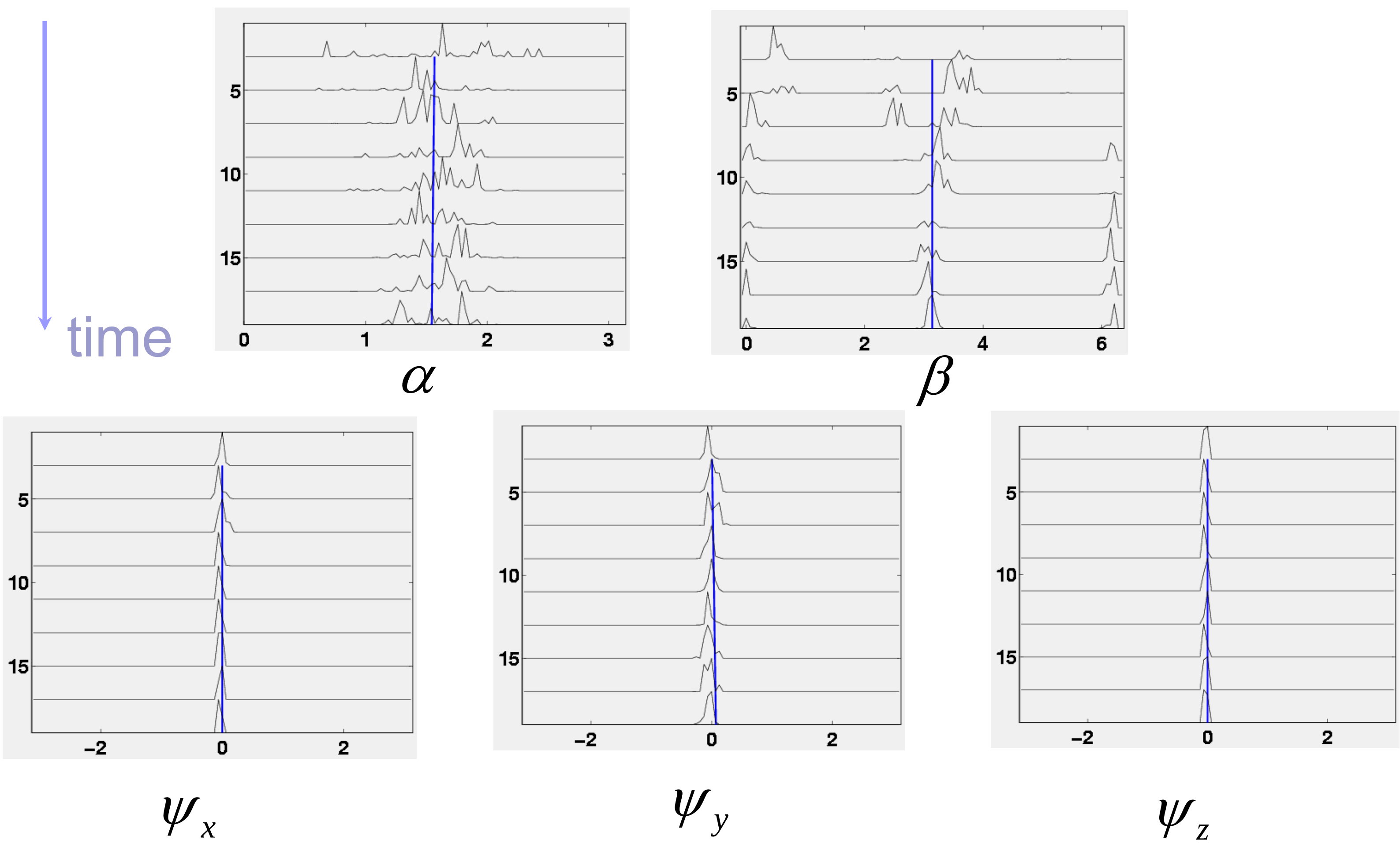
Reprojected trajectories using the false solution

Ambiguous Sequences

- Multi-mode pattern in the posterior distribution of the azimuth angle

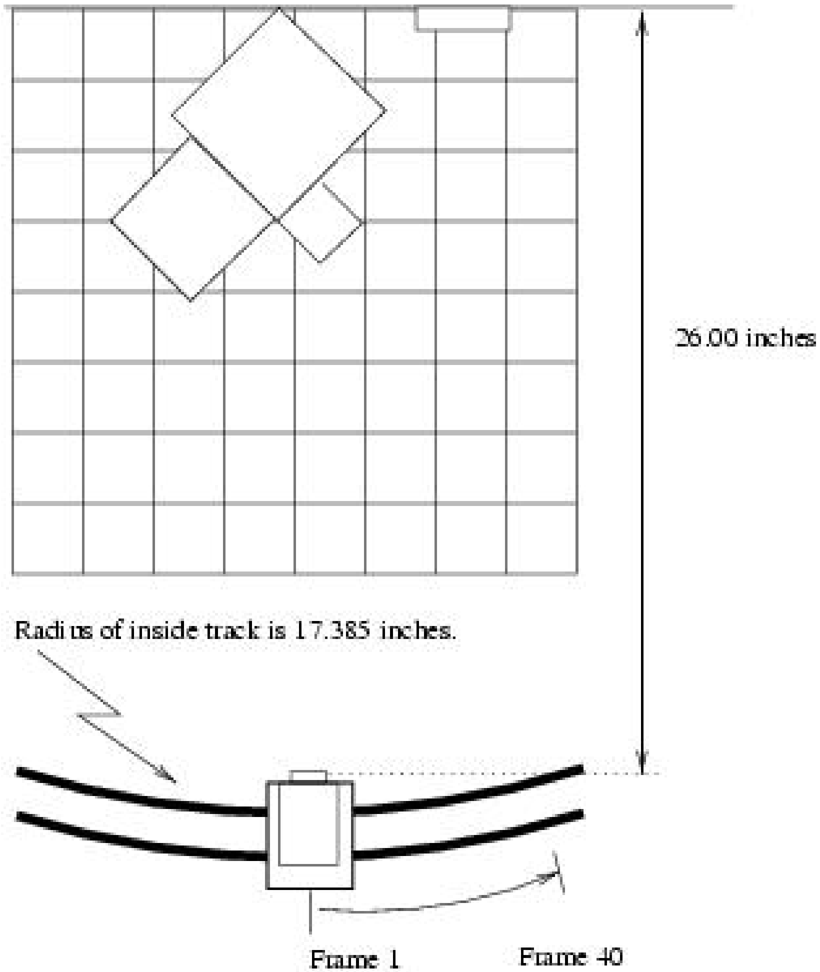


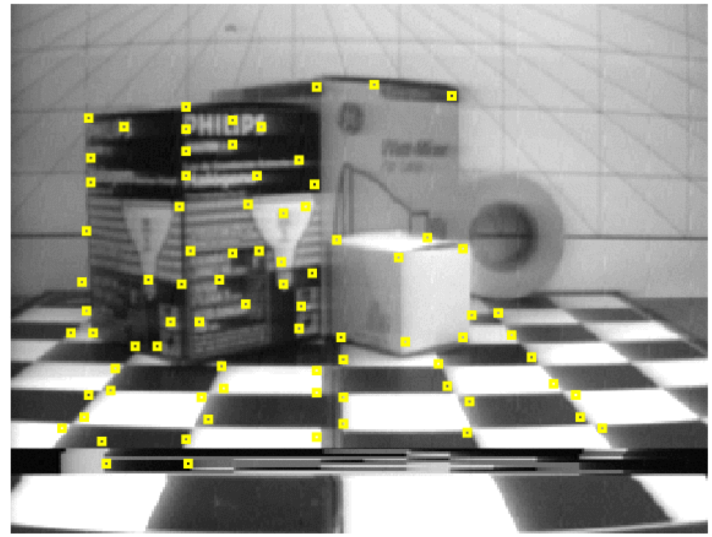
Ambiguous Sequences

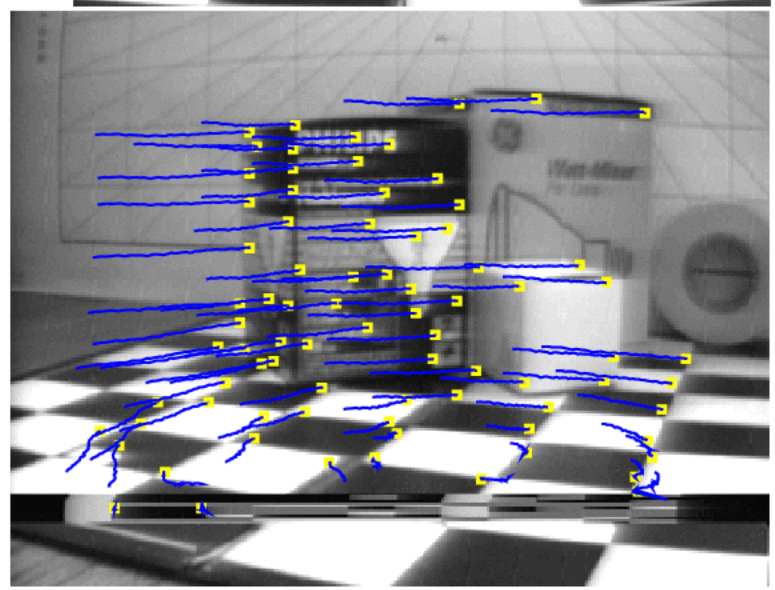


Experiment I: Indoor Sequence

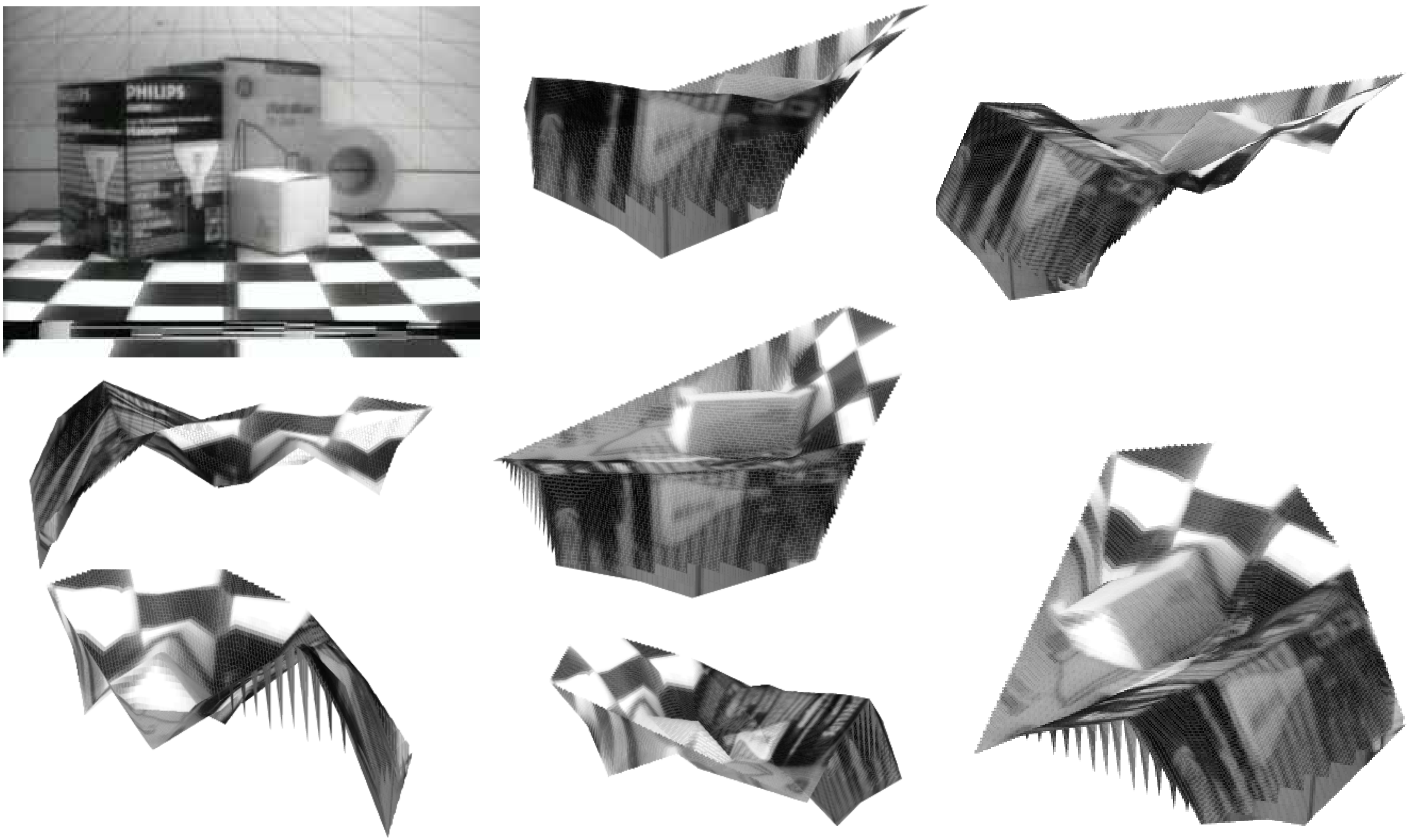
Experiment configuration







3D Model Building

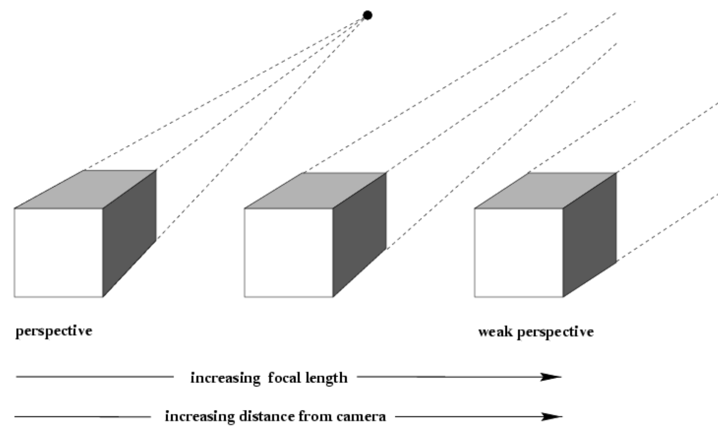


Metric Reconstruction via Factorization for Orthographic Projection

Reference: Tomasi, C., and Kanade, T., (1992) **Shape and Motion from Image Streams under Orthography: A Factorization Method,** IJCV(9), No. 2, November 1992, pp. 137-154.

Recap: Affine cameras

- Points at infinity in the 3D space are imaged to lines at infinity on the image.
- Last row of the projection matrix is (0,0,0,1).
- A valid assumption when
  - 3D points are close to a plane perpendicular to the camera looking direction, (i.e. scene structure variation is smaller compared to the distance of the camera) and
  - Projections of points are close to the principal point of the image.



Recap: Affine cameras

• Obtained by shifting the camera center along opposite camera looking direction and enlarging the focal length to keep the image size approximately unchanged.

$$d_0 = -\mathrm{r}^{3T}\tilde{\mathrm{C}}$$

$$\mathrm{P}_t = \mathrm{K}\begin{bmatrix} \mathrm{r}^{1T} & -\mathrm{r}^{1T}\left(\tilde{\mathrm{C}} - t\mathrm{r}^3\right) \\ \mathrm{r}^{2T} & -\mathrm{r}^{2T}\left(\tilde{\mathrm{C}} - t\mathrm{r}^3\right) \\ \mathrm{r}^{3T} & -\mathrm{r}^{3T}\left(\tilde{\mathrm{C}} - t\mathrm{r}^3\right) \end{bmatrix} = \mathrm{K}\begin{bmatrix} \mathrm{r}^{1T} & -\mathrm{r}^{1T}\tilde{\mathrm{C}} \\ \mathrm{r}^{2T} & -\mathrm{r}^{2T}\tilde{\mathrm{C}} \\ \mathrm{r}^{3T} & d_t \end{bmatrix}$$

$$= \mathrm{K}\begin{bmatrix} d_t/d_0 & & \\ & d_t/d_0 & \\ & & 1 \end{bmatrix}\begin{bmatrix} \mathrm{r}^{1T} & -\mathrm{r}^{1T}\tilde{\mathrm{C}} \\ \mathrm{r}^{2T} & -\mathrm{r}^{2T}\tilde{\mathrm{C}} \\ \mathrm{r}^{3T} & d_t \end{bmatrix}$$

$$= \frac{d_t}{d_0}\mathrm{K}\begin{bmatrix} \mathrm{r}^{1T} & -\mathrm{r}^{1T}\tilde{\mathrm{C}} \\ \mathrm{r}^{2T} & -\mathrm{r}^{2T}\tilde{\mathrm{C}} \\ \mathrm{r}^{3T} d_0/d_t & d_0 \end{bmatrix}$$

$$\mathrm{P}_\infty = \lim_{t\to\infty} \mathrm{P}_t = \mathrm{K}\begin{bmatrix} \mathrm{r}^{1T} & -\mathrm{r}^{1T}\tilde{\mathrm{C}} \\ \mathrm{r}^{2T} & -\mathrm{r}^{2T}\tilde{\mathrm{C}} \\ 0 & d_0 \end{bmatrix}$$

$$= \mathrm{K}\begin{bmatrix} 1 & & \\ & 1 & \\ & & d_0 \end{bmatrix}\begin{bmatrix} \mathrm{r}^{1T} & -\mathrm{r}^{1T}\tilde{\mathrm{C}} \\ \mathrm{r}^{2T} & -\mathrm{r}^{2T}\tilde{\mathrm{C}} \\ 0 & 1 \end{bmatrix}$$

$-r^{1T}\tilde{C}$ and $-r^{2T}\tilde{C}$ are first two coordinates of the world center in the camera frame. The third coordinate is infinity.

Recap: Decomposition of P_∞

$$d_0 = -r^{3T}\tilde{C} \qquad P_\infty = K\begin{bmatrix} 1 & & \\ & 1 & \\ & & d_0 \end{bmatrix}\begin{bmatrix} r^{1T} & -r^{1T}\tilde{C} \\ r^{2T} & -r^{2T}\tilde{C} \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} K_{2\times2} & \tilde{x}_0 \\ & 1 \end{bmatrix}\begin{bmatrix} I_{2\times2} & 0 \\ & d_0 \end{bmatrix}\begin{bmatrix} \tilde{R} & \tilde{t} \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} d_0^{-1}K_{2x2} & \tilde{x}_0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{R} & \tilde{t} \\ 0 & 1 \end{bmatrix}$$

$K'_{2x2} = d_0^{-1} K_{2x2}$

$$= \begin{bmatrix} K'_{2x2} & \tilde{x}_0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{R} & \tilde{t} \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} K'_{2x2}\tilde{R} & K'_{2x2}\tilde{t} + \tilde{x}_0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} K'_{2x2} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{R} & \tilde{t} + K^{-1}_{2x2}\tilde{x}_0 \\ 0 & 1 \end{bmatrix} \Rightarrow P_\infty = \begin{bmatrix} K_{2x2} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{R} & \tilde{t} \\ 0 & 1 \end{bmatrix}_{3\times4}$$

Recap: hierarchy of affine cameras

Projection along Z-axis:
Camera frame is parallel to the world frame. Camera optical axis is aligned with the Z-axis of the world frame. It is a projection along the Z-axis. K = I

$$P_\infty = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Orthographic projection:
Camera goes through a Euclidean transformation H and K = I.

$$H = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad P_\infty = \begin{bmatrix} r^{1T} & t_1 \\ r^{2T} & t_2 \\ 0 & 1 \end{bmatrix}$$

(5dof)

Scaled orthographic projection:
Camera goes through a Euclidean transformation H and K is diagonal, i.e. the skew factor of the camera is zero and the principal point on the image is known.

$$H = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

$$P_\infty = \begin{bmatrix} k & & \\ & k & \\ & & 1 \end{bmatrix} \begin{bmatrix} r^{1T} & t_1 \\ r^{2T} & t_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r^{1T} & t_1 \\ r^{2T} & t_2 \\ 0 & 1/k \end{bmatrix}$$

(6dof)

Some Insights

- The projection of a 3D point using affine camera is determined by the first two coordinates of the point in the camera frame.

$$
P_\infty \begin{bmatrix} \tilde{X} \\ 1 \end{bmatrix} = \begin{bmatrix} k & & \\ & k & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathrm{r}^{1T} & t_1 \\ \mathrm{r}^{2T} & t_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{X} \\ 1 \end{bmatrix}
$$

$$
= \begin{bmatrix} k & & \\ & k & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathrm{r}^{1T} & -\mathrm{r}^{1T}\tilde{C} \\ \mathrm{r}^{2T} & -\mathrm{r}^{2T}\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{X} \\ 1 \end{bmatrix}
$$

$$
= \begin{bmatrix} k & & \\ & k & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathrm{r}^{1T}(\tilde{X} - \tilde{C}) \\ \mathrm{r}^{2T}(\tilde{X} - \tilde{C}) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{r}^{1T}(\tilde{X} - \tilde{C}) \\ \mathrm{r}^{2T}(\tilde{X} - \tilde{C}) \\ 1/k \end{bmatrix}
$$

- Moving a camera along the optical axis of the camera won't produce any changes in the image.

SfM from an affine camera: projection model

Suppose that given trajectories of P feature points over F frames from a moving affine camera

$$(u_{fp}, v_{fp}), f = 1, \cdots, F, p = 1, \cdots, P,$$

we need to recover the 3D camera motion and scene structure. Recall that the projection matrix of a scaled orthographic camera is given by

$$\mathrm{P}_\infty = \begin{bmatrix} k & & \\ & k & \\ & & 1 \end{bmatrix} \begin{bmatrix} r^{1T} & -r^{1T}\widetilde{C} \\ r^{2T} & -r^{2T}\widetilde{C} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r^{1T} & -r^{1T}\widetilde{C} \\ r^{2T} & -r^{2T}\widetilde{C} \\ 0 & 1/k \end{bmatrix} = \begin{bmatrix} P^{1T} \\ P^{2T} \\ P^{3T} \end{bmatrix},$$

$$\begin{bmatrix} u_{fp} \\ v_{fp} \\ 1/k \end{bmatrix} = \begin{bmatrix} P_f^{1T} \\ P_f^{2T} \\ P_f^{3T} \end{bmatrix} \begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix} = \begin{bmatrix} P_f^{1T} \\ P_f^{2T} \end{bmatrix} \begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} = \begin{bmatrix} r_f^{1T}(\widetilde{X}_p - \widetilde{C}_f) \\ r_f^{2T}(\widetilde{X}_p - \widetilde{C}_f) \end{bmatrix}$$

Remark: In this step, the camera motion and scene structure at represented in an arbitrary world coordinate frame.

Projection of the object center

- The center of the features in an image is the projection of the center of the 3D point cloud.

$$\begin{bmatrix} \bar{u}_f \\ \bar{v}_f \end{bmatrix} = \frac{1}{P}\sum_{p=1}^{P}\begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix} = \frac{1}{P}\sum_{p=1}^{P}\begin{bmatrix} P_f^{1T} \\ P_f^{2T} \end{bmatrix}\begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} P_f^{1T} \\ P_f^{2T} \end{bmatrix}\frac{1}{P}\sum_{p=1}^{P}\begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} = \begin{bmatrix} P_f^{1T} \\ P_f^{2T} \end{bmatrix}\begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} r_f^{1T}(\bar{\tilde{X}} - \tilde{C}_f) \\ r_f^{2T}(\bar{\tilde{X}} - \tilde{C}_f) \end{bmatrix}$$

A new world coordinate frame

- Center the feature points in each image frame so that the image center is the center of the feature points.

$$\begin{bmatrix}\tilde{u}_{fp}\\ \tilde{v}_{fp}\end{bmatrix}=\begin{bmatrix}u_{fp}\\ v_{fp}\end{bmatrix}-\begin{bmatrix}\bar{u}_f\\ \bar{v}_f\end{bmatrix}=\begin{bmatrix}P_f^{1T}\\ P_f^{2T}\end{bmatrix}\left(\begin{bmatrix}X_p\\ Y_p\\ Z_p\\ 1\end{bmatrix}-\begin{bmatrix}\bar{X}\\ \bar{Y}\\ \bar{Z}\\ 1\end{bmatrix}\right)$$

$$=\begin{bmatrix}r_f^{1T} & -r_f^{1T}\tilde{C}_f\\ r_f^{2T} & -r_f^{2T}\tilde{C}_f\end{bmatrix}\begin{bmatrix}X_p-\bar{X}\\ Y_p-\bar{Y}\\ Z_p-\bar{Z}\\ 0\end{bmatrix}$$

$$=\begin{bmatrix}r_f^{1T} & 0\\ r_f^{2T} & 0\end{bmatrix}\begin{bmatrix}X_p-\bar{X}\\ Y_p-\bar{Y}\\ Z_p-\bar{Z}\\ 1\end{bmatrix}$$

The equivalent camera center in this projection matrix is always perpendicular to the first two axes of the camera frame, this means that the camera is always looking at the origin of the world coordinate frame.

- The resulting projections are essentially represented in a coordinate system centered at the centered of the 3D point cloud.

$$s_p=\begin{bmatrix}X_p-\bar{X}\\ Y_p-\bar{Y}\\ Z_p-\bar{Z}\end{bmatrix}$$

- The camera is always looking at the world center (also object center), e.g. a rotating object about its center.

$$\begin{bmatrix}\tilde{u}_{fp}\\ \tilde{v}_{fp}\end{bmatrix}=\begin{bmatrix}r_f^{1T}\\ r_f^{2T}\end{bmatrix}\begin{bmatrix}X_p-\bar{X}\\ Y_p-\bar{Y}\\ Z_p-\bar{Z}\end{bmatrix}=\begin{bmatrix}r_f^{1T}\\ r_f^{2T}\end{bmatrix}s_p$$

Insights

- After the step of centering the features in all images, we essentially repositioned the cameras (without rotating the cameras) perpendicularly to the camera looking directions so that after the repositioning in each image frame the camera's principal point is aligned with the projection of the world center, which is also the object center in this case.

Low-rank factorization

$$\begin{bmatrix} \tilde{u}_{fp} \\ \tilde{v}_{fp} \end{bmatrix} = \begin{bmatrix} r_f^{1T} \\ r_f^{2T} \end{bmatrix} s_p$$

Stack the registered measurement to create

$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ \tilde{V} \end{bmatrix}_{2F\times P}, \tilde{U}_{F\times P}, \tilde{V}_{F\times P},$$

$$\tilde{U}(f,p) = \tilde{u}_{fp}, \tilde{V}(f,p) = \tilde{v}_{fp}$$

$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ \tilde{V} \end{bmatrix} = \begin{bmatrix} r_1^{1T} \\ \vdots \\ r_F^{1T} \\ r_1^{2T} \\ \vdots \\ r_F^{2T} \end{bmatrix} \begin{bmatrix} s_1 & s_2 & \cdots & s_P \end{bmatrix} = \tilde{R}S$$

Recall: rank(AB) ≤min(rankA, rank B) ⇒
Rank Theorem: Without observation noise, the registered measure matrix $\tilde{W}$ is at most rank three (simply because the *motion matrix* $\tilde{R}$ and the *structure matrix S* are rank three).

With noise, the rank of the registered measurement matrix is not exactly zero, But this problem can be solved by keeping only components related to the first three largest singular value through singular value decomposition (SVD).

Rank-3 factorization

- Factorize **the registered measurement matrix** through singular value decomposition

$$\widetilde{W} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}} = \widetilde{R}S$$

- An **affine reconstruction** is obtained by keeping space spanned by the singular vectors corresponding to the first three largest singular values.

$$\hat{R} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix}^{1/2},$$

$$\hat{S} = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix}^{1/2} \begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \end{bmatrix}$$

Why affine reconstruction ?

Metric Upgrade

A metric reconstruction can be obtained as follows by finding Q

$$\tilde{R} = \hat{R}Q, S = Q^{-1}\hat{S}$$

Where Q is computed from according to the following constraint for a rotation matrix

$$r_f^{1T} r_f^1 = 1, \quad r_f^{2T} r_f^2 = 1, \quad r_f^{1T} r_f^2 = 0$$

$$\Downarrow$$

$$\hat{r}_f^{1T} QQ^T \hat{r}_f^1 = 1, \quad \hat{r}_f^{2T} QQ^T \hat{r}_f^2 = 1, \quad \hat{r}_f^{1T} QQ^T \hat{r}_f^2 = 0$$

- Three linear equations per view on symmetric matrix $C=QQ^T$ (6DOF)

- Q can be then obtained from C through Cholesky factorization.

- Select the world system aligned with the first camera frame, i.e.

$r_1^{1T}=[1,0,0]$ and $r_1^{2T}=[0,1,0]$.

Recovery of Camera Translation

- Camera translation can only be recovered partially.

Raw image feature points

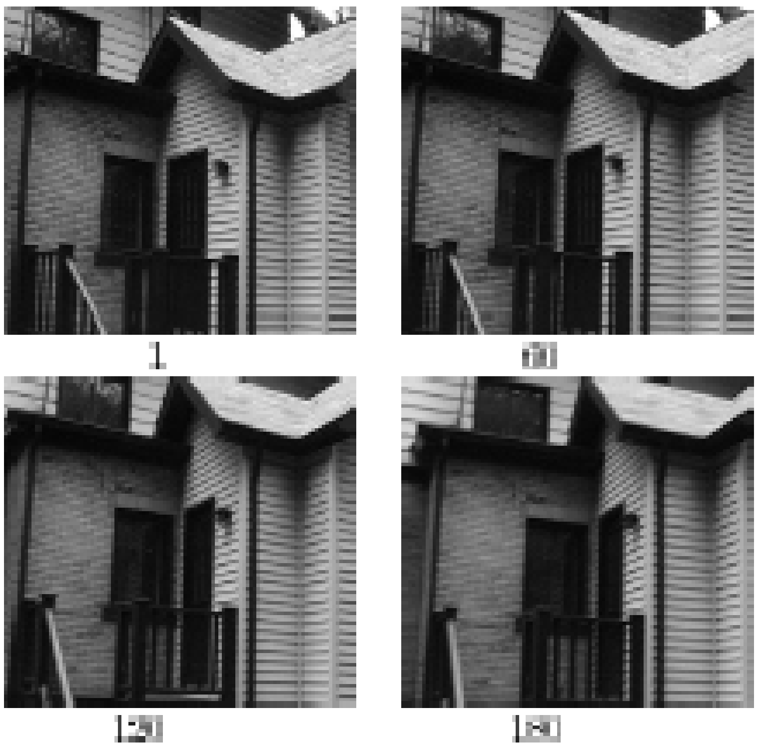
$$\begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix} = \begin{bmatrix} \tilde{u}_{fp} \\ \tilde{v}_{fp} \end{bmatrix} + \begin{bmatrix} \bar{u}_f \\ \bar{v}_f \end{bmatrix} = \begin{bmatrix} r_f^{1T} \\ r_f^{2T} \end{bmatrix} s_p + \begin{bmatrix} \bar{u}_f \\ \bar{v}_f \end{bmatrix}$$

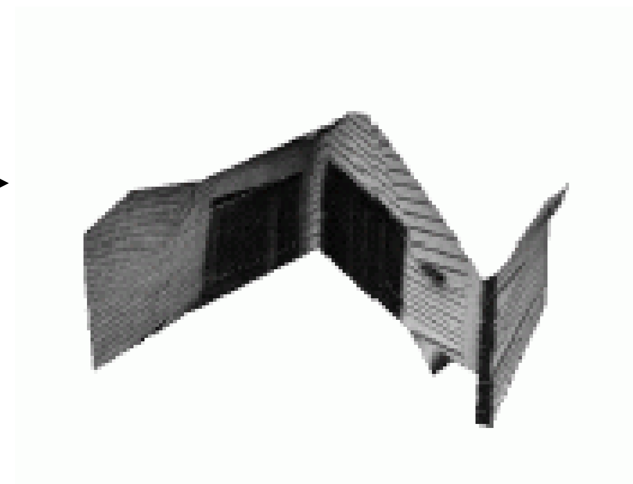
Projection matrix for the original raw feature points in the world coordinate with origin at the object center.

$$= \begin{bmatrix} r^{1T} & \bar{u}_f \\ r^{2T} & \bar{v}_f \end{bmatrix} \begin{bmatrix} s_p \\ 1 \end{bmatrix} = \begin{bmatrix} r^{1T} & -r_f^{1T}\tilde{C}_f \\ r^{2T} & -r_f^{2T}\tilde{C}_f \end{bmatrix} \begin{bmatrix} s_p \\ 1 \end{bmatrix} \Rightarrow -r_f^{1T}\tilde{C}_f = \bar{u}_f, \quad -r_f^{2T}\tilde{C}_f = \bar{v}_f$$

Examples

3D Pose and
3D rigid Object





Projective Reconstruction from Uncalibrated Images

The projective reconstruction theorem

If a set of point correspondences in two views determine the fundamental matrix uniquely, then the scene and cameras may be reconstructed from these correspondences alone, and any two such reconstructions from these correspondences are projectively equivalent

$$x_i \leftrightarrow x'_i \qquad \left(P_1, P_1', \{X_{1i}\}\right) \quad \left(P_2, P_2', \{X_{2i}\}\right)$$

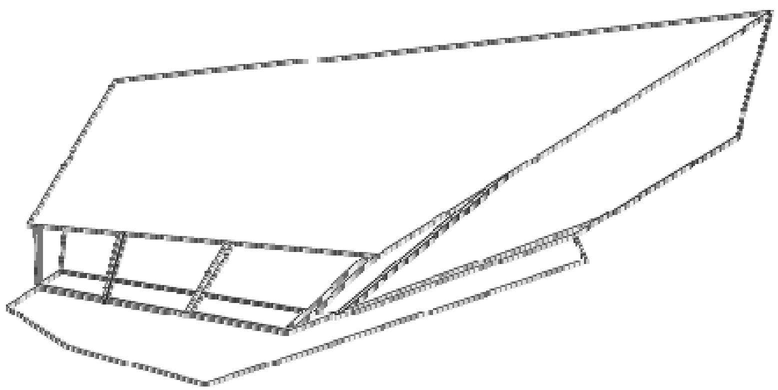
$$\underbrace{P_2 = P_1 H^{-1} \quad P_2' = P_1' H^{-1}}_{\text{theorem from last class}} \quad X_2 = HX_1 \qquad \left(\text{except}: Fx_i = x_i' F = 0\right)$$

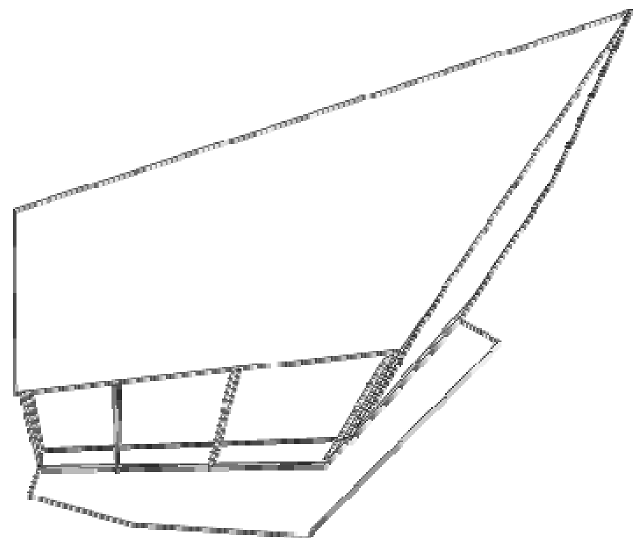
$$P_2\left(HX_{1i}\right) = P_1 H^{-1} H X_{1i} = P_1 X_{1i} = x_i = P_2 X_{2i}$$

$\Rightarrow$ along same ray of P_2, idem for P'_2

two possibilities: X_{2i}=HX_{1i}, or points along baseline

key result: allows reconstruction from pair of uncalibrated images





Projective ambiguity

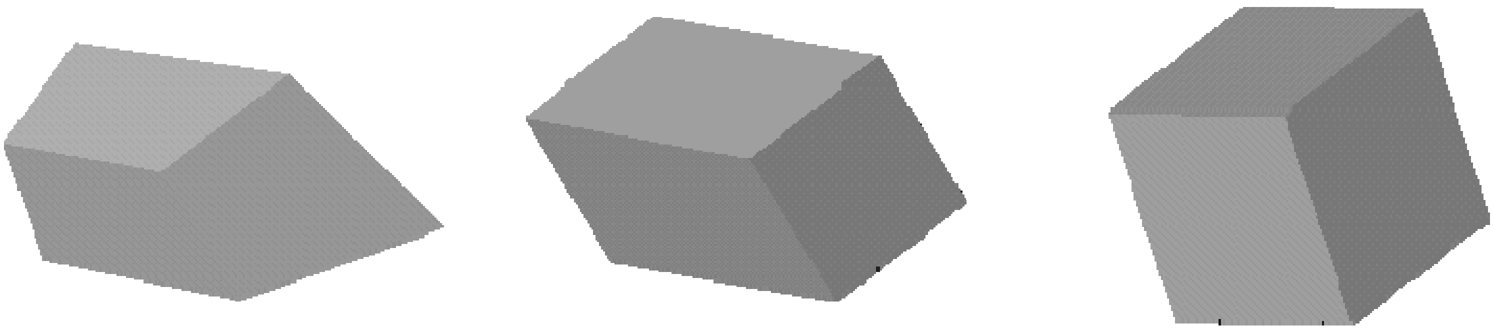
Reconstruction from uncalibrated images

⇒ the projective transformation is the only ambiguity between from a projective reconstruction to a metric construction.

$$\mathrm{m} = \mathbf{P}\,\mathrm{M} = (\mathbf{P}\mathbf{T}^{-1})(\mathbf{T}\,\mathrm{M})$$

Stratification of geometry

Projective	Affine	Metric



15 DOF	12 DOF	7 DOF
	plane at infinity	*absolute conic*
	parallelism	angles, rel.dist.

More general ←

More structure →

Projective reconstruction

- Factorization-based methods
  - M. Han and T. Kanade. **Creating 3D models with uncalibrated cameras**. In *Proc. WACV*, 2000.
  - B. Triggs. **Factorization methods for projective structure and motion**. *CVPR*, 1996.
  - P.F.Sturm, B. Triggs. **A Factorization Based Algorithm for Multi-Image Projective Structure and Motion**, *ECCV96*

- Through computation of fundamental matrix
  - M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, *Visual modeling with a hand-held camera*, International Journal of Computer Vision 59(3), 207-232, 2004.

Projective Reconstruction via Perspective Factorization

M. Han and T. Kanade. **Creating 3D models with uncalibrated cameras**. In *Proc*. *WACV*, 2000.

B. Triggs. **Factorization methods for projective structure and motion**. *CVPR*, 1996.

P.F.Sturm, B. Triggs. **A Factorization Based Algorithm for Multi-Image Projective Structure and Motion**, *ECCV96*

Goal

- Obtain a projective reconstruction of the scene and the projection matrices using a set of feature correspondences through matrix factorization

Perspective factorization

The camera equations

$$\lambda_{fp} \mathrm{m}_{fp} = \mathbf{P}_f \, \mathrm{M}_p, \; f = 1,\ldots,F, \; p = 1,\ldots,P$$

for a fixed image *f* can be written in matrix form as

$$\mathbf{m}_f \Lambda_f = \mathbf{P}_f \mathbf{M}$$

where

$$\mathbf{m}_f = \left[m_{f1}, m_{f2}, \ldots, m_{fP} \right], \; \mathbf{M} = \left[\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_P \right]$$

$$\Lambda_f = \mathrm{diag}\left(\lambda_{f1}, \lambda_{f2}, \ldots, \lambda_{fP} \right)$$

projective depth

Remark: for the orthographic projection, the projective depth of all the image points in all image frames are identical. Hence, projective depth is not taken into account in the factorization method for SfM using an orthographic projection model.

Rank-4 perspective factorization

All equations can be collected for all f as

$$\mathbf{W} = \mathbf{PM}$$

where

$$\mathbf{W} = \begin{bmatrix} \mathbf{m}_1 \Lambda_1 \\ \mathbf{m}_2 \Lambda_2 \\ \cdots \\ \mathbf{m}_F \Lambda_F \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \cdots \\ \mathbf{P}_F \end{bmatrix} \quad \mathbf{M} = [\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_P]$$

Remarks:

- In these formulas m_f are known, but Λ_f,$\mathbf{P}$ and $\mathbf{M}$ are all unknown.
- Observe that $\mathbf{W}$=$\mathbf{PM}$ is a product of a 3Fx4 matrix and a 4xP matrix, i.e. it is a rank 4 matrix.

Rank-4 factorization

Assume that $\{\Lambda_f\}$ f=1,…,F are known, then **W** is known.

Use the singular value decomposition

$$\mathbf{W}=\mathbf{PM}=U\Sigma V^T$$

In the noise-free case:

$$S=\mathrm{diag}(\sigma_1,\sigma_2,\sigma_3,\sigma_4,0, \ldots ,0)$$

and a reconstruction can be obtained by setting:

P=the first four columns of $U\Sigma$.

M=the first four rows of V.

Iterative rank-4 perspective factorization

When $\{ \Lambda_f \}$ are unknown the following algorithm can be used:

1. Set λ_{fp}=1 (affine approximation).

2. Factorize $W=\mathbf{PM}$ and obtain an estimate of $\mathbf{P}$ and $\mathbf{M}$.

3. Use $\mathbf{m}$, $\mathbf{P}$ and $\mathbf{M}$ to update Λ_i from the camera equations (linearly) so that $\mathbf{m}_f \Lambda_f = \mathbf{P}_f \mathbf{M}$.

If $\{\Lambda_f\}$ are the same as the previous iteration or σ_5 is sufficiently small, STOP the iteration. Otherwise goto 2.

Projective reconstruction Using F: Outline

(i) Compute F from correspondences
(ii) Compute camera matrices from F
(iii) Compute 3D point for each pair of corresponding points

computation of F
use $x'_i F x_i = 0$ equations, linear in coeff. F
8 points (linear), 7 points (non-linear), 8+ (least-squares)
(more on this next class)

computation of camera matrices
use $P = [I \mid 0] \quad P' = [[e']_\times F + e' v^T \mid \lambda e']$

triangulation
compute intersection of two backprojected rays using the method introduced in the begining of the lecture

Automatic Metric Upgrading from Projective Reconstruction (Self-Calibration)

Projective ambiguity

Reconstruction from uncalibrated images

$\Rightarrow$ Recall that the projective transformation is the only ambiguity between from a projective reconstruction to a metric construction.

$$\mathrm{m} = \mathbf{P}\,\mathrm{M} = (\mathbf{P}\mathbf{T}^{-1})(\mathbf{T}\,\mathrm{M})$$

Motivation

- Allow flexible acquisition
  - No prior calibration necessary
  - Possibility to vary intrinsic parameters
  - Use archive footage

Constraints ?

- Scene constraints
  - Parallellism, vanishing points, horizon, ...
  - Distances, positions, angles, ...

Unknown scene → no constraints

- Camera extrinsics constraints
  - Pose, orientation, ...

Unknown camera motion → no constraints

- Camera intrinsics constraints
  - Focal length, principal point, aspect ratio & skew

Knowledge about the camera intrinsic parameters
→ some constraints

Euclidean projection matrix

Factorization of Euclidean projection matrix

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R}^{\top} \quad -\mathbf{R}^{\top}\mathbf{t}\right]$$

Intrinsics: $\mathbf{K} = \begin{bmatrix} f_x & s & u_x \\ & f_y & u_y \\ & & 1 \end{bmatrix}$ (camera geometry)

Extrinsics: $(\mathbf{R}, \mathrm{t})$ (camera motion)

Note: every projection matrix can be factorized,
but only meaningful for euclidean projection matrices

Constraints on intrinsic parameters

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_x \\ & f_y & u_y \\ & & 1 \end{bmatrix}$$

- Constant

  e.g. fixed camera: $\mathbf{K}_1 = \mathbf{K}_2 = \cdots$

- Known

  e.g. rectangular pixels: $s = 0$

  square pixels: $f_x = f_y, s = 0$

  principal point known:

  $$(u_x, u_y) = \left(\frac{w}{2}, \frac{h}{2}\right)$$

Self-calibration

Upgrade from *projective* structure to *metric* structure using *constraints on intrinsic* camera parameters

- Constant intrinsics

(Faugeras et al. ECCV´92, Hartley´93, Triggs´97, Pollefeys et al. PAMI´98, ...)

- Some known intrinsics, others varying

(Heyden&Astrom CVPR´97, Pollefeys et al. ICCV´98)

- Constraints on intrincs and restricted motion
  (e.g. pure translation, pure rotation, planar motion)

(Moons et al.´94, Hartley ´94, Armstrong ECCV´96, ...)

Conics & Quadrics

conics

$$m^T \mathbf{C} m = 0 \qquad l^T \mathbf{C}^* l = 0$$

$$\mathbf{C}^* = \mathbf{C}^{-1}$$

quadrics

$$M^T \mathbf{Q} M = 0 \qquad \Pi^T \mathbf{Q}^* \Pi = 0$$

$$\mathbf{Q}^* = \mathbf{Q}^{-1}$$

transformations

$$\mathbf{C} \mapsto \mathbf{H}^{-T} \mathbf{C} \mathbf{H}^{-1}$$

$$\mathbf{C}^* \mapsto \mathbf{H} \mathbf{C}^* \mathbf{H}^T$$

$$\mathbf{Q} \mapsto \mathbf{T}^{-T} \mathbf{Q} \mathbf{T}^{-1}$$

$$\mathbf{Q}^* \mapsto \mathbf{T} \mathbf{Q}^* \mathbf{T}^T$$

projection

$$\mathbf{C}^* \sim \mathbf{P} \mathbf{Q}^* \mathbf{P}^T$$

The Absolute Conic

Ω_∞ is a specific imaginary conic on π_∞,

$$X_1^2 + X_2^2 + X_3^2 = 0 \text{ and } X_4 = 0$$

for canonical mapping (metric frame)

or $$(X_1, X_2, X_3)\mathbf{I}(X_1, X_2, X_3)^T \text{ in } \pi_\infty$$

Remember, the absolute conic is fixed under H if, and only if, H is a similarity transformation

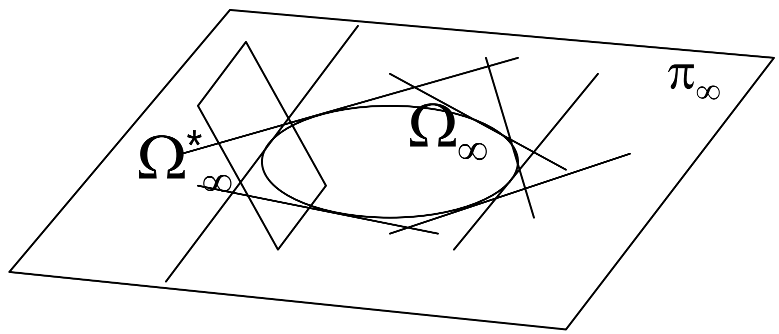
Image related to intrinsic parameters

The Absolute Dual Quadric

(Triggs CVPR´97)

Degenerate dual quadric Ω^*_∞

Encodes both absolute conic Ω_∞ and π_∞



for metric frame: $$\pi^T \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \pi = 0$$

The image of the absolute conic (IAC)

$$C \mapsto H^{-T} C H^{-1}$$

$$\Omega_\infty = I, \ H = KR$$

$$\omega = K^{-T} R^{-T} I R^{-1} K^{-1} = K^{-T} K^{-1} = \left(KK^T\right)^{-1}$$

(i) IAC depends only on intrinsic parameters.
(ii) angle between two rays (i.e. two directions)

$$\cos\theta = \frac{x_1^T \omega x_2}{\sqrt{\left(x_1^T \omega x_1\right)\left(x_2^T \omega x_2\right)}}$$

Dual Image of the Absolute Conic is
(DIAC) = ω^* = KK^T
It is also the image of the absolute dual quadric (IADQ) DIAC=IADQ

ω **⇔ K (Cholesky factorization).** ω **Can be determined by images of circular points.**

Absolute Dual Quadric and Self-calibration

Projection of dual quadric

$$\mathbf{P}\Omega_\infty^* \mathbf{P}^\mathsf{T} \propto \mathbf{K}\mathbf{K}^\mathsf{T} \qquad \Omega_\infty^* = \mathrm{diag}(1,1,1,0)$$

Abs.Dual Quadric also exists in projective world

$$\mathbf{K}\mathbf{K}^\mathsf{T} \propto \mathbf{P}\Omega_\infty^* \mathbf{P}^\mathsf{T} \propto (\mathbf{P}\mathbf{T}^{-1})(\mathbf{T}\Omega_\infty^* \mathbf{T}^\mathsf{T})(\mathbf{T}^{-\mathsf{T}}\mathbf{P}^\mathsf{T})$$

$$\propto \mathbf{P}'\Omega'^*_\infty \mathbf{P}'^\mathsf{T}$$

8 d.o.f.

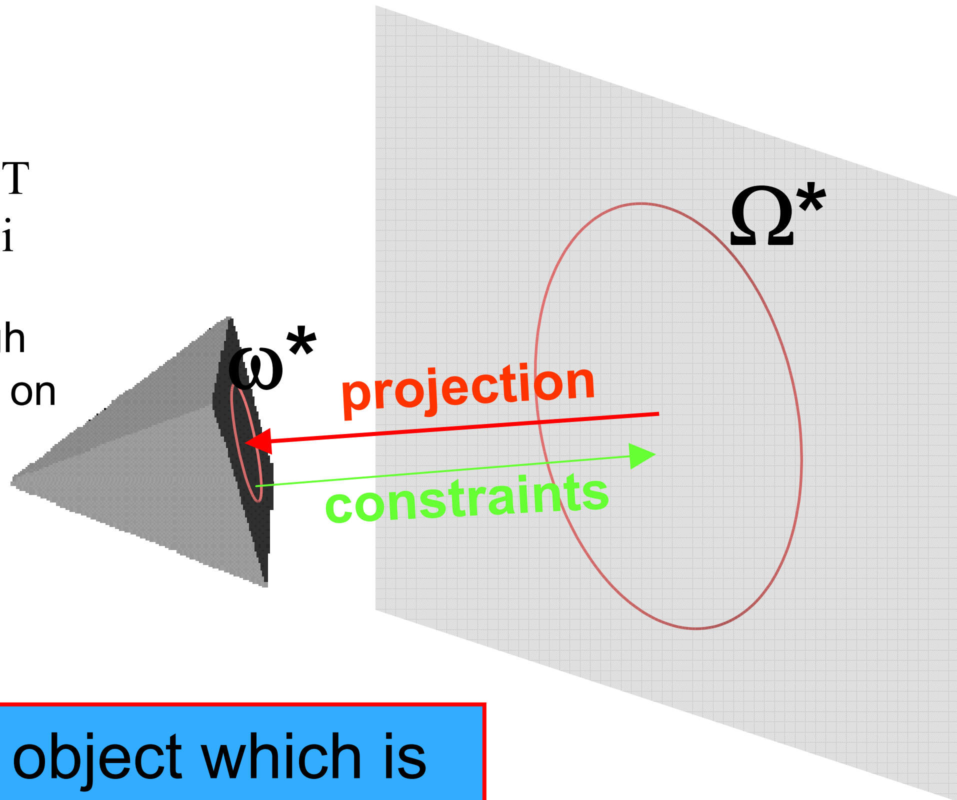
Transforming world so that $\Omega'^*_\infty \to \Omega_\infty^*$ reduces ambiguity to metric

Absolute Dual Quadric and Self-calibration

Projection equation:

$$\omega_i^* \propto \mathbf{P}_i \Omega^* \mathbf{P}_i^T \propto \mathbf{K}_i \mathbf{K}_i^T$$

Translate constraints on **K** through projection equation to constraints on Ω^*



Absolute conic = calibration object which is always present but can only be observed through constraints on the intrinsics

Constraints on ω^*_∞

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix} \qquad \omega^*_\infty = \begin{bmatrix} f_x^2 + s^2 + c_x^2 & sf_y + c_x c_y & c_x \\ sf_y + c_x c_y & f_y^2 + c_y^2 & c_y \\ c_x & c_y & 1 \end{bmatrix}$$

m, number of frames

condition	constraint	type	#constraints
Zero skew	$\omega^*_{12}\omega^*_{33} = \omega^*_{13}\omega^*_{23}$	quadratic	m
Principal point	$\omega^*_{13} = \omega^*_{23} = 0$	linear	$2m$
Zero skew (& p.p.)	$\omega^*_{12} = 0$	linear	m
Fixed (unknown) aspect ratio (& p.p.& Skew)	$\omega^*_{11}\omega'^*_{22} = \omega^*_{22}\omega'^*_{11}$	quadratic	m-1
Known aspect ratio (& p.p.& Skew)	$\omega^*_{11} = \omega^*_{22}$	linear	m

Linear algorithm

(Pollefeys et al.,ICCV´98/IJCV´99)

Assume everything known, except focal length, which could be varying.

$$\omega_\infty^* \approx \begin{bmatrix} \hat{f}^2 & 0 & 0 \\ 0 & \hat{f}^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \propto \mathbf{P}\Omega^*\mathbf{P}^\mathrm{T}$$

$$\left(\mathbf{P}\Omega^*\mathbf{P}^\mathrm{T}\right)_{11} - \left(\mathbf{P}\Omega^*\mathbf{P}^\mathrm{T}\right)_{22} = 0$$
$$\left(\mathbf{P}\Omega^*\mathbf{P}^\mathrm{T}\right)_{12} = 0$$
$$\left(\mathbf{P}\Omega^*\mathbf{P}^\mathrm{T}\right)_{13} = 0$$
$$\left(\mathbf{P}\Omega^*\mathbf{P}^\mathrm{T}\right)_{23} = 0$$

Yields 4 constraint per image

Note that rank-3 constraint is not enforced

Linear algorithm revisited

(Pollefeys et al., ECCV'02)

Weighted linear equations

$$\mathbf{KK}^{\mathrm{T}} \approx \begin{bmatrix} \hat{f}^2 & 0 & 0 \\ 0 & \hat{f}^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\hat{f} \approx 1$$

$$\frac{1}{0.2}\left(\mathbf{P}\Omega^*\mathbf{P}^{\mathrm{T}}\right)_{11} - \left(\mathbf{P}\Omega^*\mathbf{P}^{\mathrm{T}}\right)_{22} = 0$$
$$\frac{1}{0.01}\left(\mathbf{P}\Omega^*\mathbf{P}^{\mathrm{T}}\right)_{12} = 0$$
$$\frac{1}{0.1}\left(\mathbf{P}\Omega^*\mathbf{P}^{\mathrm{T}}\right)_{13} = 0$$
$$\frac{1}{0.1}\left(\mathbf{P}\Omega^*\mathbf{P}^{\mathrm{T}}\right)_{23} = 0$$
$$\frac{1}{9}\left(\mathbf{P}\Omega^*\mathbf{P}^{\mathrm{T}}\right)_{11} - \left(\mathbf{P}\Omega^*\mathbf{P}^{\mathrm{T}}\right)_{33} = 0$$
$$\frac{1}{9}\left(\mathbf{P}\Omega^*\mathbf{P}^{\mathrm{T}}\right)_{22} - \left(\mathbf{P}\Omega^*\mathbf{P}^{\mathrm{T}}\right)_{33} = 0$$

assumptions

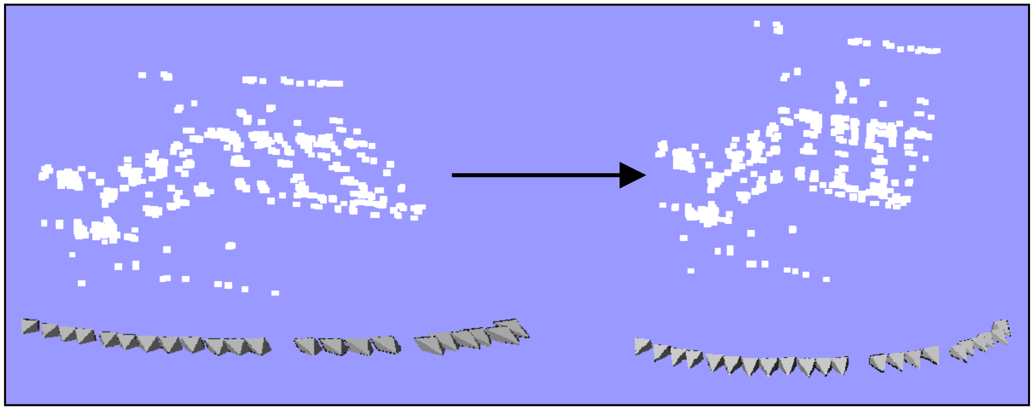
$$\log(\hat{f}) \approx \log(1) \pm \log(3) \qquad c_x \approx 0 \pm 0.1 \qquad s = 0$$
$$\log\left(\frac{\hat{f}_x}{\hat{f}_y}\right) \approx \log(1) \pm \log(1.1) \qquad c_y \approx 0 \pm 0.1$$

Projective to metric

Compute $\mathbf{T}$ from $\tilde{\mathbf{I}} = \mathbf{T}\Omega_\infty^* \mathbf{T}^\mathsf{T}$ or $\mathbf{T}^{-1}\tilde{\mathbf{I}}\mathbf{T}^{-\mathsf{T}} = \Omega_\infty^*$ with $\tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{I} & 0 \\ 0^\mathsf{T} & 0 \end{bmatrix}$

using eigen decomposition of Ω_∞^*

and then obtain metric reconstruction as $\mathbf{PT}^{-1}$ and $\mathbf{TM}$



Critical motion sequences

(Sturm, CVPR´97, Kahl, ICCV´99, Pollefeys,PhD´99)

- Self-calibration depends on camera motion
- Motion sequence is not always general enough

- Critical Motion Sequences have more than one potential absolute conic satisfying all constraints
- Possible to derive classification of CMS

Critical motion sequences: constant intrinsic parameters

Most important cases for constant intrinsics

Critical motion type	ambiguity
pure translation	affine transformation (5DOF)
pure rotation	arbitrary position for Π_∞ (3DOF)
orbital motion	proj.distortion along rot. axis (2DOF)
planar motion	scaling axis $\perp$ plane (1DOF)

Note relation between *critical motion sequences* and *restricted motion algorithms*

Critical motion sequences: varying focal length

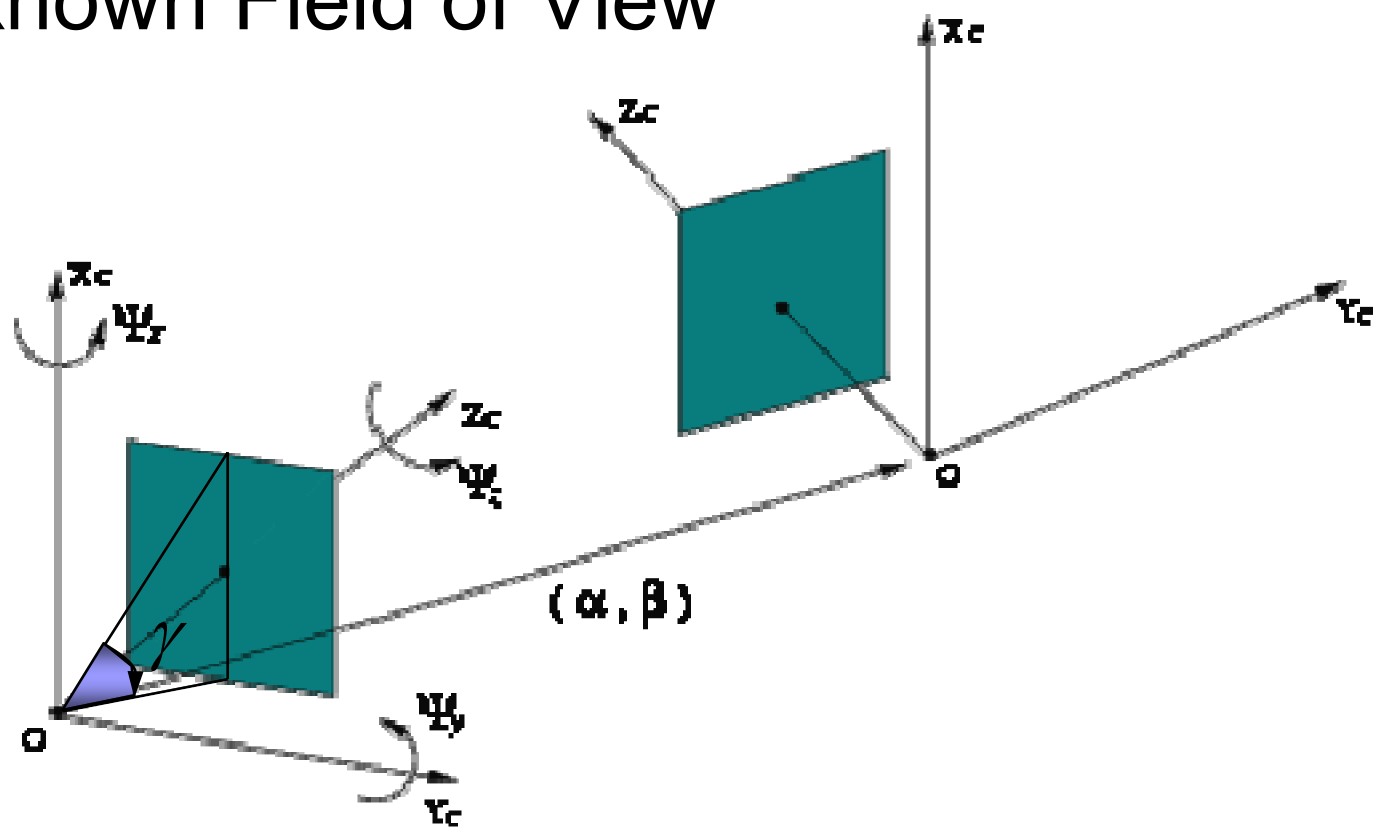
Most important cases for varying focal length (other parameters known)

Critical motion type	ambiguity
pure rotation	arbitrary position for Π_∞ (3DOF)
forward motion	proj.distortion along opt. axis (2DOF)
translation and rot. about opt. axis	scaling optical axis (1DOF)
hyperbolic and/or elliptic motion	one extra solution

SfM with direct estimation of unknown camera parameters

Reference: G. Qian and R. Chellappa, "Bayesian Self-Calibration of a Moving Camera," Computer Vision and Image Understanding, vol. 95, No. 3, pp. 287-316, September 2004

Partially Uncalibrated Camera: Unknown Field of View



$$m_t = (\Psi_x, \Psi_y, \Psi_z, \alpha, \beta, \gamma)$$

Motivation

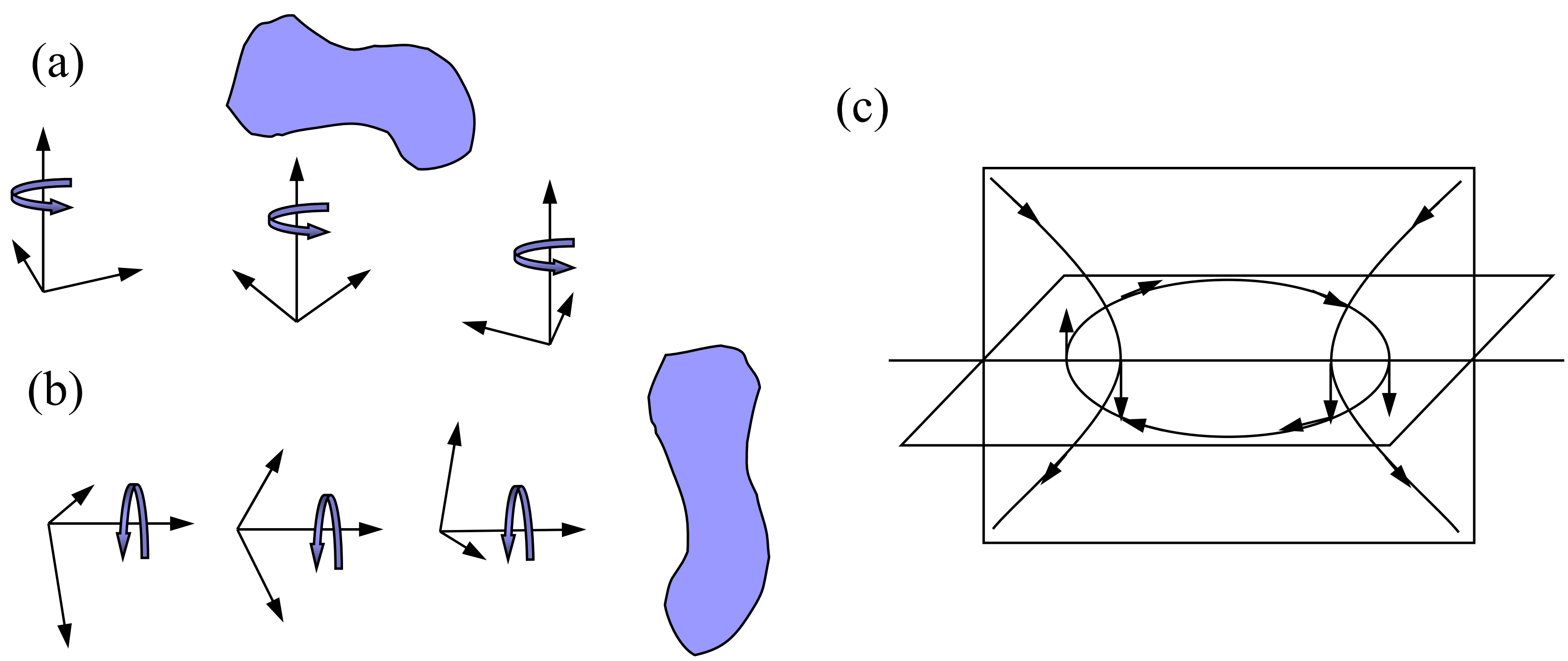
- Difficulties of Current Methods
  - Sensitivity to measurement noise
  - Requirement of good initial guess
- Challenges
  - Critical motion sequences

Critical Motion Sequences

- Not all camera motion sequences lead to unique camera intrinsic parameters and 3D Euclidean scene reconstruction.
- Camera motion sequences that produce ambiguous Euclidean reconstruction are called critical motion sequences (CMS).
- Identification of CMS
  - Constant calibration [Strum CVPR 1997]
  - Variable calibration [Kahl, Triggs & Astrom JMIV 2000]
  - Unknown (constant / variable) focal length [Strum, BMVC 1999]

CMS for Unknown Focal Length

- Translation with any rotation about the optical axis
- Translation along the optical axis with arbitrary rotation about the center projections at most twice in the sequence
- Translation along an ellipse or a hyperbola with the optical axis tangent to the ellipse or hyperbola.



Why *not* Unique?

$$m \cong Pw = A[R \mid -Rt]w$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$m \cong PT \cdot T^{-1} w$$

$PT \rightarrow P'$, $T^{-1}w \rightarrow w'$

$$T = \begin{bmatrix} \Delta A & 0 \\ 0 & s \end{bmatrix} \quad \Delta A = \begin{bmatrix} \Delta f & 0 & 0 \\ 0 & \Delta f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R \cdot \Delta A = \Delta A \cdot R$$

$$P' = PT$$

$$= A[R \mid -Rt]\begin{bmatrix} \Delta A & 0 \\ 0 & s \end{bmatrix}$$

$$= A[R \cdot \Delta A \mid -sRt]$$

$$= A[R \cdot \Delta A \mid -sR \cdot \Delta A \Delta A^{-1} t]$$

$$= A[\Delta A \cdot R \mid -\Delta A \cdot sR\Delta A^{-1} t]$$

$$= A \cdot \Delta A[R \mid -Rs\Delta A^{-1} t]$$

$A \cdot \Delta A \rightarrow A'$, $s\Delta A^{-1} t \rightarrow t'$

$$m \cong P'w' = A'[R \mid -Rt']w'$$

True v.s. False

$$f_\Phi = \Delta f \cdot f_e$$

$$T = \begin{bmatrix} \Delta f & 0 & 0 \\ 0 & \Delta f & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

$$t_\Phi = \kappa \cdot T \cdot t_e$$

$$\alpha_\Phi = \cos^{-1} \frac{\Delta f \cdot \cos \alpha_e}{\sqrt{\Delta f^2 \cdot \cos^2 \alpha_e + \sin^2 \alpha_e}}$$

$$\beta_\Phi = \beta_e$$

$$s_\Phi = \kappa \cdot T \cdot s_e$$

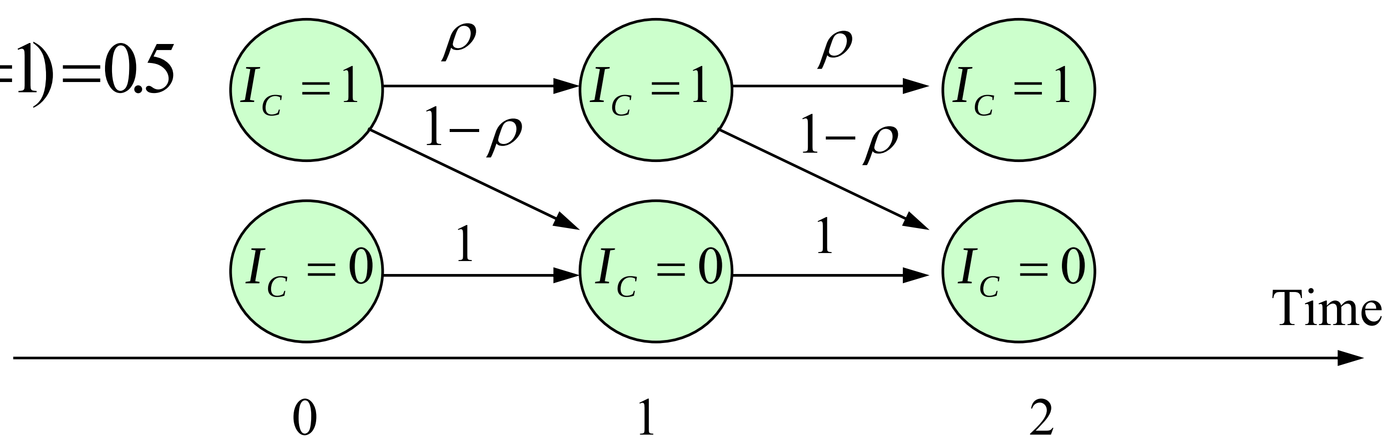
Processing CMS

- Detection of CMS as a hypothesis testing

$$I_C = \begin{cases} 1, \text{ the motion sequence is critical} \\ 0, \text{ the motion sequence is not cirtical} \end{cases}$$

Modeling of criticalness transition using a Markovian chain

$$\pi_0(I_C = 1) = 0.5$$



$$\rho = P(I_C(t+1) = 0 \mid I_C(t) = 1) = 0.5$$

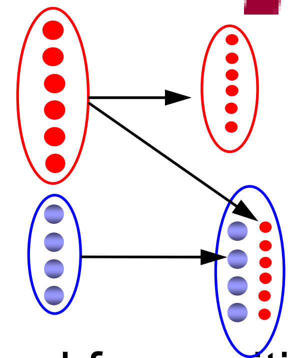
Posterior Probability of Criticalness

- Critical and general motion sample sets

- Posterior probability of the criticalness

$$p(I_C = 1) = \lim_{N \to \infty} \frac{\sum w_c}{\sum w_c + \sum w_g}$$

Sample Transfer and Uniforming



- To implement the criticality transition of a motion sequence, samples in the critical set are transferred to the general set.

- Let $\{x_T^{(j)}, w_T^{(j)}\}_{j=1}^{n}$ be the samples and weights transferred from critical sample set to the non-critical set. For each sample

  (A) Uniformly draw $\Gamma_j = \left\{\gamma_j^{(i)}\right\}_{i=1}^{n_j}$ from $[0, \pi]$ with

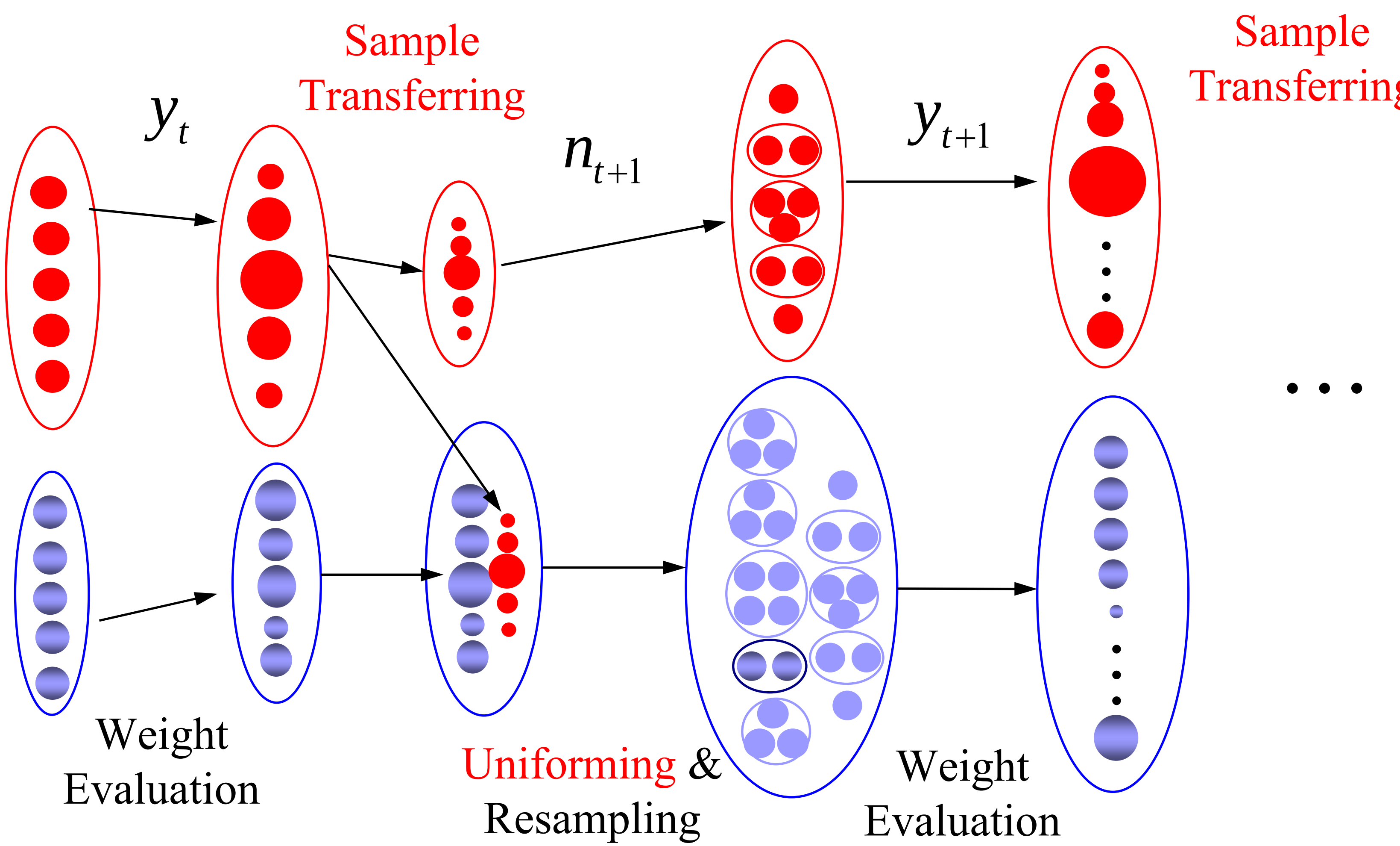
$$x_T^{(j)} = \left(0,0,\psi_z,\alpha,\beta,\gamma\right)$$

  (B) For each sample $\gamma_j^{(i)}$, a new sample can be obtained.

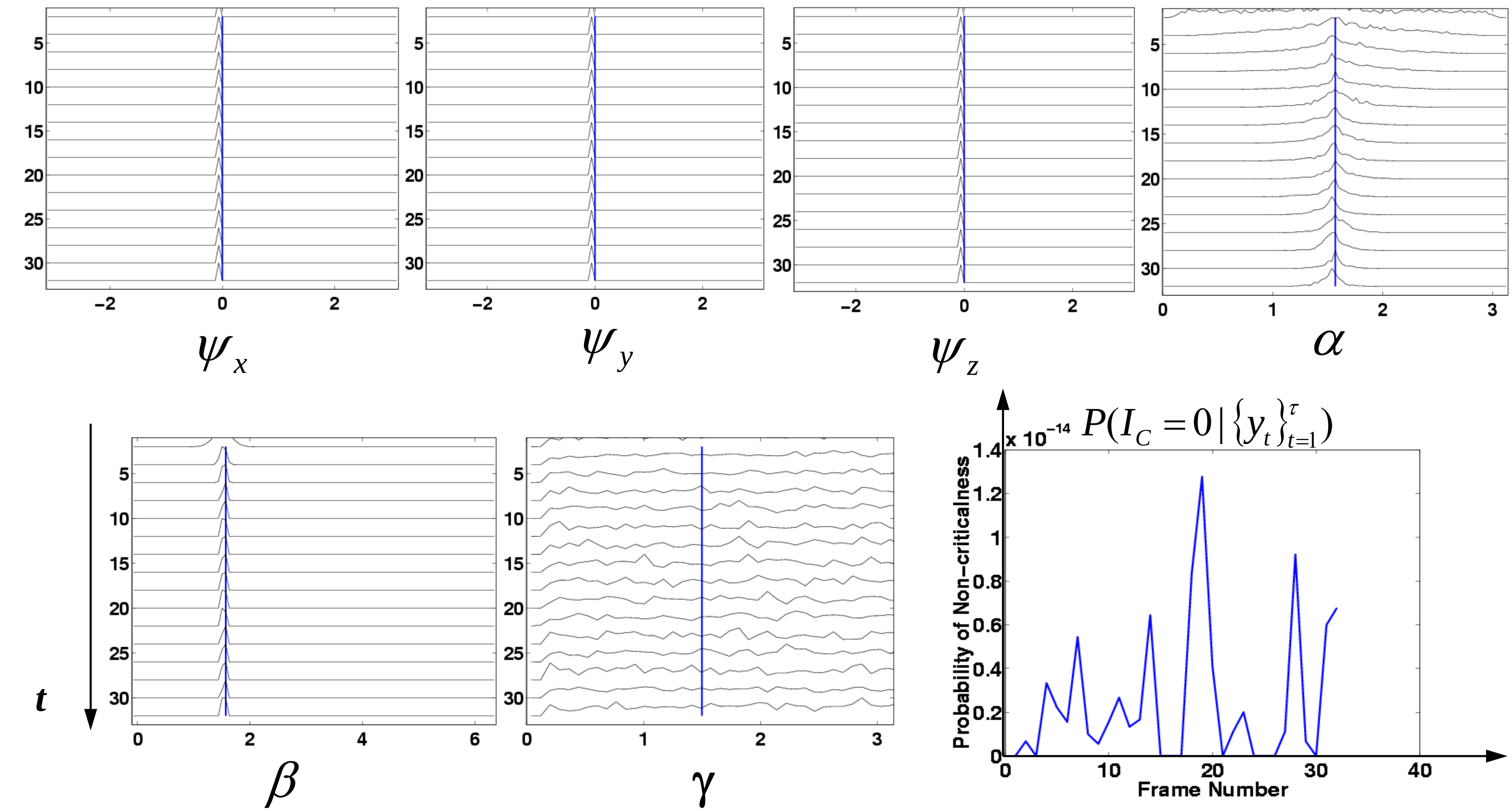
$$n_j = \frac{n w_T^{(j)}}{\sum w_C^{(l)} + \sum w_G^{(l)}}$$

$$x_j^{(i)} = \left(0,0,\psi_z,\alpha_j^{(i)},\beta,\gamma_j^{(i)}\right)$$

The Self-Calibration Algorithm

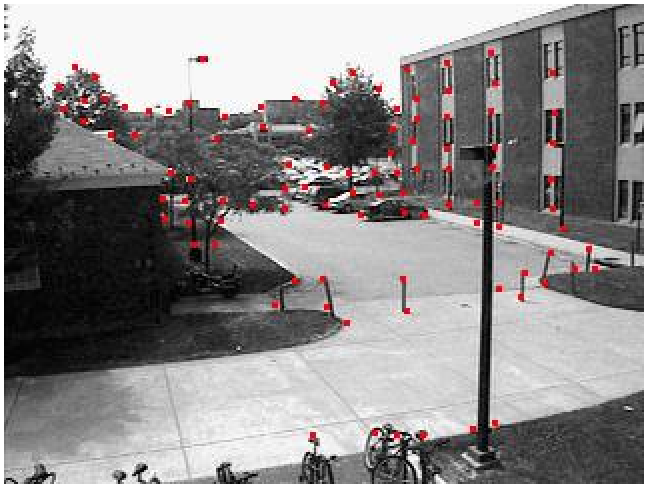


A Critical Motion Sequence: Horizontal Translation

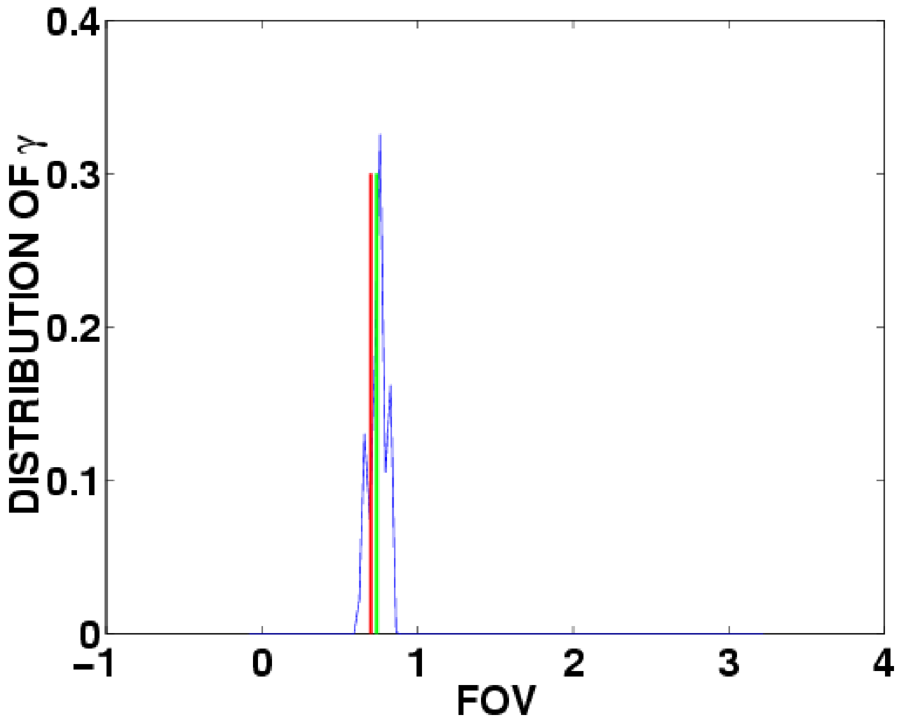


Outdoor Camera

Input video

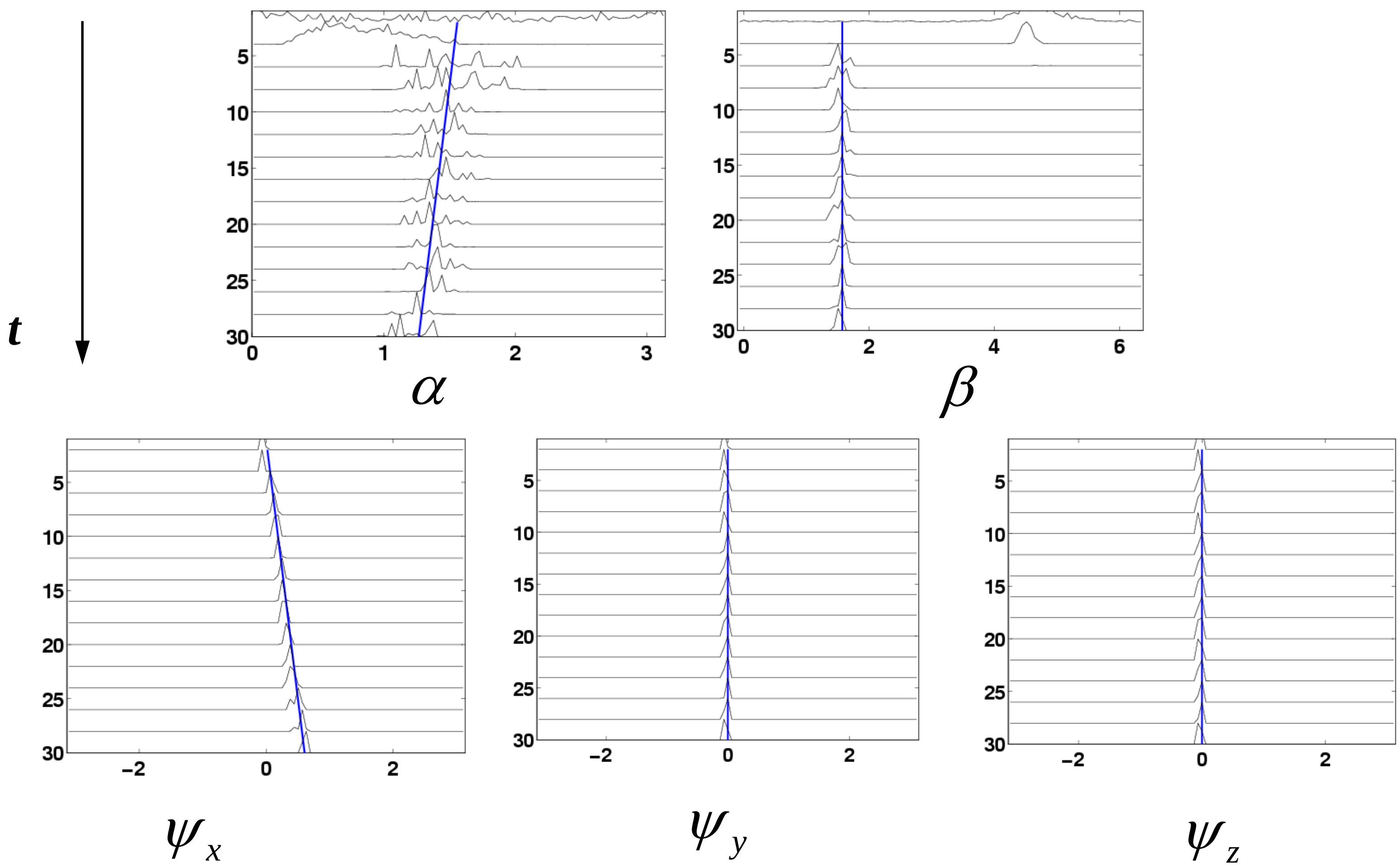


Feature points

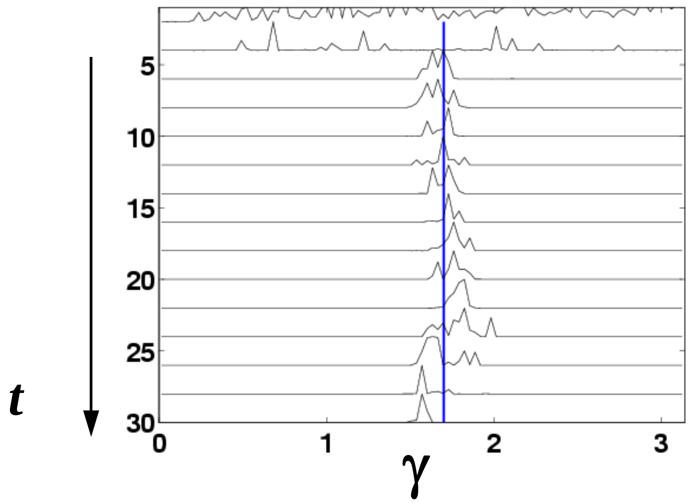


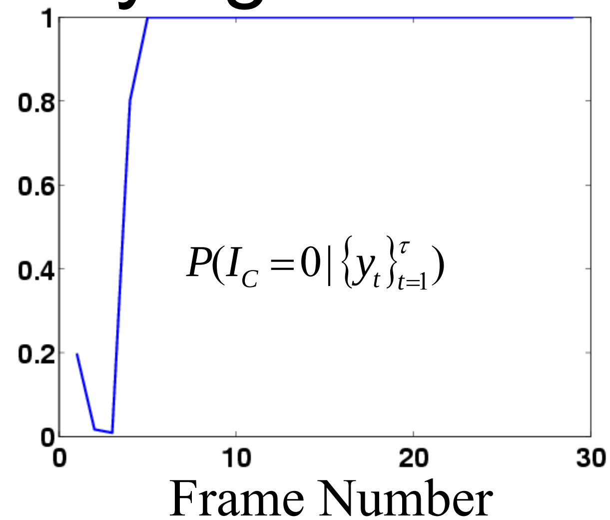
Posterior distribution of field of view

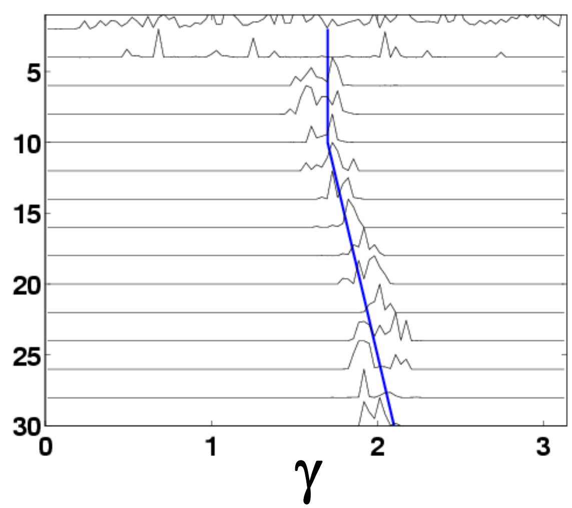
A Non-Critical Motion Sequence: Circular Motion With Varying FOV

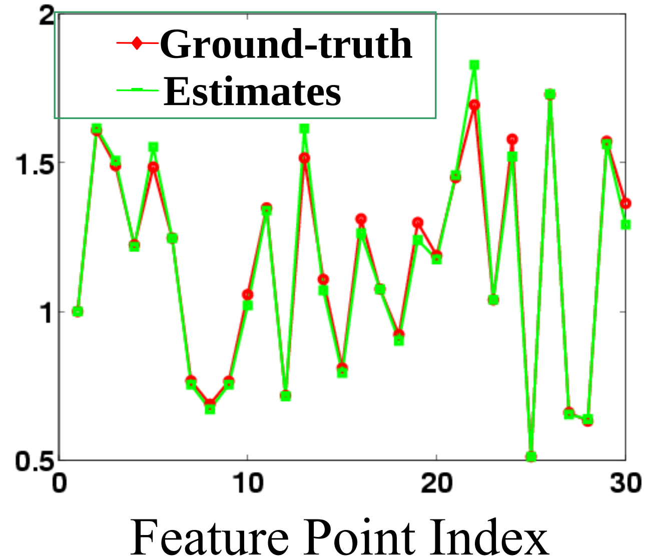


A Non-Critical Motion Sequence: Circular Motion With Varying FOV

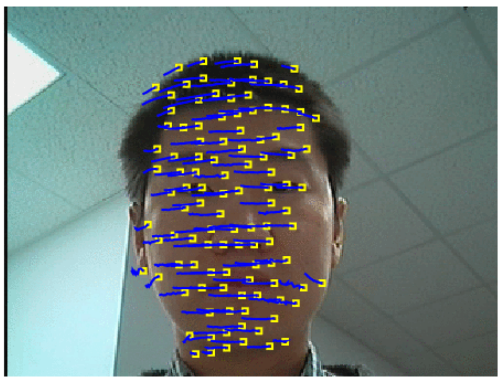




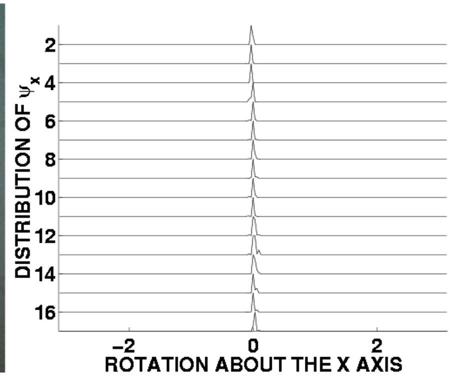




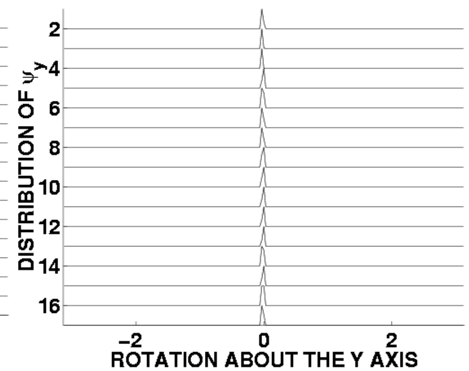
Face Modeling: Motion Estimates



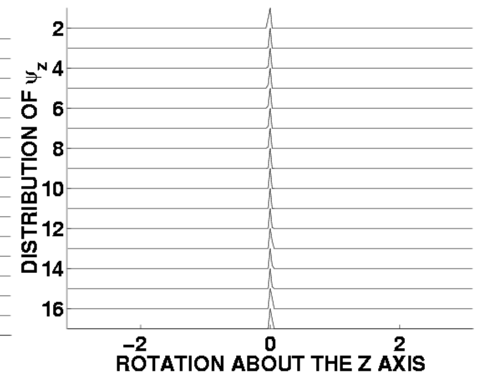
Feature Trajectories



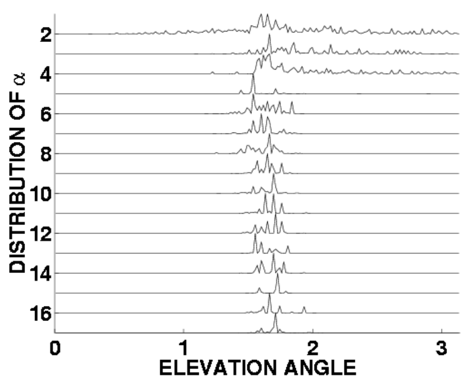
ψ_x



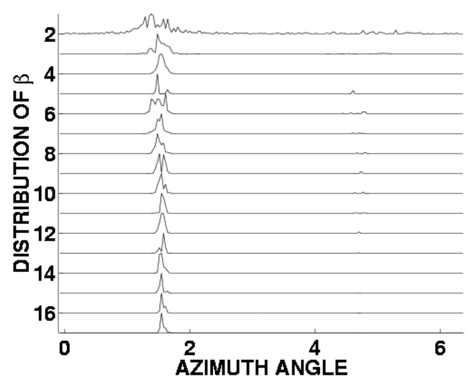
ψ_y



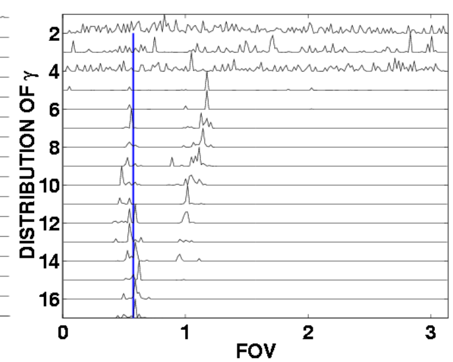
ψ_z



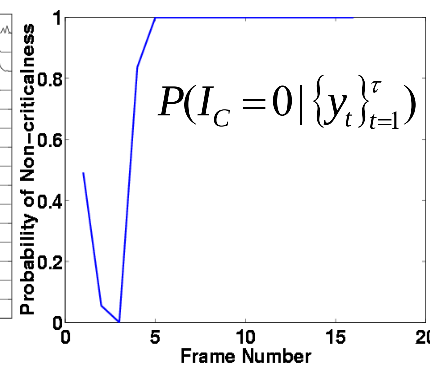
α



β

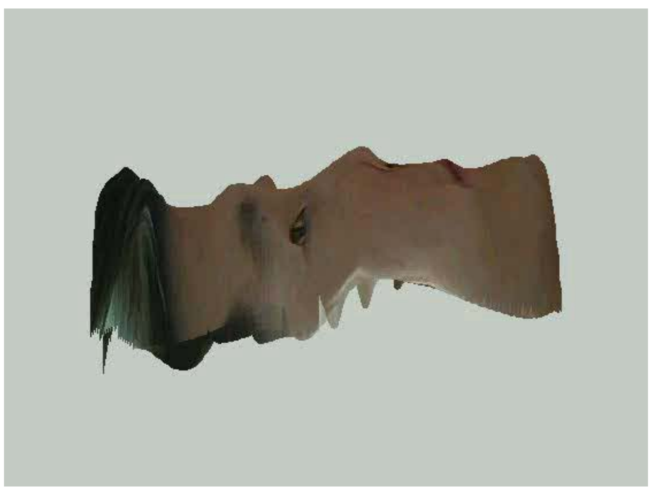


λ

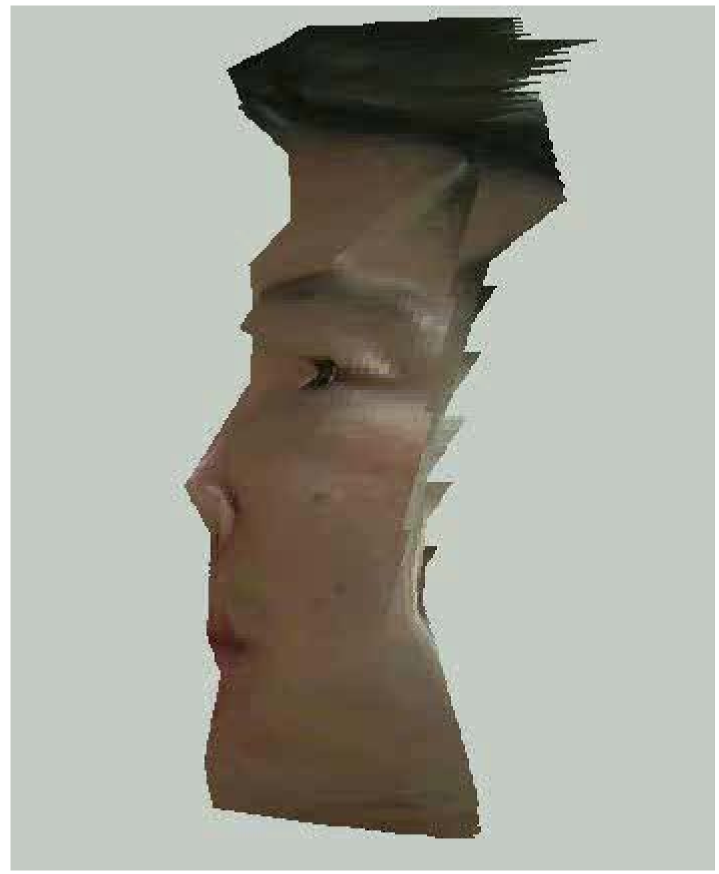


Face Modeling: 3D Reconstruction

Original video



Reconstruction



Reconstruction