IEEE VR 2003 tutorial 1

Recent Methods for Image-based Modeling and Rendering

Lecture 3: Modeling from images

Darius Burschka
Johns Hopkins University

Martin Jagersand
University of Alberta

Dana Cobzas
University of Alberta

Keith Yerex
Virtual Universe Corporation

Zach Dodds
Harvey Mudd College

Pinhole camera

- Central projection

$$(X,Y,Z)^T \to (fX/Z,\, fY/Z)^T$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

- Principal point & aspect

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} \frac{1}{p_x} x + c_x \\ \frac{1}{p_y} y + c_y \\ 1 \end{pmatrix} = \begin{bmatrix} \frac{1}{p_x} & 0 & c_x \\ 0 & \frac{1}{p_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

The projection matrix:

$$\mathbf{x} = \left[\boxed{\begin{matrix} \frac{f}{p_x} & 0 & c_x \\ 0 & \frac{f}{p_y} & c_y \\ 0 & 0 & 1 \end{matrix}} \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \right] \mathbf{X}_{cam}$$

$$\mathbf{x} = K[I \mid \mathbf{0}]\mathbf{X}_{cam}$$

Projective camera

- Camera rotation and translation

$$\mathbf{X} = [R \quad \mathbf{t}]\mathbf{X}_{cam} \qquad \mathbf{X}_{cam} = [R^T \quad -R^T\mathbf{t}]\mathbf{X}$$

- The projection matrix

$$\mathbf{x} = \underbrace{KR^T[I \quad -\mathbf{t}]}_{P}\mathbf{X}$$

In general:

•P is a 3x4 matrix with 11 DOF

•Finite: left 3x3 matrix non-singular

•Infinite: left 3x3 matrix singular

Properties: P=[M p_4]

•Center: $P\mathbf{C}=0$

$$\mathbf{C} = \begin{pmatrix} -M^{-1}\mathbf{p}_4 \\ 1 \end{pmatrix} \qquad \mathbf{C} = \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix}, M\mathbf{d} = 0$$

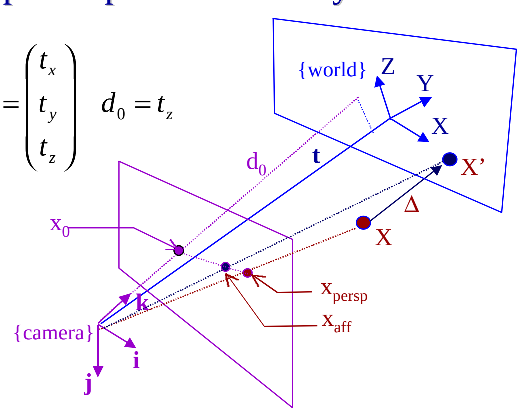
•Principal ray (projection direction)

$$\mathbf{v} = \det(M)\mathbf{m}^3$$

Affine cameras

- Infinite cameras where the last row of P is (0,0,0,1)
- Points at infinity are mapped to points at infinity

$$P_\infty = K\begin{bmatrix} \mathbf{i} & t_x \\ \mathbf{j} & t_y \\ \mathbf{0}^T & d_0 \end{bmatrix} \quad R = \begin{pmatrix} \mathbf{i} \\ \mathbf{j} \\ \mathbf{k} \end{pmatrix} \quad \mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad d_0 = t_z$$



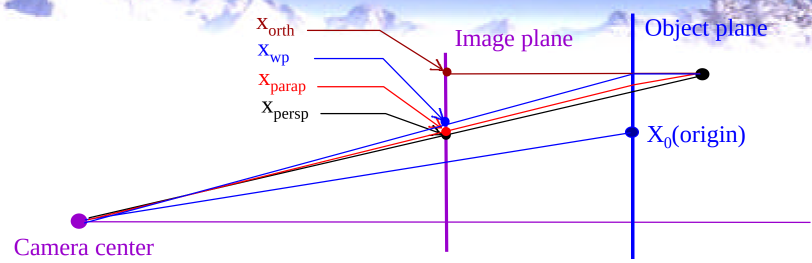
- Error

$$\mathbf{x}_{aff} - \mathbf{x}_{persp} = \frac{\Delta}{d_0}(\mathbf{x}_{proj} - \mathbf{x}_0)$$

Good approximation:

- Δ small compared to d_0
- point close to principal ray

Hierarchy of cameras



Perspective:

$$P_{persp} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{i} & t_x \\ \mathbf{j} & t_y \\ \mathbf{k} & t_z \end{bmatrix}$$

Weak perspective:

$$P_{wp} = \begin{bmatrix} k & & \\ & k & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{i} & t_x \\ \mathbf{j} & t_y \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Orthographic:

$$P_{orth} = \begin{bmatrix} \mathbf{i} & t_x \\ \mathbf{j} & t_y \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Para-perspective:

First order approximation of perspective

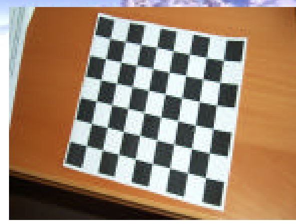
Examples of camera projections

perspective

Orthographic (parallel)

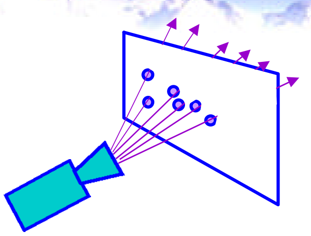
Camera calibration

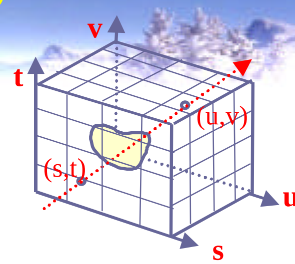
$$\underset{\text{known}}{\mathbf{x}_i} = \underset{?}{P}\,\underset{\text{known}}{\mathbf{X}_i}$$



- 11 DOF => at least 6 points
- Linear solution $\begin{cases} \min A\mathbf{p} = 0 \\ \|\mathbf{p}\| = 1 \end{cases}$
  - Normalization required
  - Minimizes algebraic error
- Nonlinear solution
  - Minimize geometric error (pixel re-projection)
- Radial distortion $dr = 1 + K_1 r + K_2 r^2 + \ldots$
  - Small near the center, increase towards periphery

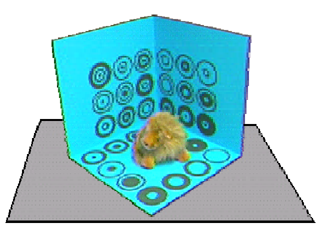
Application: raysets

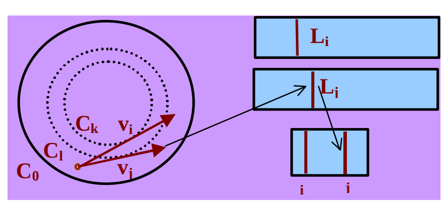




Gortler and *al.*; Microsoft
Lumigraph

H-Y Shum, L-W He; Microsoft
Concentric mosaics



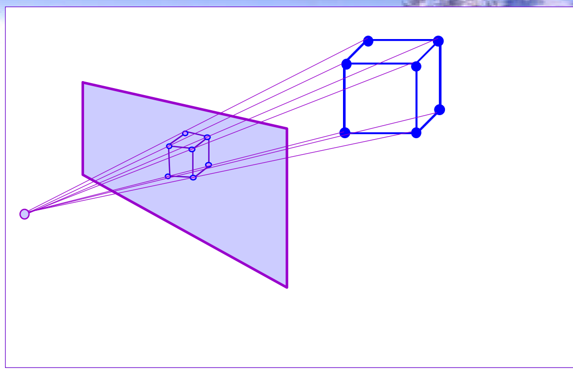


Multi-view geometry - resection

- Projection equation

  $x_i = P_i X$

- Resection:
  - $x_i, X \longrightarrow P_i$



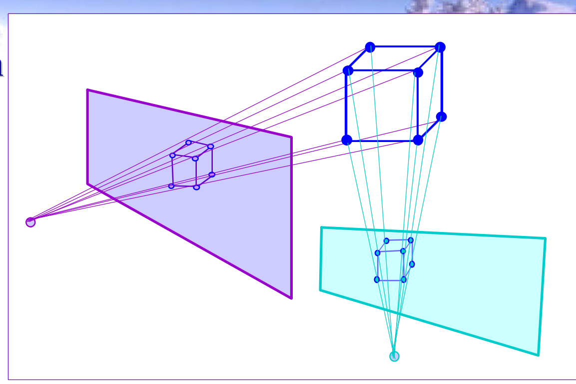
Given image points and 3D points calculate camera projection matrix.

Multi-view geometry - intersection

- Projection equation

  $x_i = P_i X$

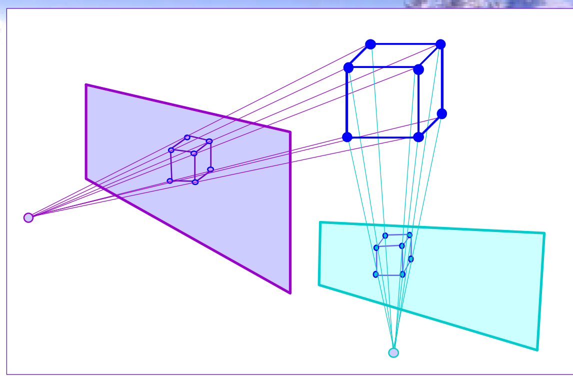
- Intersection:
  - $x_i, P_i \longrightarrow X$



Given image points and camera projections in at least 2 views calculate the 3D points (structure)

Multi-view geometry - SFM

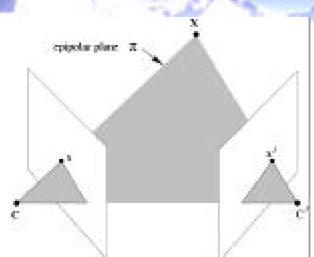
- Projection equation
  $x_i = P_i X$
- Structure from motion (SFM)
  - $x_i \longrightarrow P_i, X$

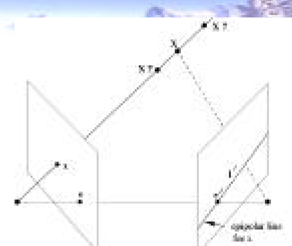


Given image points in at least 2 views calculate the 3D points (structure) and camera projection matrices (motion)

- Estimate projective structure
- Rectify the reconstruction to metric (autocalibration)

2 view geometry (Epipolar geometry)

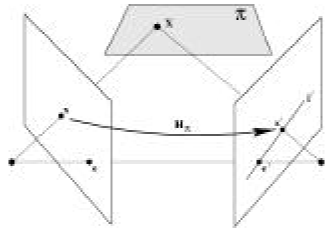




Fundamental matrix

[Faugeras ’92, Hartley ’92]

- Algebraic representation of epipolar geometry



Step 1: X on a plane π $\mathbf{x}' = H\mathbf{x}$

Step 2: epipolar line l’ $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

$= [\mathbf{e}']_\times H\mathbf{x} = F\mathbf{x}$

$$\boxed{\mathbf{x}'^T F\mathbf{x} = 0}$$

F

- 3x3, Rank 2, det(F)=0
- Linear sol. – 8 corr. Points (unique)
- Nonlinear sol. – 7 corr. points (3sol.)
- Very sensitive to noise & outliers

Epipolar lines: $\mathbf{l}' = F\mathbf{x} \quad \mathbf{l} = F^T\mathbf{x}'$

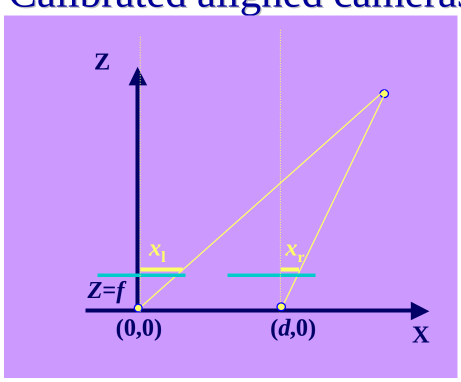
Epipoles: $F\mathbf{e} = 0 \quad F^T\mathbf{e}' = 0$

Projection matrices: $P = [I \mid \mathbf{0}]$

$P' = \left[[\mathbf{e}']_\times F + \mathbf{e}'\mathbf{v}^T \mid l\mathbf{e}'\right]$

Depth from stereo

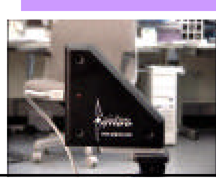
- Calibrated aligned cameras



Disparity d

$$Z = \frac{f}{x_l} X = \frac{f}{x_r}(X - d)$$

$$Z = \frac{df}{x_l - x_r}$$



Trinocular Vision System

(Point Grey Research)

Application: depth based reprojection

3D warping, McMillan

Plenoptic modeling, McMillan & Bishop

Application: depth based reprojection

Layer depth images, Shade et al.

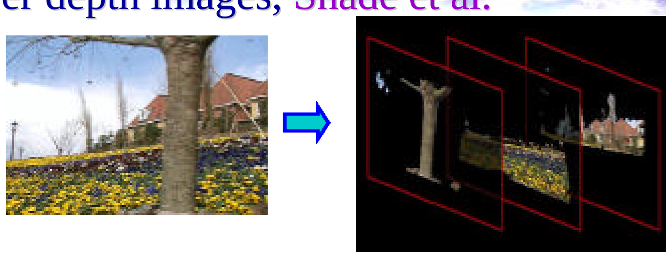


Image based objects, Oliveira & Bishop

3,4,N view geometry

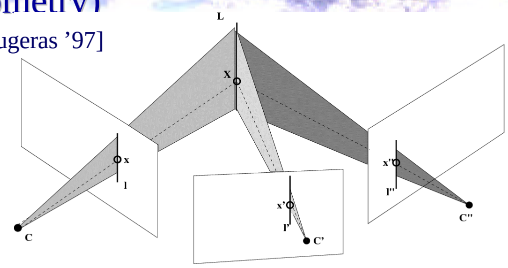
- Trifocal tensor (3 view geometry)

  [Hartley ’97][Torr & Zisserman ’97][Faugeras ’97]

  $\mathbf{T}:[T_1,T_2,T_3]$ 3x3x3 tensor;
  27 params. (18 indep.)

  $\mathbf{l}'[T1,T2,T3]\mathbf{l}''=\mathbf{l}^T$ lines

  $[\mathbf{x}']_\times(\sum_i \mathbf{x}'^i T_i)[\mathbf{x}'']_\times = 0$ points



- Quadrifocal tensor (4 view geometry) [Triggs ’95]
- Multiview tensors [Hartley’95][Hayden ‘98]

There is no additional constraint between more than 4 images. All the constraints can be expressed using F,triliear tensor or quadrifocal tensor.

Minimal cases

- 2 images (epipolar geometry,F):
  - 8 points – linear solution
  - 7 points – nonlinear solution (3 solutions)
- 3 images (trifocal tensor)
  - 7 points – linear solution
  - 6 points – 3 solutions
- 4 images (quadrifocal tensor)
  - 6 points – linear solution

N-view geometry
Affine factorization

[Tomasi &Kanade '92]

- Affine camera

$P_\infty = [M \mid \mathbf{t}]$ M 2x3 matrix; t 2D vector

- Projection $\begin{pmatrix} x \\ y \end{pmatrix} = M \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t}$

- *n* points, *m* views: measurement matrix

$$W = \begin{bmatrix} \mathbf{x}_1^1 & \dots & \mathbf{x}_n^1 \\ \vdots & \ddots & \vdots \\ \mathbf{x}_1^m & \dots & \mathbf{x}_n^m \end{bmatrix} = \begin{bmatrix} M^1 \\ \vdots \\ M^m \end{bmatrix} [\mathbf{X}_1 \quad \cdots \quad \mathbf{X}_n]$$

W: Rank 3

$$W = UDV^T$$
$$\hat{W} = \boxed{U_{2m\times 3} D_{3\times 3}} \boxed{V_{n\times 3}^T} = \hat{M}\hat{X}$$

Assuming isotropic zero-mean Gaussian noise, factorization achieves ML affine reconstruction.

Weak perspective factorization

[D. Weinshall]

- Weak perspective camera $M = \begin{bmatrix} s\mathbf{i} \\ s\mathbf{j} \end{bmatrix}$

- Affine ambiguity $\hat{W} = \hat{M}QQ^{-1}\hat{X} = (\hat{M}Q)(Q^{-1}\hat{X})$

- Metric constraints

$$s\hat{\mathbf{i}}^T QQ^T s\hat{\mathbf{i}} = s\hat{\mathbf{j}}^T QQ^T s\hat{\mathbf{j}} = s$$
$$s\hat{\mathbf{i}}^T QQ^T s\hat{\mathbf{j}} = 0$$

Extract motion parameters

- Eliminate scale
- Compute direction of camera axis k = i x j
- parameterize rotation with Euler angles

Projective factorization

[Sturm & Triggs'96][Heyden '97]

- Measurement matrix

$$W = \begin{bmatrix} l_1^1\mathbf{x}_1^1 & \dots & l_n^1\mathbf{x}_n^1 \\ \vdots & \ddots & \vdots \\ l_1^m\mathbf{x}_1^m & \dots & l_n^m\mathbf{x}_n^m \end{bmatrix} = \begin{bmatrix} P^1 \\ \vdots \\ P^m \end{bmatrix} [\mathbf{X}_1 \;\; \cdots \;\; \mathbf{X}_n]$$

3mxn matrix

Rank 4

- Known projective depth l_j^i

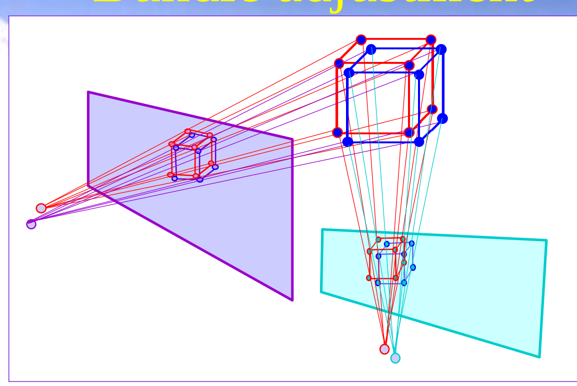
$$W = UDV^T$$

$$\hat{W} = U_{2m\times 4} D_{4\times 4} V_{n\times 4}^T = \hat{P}\hat{X}$$

  – Projective ambiguity

- Iterative algorithm
  – Reconstruct with $l_j^i = 1$
  – Reestimate depth l_j^i and iterate

Bundle adjustment



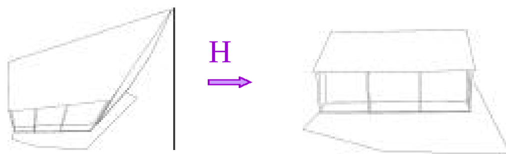
- Refine structure X_j and motion P^i
- Minimize geometric error
- ML solution, assuming noise is Gaussian
- Tolerant to missing data

$$\min \sum_{i,j} d(\hat{P}^i \hat{\mathbf{X}}_j, \mathbf{x}_j^i)^2$$

Projective ambiguity

Given an uncalibrated image sequence with corresponding point it is possible to reconstruct the object up to an unknown projective transformation

- Autocalibration (self-calibration): Determine a projective transformation H that upgrades the projective reconstruction to a metric one.
- This homography transforms the absolute conic (absolute dual quadric) in their canonical configurations.



Conics

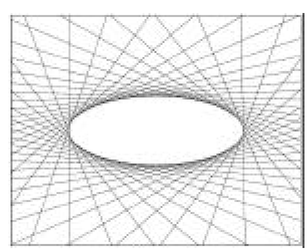
- Conic:
  - Euclidean geometry: hyperbola, ellipse, parabola & degenerate
  - Projective geometry: equivalent under projective transform
  - Defined by 5 points

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

$$\mathbf{x}^T C \mathbf{x} = 0$$

$$C = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}$$

- Tangent $\mathbf{l} = C\mathbf{x}$
- Dual conic C* $\mathbf{l}^T C^* \mathbf{l} = 0$



Quadrics

Quadrics: Q

4x4 symmetric matrix

9 DOF (defined by 9 points in general pose) $\mathbf{X}^T Q \mathbf{X} = 0$

•Dual: Q*

Planes tangent to the quadric $\text{ð}^T Q^* \text{ð} = 0$

The absolute conic

- Absolute conic Ω_∞ is a imaginary circle on ð_∞
- The absolute dual quadric (rim quadric) Ω^*_∞
- In a metric frame

$$\text{ð}_\infty = (0,0,0,1)$$

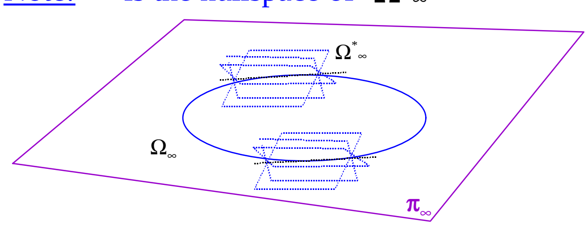
$$\Omega_\infty \quad \left.\begin{matrix} x_1^2 + x_2^2 + x_3^2 \\ x_4 \end{matrix}\right\} = 0$$

On ð_∞: $(x_1, x_2, x_3) I (x_1, x_2, x_3)^T = 0$

$$\Omega^*_\infty = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}$$

$$p^T \Omega^*_\infty p = 0$$

Note: is the nullspace of Ω^*_∞



Ω_∞ Fixed under similarity transf.

Self-calibration

- Theoretically formulated by [Faugeras '92]
- 2 basic approaches
  - Stratified: recover ð$_\infty$ Ω_∞
  - Direct: recover Ω^*_∞ [Triggs'97]
- Constraints:
  - Camera internal constraints
    - Constant parameters [Hartley'94][Mohr'93]
    - Known skew and aspect ratio [Hayden&Åström'98][Pollefeys'98]
  - Scene constraints (angles, ratios of length)
- Choice of H:

  Knowing camera K and ð$_\infty$

$$H = \begin{bmatrix} K & \mathbf{0} \\ -\mathbf{p}^T K & 1 \end{bmatrix}, \quad p_\infty = (\mathbf{p}^T, 1)^T$$

Self calibration based on the IADC

- Calibrated camera
  - Dual absolute quadric (DAC) $\tilde{I} = diag(1,1,1,0)$
  - Dual image of the absolute conic (DIAC) $w^* = KK^T$
- Projective camera
  - DAC $Q^*_\infty = H\tilde{I}H^T$
  - DIAC $w^{*i} = P^i Q^*_\infty P^{iT} = K_i K_i^T$
- Autocalibration
  - Determine Ω^*_∞ based on constraints on w^{*i}
  - Decompose $Q^*_\infty = H\tilde{I}H^T$

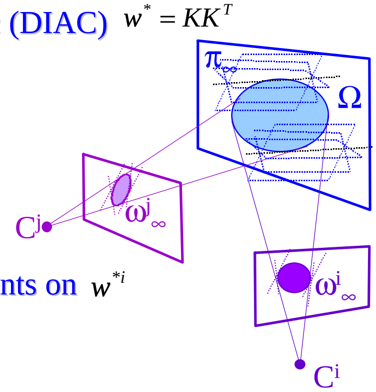
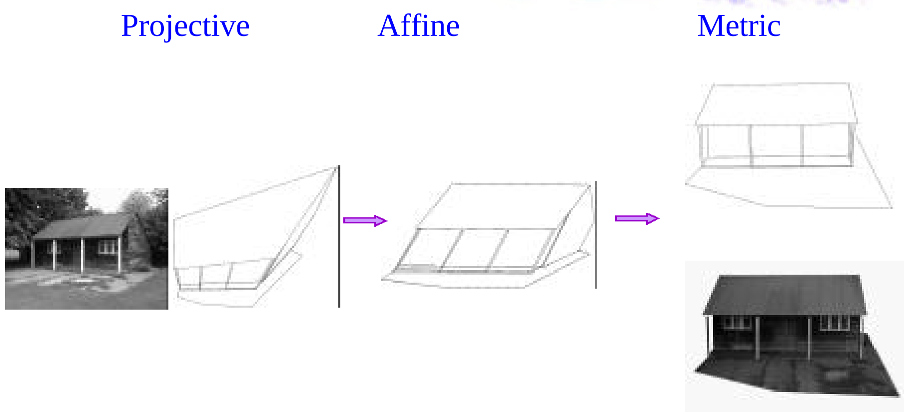


Illustration of self-calibration



Degenerate configurations

- Pure translation: affine transformation (5 DOF)
- Pure rotation: arbitrary pose for ð$_{\infty}$ (3 DOF)
- Planar motion: scaling axis perpendicular to plane (1DOF)
- Orbital motion: projective distortion along rotation axis (2DOF)

Not unique solution !

A complete modeling system

Sequence of frames → scene structure

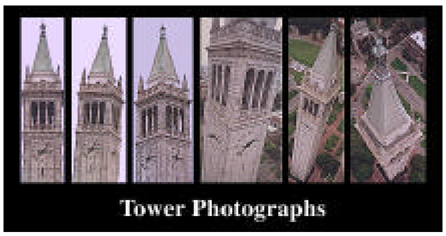
1. Get corresponding points (tracking).
2. 2,3 view geometry: compute F,T between consecutive frames (recompute correspondences).
3. Initial reconstruction: get an initial structure from a subsequence with big baseline (trilinear tensor, factorization …) and bind more frames/points using resection/intersection.
4. Bundle adjustment.
5. Self-calibration.

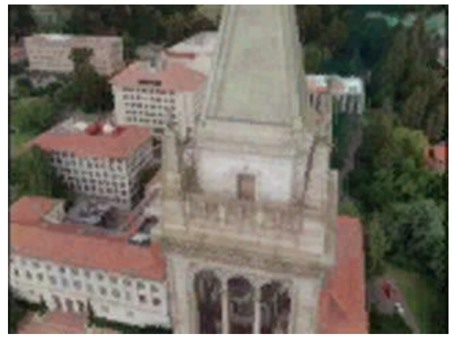
Examples – modeling with dynamic texture

Cobzas,Yerex,Jagersand

Examples: geometric modeling

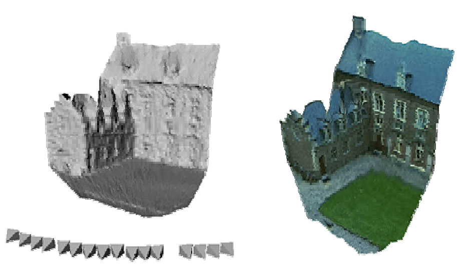
Debevec,Camillo: Façade





Examples: geometric modeling

Pollefeys: Arenberg Castle



Examples: geometric modeling

INRIA –VISIRE project

Examples: geometric modeling

CIP Prague –

Projective Reconstruction Based on Cake Configuration