Multiple View Geometry

Andrew Zisserman, University of Oxford
az@robots.ox.ac.uk
Marc Pollefeys, University of North Carolina at Chapel Hill
marc@cs.unc.edu

Part 3

Three and more views (45 min + questions)

- trifocal tensor
- factorization
- bundle adjustment
- auto-calibration

Three View Geometry

Cameras $\mathrm{P}, \mathrm{P}', \mathrm{P}''$ such that

$$\mathbf{x} = \mathrm{P}\mathbf{X} \qquad \mathbf{x}' = \mathrm{P}'\mathbf{X} \qquad \mathbf{x}'' = \mathrm{P}''\mathbf{X}$$

Main new result: The Trifocal Tensor

- Defined for three views.
- Plays a similar rôle to F for two views.
- Unlike F, trifocal tensor also relates lines in three views.

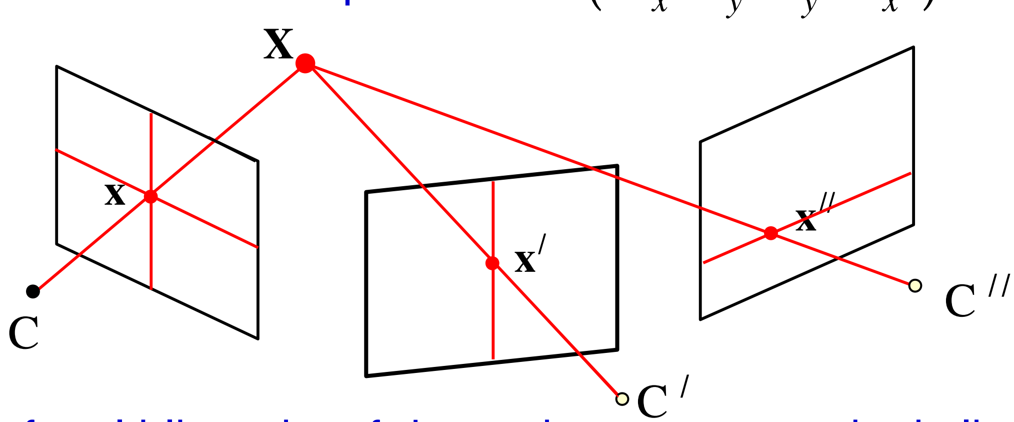
lines: $\quad l^{T} x = 0 \qquad l = [x]_{x}\, x' \qquad x = [l]_{x}\, l'$

planes: $\quad \Pi^{T} X = 0 \qquad \Pi = \mathbf{P}^{T} l$

Tri-linear relation

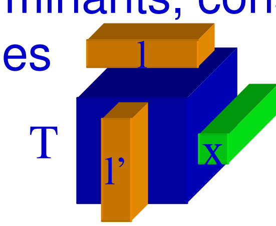
back-projected lines passing through corresponding points should all intersect $\quad \Pi_x = \mathbf{P}^{T} l_x \qquad l_x = [v_x]_{\times} x$

non-trivial constraint for 4 planes $\quad \det(\Pi_x \Pi_y \Pi_{y'} \Pi_{x''}) = 0$



because of multi-linearity of determinants, constraint is linear in coordinates of points and/or lines

3^3 coefficients: Trifocal Tensor



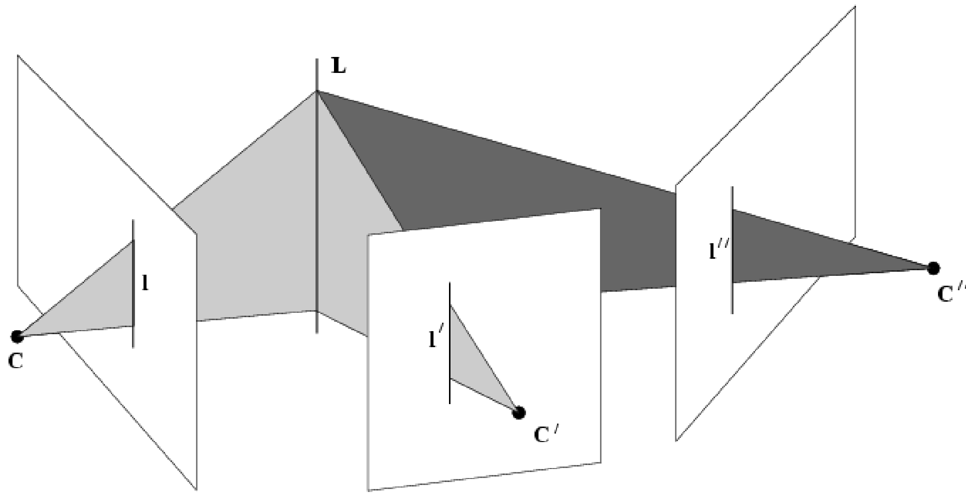
$$\mathbf{l}'^{\top}\left(\sum_i x^i \mathrm{T}_i\right)\mathbf{l}''^{\top} = 0$$

Trifocal tensor

Tri-linear relation can be represented efficiently as tensor
Easy to express line and point coincidence relations

(i) Line–line–line correspondence

$$\mathbf{l}'^{\top}[\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\mathbf{l}'' = \mathbf{l}^{\top} \quad \text{or} \quad \left(\mathbf{l}'^{\top}[\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\mathbf{l}''\right)[\mathbf{l}]_{\times} = \mathbf{0}^{\top}$$



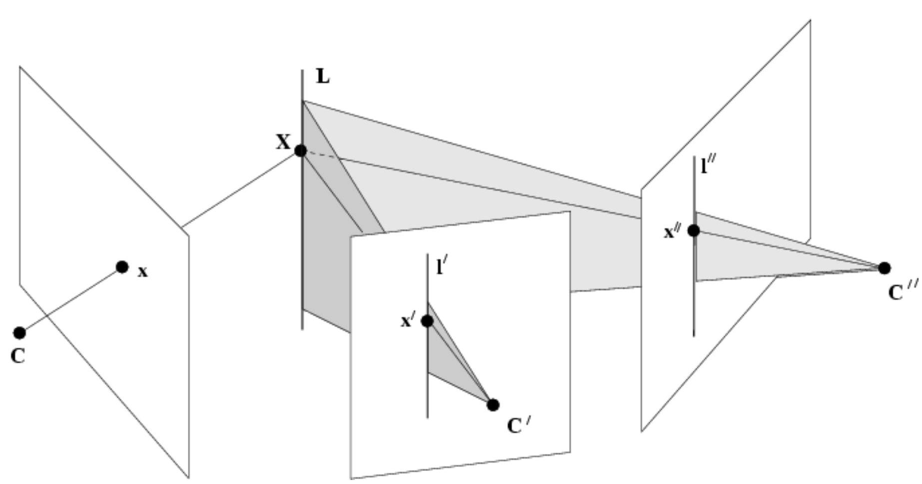
allows line transfer

Trifocal tensor

Tri-linear relation can be represented efficiently as tensor
Easy to express line and point coincidence relations

(ii) Point–line–line correspondence

$$\mathbf{l}'^{\top}\Big(\sum_i x^i \mathrm{T}_i\Big)\mathbf{l}'' = 0 \quad \text{for a correspondence } \mathbf{x} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$$



Trifocal tensor

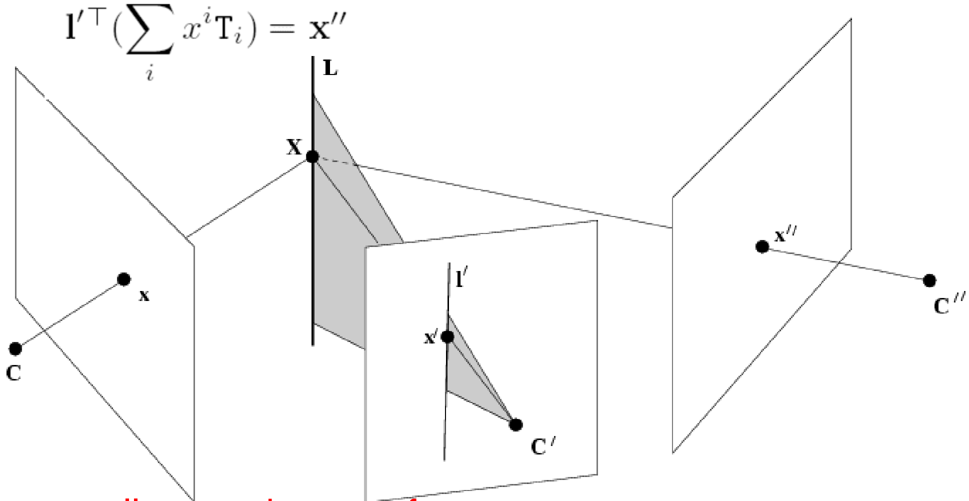
Tri-linear relation can be represented efficiently as tensor

Easy to express line and point coincidence relations

(iii) Point–line–point correspondence

$$\mathbf{l}'^{\top}(\sum x^i \mathrm{T}_i)[\mathbf{x}'']_\times = \mathbf{0}^\top \quad \text{for a correspondence } \mathbf{x} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{x}''$$

$$\mathbf{l}'^{\top}(\sum_i x^i \mathrm{T}_i) = \mathbf{x}''$$



allows point transfer

Application: image warping T(1,2,N)

(Avidan and Shashua `97)

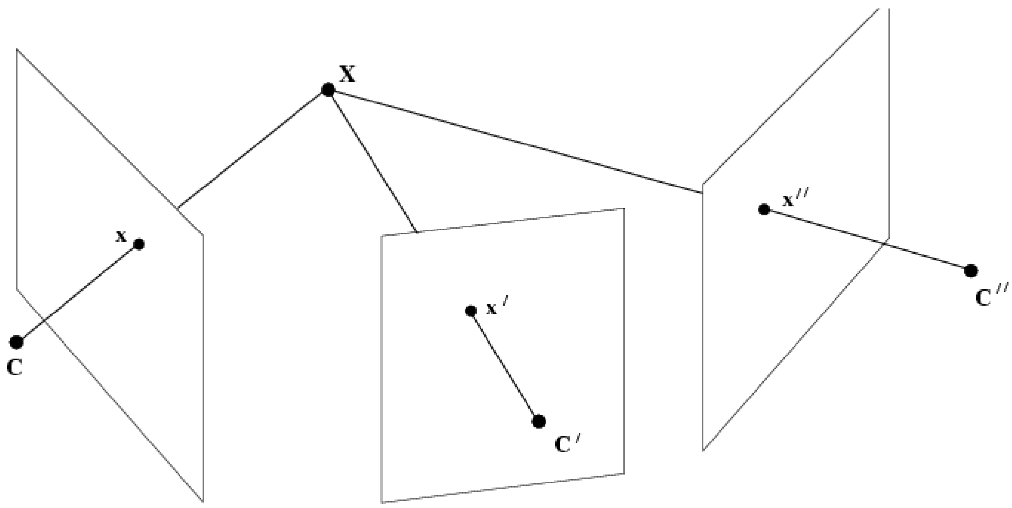
Trifocal tensor

Tri-linear relation can be represented efficiently as tensor

Easy to express line and point coincidence relations

(v) Point–point–point correspondence

$$[\mathbf{x}']_\times \left(\sum_i x^i \mathrm{T}_i\right)[\mathbf{x}'']_\times = 0_{3\times 3}$$



More views …

Problem statement – Structure and Motion Estimation

Given: n matching image points $\mathbf{x}^i{}_j$ over m views

Find: the cameras P^i and the 3D points $\mathbf{X}_j$ such that $\mathbf{x}^i{}_j = P^i\,\mathbf{X}_j$

$$\min_{\mathrm{P}^i\,\mathbf{X}_j} \sum_{j \in \text{points}} \sum_{i \in \text{views}} d\left(\mathbf{x}_j^i,\ \mathrm{P}^i\mathbf{X}_j\right)^2$$

minimizing reprojection error corresponds to
a maximum likelihood estimation (MLE) under
the assumption of zero mean Gaussian noise

Factorization

Factorization

Factorise observations in structure of the scene and motion/calibration of the camera

Use **all points** in **all images** at the same time

Affine factorisation

Projective factorisation

Affine camera

The affine projection equations are

$$\begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} = \begin{bmatrix} \overline{P}_i^{\,x} \\ \overline{P}_i^{\,y} \end{bmatrix} \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

how to find the origin? or for that matter a 3D reference point?

affine projection preserves center of gravity

$$\tilde{x}_{ij} = x_{ij} - \sum_i x_{ij} \qquad \tilde{y}_{ij} = y_{ij} - \sum_i y_{ij}$$

The orthographic projection equations are

$$\overline{\mathbf{x}}_{ij} = \overline{\mathbf{P}}_i \overline{\mathrm{X}}_j, i = 1,\ldots,m,\ j = 1,\ldots,n$$

where

$$\overline{\mathbf{x}}_{ij} = \begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix}, \overline{\mathbf{P}}_i = \begin{bmatrix} \overline{P}_i^x \\ \overline{P}_i^y \end{bmatrix}, \overline{\mathrm{X}}_j = \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix}$$

All equations can be collected for all i and j

$$\overline{\mathbf{x}} = \overline{\mathbf{P}\mathbf{X}}$$

where

$$\overline{\mathbf{x}} = \begin{bmatrix} \overline{\mathrm{x}}_{11} & \overline{\mathrm{x}}_{12} & \cdots & \overline{\mathrm{x}}_{1n} \\ \overline{\mathrm{x}}_{21} & \overline{\mathrm{x}}_{22} & \cdots & \overline{\mathrm{x}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\mathrm{x}}_{m1} & \overline{\mathrm{x}}_{m2} & \cdots & \overline{\mathrm{x}}_{mn} \end{bmatrix}, \quad \overline{\mathbf{P}} = \begin{bmatrix} \overline{\mathbf{P}}_1 \\ \overline{\mathbf{P}}_2 \\ \vdots \\ \overline{\mathbf{P}}_m \end{bmatrix}, \overline{\mathbf{X}} = [\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n]$$

Note that $\mathbf{P}$ and $\mathbf{X}$ are resp. 2mx3 and 3xn matrices and therefore the rank of $\mathbf{x}$ is at most 3

Orthographic factorization

Factorize $\mathbf{m}$ through singular value decomposition

$$\overline{\mathbf{x}} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$$

An affine reconstruction is obtained as follows

$$\tilde{\mathbf{P}} = \mathbf{U}, \tilde{\mathbf{X}} = \Sigma\mathbf{V}^{\mathrm{T}}$$

Closest rank-3 approximation yields MLE!

$$\min \left\| \begin{bmatrix} \overline{\mathrm{x}}_{11} & \overline{\mathrm{x}}_{12} & \cdots & \overline{\mathrm{x}}_{1n} \\ \overline{\mathrm{x}}_{21} & \overline{\mathrm{x}}_{22} & \cdots & \overline{\mathrm{x}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\mathrm{x}}_{m1} & \overline{\mathrm{x}}_{m2} & \cdots & \overline{\mathrm{x}}_{mn} \end{bmatrix} - \begin{bmatrix} \overline{\mathbf{P}}_1 \\ \overline{\mathbf{P}}_2 \\ \vdots \\ \overline{\mathbf{P}}_m \end{bmatrix} [\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n] \right\|$$

Examples

Tomasi Kanade'92,
Poelman & Kanade'94

Examples

Tomasi Kanade'92,
Poelman & Kanade'94

Perspective factorization

The camera equations

$$\lambda_{ij}\, \mathrm{x}_{ij} = \mathbf{P}_i \mathrm{X}_j, i = 1,\ldots,m,\ j = 1,\ldots,n$$

for a fixed image i can be written in matrix form as

$$\mathbf{x}_i \Lambda_i = \mathbf{P}_i \mathbf{X}$$

where $\mathbf{x}_i = [\mathrm{x}_{i1}, \mathrm{x}_{i2}, \ldots, \mathrm{x}_{im}],\ \mathbf{X} = [\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_m]$

$$\Lambda_i = \mathrm{diag}(\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{im})$$

Perspective factorization

All equations can be collected for all i as

$$\mathbf{x} = \mathbf{P}\mathbf{X}$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \Lambda_1 \\ \mathbf{x}_2 \Lambda_2 \\ \ldots \\ \mathbf{x}_n \Lambda_n \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \ldots \\ \mathbf{P}_m \end{bmatrix}$$

In these formulas $\mathbf{x}$ are known, but Λ_i,$\mathbf{P}$ and $\mathbf{X}$ are unknown

Observe that $\mathbf{PX}$ is a product of a 3mx4 matrix and a 4xn matrix, i.e. it is a rank-4 matrix

Perspective factorization algorithm

Assume that Λ_i are known, then $\mathbf{PX}$ is known.

Use the singular value decomposition

$$\mathbf{PX} = U\Sigma V^T$$

In the noise-free case

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

and a reconstruction can be obtained by setting:

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

$\mathbf{X}$ = the first four rows of V.

Iterative perspective factorization

When Λ_i are unknown the following algorithm can be used:

1. Set $\lambda_{ij}=1$ (affine approximation).

2. Factorize $\mathbf{PX}$ and obtain an estimate of $\mathbf{P}$ and $\mathbf{X}$.

   If σ_5 is sufficiently small then STOP.

3. Use $\mathbf{x}$, $\mathbf{P}$ and $\mathbf{X}$ to estimate Λ_i from the camera equations (linearly) $\mathbf{x}_i \Lambda_i = \mathbf{P}_i \mathbf{X}$

4. Goto 2.

In general the algorithm minimizes the *proximity measure* $P(\Lambda, \mathbf{P}, \mathbf{X}) = \sigma_5$

Note that structure and motion recovered up to an arbitrary projective transformation

Further Factorization work

Factorization with uncertainty (Irani & Anandan, IJCV'02)

Multi-body factorization (Costeira and Kanade '94)

Factorization for dynamic scenes (Bregler et al. 2000, Brand 2001)

Bundle adjustment

global refinement of

recovered structure and motion

Bundle adjustment - refining structure and motion

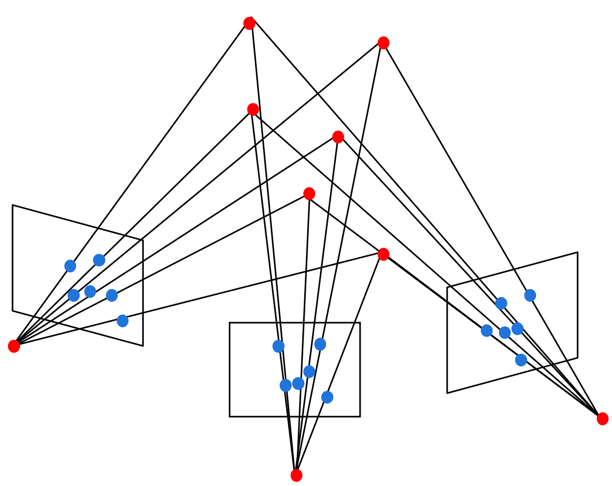
Minimize reprojection error

$$\min_{\hat{P}_i,\hat{X}_j} \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(x_{ij}, \hat{P}_i\hat{X}_j\right)^2$$

- Maximum Likelyhood Estimation
  (if error zero-mean Gaussian noise)
- Huge problem but can be solved efficiently
  (Bundle adjustment)

Bundle adjustment

Developed in photogrammetry in 50´s



Non-linear least squares

Linear approximation of residual $e_0 - J\Delta$

allows quadratic approximation of sum-of-squares

$$\left(e_0 - J\Delta\right)^T \left(e_0 - J\Delta\right)$$

Minimization corresponds to finding zeros of derivative

$$2J^T J\Delta - 2J^T e_0 = 0$$

$$\Rightarrow \Delta = \underbrace{\left(J^T J\right)^{-1}}_{\mathbf{N}} J^T e_0$$

Levenberg-Marquardt: extra term to deal with singular **N**
(decrease/increase λ if success/failure to descent)

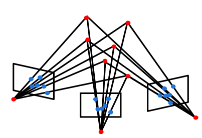
(extra term = descent term)

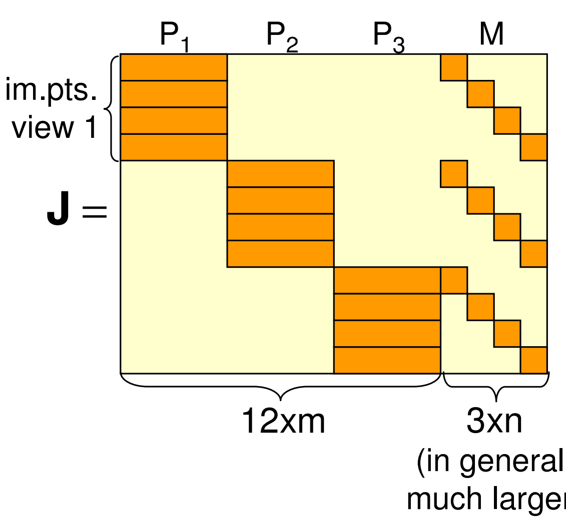
$$N' = J^T J + \lambda \operatorname{diag}(J^T J)$$

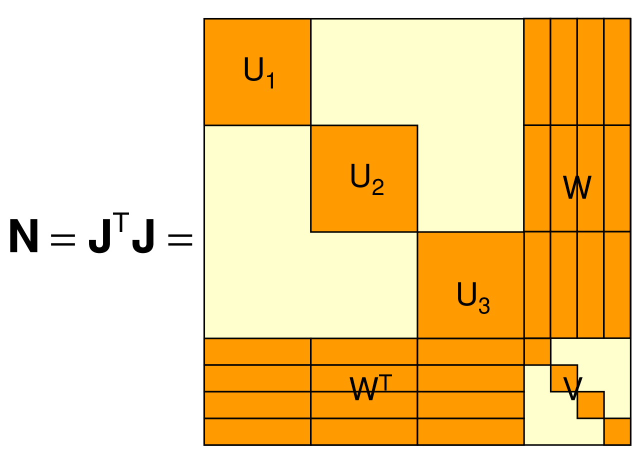
Bundle adjustment

Jacobian of $\sum_{i=1}^{m} \sum_{j=1}^{n} D\left(\mathrm{m}_{ij}, \hat{\mathrm{P}}_i\left(\hat{\mathrm{M}}_j\right)\right)^2$ has sparse block structure

cameras independent of other cameras,
points independent of other points





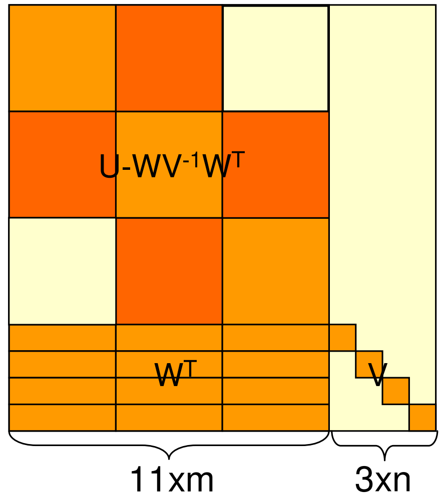


Needed for non-linear minimization

Eliminate dependence of camera/motion parameters on structure parameters

Note in general 3n >> 11m

$$\begin{bmatrix} \mathbf{I} & -\mathbf{W}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \times \mathbf{N} =$$



Allows much more efficient computations

e.g. 100 views,10000 points,

solve ±1000x1000, not ±30000x30000

Often still band diagonal

use sparse linear algebra algorithms

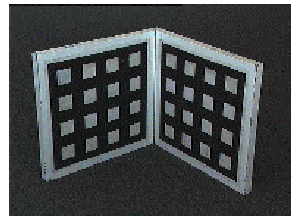
Self-calibration

from projective to metric

Self-calibration: Motivation

Avoid explicit calibration procedure

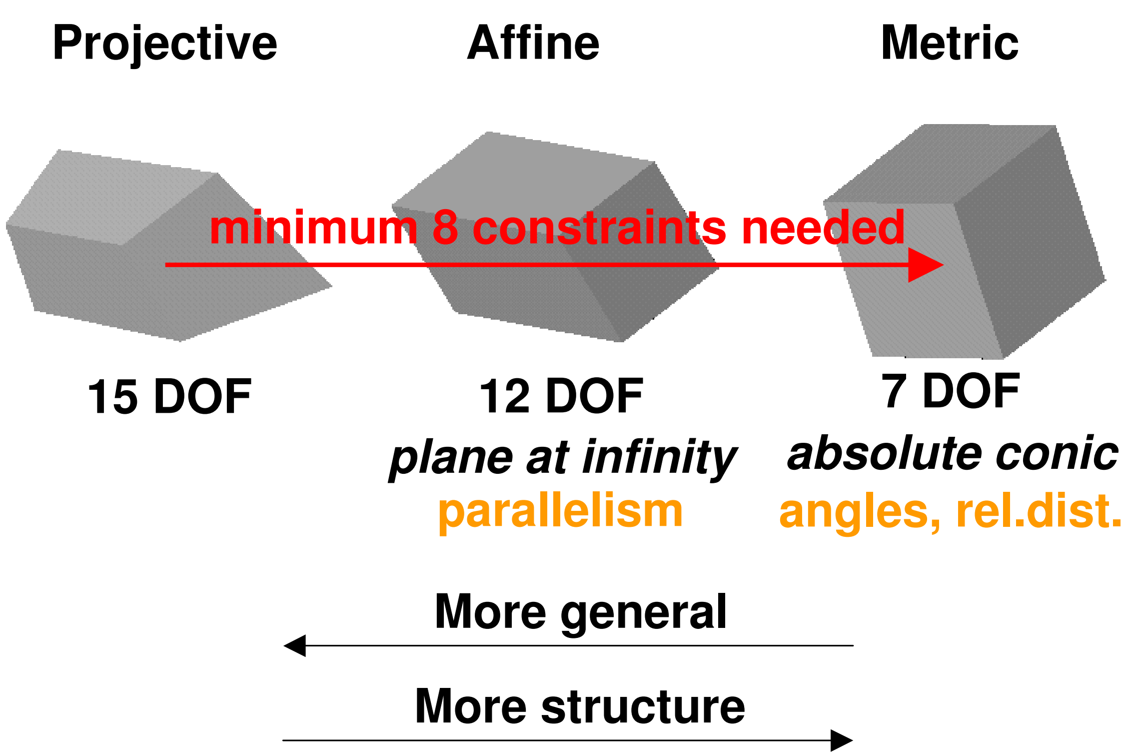
- Complex procedure
- Need for calibration object
- Need to maintain calibration



Allow flexible acquisition

- No prior calibration necessary
- Possibility to vary intrinsics
- Use archive footage

Stratification of geometry



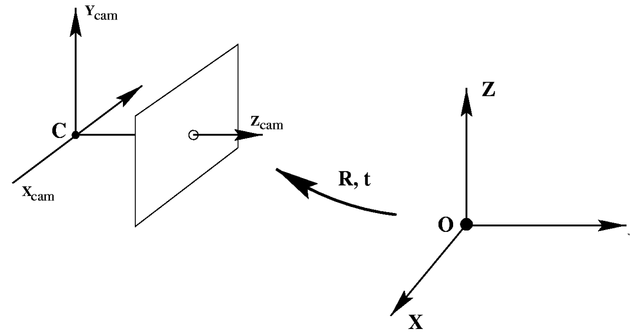
$$P = K [R \mid t]$$

internal calibration

rotation

translation

from world to camera coordinate frame



K is a 3×3 upper triangular matrix, called the camera calibration matrix:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

There are five parameters:

1. The scaling in the image x and y directions, α_x and α_y.

2. The principal point (x_0, y_0), which is the point where the optic axis intersects the image plane.

3. The skew s, which is the angle between the image x and y axes.

Self-calibration: conceptual algorithm

Given projective structure and motion $\{\mathbf{P}_j, \mathbf{X}_i\}$, then the metric structure and motion can be obtained as $\{\mathbf{P}_j\mathbf{T}^{-1}, \mathbf{T}\mathbf{X}_i\}$, with

$$\mathbf{T} = \arg\min_{\mathbf{T}} C\left(K\left(\mathbf{P}_1\mathbf{T}^{-1}\right), K\left(\mathbf{P}_2\mathbf{T}^{-1}\right), \ldots, K\left(\mathbf{P}_n\mathbf{T}^{-1}\right)\right)$$

$C\left(\mathbf{K}_1, \mathbf{K}_2, \ldots, \mathbf{K}_n\right)$ criterium expressing constraints

$K(\mathbf{P})$ function extracting intrinsics from projection matrix

Absolute Dual Quadric and Self-calibration

Eliminate extrinsics from equation

$$\mathbf{K}[\mathbf{R} \quad -\cancel{\mathbf{Rt}}] \rightarrow \mathbf{K}\cancel{\mathbf{R}\,\mathbf{R}^{\mathsf{T}}}\mathbf{K} \rightarrow \mathbf{K}\mathbf{K}^{\mathsf{T}}$$

Equivalent to projection of dual quadric

$$\mathbf{P}\Omega^*_\infty\mathbf{P}^{\mathsf{T}} \propto \mathbf{K}\mathbf{K}^{\mathsf{T}} \qquad \Omega^*_\infty = \mathrm{diag}(1110)$$

Abs.Dual Quadric also exists in projective world

$$\mathbf{K}\mathbf{K}^{\mathsf{T}} \propto \mathbf{P}\Omega^*_\infty\mathbf{P}^{\mathsf{T}} \propto (\mathbf{P}\mathbf{T}^{-1})(\mathbf{T}\Omega^*_\infty\mathbf{T}^{\mathsf{T}})(\mathbf{T}^{-\mathsf{T}}\mathbf{P}^{\mathsf{T}})$$
$$\propto \mathbf{P}'\Omega'^*_\infty\mathbf{P}'^{\mathsf{T}}$$

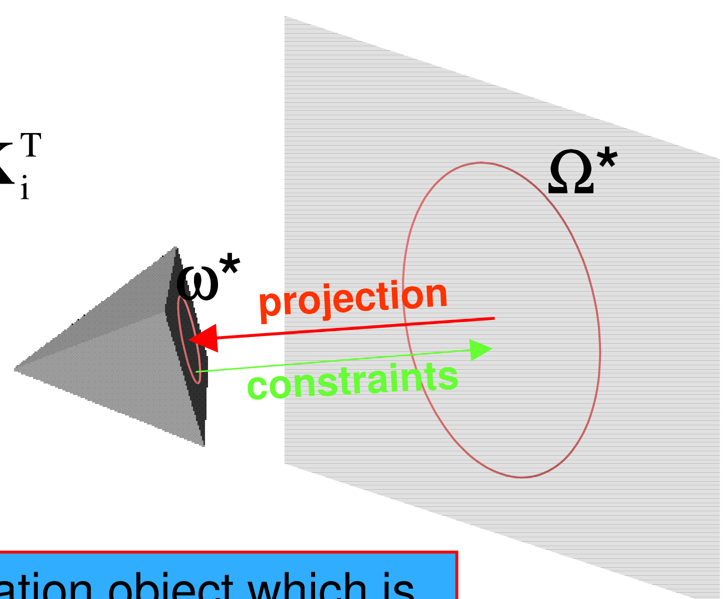
Transforming world so that $\Omega'^*_\infty \rightarrow \Omega^*_\infty$
reduces ambiguity to metric

Absolute Dual Quadric and Self-calibration

Projection equation:

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

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^*_∞

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

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 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
Focal length (& p.p. & Skew)	$\omega^*_{33} = \omega^*_{11}$	linear	m

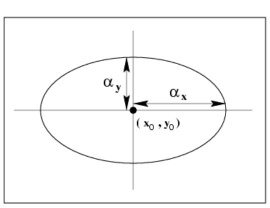
Linear algorithm

Assume everything known, except focal length

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

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

Yields 4 constraint per image



Note that rank-3 constraint is not enforced!

Linear algorithm revisited

Weighted linear equations

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

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

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

assumptions

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

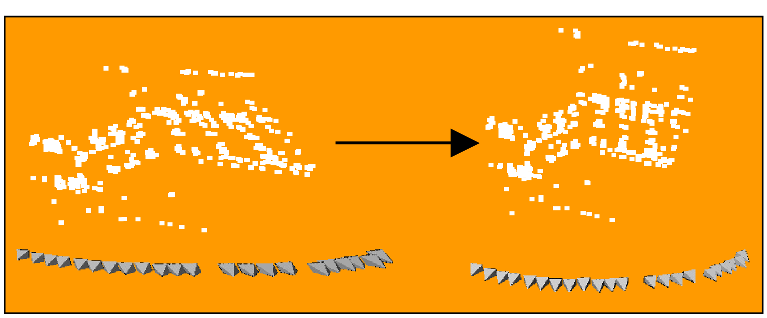
Projective to metric

Compute $\mathbf{T}$ from

$$\tilde{\mathbf{I}} = \mathbf{T}\Omega^*_\infty\mathbf{T}^\top \text{ or } \mathbf{T}^{-1}\tilde{\mathbf{I}}\mathbf{T}^{-\top} = \Omega^*_\infty \text{ with } \tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{I} & 0 \\ 0^\top & 0 \end{bmatrix}$$

using eigenvalue decomposition of Ω^*_∞
and then obtain metric reconstruction as

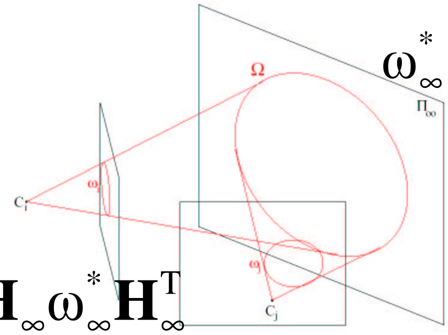
$$\mathbf{PT}^{-1} \text{ and } \mathbf{TX}$$



Alternatives: (Dual) image of absolute conic

Equivalent to Absolute Dual Quadric

$$\omega_\infty^* \propto \mathbf{P}\Omega_\infty^*\mathbf{P}^\mathsf{T}$$



$$\omega_\infty^* \propto \mathbf{H}_\infty \omega_\infty^* \mathbf{H}_\infty^\mathsf{T} \qquad (\mathbf{H}_\infty = \mathbf{H} + \mathrm{ea})$$

Practical when $\mathbf{H}_\infty$ can be computed first

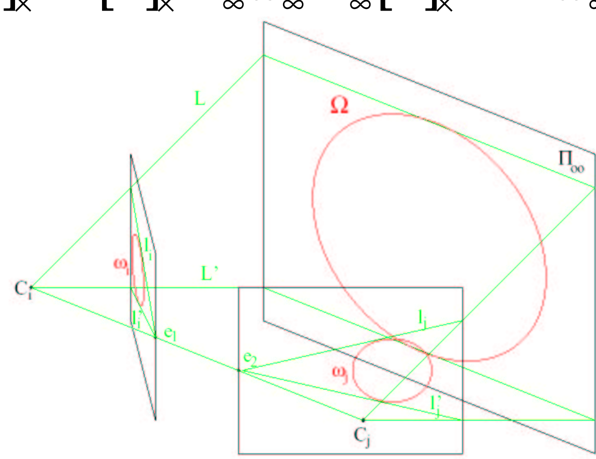
- Pure rotation
- Vanishing points, pure translations, modulus constraint, …

Kruppa equations

Limit equations to epipolar geometry

Only 2 independent equations per pair,
but independent of plane at infinity

$$[\mathrm{e}']_\times \omega_\infty^* [\mathrm{e}']_\times^\mathsf{T} \propto [\mathrm{e}']_\times \mathbf{H}_\infty \omega_\infty^* \mathbf{H}_\infty^\mathsf{T} [\mathrm{e}']_\times^\mathsf{T} \propto \mathbf{F}\omega_\infty^*\mathbf{F}^\mathsf{T}$$



Refinement

Metric bundle adjustment

$$\arg\min_{\mathbf{P}_k, \mathrm{X}_i} \sum_{k=1}^{m} \sum_{i=1}^{n} D\left(\mathrm{x}_{ki}, \mathbf{P}_k\left(\mathrm{X}_i\right)\right)^2$$

Enforce constraints or priors
on intrinsics during minimization
(this is „self-calibration" for photogrammetrist)

Part 4

Computing reconstructions from multiple views and applications

(45 min + questions)

- correspondences for sequences
- augmenting video sequences
- surface reconstruction from sequence

Camera motion computation from image sequences

Structure and motion recovery

Sequential approach

- Initialize motion from two images
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Initial projective camera motion

Choose P and P´compatible with F

$$\mathbf{P} = [\mathbf{I}_{3\times3} \quad 0_3]$$

$$\mathbf{P}' = [e'\times\mathbf{F} + e'a^{\mathrm{T}} \quad e']$$

(reference plane;arbitrary)

Reconstruction up to projective ambiguity

Same for more views?

$$\mathbf{P}'' \stackrel{?}{=} \cancel{[e''\times\mathbf{F}'' + e''a^{\mathrm{T}} \quad e'']}$$

different projective basis

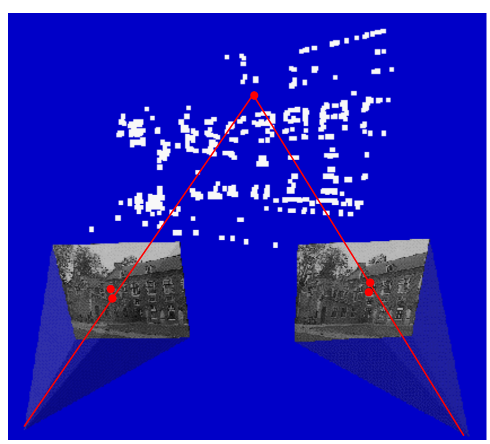
- Initialize motion
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Initializing projective structure

Reconstruct matches in projective frame by minimizing the reprojection error

$$D(x_1, P_1X)^2 + D(x_2, P_2X)^2$$

Non-iterative optimal solution



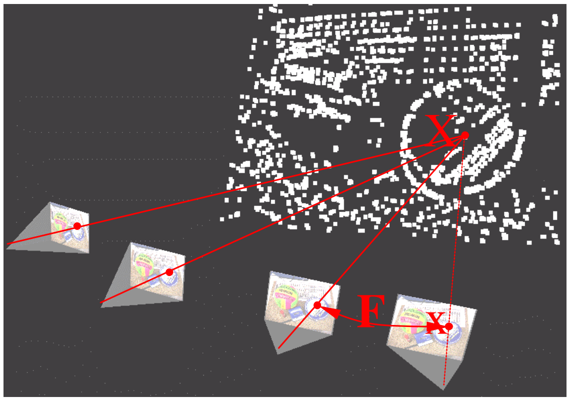
- Initialize motion
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Projective pose estimation

Infere 2D-3D matches from 2D-2D matches

Compute pose from $x \sim \mathbf{P}X$ (RANSAC,6pts)

$$\begin{bmatrix} X^\top & 0 & X^\top x \\ 0 & X^\top & X^\top y \end{bmatrix} \mathbf{p} = 0$$



Inliers:
$$\exists X \, \forall x_i \; D(\mathbf{P}_i X, x_i) < D_{\text{in}}$$

- Initialize motion
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Refining and extending structure

Refining structure

$$\frac{1}{P_3 \tilde{X}} \begin{bmatrix} P_3 x - P_1 \\ P_3 y - P_1 \end{bmatrix} X = 0$$

(Iterative linear)

Extending structure

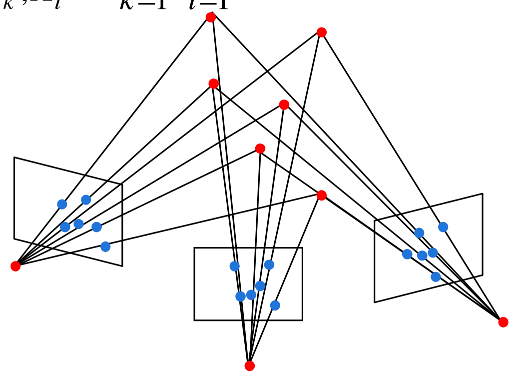
2-view triangulation

- Initialize motion
- Initialize structure
- For each additional view
  - Determine pose of camera
  - Refine and extend structure
- Refine structure and motion

Refining structure and motion

use bundle adjustment

$$\arg\min_{\mathbf{P}_k, \mathrm{X}_i} \sum_{k=1}^{m} \sum_{i=1}^{n} D\left(\mathrm{x}_{ki}, \mathbf{P}_k(\mathrm{X}_i)\right)^2$$



Also model radial distortion to avoid bias!

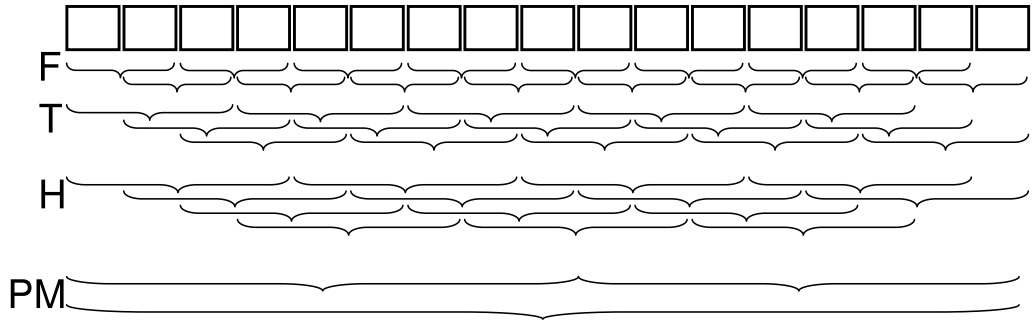
Hierarchical structure and motion recovery

Compute 2-view

Compute 3-view

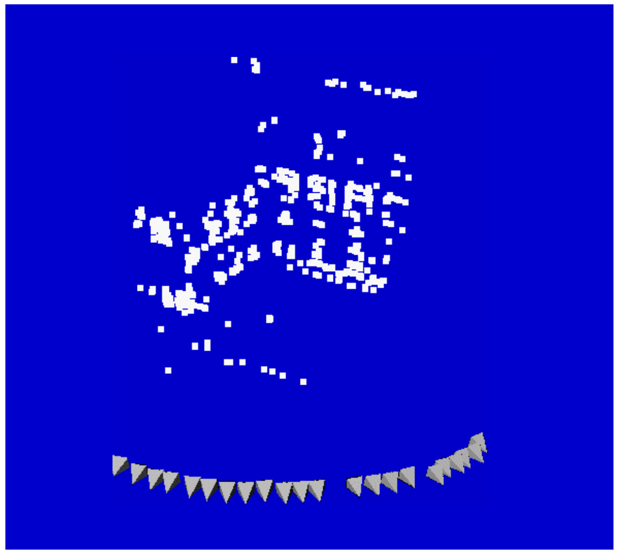
Stitch 3-view reconstructions

Merge and refine reconstruction



Metric structure and motion

use self-calibration



Note that a fundamental problem of the uncalibrated approach is
that it fails if a purely planar scene is observed (in one or more views)

(solution possible based on model selection)

Application I : Augmented reality

Surface reconstruction

from image sequences

Dense model reconstruction

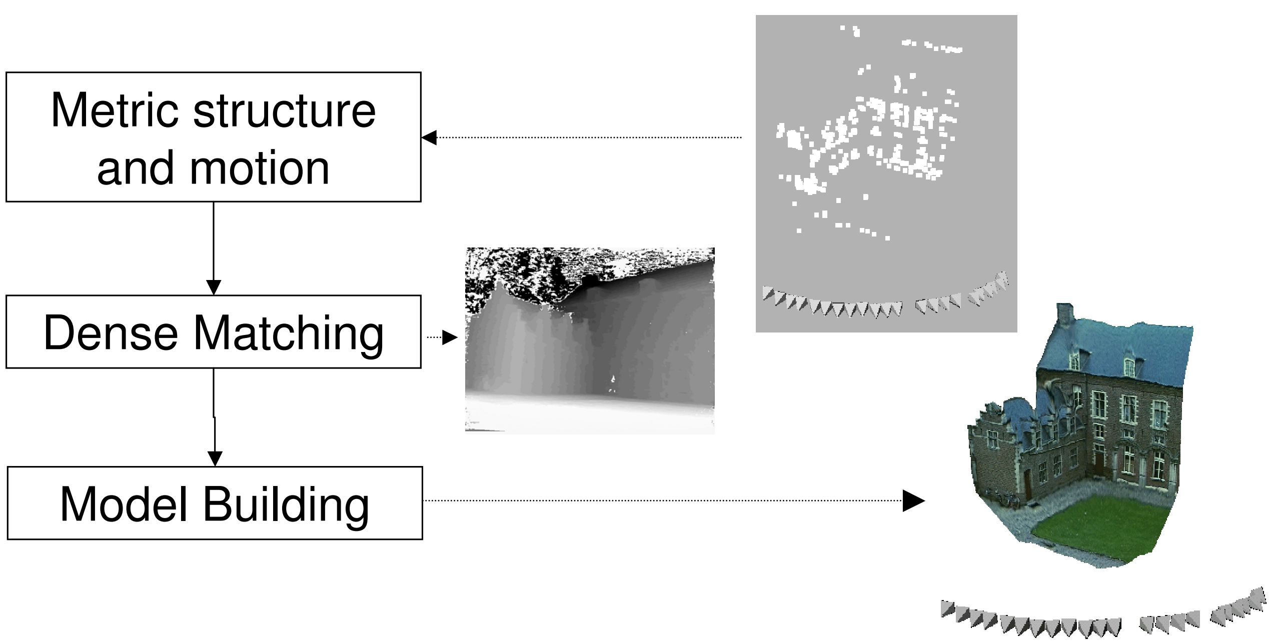


Image rectification

Resample image to simplify matching process

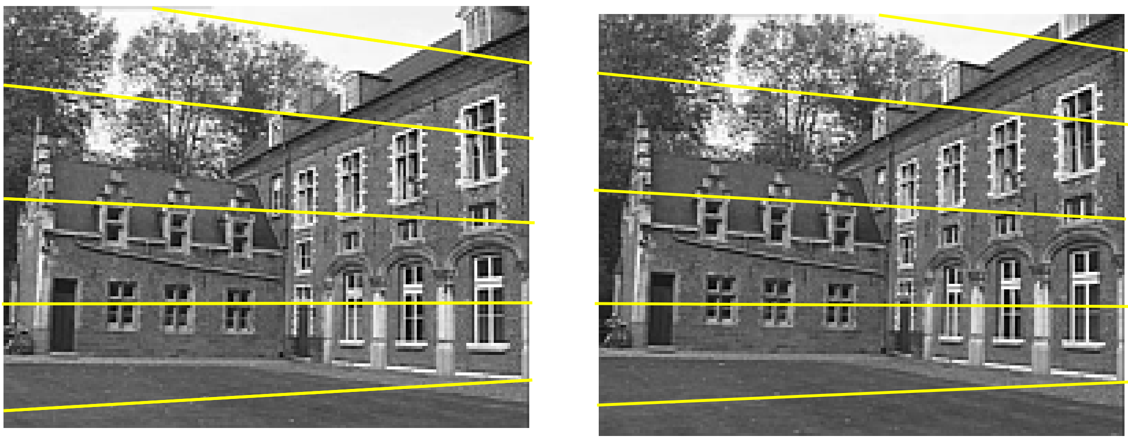
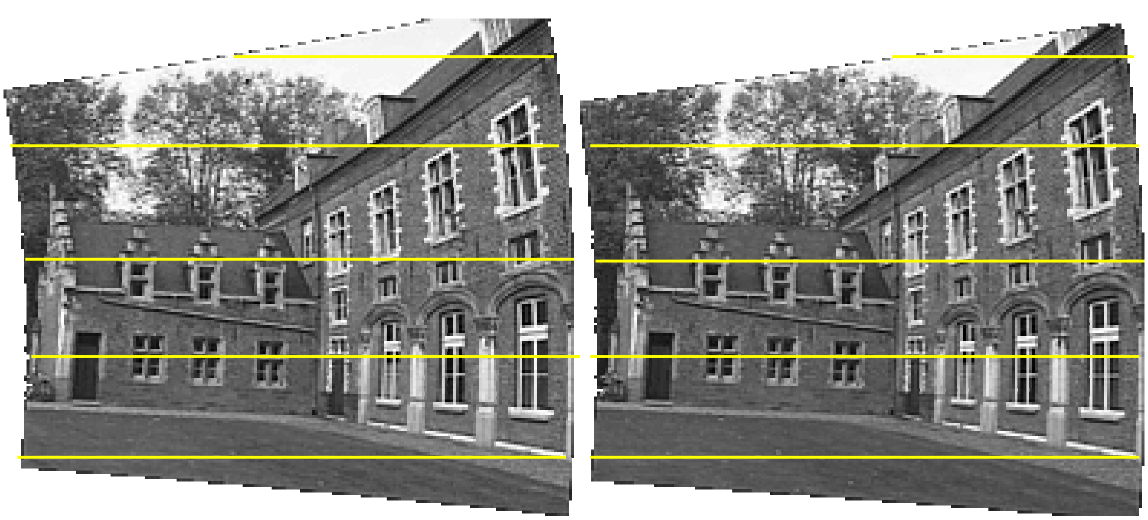


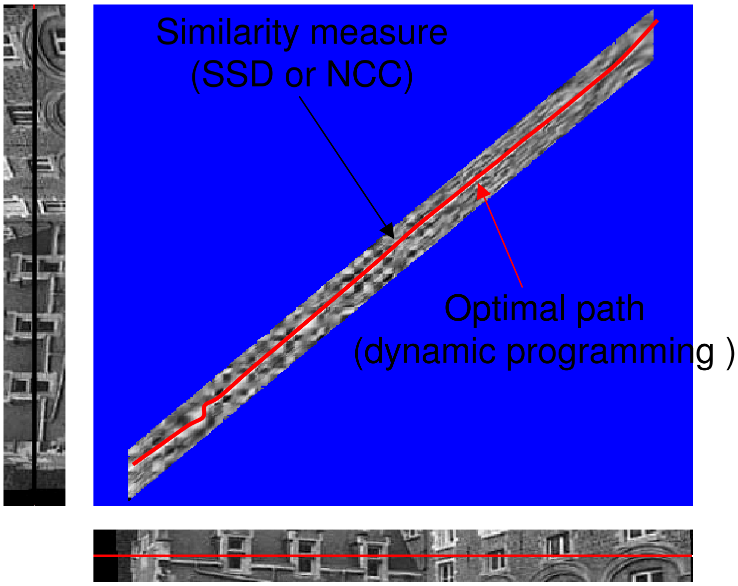
Image rectification

Resample image to simplify matching process



Approach that works for all relative motions

Stereo matching

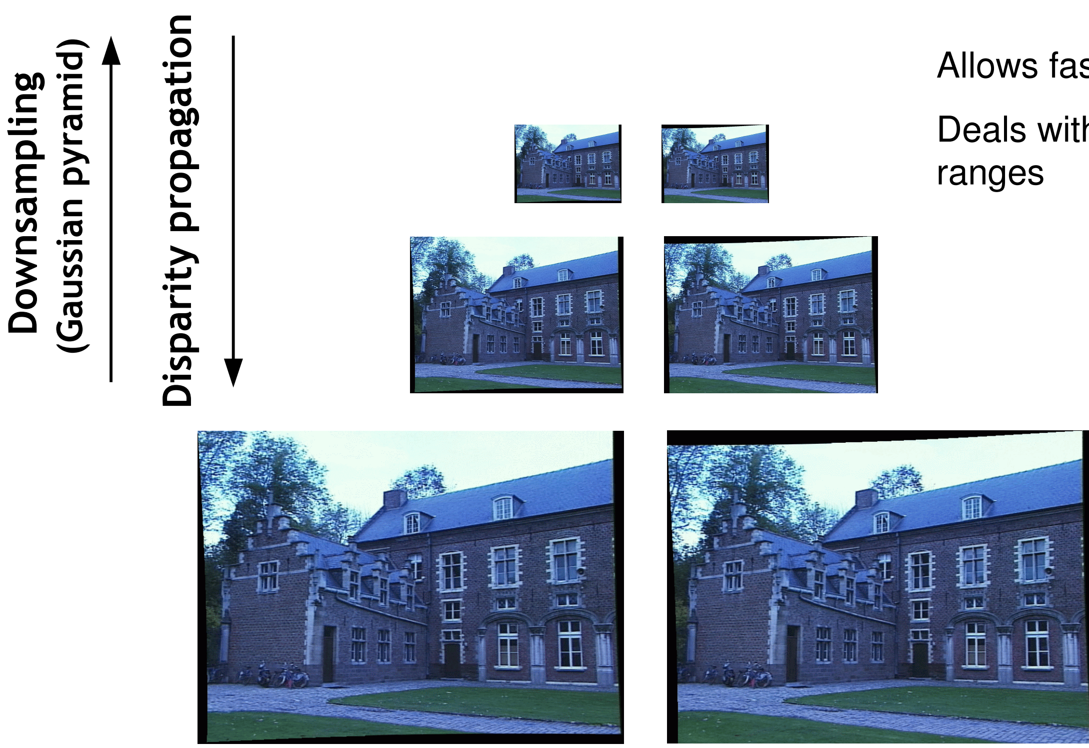


Constraints
- epipolar
- ordering
- uniqueness
- …

Trade-off
- Matching cost
- Discontinuities

besides line-to-line optimization, also point-by-point optimization (real-time),
or image-to-image (e.g. graph-cut)

Hierarchical stereo matching



Allows faster computation

Deals with large disparity ranges

Disparity map

image $I(x,y)$

Disparity map $D(x,y)$

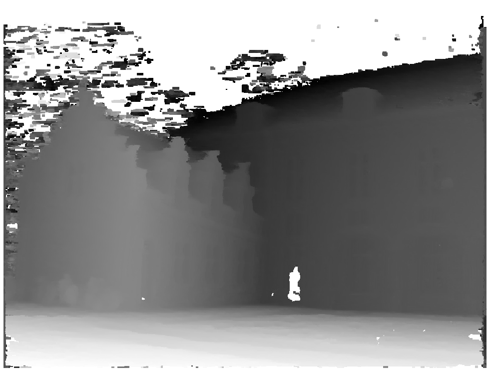
image $I'(x',y')$

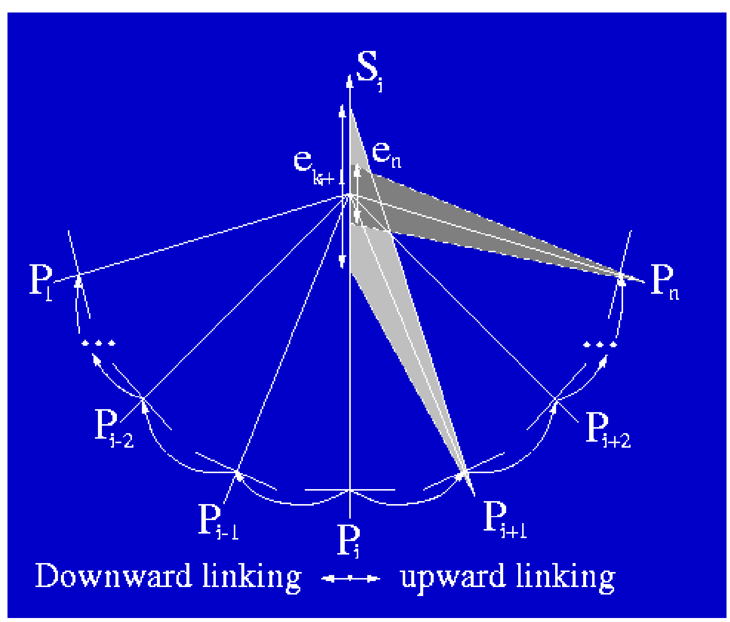
$$(x',y')=(x+D(x,y),y)$$

Multi-view depth fusion

Compute depth for every pixel of reference image

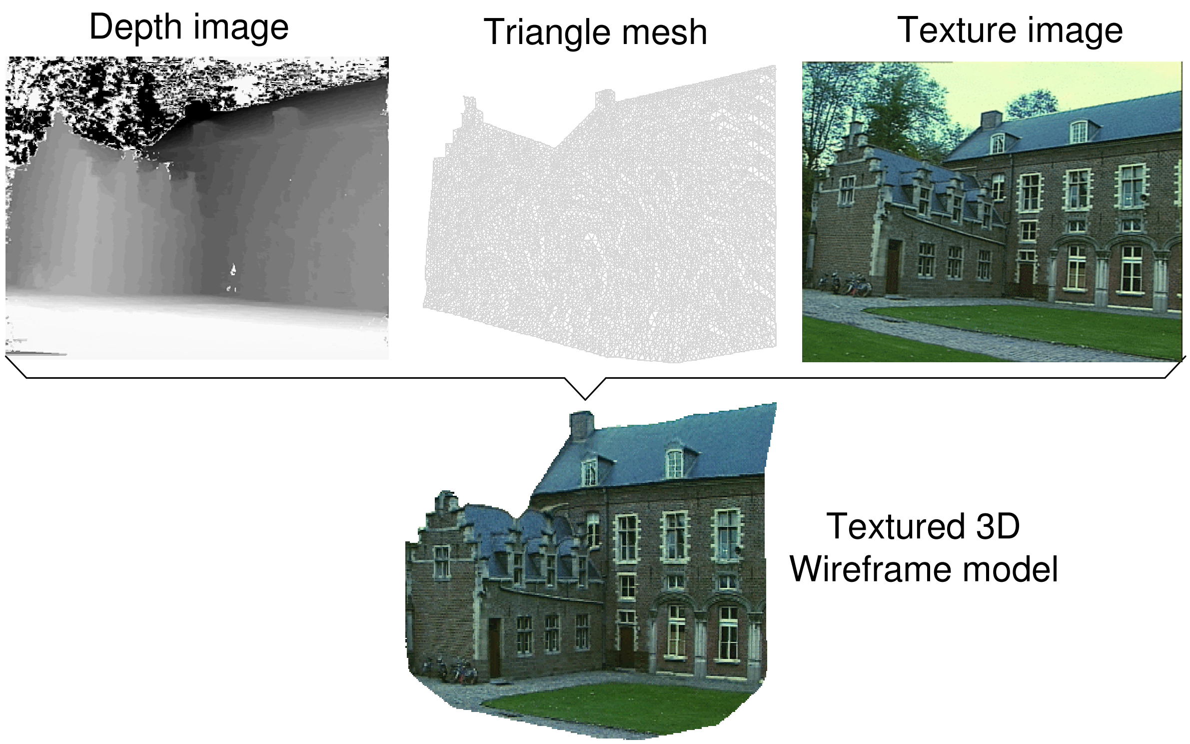
- Triangulation
- Use multiple views
- Up- and down sequence
- Use Kalman filter





Allows to compute robust texture

3D surface model



Part of Jain temple

Recorded during tourist trip in India

(Nikon F50; Scanned)

Recording archaeological artefacts

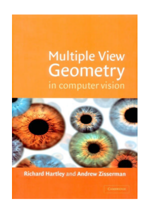
(Medusa head on Antonine Nymphaeum of Sagalassos)

Figures and videos courtesy of:

- David Capel, Andrew Fitzgibbon, Eugénie von Tunzelmann
- 2d3
- Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, Luc Van Gool, Reinhard Koch, Benno Heigl

Further reading:

- Hartley & Zisserman “Multiple View Geometry in Computer Vision”, Second edition, CUP, 2003



- Notes and more available at :

  http://www.robots.ox.ac.uk/~az/tutorials/
  http://www.cs.unc.edu/~marc/tutorial/