Computer Vision (CS 763)

Absolute Orientation and Bundle Adjustment

Arjun Jain

Recap

Motivation

Given the relative orientation of two images, compute the points in 3D

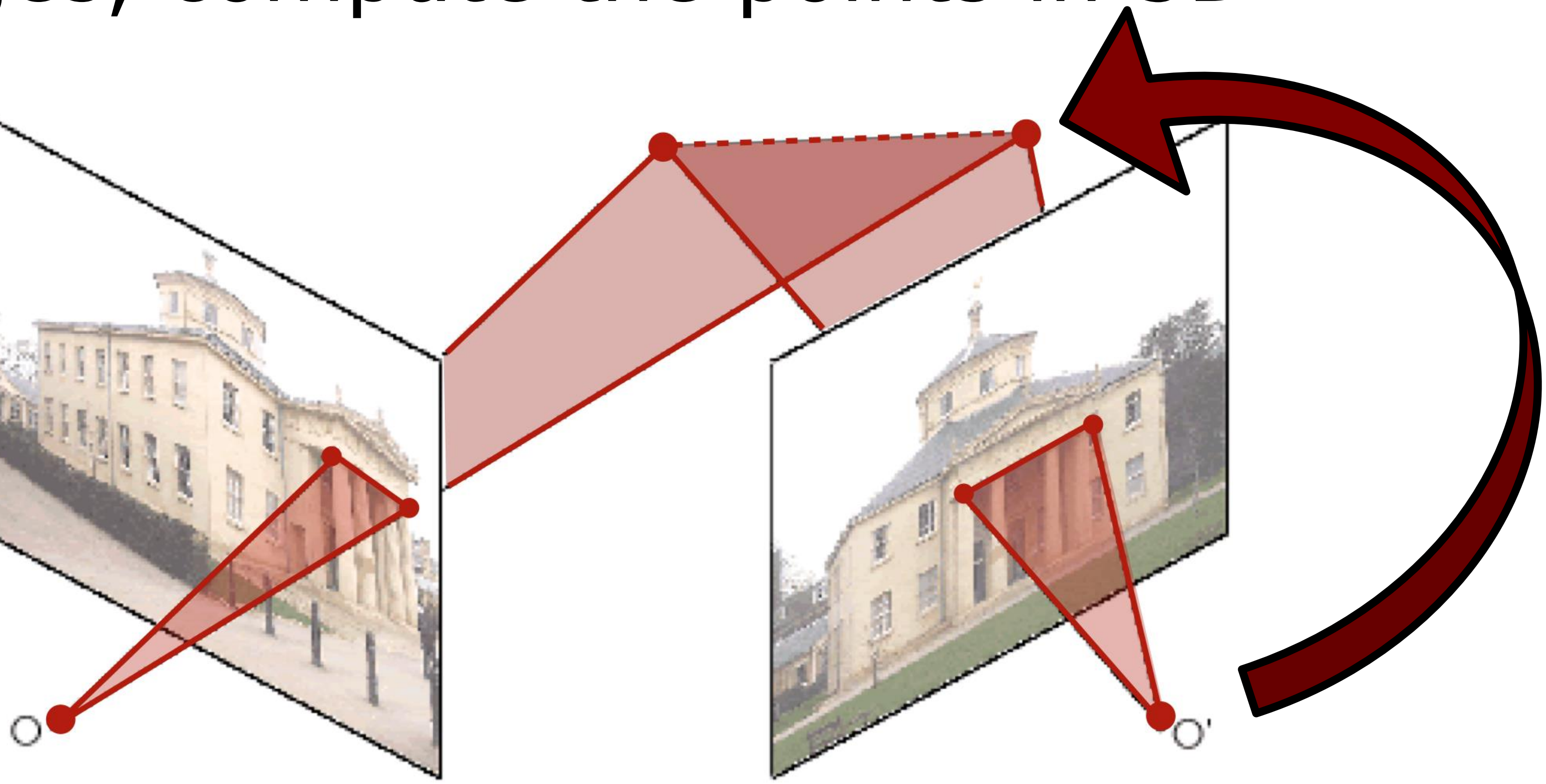


Image courtesy: Schindler

Topic

Last lectures

Computing the relative orientation of two images

Today

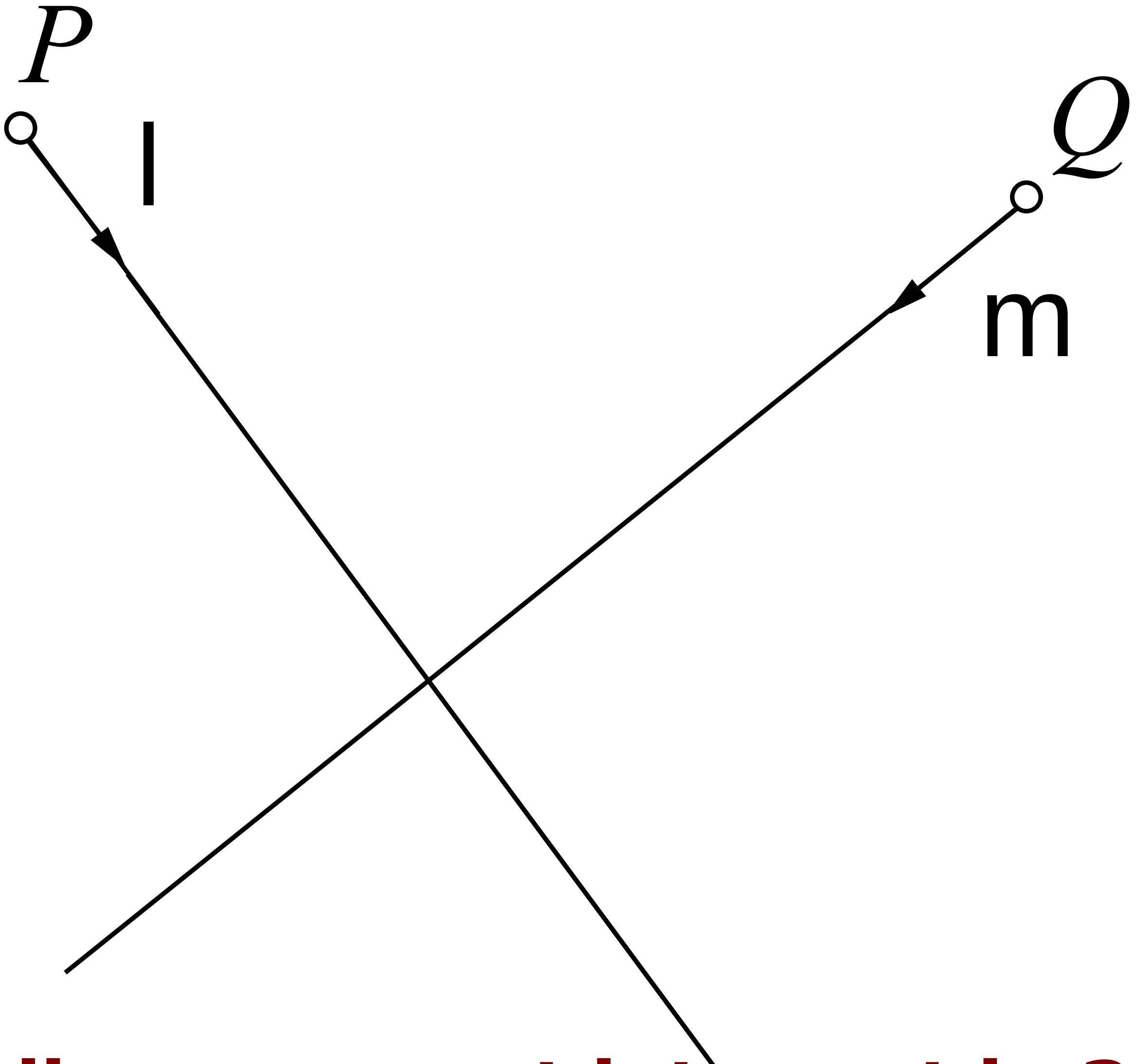
Given the relative orientation of the images, compute the **3D location of corresponding points**

Triangulation / forward intersection

Table of Contents

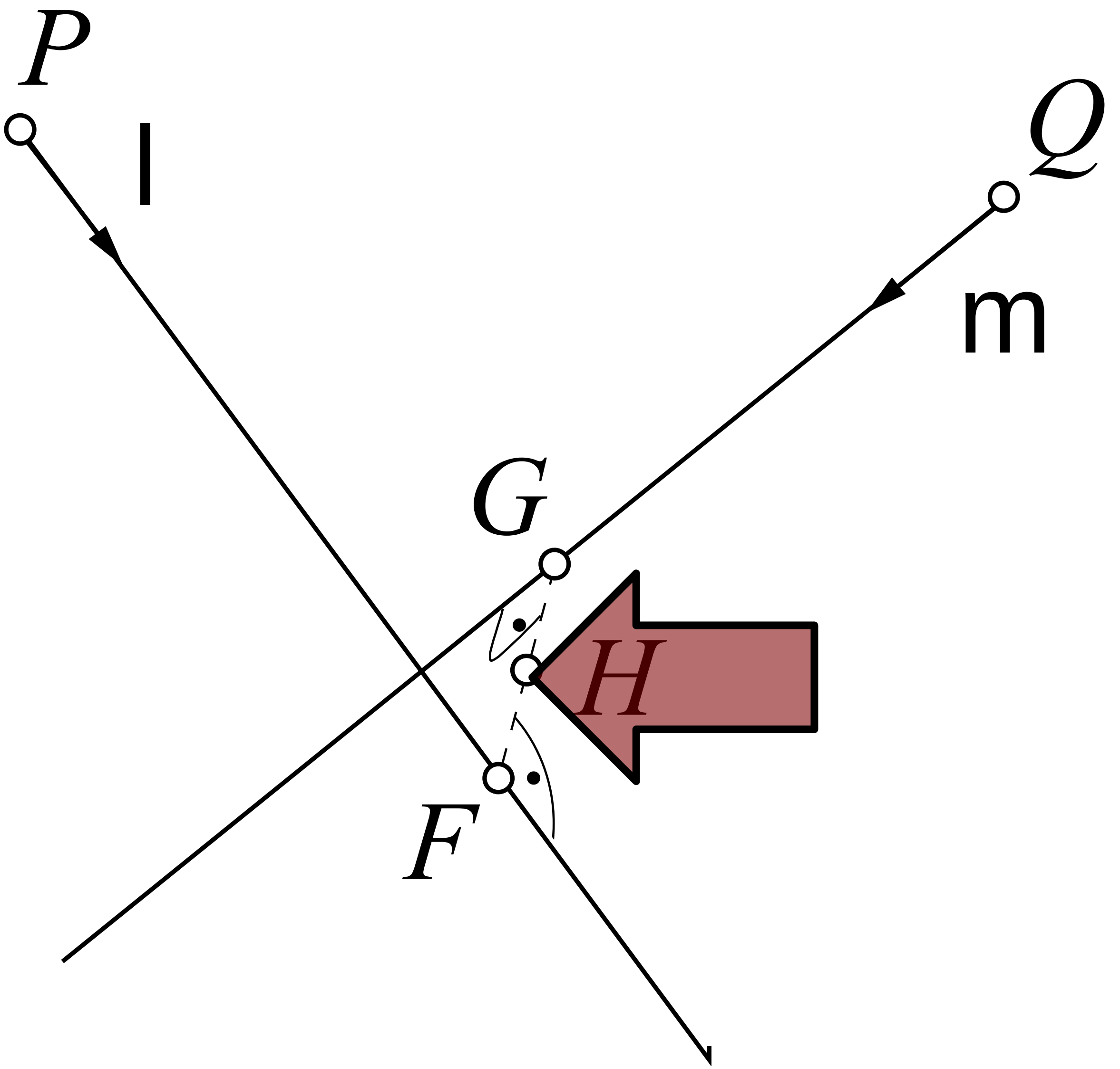
1.
Geometric Approach

The Problem



The lines may not intersect in 3D!

Find the Point H



Refresher from Camera Geometry

- Starting from $\lambda \mathbf{x} = \mathsf{K}R\boldsymbol{X} - \mathsf{K}R\boldsymbol{X}_O$
- we obtain

$$
\begin{aligned}
\boldsymbol{X} &= (\mathsf{K}R)^{-1}\mathsf{K}R\boldsymbol{X}_O + \lambda(\mathsf{K}R)^{-1}\mathbf{x} \\
&= \boldsymbol{X}_O + \lambda(\mathsf{K}R)^{-1}\mathbf{x}
\end{aligned}
$$

3x1 Euclidean

- The term $(\mathsf{K}R)^{-1}\mathbf{x}$ describes the direction of the ray from the camera origin $\boldsymbol{X}_O$ to the 3D point $\boldsymbol{X}$

Geometric Solution

- Equation for two lines in 3D

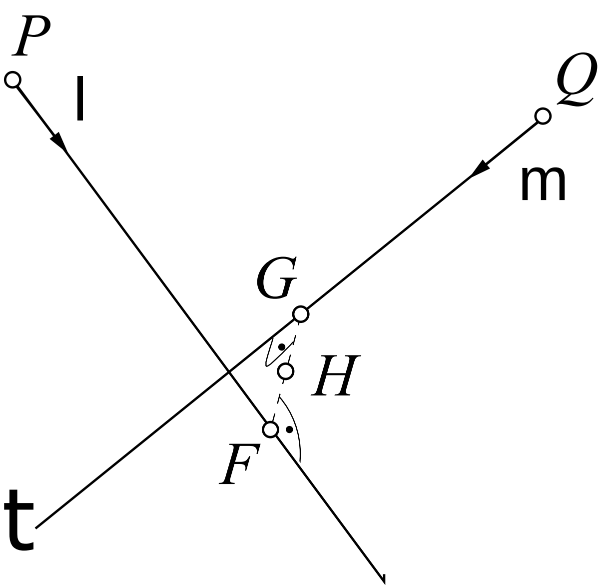
$$\boldsymbol{f} = \boldsymbol{p} + \lambda \boldsymbol{r} \qquad \boldsymbol{g} = \boldsymbol{q} + \mu \boldsymbol{s}$$

- with the points $\boldsymbol{p} = \boldsymbol{X}_{O'} \quad \boldsymbol{q} = \boldsymbol{X}_{O''}$
- and the directions (calibrated camera)

$$\boldsymbol{r} = \mathsf{R}'^{\mathsf{T}}\ {}^k\mathbf{x}' \qquad \boldsymbol{s} = \mathsf{R}''^{\mathsf{T}}\ {}^k\mathbf{x}''$$

- with ${}^k\mathbf{x}' = (x', y', c)^{\mathsf{T}} \quad {}^k\mathbf{x}'' = (x'', y'', c)^{\mathsf{T}}$

Geometric Solution



- The shortest connection requires that FG is orthogonal to both lines
- This leads to the constraint

$$(\boldsymbol{f}-\boldsymbol{g})\cdot\boldsymbol{r}=0 \qquad (\boldsymbol{f}-\boldsymbol{g})\cdot\boldsymbol{s}=0$$

  which directly leads to

$$\begin{aligned}(\boldsymbol{q}+\lambda\boldsymbol{s}-\boldsymbol{p}-\mu\boldsymbol{r})\cdot\boldsymbol{s} &= 0\\ (\boldsymbol{q}+\lambda\boldsymbol{s}-\boldsymbol{p}-\mu\boldsymbol{r})\cdot\boldsymbol{r} &= 0\end{aligned}$$

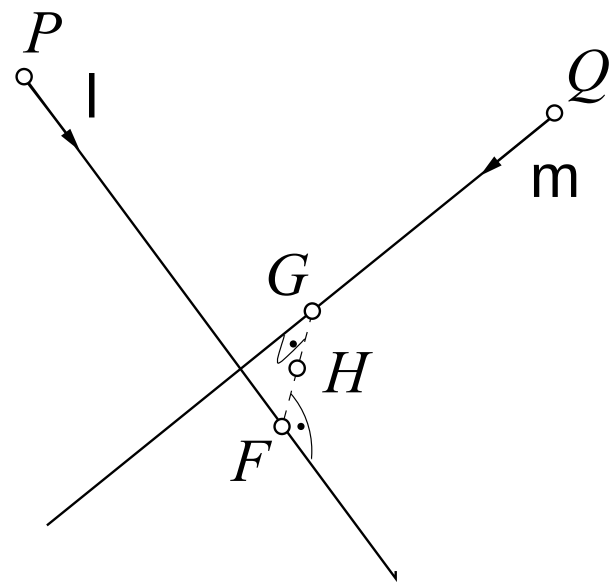
- Two equations, two unknowns

Geometric Solution

- By solving the equations

$$\begin{aligned}(\boldsymbol{q}+\lambda\boldsymbol{s}-\boldsymbol{p}-\mu\boldsymbol{r})\cdot\boldsymbol{s} &= 0\\(\boldsymbol{q}+\lambda\boldsymbol{s}-\boldsymbol{p}-\mu\boldsymbol{r})\cdot\boldsymbol{r} &= 0\end{aligned}$$

  we obtain λ, μ
- λ, μ directly yield F and G
- We compute H as the middle of the line connecting F and G



2.
For the Stereo Normal Case

Stereo Normal Case

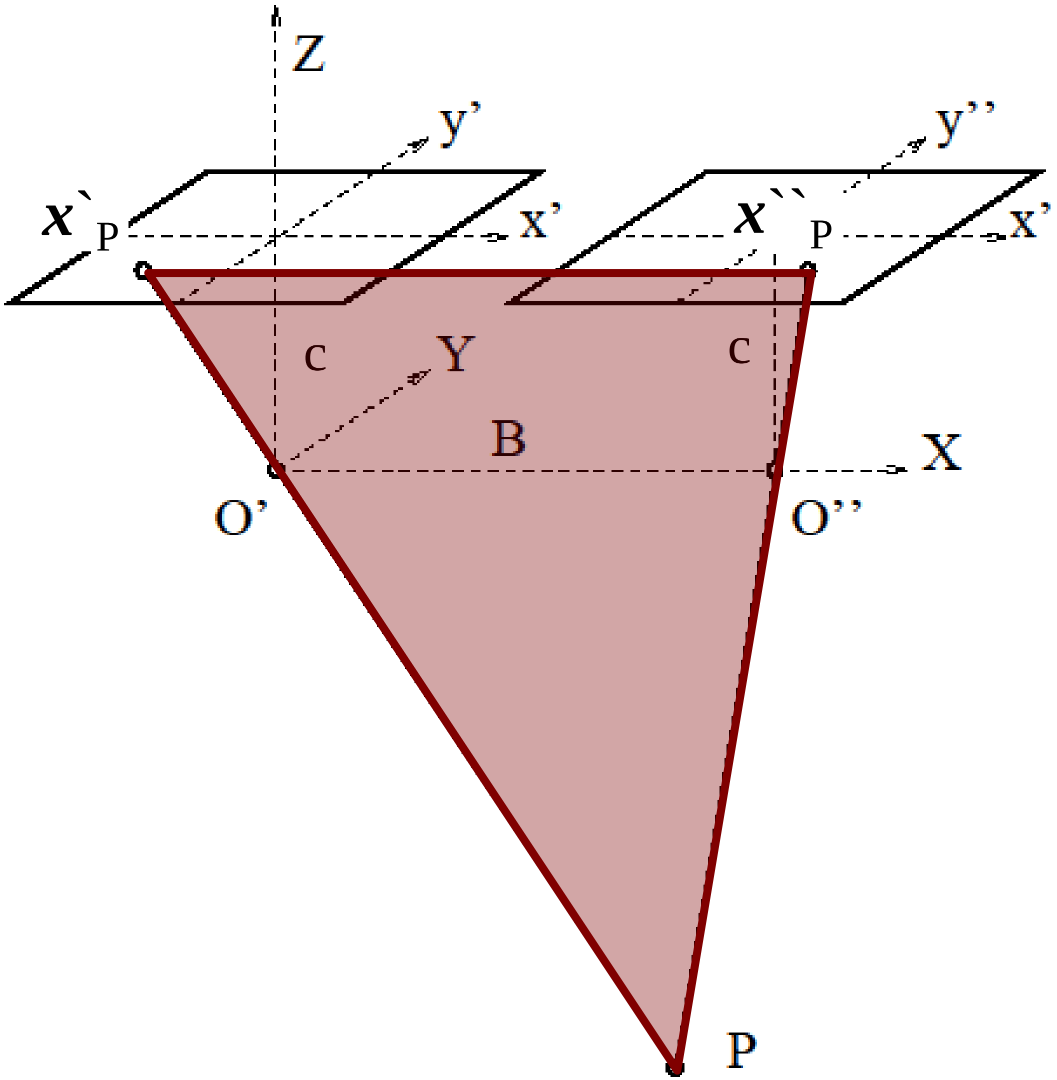


Image courtesy: Förstner

Stereo Normal: Intersection

1. Z-coordinate from intercept theorem

$$\frac{Z}{c} = \frac{B}{-\underbrace{(x'' - x')}_{p_x}}$$

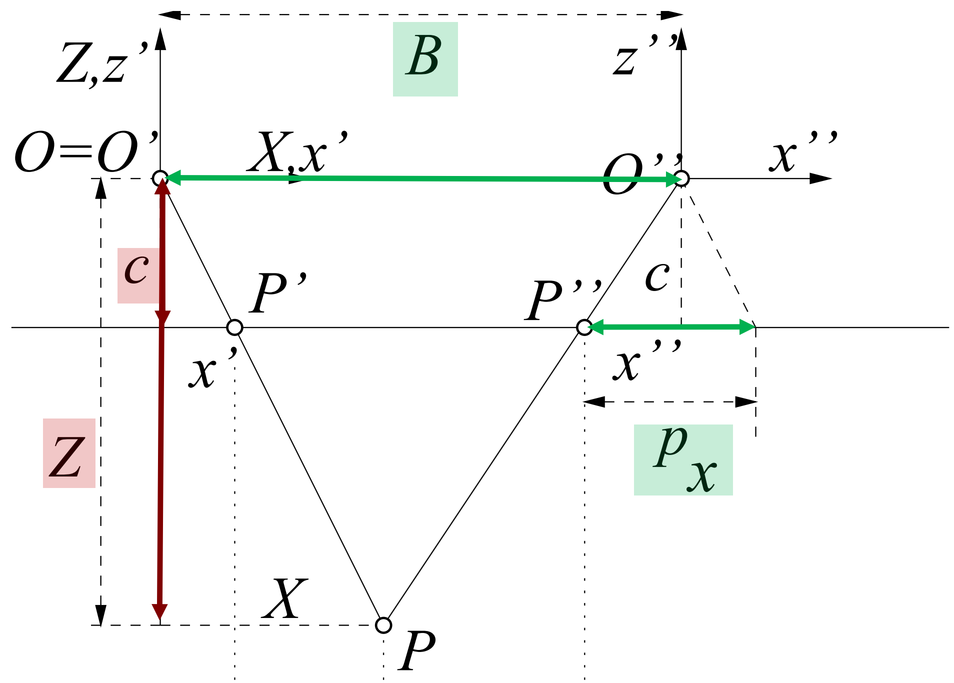


Image courtesy: Förstner

Stereo Normal: Intersection

1. Z-coordinate from intercept theorem

$$\frac{Z}{c} = \frac{B}{\underbrace{-(x'' - x')}_{p_x}}$$

$$Z = c\frac{B}{-(x'' - x')}$$

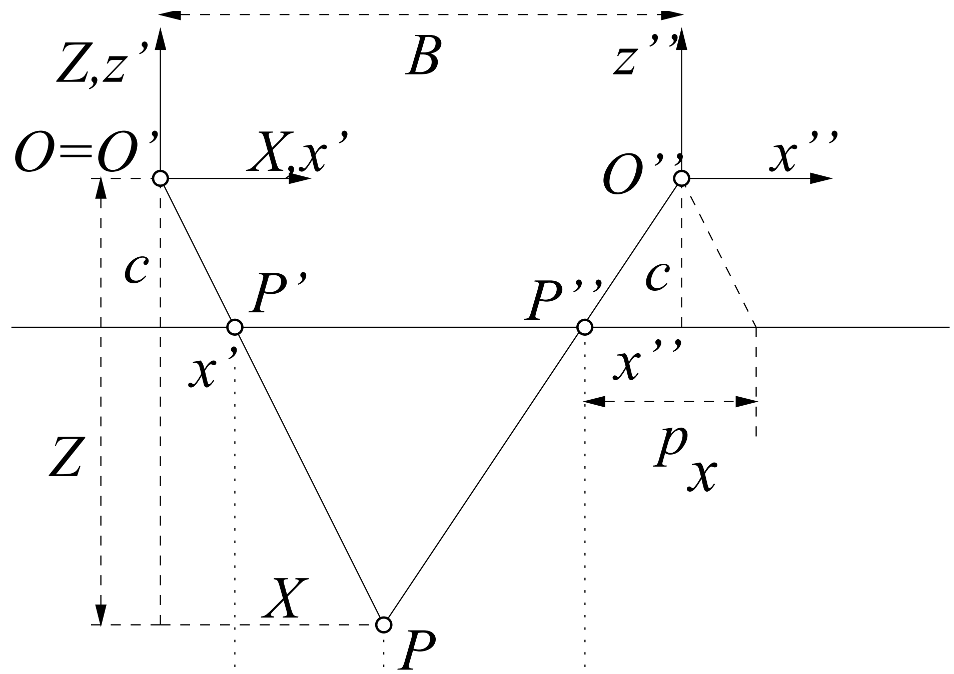


Image courtesy: Förstner

Stereo Normal: Intersection

1. Z-coordinate from intercept theorem

$$\frac{Z}{c} = \frac{B}{-(x'' - x')}$$

2. X-coordinate

$$\frac{X}{x'} = \frac{Z}{c}$$

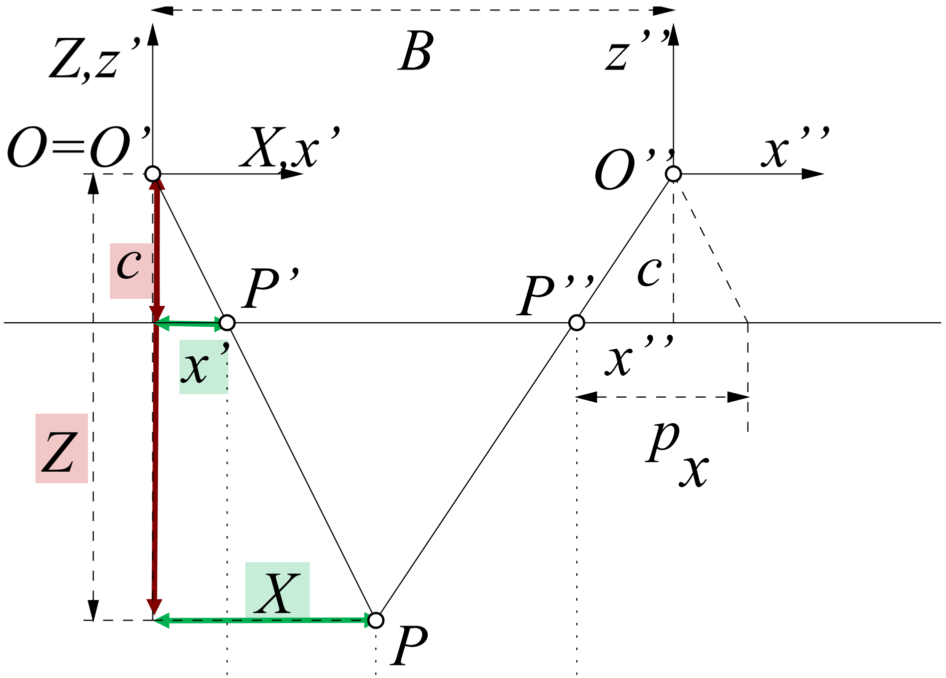


Image courtesy: Förstner

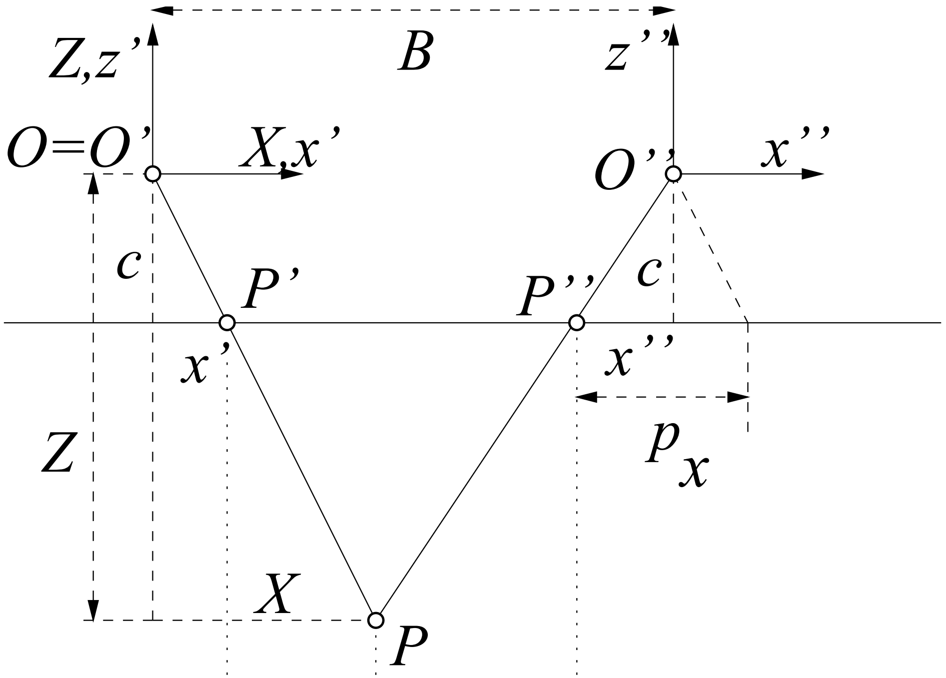
Stereo Normal: Intersection

1. Z-coordinate from intercept theorem

$$\frac{Z}{c} = \frac{B}{-(x'' - x')}$$

2. X-coordinate

$$\frac{X}{x'} = \frac{Z}{c}$$



$$\Rightarrow \quad X = x' \frac{B}{-(x'' - x')}$$

Image courtesy: Förstner

Stereo Normal: Intersection

1. Z-coordinate from intercept theorem

$$\frac{Z}{c} = \frac{B}{-(x'' - x')}$$

2. X-coordinate

$$\frac{X}{x'} = \frac{Z}{c}$$

3. Y-coordinate by mean

$$\frac{Y}{X} = \frac{\frac{y'+y''}{2}}{x'}$$

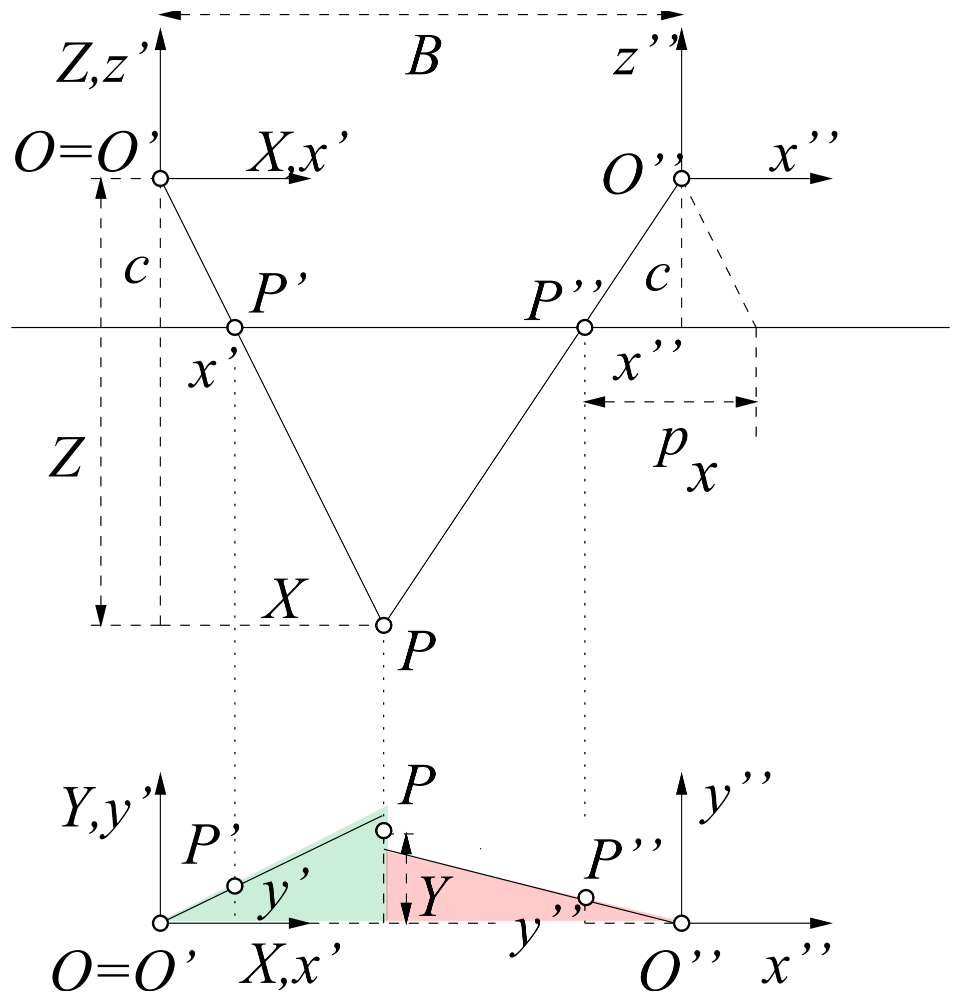


Image courtesy: Förstner

Stereo Normal: Intersection

1. Z-coordinate from intercept theorem

$$\frac{Z}{c} = \frac{B}{-(x'' - x')}$$

2. X-coordinate

$$\frac{X}{x'} = \frac{Z}{c}$$

3. Y-coordinate by mean

$$\frac{Y}{X} = \frac{\frac{y'+y''}{2}}{x'}$$

$$Y = \frac{y' + y''}{2} \frac{B}{-(x'' - x')}$$

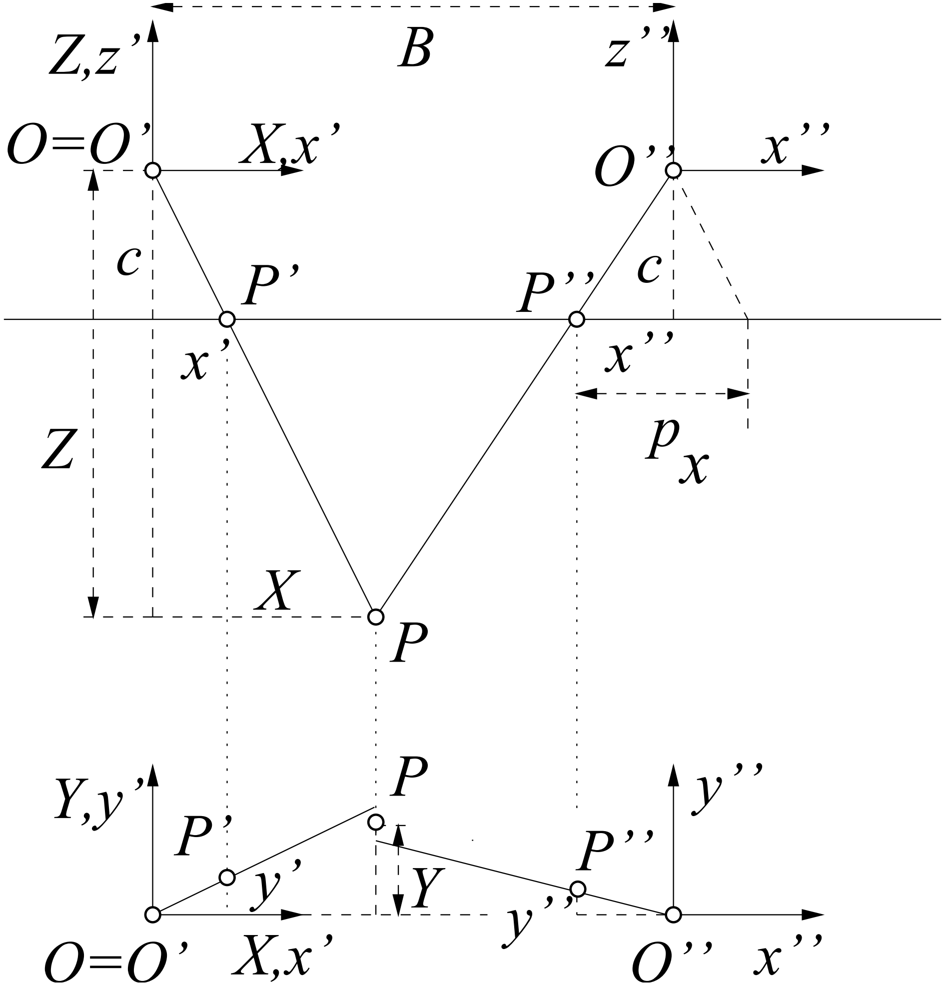


Image courtesy: Förstner

Intersection of Two Rays for the Stereo Normal Case

$$X = x' \frac{B}{-(x'' - x')} \qquad Y = \frac{y' + y''}{2} \frac{B}{-(x'' - x')} \qquad Z = c \frac{B}{-(x'' - x')}$$

- x-parallax $p_x = x'' - x'$ corresponds to depth Z
- y-parallax $p_y = y'' - y'$ corresponds to the consistency of image points and should be small (due to stereo normal case)
- The parallax is also called disparity

X-Parallax (X-Disparity)

- The x-parallax is a key element

$$
\begin{aligned}
X &= x' \frac{B}{-(x'' - x')} \\
Y &= \frac{y' + y''}{2} \frac{B}{-(x'' - x')} \\
Z &= c \frac{B}{-(x'' - x')}
\end{aligned}
$$

X-Parallax and Scale Number

- The x-parallax is a key element

$$
\begin{aligned}
X &= x' \boxed{\frac{B}{-(x''-x')}} \\
Y &= \frac{y'+y''}{2} \boxed{\frac{B}{-(x''-x')}} \\
Z &= c \boxed{\frac{B}{-(x''-x')}}
\end{aligned}
$$

image scale number

X-Parallax and Scale Number

- The x-parallax is a key element

$$
\begin{aligned}
X &= x' \boxed{\frac{B}{-(x'' - x')}} \\
Y &= \frac{y' + y''}{2} \boxed{\frac{B}{-(x'' - x')}} \\
Z &= c \boxed{\frac{B}{-(x'' - x')}}
\end{aligned}
$$

image scale number

- Image scale number: $M = \frac{-B}{x'' - x'} = \frac{Z}{c}$

$$
\boxed{X = Mx' \qquad Y = M\frac{y' + y''}{2} \qquad Z = Mc}
$$

Intersection of Two Rays for the Stereo Normal Case

- If the y-parallax is zero, we obtain

$$X = x' \frac{B}{-p_x} \qquad Y = y' \frac{B}{-p_x} \qquad Z = c \frac{B}{-p_x}$$

Intersection of Two Rays for the Stereo Normal Case

- If the y-parallax is zero, we obtain

$$X = x' \frac{B}{-p_x} \qquad Y = y' \frac{B}{-p_x} \qquad Z = c \frac{B}{-p_x}$$

- We can write this as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -\frac{B}{p_x} & 0 & 0 \\ 0 & -\frac{B}{p_x} & 0 \\ 0 & 0 & -\frac{B}{p_x} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ c \end{bmatrix}$$

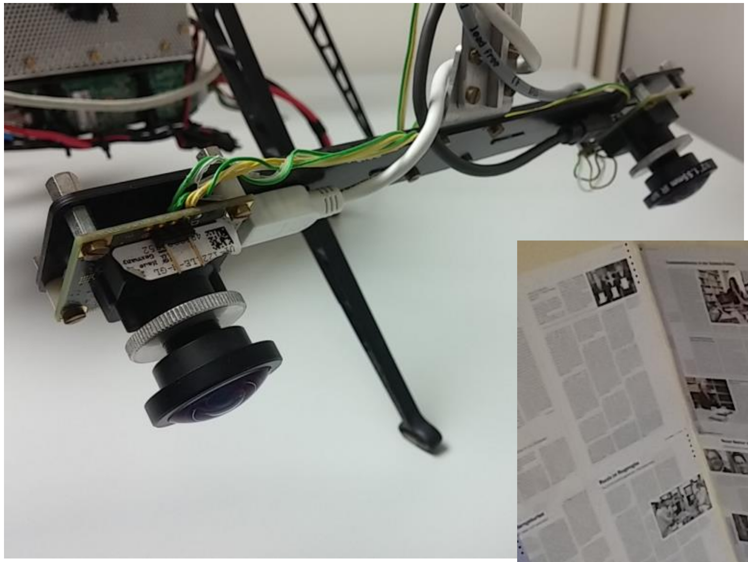
Parallax Map

- Using H.C. and the parallax as input

$$\begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix} = \begin{bmatrix} B & 0 & 0 & 0 \\ 0 & B & 0 & 0 \\ 0 & 0 & Bc & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \\ p_x \end{bmatrix}$$

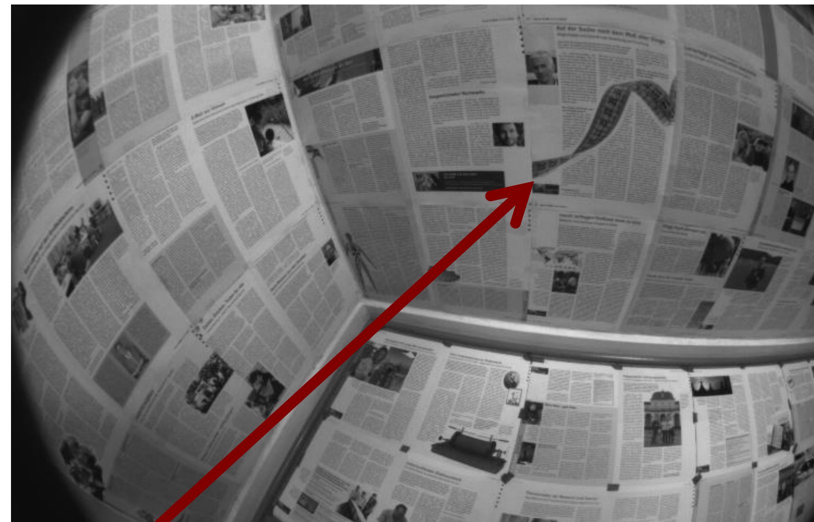
- For a set of points $\{x', y'\}$ in the first image, $\{x', y', p_x\}$ is called **parallax map**
- **The parallax map directly leads to the 3D coordinates of the point**

Example – Setup



Example – Image Pair

Example – Parallax Map

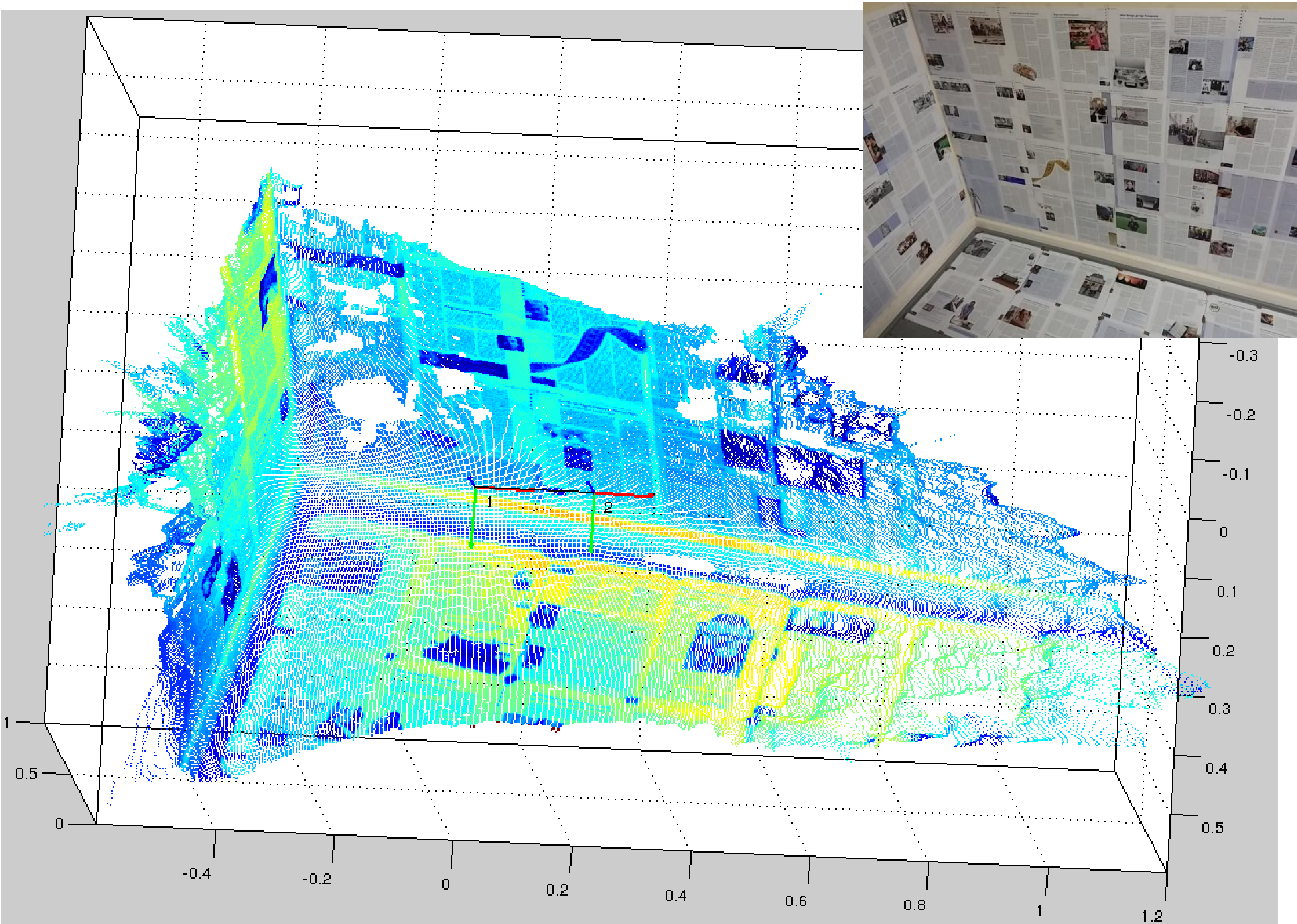
parallax map $\{x', y', p_x\}_1 \ldots \{x', y', p_x\}_N$

Example – Parallax Map

parallax map $\{x', y', p_x\}_1 \ldots \{x', y', p_x\}_N$

Example – 3D Point Cloud



3.
Quality of the 3D Points

Quality of the 3D Points

What influences the quality of the 3D points obtained in the normal case?

Quality of the 3D Points

What influences the quality of the 3D points obtained in the normal case?

A. Quality of the orientation parameters

B. Quality of the measured image coordinates

Quality of the 3D Points

What influences the quality of the 3D points obtained in the normal case?

A. Quality of the orientation parameters

B. Quality of the measured image coordinates

Quality of the 3D Points

- Assuming that we measure the image coordinates in x/y with $\sigma_{x'} = \sigma_{y'}$
- Starting from

$$X = Mx' \qquad Y = M\frac{y' + y''}{2}$$

- Directly yields the uncertainty in x/y

$$\boxed{\sigma_X = M\sigma_{x'} = \frac{Z}{c}\sigma_{x'}} \qquad \boxed{\sigma_Y = \frac{\sqrt{2}}{2}M\sigma_{y'} = \frac{\sqrt{2}}{2}\frac{Z}{c}\sigma_{y'}}$$

Quality of the 3D Points

- For the Z coordinate, we start with

$$Z = Mc = -\frac{B}{p_x}c \qquad \rightarrow \qquad Z\,p_x = -B\,c$$

- and obtain for the relative precision

$$\boxed{\frac{\sigma_Z}{Z} = \frac{\sigma_{p_x}}{p_x}}$$

The relative precision of the height is the relative precision of the x-parallax

Height/Depth Precision

- Starting from $\frac{\sigma_Z}{Z} = \frac{\sigma_{p_x}}{p_x}$ we obtain:

$$\sigma_Z = \frac{Z}{p_x}\sigma_{p_x} = \frac{cB}{p_x^2}\sigma_{p_x} = \frac{Z^2}{cB}\sigma_{p_x} = \frac{Z}{c\;B/Z}\sigma_{p_x}$$

$$Z = \frac{cB}{p_x} \qquad \frac{1}{p_x} = \frac{Z}{cB} \qquad Z = \frac{1}{1/Z}$$

Height/Depth Precision

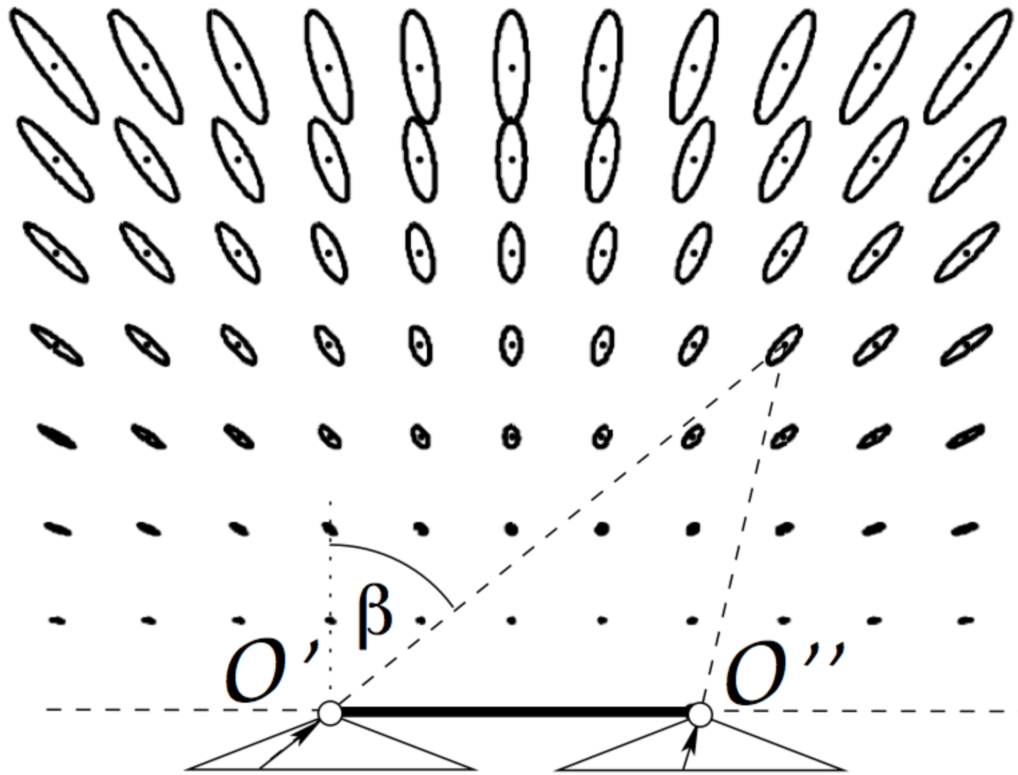
- Starting from $\frac{\sigma_Z}{Z} = \frac{\sigma_{p_x}}{p_x}$ we obtain:

$$\sigma_Z = \frac{Z}{p_x}\sigma_{p_x} = \frac{cB}{p_x^2}\sigma_{p_x} = \frac{Z^2}{cB}\sigma_{p_x} = \frac{Z}{c\ B/Z}\sigma_{p_x}$$

Standard deviation of Z depends

- on the **x-parallax standard deviation**
- inverse quadratically on the **x-parallax**
- quadratically on the **depth**
- inversely on the **base/depth ratio**

Stereo Uncertainty Field



$$x', x''$$

$$\sigma_{x'}^2 = \sigma_{x''}^2$$

Image courtesy: Förstner and Wrobel

Summary - Triangulation

- We can estimate 3D point locations (in the camera frame) given corresponding points and orientation parameters through triangulation
- Geometric approach
- Triangulation for the stereo normal case
- Quality of the 3D Points for the stereo normal case

Part II
Absolute Orientation

"Where are the points in the world?"

Relative Orientation

- The result of the R.O. is the so-called **photogrammetric model**
- It contains the
  - parameters of the relative orientation of both cameras
  - 3D coordinates of N points in a local coordinate frame

$$ {}^{m}\boldsymbol{X}_{n} = ({}^{m}X_{n}, {}^{m}Y_{n}, {}^{m}Z_{n})^{\mathsf{T}} \qquad n = 1, \cdots, N $$

- Known up to a similarity transform (for calibrated cameras)

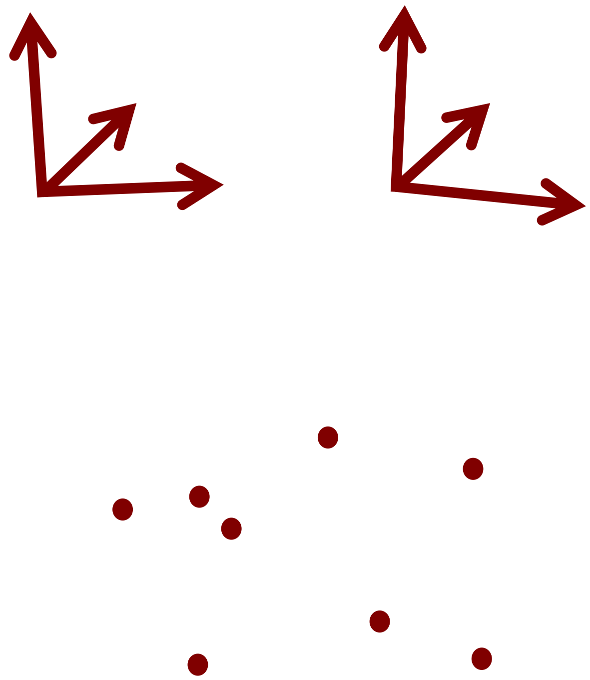
Absolute Orientation

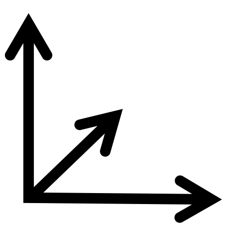
- A similarity transform maps the photogrammetric model into the object reference frame

$$^{o}\boldsymbol{X}_n = \underline{\lambda R}\, ^{m}\boldsymbol{X}_n + \underline{\boldsymbol{T}}$$

- 7 DoF for the similarity transform (3 rotation, 3 translation, 1 scale)
- **Control points** are required

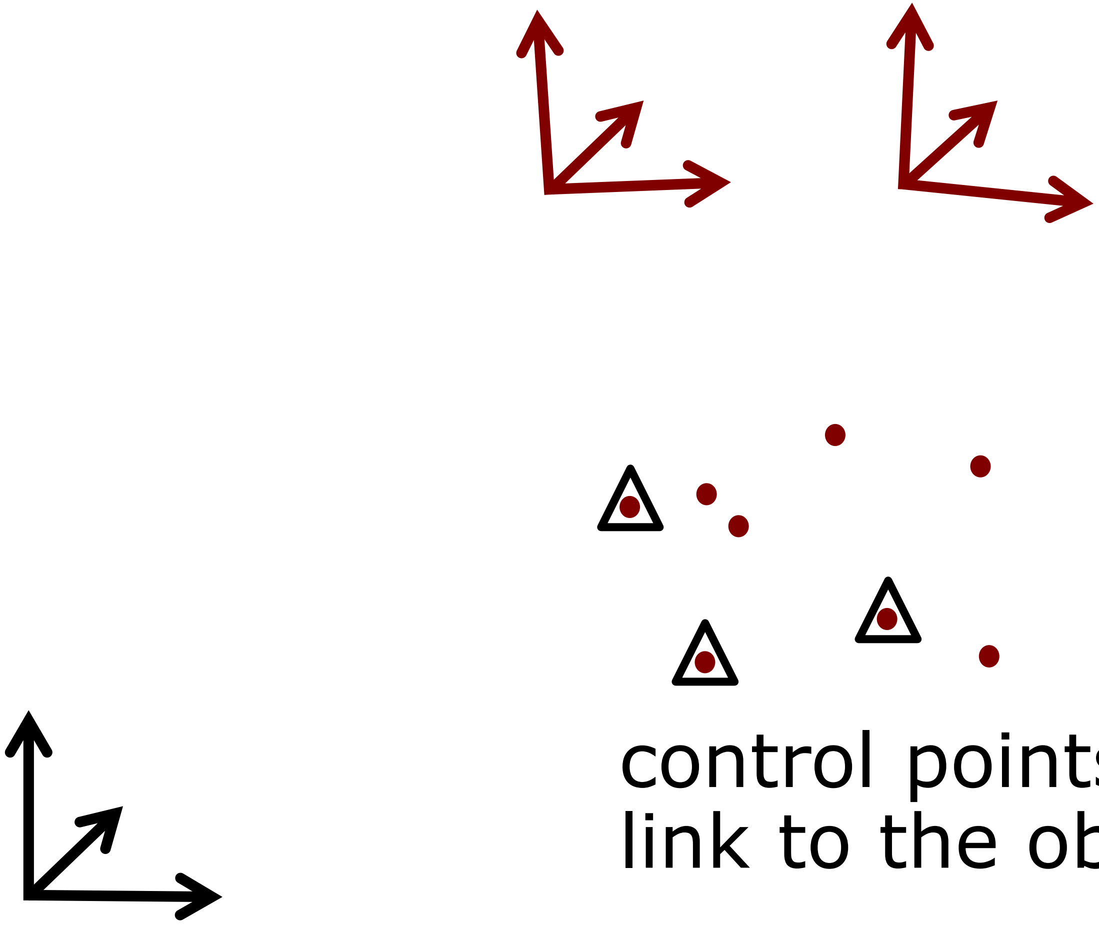
Photogrammetric Model





missing

Object Reference Frame

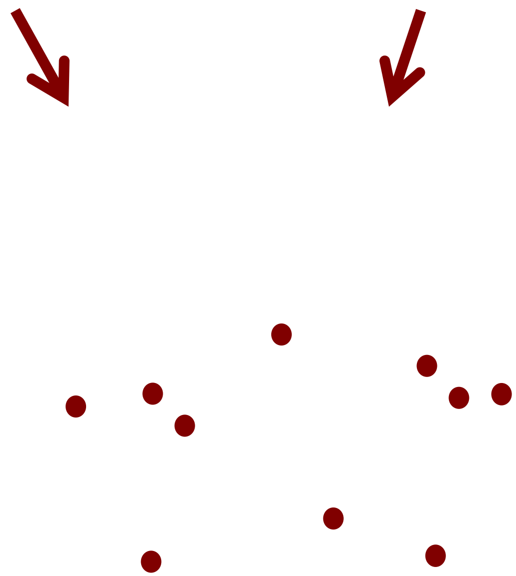


control points create the
link to the object frame

Least Squares Solution

- Non-linear least squares solution
- At least 3 control points (X,Y,Z known)
- Solution: find a similarity transform between the control points and the photogrammetric model points
- See: Förstner, Photo II, Chapter 1.5

Overview – Initial Stage

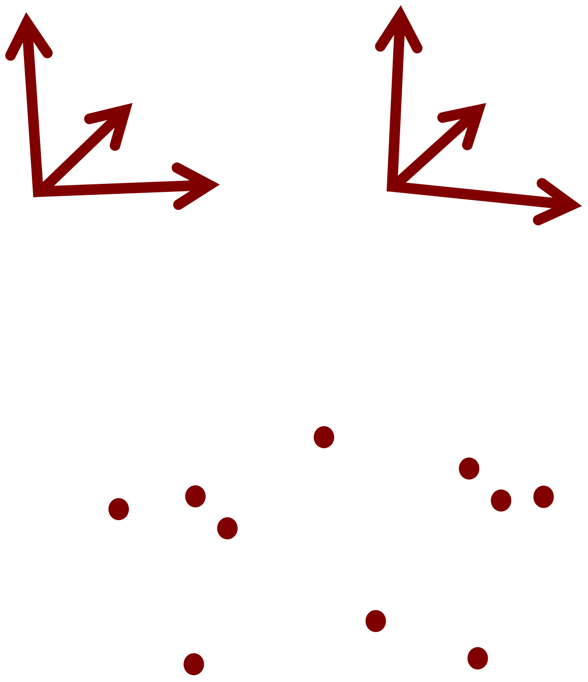


two cameras

+

corresponding points in both images

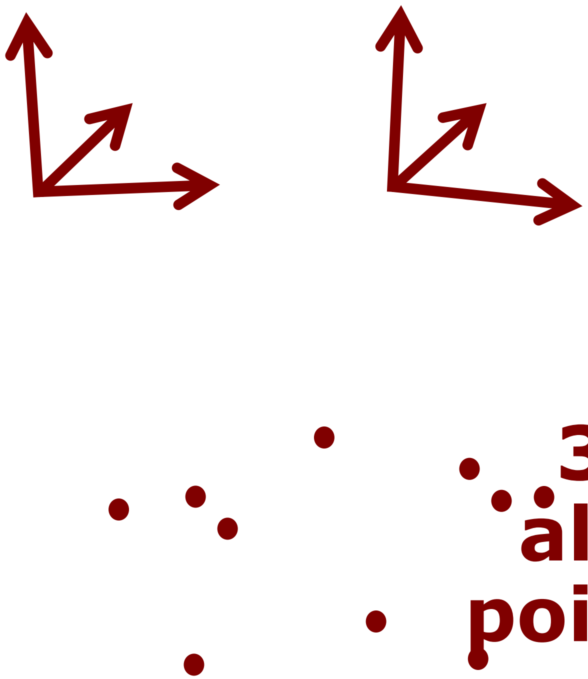
Overview – 1st Step



relative orientation

corresponding points in both images

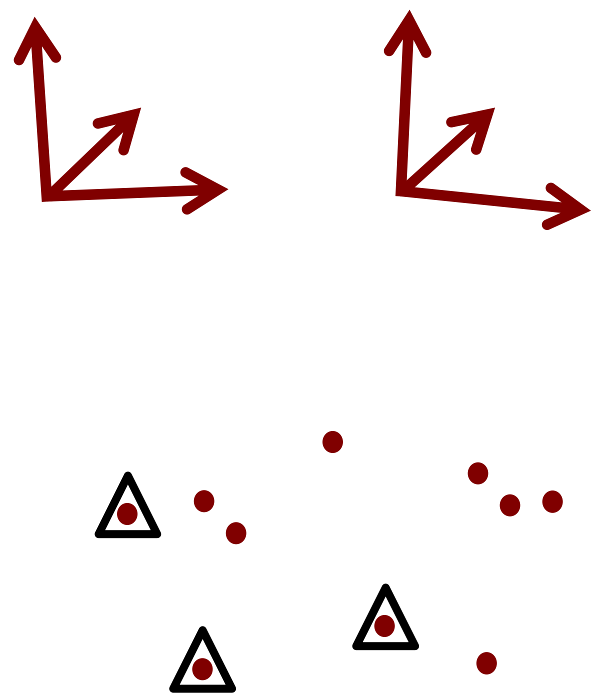
Overview – 1st Step



relative orientation

3D locations of all corresponding points (local frame)

Overview – 2nd Step

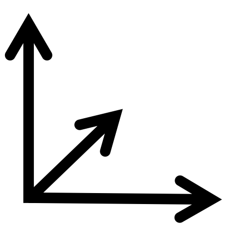


relative orientation

+

control points

3D location of all corresponding points in the object frame



2-Step Solution

- By combining all techniques we learnt for Geometric Stereo, we obtain
- **Relative orientation** without control points and 3D location of correspond. points in a local frame
- **Absolute orientation** of cameras and corresponding points through control points

2-Step Solution

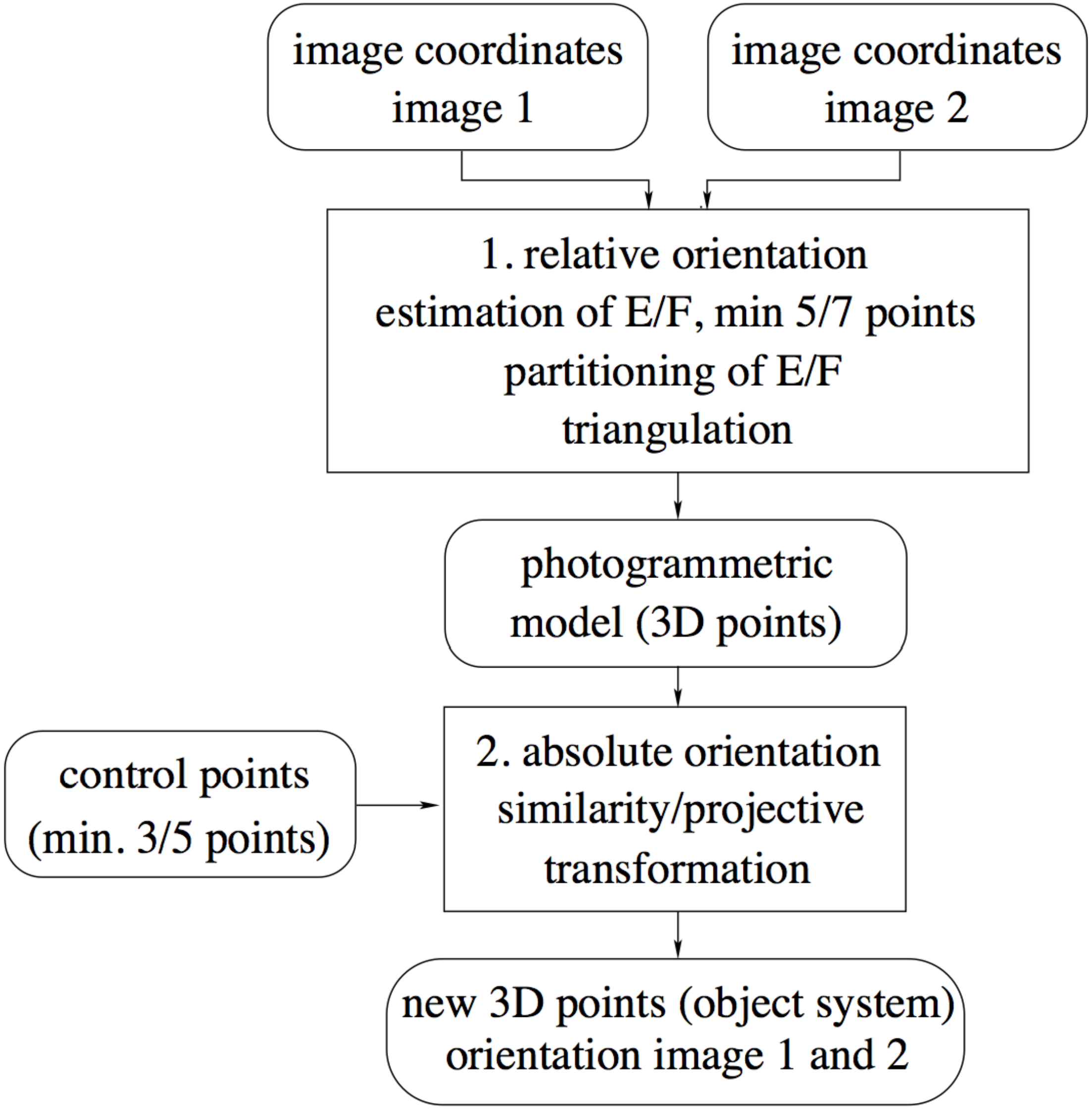


Image courtesy: Förstner and Wrobel

Discussion:

Which Other Orientation Approaches Do We Know?

We have studied

- Direct linear transform (DLT)
- Relative orientation
- Triangulation
- Absolute orientation

We have studied

- Direct linear transform (DLT)
- Relative orientation
- Triangulation
- Absolute orientation

How could we achieve the same using the techniques listed above?

Other Possibilities

Option 1

- DLT for each camera using control pts
- Triangulation for all corresponding pts

Option 2

- One big least squares approach (bundle adjustment)

Option 1 – DLT

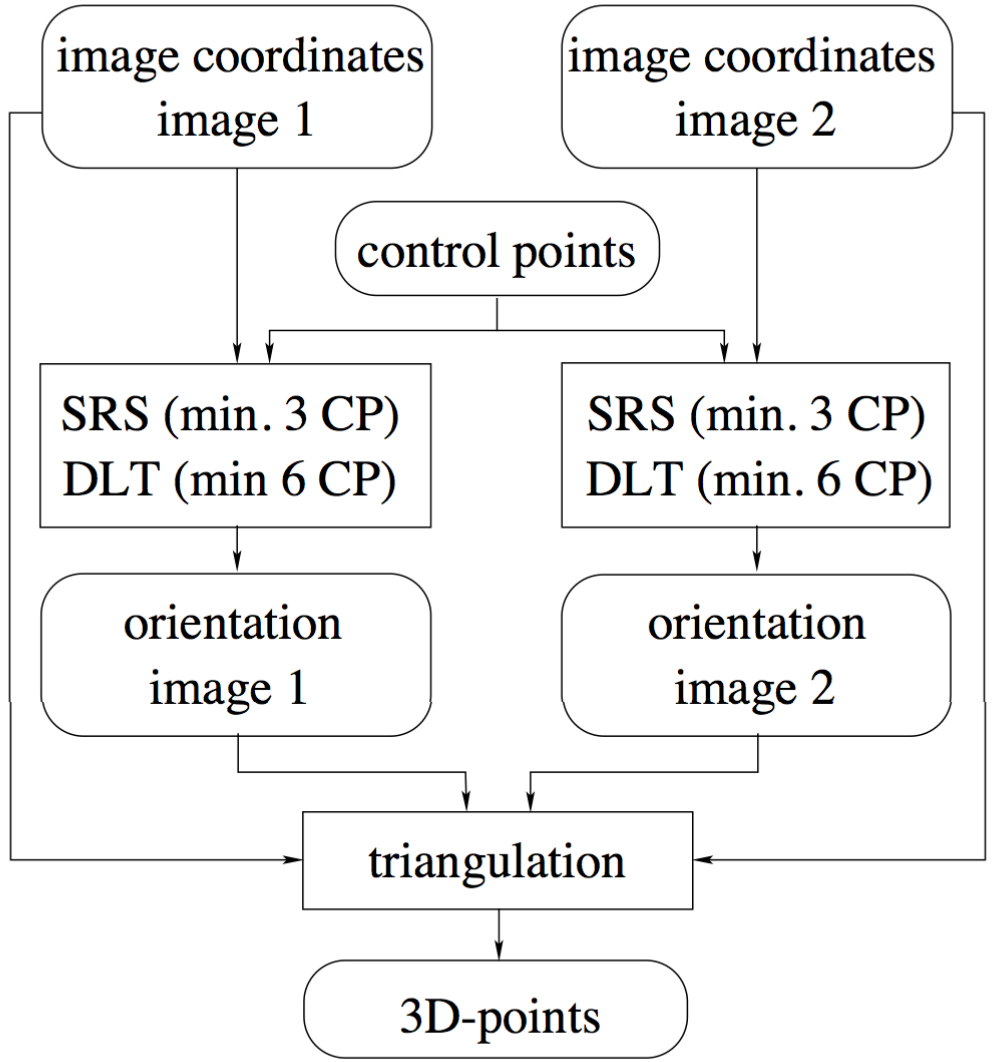


Image courtesy: Förstner and Wrobel

Option 2 – Bundle Adjustment

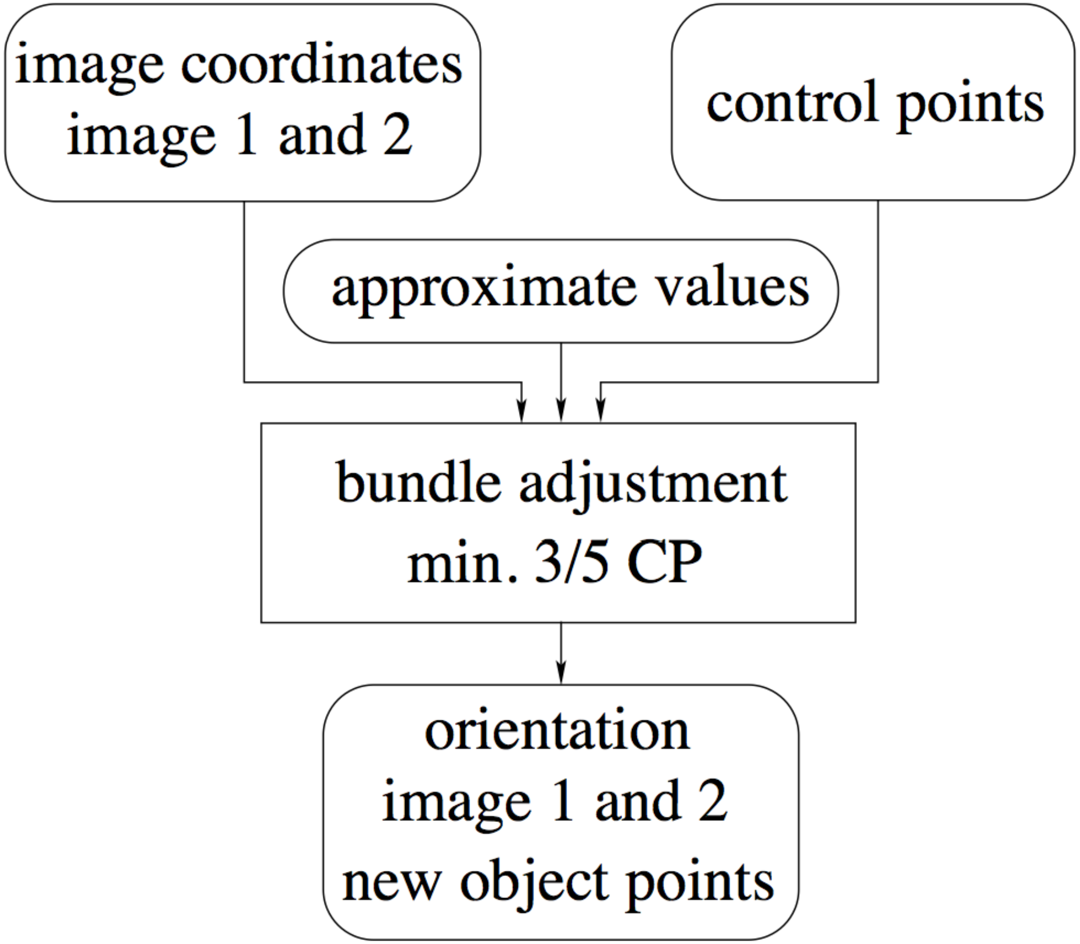


Image courtesy: Förstner and Wrobel

Summary

- Absolute Orientation transforms the photog. model to the object frame
- Different ways for orienting points absolutely (DLT, 2-Step, BA)
- Bundle adjustment is the optimal solution in a statistical sense but requires a (good) initial guess

Multi-View Geometry

Multi-View Geometry

So Far…

Compute the (absolute) orientation of **two** images and observed points

Now

Compute the (absolute) orientation of **multiple** images and observed points

Bundle Adjustment

Why
Multi-View
Reconstruction?

Why Multi-View Reconstruction

- Multiple images are needed to cover the whole surface of the object
- If Precision requirements higher than those of a single image pair
- Level of detail higher than one image
- Estimate the motion of a vehicle/robot and the map of the environment (SLAM, simultaneous localization and mapping)

Example

Video courtesy: van Gool / GeoAutomation

Bundle Block Adjustment

- We used bundle adjustment as the least squares solution for orienting images and measured points
- Bundle block adjustment: multiple images are corrected “en bloc”
- Note: often the term “block” is dropped in modern literature

Bundle Adjustment (BA) Details

- We can formulate BA through

a="arbitrary frame"　img pix.　project. params　distortion

$$
{}^{a}\mathbf{x}'_{ij} + {}^{a}\widehat{\mathbf{v}}_{x'_{ij}} = \widehat{\lambda}_{ij}\ {}^{a}\widehat{\mathsf{P}}_{j}(\boldsymbol{x}_{ij}, \boldsymbol{p}, \boldsymbol{q})\ \widehat{\mathbf{X}}_{i}
$$

ij = point i observed in image j　corrections　scale factor　projection matrix (w/ non-lin. calib.)　3D point

Bundle Adjustment (BA) Details

- We can formulate BA through

$$ {}^{a}\mathbf{x}'_{ij} + {}^{a}\widehat{\mathbf{v}}_{x'_{ij}} = \widehat{\lambda}_{ij}\, {}^{a}\widehat{\mathsf{P}}_{j}(\boldsymbol{x}_{ij}, \boldsymbol{p}, \boldsymbol{q})\, \widehat{\mathbf{X}}_{i} $$

- Encodes the **projection** from the 3D world to the image coordinate system

Bundle Adjustment (BA) Details

- We can formulate BA through

$$ {}^{a}\mathbf{x}'_{ij} + {}^{a}\widehat{\mathbf{v}}_{x'_{ij}} = \widehat{\lambda}_{ij}\ {}^{a}\widehat{\mathsf{P}}_{j}(\boldsymbol{x}_{ij}, \boldsymbol{p}, \boldsymbol{q})\ \widehat{\mathbf{X}}_{i} $$

Unknowns:

- 3D locations of new points $\widehat{\mathbf{X}}_i$
- 1D scale factor $\widehat{\lambda}_{ij}$
- 6D exterior orientation
- 5D projection parameters (interior o.)
- Non-linear distortion parameters $\boldsymbol{q}$

Bundle Adjustment (BA) Details

- We can formulate BA through

$$ {}^{a}\mathbf{x}'_{ij} + {}^{a}\widehat{\mathbf{v}}_{x'_{ij}} = \widehat{\lambda}_{ij}\ {}^{a}\widehat{\mathsf{P}}_{j}(\boldsymbol{x}_{ij}, \boldsymbol{p}, \boldsymbol{q})\ \widehat{\mathbf{X}}_{i} $$

$$ {}^{a}\mathbf{x}'_{ij} + {}^{a}\widehat{\mathbf{v}}_{x'_{ij}} = \widehat{\lambda}_{ij}\widehat{\mathsf{K}}(\boldsymbol{x}_{ij}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{q}})\ \widehat{R}_{j}\ [I_3 | -\widehat{\boldsymbol{X}}_{0j}]\ \widehat{\mathbf{X}}_{i} $$

calibration (interior orientation)

exterior orientation

An Example

- 10k images, 1k points per image
- Each point seen 10 times on avg.
- **How many unknowns do we have?**

$$ {}^{a}\mathbf{x}'_{ij} + {}^{a}\widehat{\mathbf{v}}_{x'_{ij}} = \widehat{\lambda}_{ij} \; {}^{a}\widehat{\mathsf{P}}_{j}(\boldsymbol{x}_{ij}, \boldsymbol{p}, \boldsymbol{q}) \; \widehat{\mathbf{X}}_{i} $$

An Example

- 10k images, 1k points per image
- Each point seen 10 times on avg.
- **How many unknowns do we have?**

$$ {}^{a}\mathbf{x}'_{ij} + {}^{a}\widehat{\mathbf{v}}_{x'_{ij}} = \widehat{\lambda}_{ij}\; {}^{a}\widehat{\mathsf{P}}_{j}(\boldsymbol{x}_{ij}, \boldsymbol{p}, \boldsymbol{q})\; \widehat{\mathbf{X}}_{i} $$

- DoF observations: 2 x 10k x 1k = 20M
- Unknowns: ~1M (pts) w/ 3 DoF + 10M (scale) + 10k (orientations) w/ 6 DoF
- **~13 Mio unknown dimensions and ~20 Mio observations**

Eliminating the Scale Factors

We can eliminate the per-point scale factor by using Euclidean coordinates (instead of homogenous coordinates)

$$ {}^{a}\mathbf{x}'_{ij} + {}^{a}\widehat{\mathbf{v}}_{x'_{ij}} = \widehat{\lambda}_{ij}\ {}^{a}\widehat{\mathsf{P}}_j(\boldsymbol{x}_{ij}, \boldsymbol{p}, \boldsymbol{q})\ \widehat{\mathbf{X}}_i $$

$$ {}^{a}\boldsymbol{x}'_{ij} + {}^{a}\widehat{\boldsymbol{v}}_{x'_{ij}} = \frac{{}^{a}\widehat{\mathsf{P}_{1:2}}_{j}(\boldsymbol{x}_{ij}, \boldsymbol{p}, \boldsymbol{q})\ \widehat{\mathbf{X}}_i}{{}^{a}\widehat{\mathsf{P}_{3}}_{j}(\boldsymbol{x}_{ij}, \boldsymbol{p}, \boldsymbol{q})\ \widehat{\mathbf{X}}_i} $$

Example: ~13M unknowns to ~3M unknowns

Least Squares Problem

- Minimize the Reprojection Error:
  - $D(x, P, X) = (x - PX)^2$
  - $P, X = argmin(D(x, P, X))$
  - Choice of $D(x, P, X)$ may vary.
- Solve using Iterative solvers:
  - Gauss-Newton
  - Levenberg-Marquardt
- But requires an initial guess.
  - Where can we get it from?

Key Properties of Bundle Adjustment

Optimality

- BA is **statistically optimal**
- Exploits all observations and considers the uncertainties and correlations
- Exploits all available information
- Computes orientations, calibration parameters, distortion parameters and point locations with highest precision
- Requires an initial estimate
- Assumes Gaussian noise model

Gross Errors / Outliers

Reasons for gross errors

- **Wrong correspondences**
- (Wrong point measurements)

Gross Errors / Outliers

Reasons for gross errors

- **Wrong correspondences**
- (Wrong point measurements)

How many observations per point do we need?

Gross Errors / Outliers

Reasons for gross errors

- **Wrong correspondences**
- (Wrong point measurements)

Observations per point

- At **least 4** views to identify the observation with a gross error
- Observed points from **5 to 6 different views** typically yield good estimates

Numeric of the
Bundle Adjustment

We Cannot Solve the Linear System of BA in a Straightforward Manner

Why?

We Cannot Solve the LS in a Straightforward Manner

The linear system is **too large**

Example

- 20,000 images, 18 points per image
- Every point is measured on avg. 3 times
- 120k points yield 360k location parameters
- 120k orientation parameters
- 480k parameters from 720k observations

Least Squares Problem

- Minimize the Reprojection Error:
  - $D(x, P, X) = (x \ - PX)^2$
  - $P, X = argmin(D(x, P, X))$
  - Choice of $D(x, P, X)$ may vary.
- Solve using Iterative solvers:
  - Gauss-Newton
  - Levenberg-Marquardt

Too Large System

Solution?

- **Reduce points and views:**
  - Divide and Conquer (partition into several sets and perform BA separately)
  - Use only some points for BA and fill the gaps by Triangulation and DLT (Hybrid approach)
- **Interleave:**
  - Minimize for points (X) and projection parameters (P) alternatively!
  - Each point is estimated independently given fixed cameras; and each camera is estimated independently given fixed points, large matrix inversions can be avoided.

Summary

- Bundle Adjustment = least squares solution to relative and absolute orientation considering uncertainties
- Statistically optimal solution
- BA leads to sparse matrices
- Often sequential solution of orientation parameters and point coordinates

Literature

- Förstner, Wrobel: Photogrammetric Computer Vision, Ch. 14.1-14.3
- Triggs, McLauchlan, Hartley, Fitzgibbon: "Bundle Adjustment — A Modern Synthesis"

Slide Information

- The slides have been created by Cyrill Stachniss as part of the photogrammetry and robotics courses.
- I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.
- The photogrammetry material heavily relies on the very well written lecture notes by Wolfgang Förstner and the Photogrammetric Computer Vision book by Förstner & Wrobel.
- Parts of the robotics material stems from the great Probabilistic Robotics book by Thrun, Burgard and Fox.

Arjun Jain, ajain@cse.iitb.ac.in