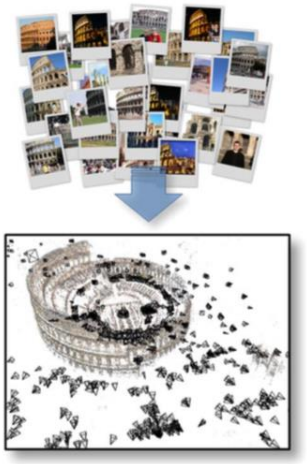
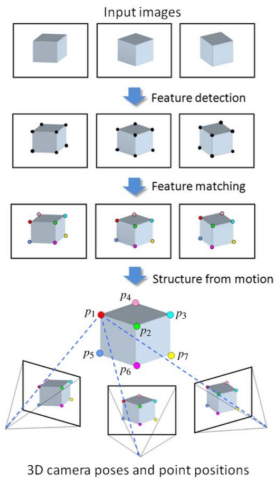
Topic 04:

Structure from Motion (SFM)

- big picture
- Euclidean & affine structure
- the factorization algorithm
- projective structure
- projective factorization
- SFM in Photo Tourism
  - projective matrices & Euclidean upgrade from F
  - incremental SFM

modern SFM





Snavely et al, "Scene Reconstruction and Visualization from community photo collections", Proc IEEE, 2010

example results: colliseum

modern SFM

Agarwal et al, Building Rome in a Day, ICCV09

ultimate objective: bundle adjustment

We wish to solve for the point positions $\mathcal{P} \equiv \{\vec{P}_n\}_{n=1}^N$ and the camera matrices $\mathcal{M} \equiv \{M_j\}_{j=1}^J$ by minimizing

$$\mathcal{O}(\mathcal{M}, \mathcal{P}) \equiv \sum_{j,n} \left\| \begin{pmatrix} x_{j,n} \\ y_{j,n} \end{pmatrix} - \frac{1}{\vec{e}_3^T M_j \vec{P}_n} \left(I_2, \vec{0} \right) M_j \vec{P}_n \right\|^2 , \quad (4)$$

where the camera matrices M_j must be of the form $M_{in,j} M_{ex,j}$, as above in (2) and (3).

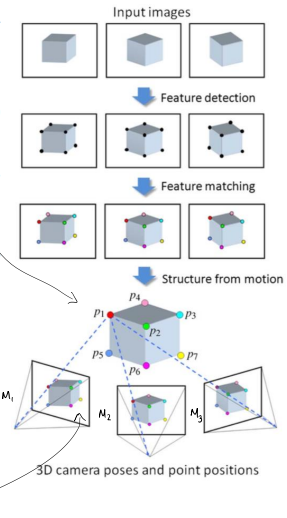
This nonlinear LS optimization problem is called **bundle adjustment**. For Gaussian IID measurement noise it is a ML estimator.

ultimate objective: bundle adjustment

The image points, $\vec{p}_{j,n}$, and the 3D scene points, $\vec{P}_n$, are related by perspective projection,

$$\vec{p}_{j,n} = \frac{1}{z_{j,n}} M_j \vec{P}_n . \quad (1)$$

- $\vec{p}_{j,n} = (x_{j,n}, y_{j,n}, 1)^T$ is given in homogeneous pixel coordinates;
- $\vec{P}_n = (P_{n,1}, P_{n,2}, P_{n,3}, 1)^T$ is also in homogeneous coordinates;
- $M_j = M_{in,j} M_{ex,j}$ is the 3×4 camera matrix formed from the product of the intrinsic and extrinsic calibration matrices;
- $z_{j,n}$ is the projective depth, $z_{j,n} = \vec{e}_3^T M_j \vec{P}_n$, where $\vec{e}_3^T = (0, 0, 1)$ (i.e., $\vec{e}_3$ is the third standard unit vector).



basic process

keypoint detection & matching, image selection

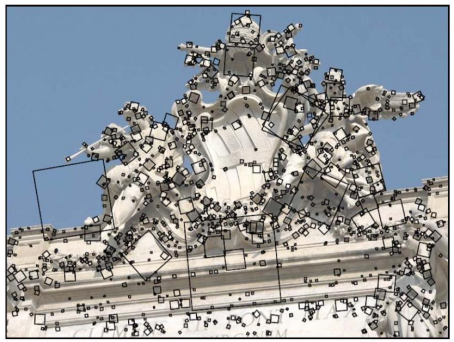
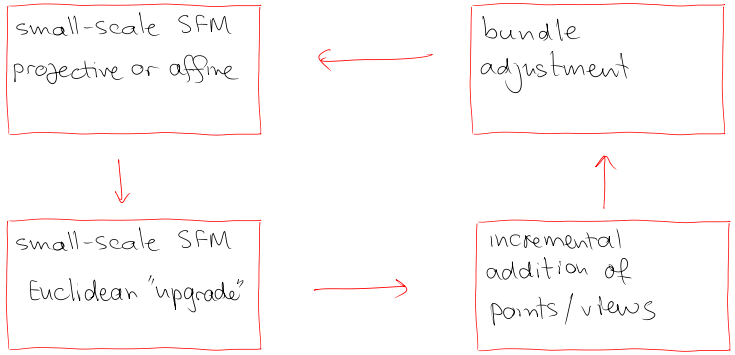


Fig. 4. *Example set of detected SIFT features. Each detected SIFT feature is displayed as a black box centered on the detected feature location. SIFT detects a canonical scale and orientation for each feature, depicted by scaling and rotating each box.*

basic process

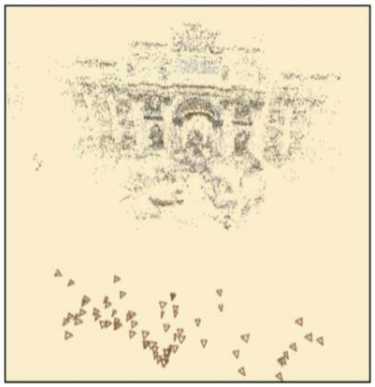
keypoint detection & matching, image selection



See Agarwal et al, Building Rome in a Day, ICCV 09, Computer 2010

basic process

keypoint detection & matching, image selection



bundle adjustment

↑

incremental addition of points / views

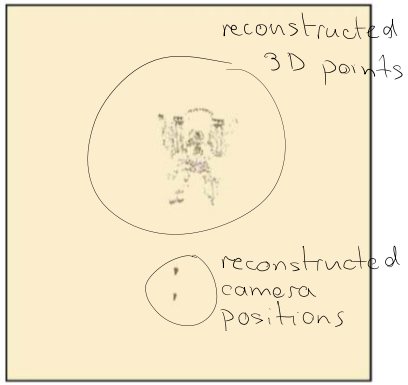
basic process

keypoint detection & matching, image selection

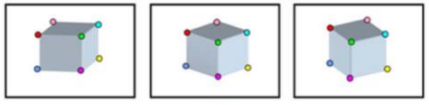
small-scale SFM
projective or affine

↓

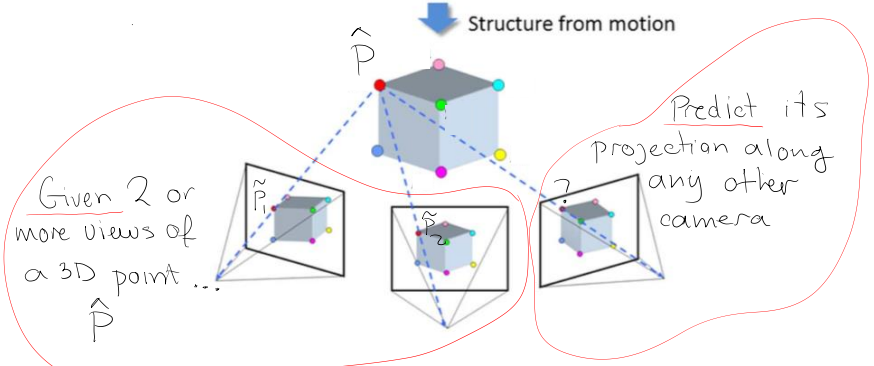
small-scale SFM
Euclidean "upgrade"



SFM: intuitive/geometric definition



Structure from motion



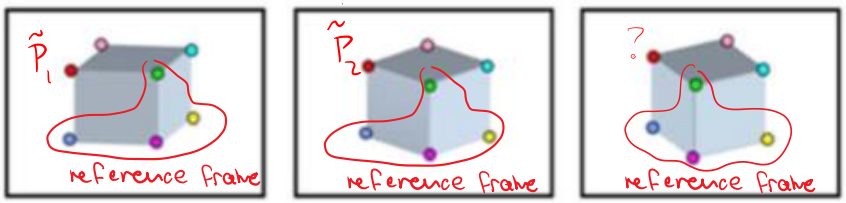
Assumptions : ① we can measure the distance between two points in the same image
② projection model (affine / perspective..)

Topic 04:

Structure from Motion (SFM)

- big picture
- **Euclidean & affine structure: definitions**
- computing affine structure from image projections
- the factorization algorithm
- projective structure
- projective factorization
- SFM in Photo Tourism
  - projective matrices & Euclidean upgrade from F
  - incremental SFM

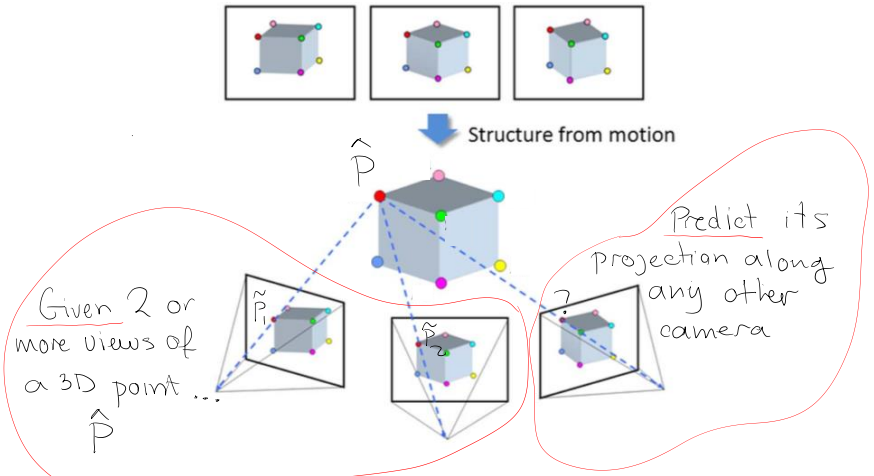
SFM: intuitive/geometric definition



Geometric intuition of the SFM process

1. identify a set of image points that will serve as a reference frame
2. express point $\hat{P}$ in terms of the reference frame (ie. assign it coordinates)
3. use those coordinates and the projection of the reference points to predict $\hat{P}$'s projection

SFM: intuitive/geometric definition



Assumptions:
1. we can measure the distance between two points in the same image
2. affine 1D, affine 2D, affine 3D, perspective

Euclidean reference frames in 2D

Euclidean

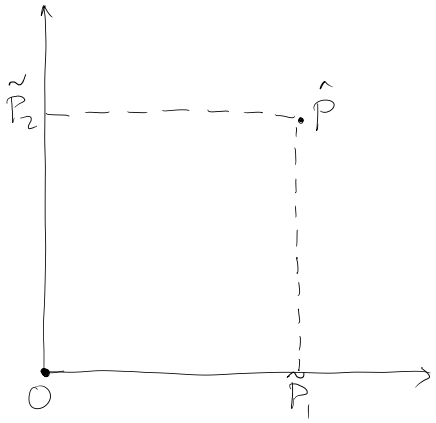
$\hat{P} = (P_1, P_2)$

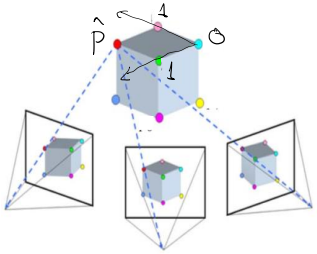
Conventions:

$\hat{P}$: 3D point

$\tilde{P}$: image projection

P_i: $\hat{P}$'s coordinates





SFM: intuitive/geometric definition

In most cases the reference points are not chosen explicitly — they are implicit in the way SFM is formulated in various algorithms

Geometric intuition of the SFM process

1. identify a set of image points that will serve as a reference frame
2. express point $\hat{P}$ in terms of the reference frame (ie. assign it coordinates)
3. use those coordinates and the projection of the reference points to predict $\hat{P}$'s projection

Euclidean reference frames in 2D

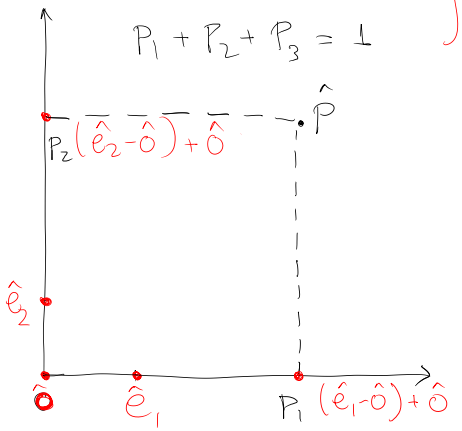
Euclidean

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{e}_2 + P_3 \hat{O}$

s.t.

$P_1 + P_2 + P_3 = 1$

called an affine sum of 3 points



To be Euclidean, the representation of $\hat{p}$ must

1. be an affine sum of 3 basis points
2. the lengths of line segments $\hat{O}\hat{e}_1$ and $\hat{O}\hat{e}_2$ must be 1
3. angle between line segments $\hat{O}\hat{e}_1$ and $\hat{O}\hat{e}_2$ is $\frac{\pi}{2}$

Euclidean reference frames in 2D

Euclidean

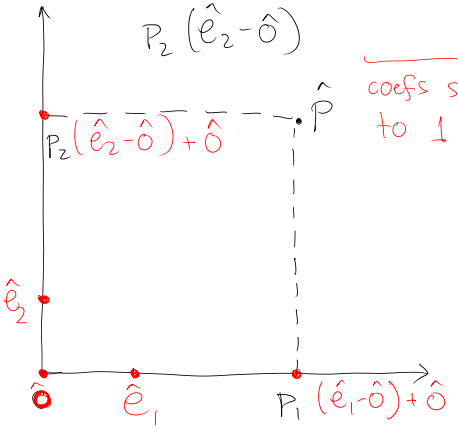
$\hat{P} = \hat{O} + P_1(\hat{e}_1 - \hat{O}) + P_2(\hat{e}_2 - \hat{O}) = (1 - P_1 - P_2)\hat{O} + P_1\hat{e}_1 + P_2\hat{e}_2$

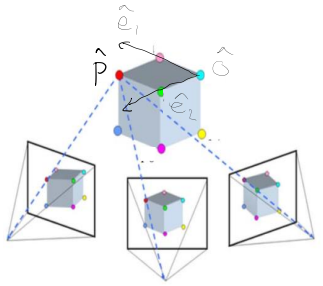
origin

unit along $\hat{O}\hat{e}_1$

unit along $\hat{O}\hat{e}_2$

coefs sum to 1





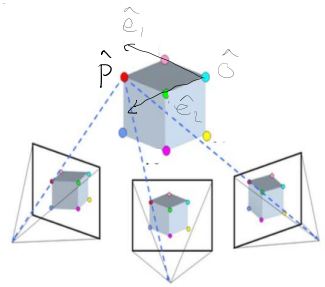
Euclidean reference frames in 2D

Euclidean

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{e}_2 + P_3 \hat{O}$

s.t.

$P_1 + P_2 + P_3 = 1$



$\hat{O}, \hat{e}_1, \hat{e}_2$ define a Euclidean frame of reference when ② & ③ hold

To be Euclidean, the representation of $\hat{p}$ must

1. be an affine sum of 3 basis points
2. the lengths of line segments $\hat{O}\hat{e}_1$ and $\hat{O}\hat{e}_2$ must be 1
3. angle between line segments $\hat{O}\hat{e}_1$ and $\hat{O}\hat{e}_2$ is $\frac{\pi}{2}$

Euclidean reference frames in 2D

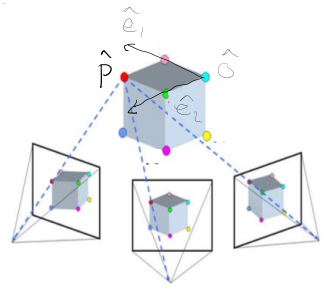
Euclidean

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{e}_2 + P_3 \hat{O}$

s.t.

$P_1 + P_2 + P_3 = 1$

called an affine sum of 3 points



To be Euclidean, the representation of $\hat{p}$ must

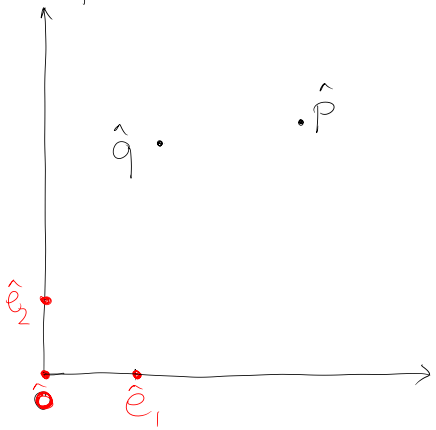
1. be an affine sum of 3 basis points
2. the lengths of line segments $\hat{O}\hat{e}_1$ and $\hat{O}\hat{e}_2$ must be 1
3. angle between line segments $\hat{O}\hat{e}_1$ and $\hat{O}\hat{e}_2$ is $\frac{\pi}{2}$

Euclidean reference frames in 2D

Euclidean

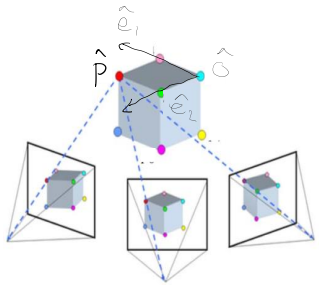
$\hat{p} = p_1 \hat{e}_1 + p_2 \hat{e}_2 + p_3 \hat{o}$

$\hat{q} = q_1 \hat{e}_1 + q_2 \hat{e}_2 + q_3 \hat{o}$



$\hat{o}, \hat{e}_1, \hat{e}_2$ define a Euclidean frame of reference

e.g. suppose we rigidly rotate the cube

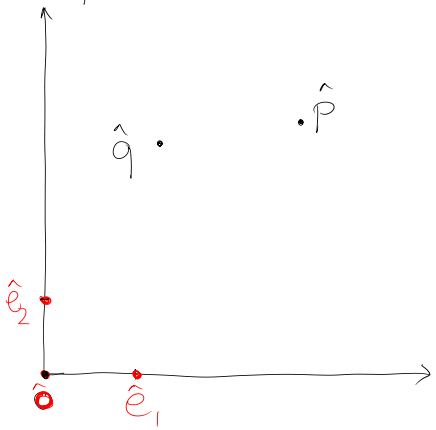


Euclidean reference frames in 2D

Euclidean

$\hat{p} = p_1 \hat{e}_1 + p_2 \hat{e}_2 + p_3 \hat{o}$

$\hat{q} = q_1 \hat{e}_1 + q_2 \hat{e}_2 + q_3 \hat{o}$



$\hat{o}, \hat{e}_1, \hat{e}_2$ define a Euclidean frame of reference

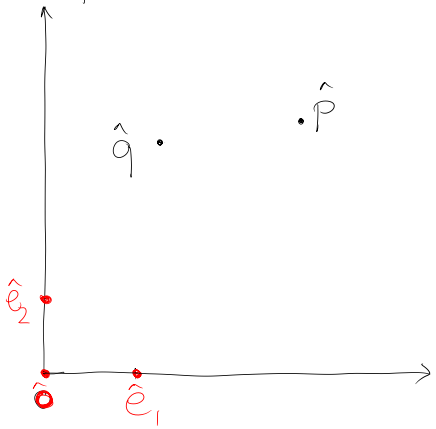
Since $\hat{p}, \hat{q}$ expressed in terms of $\hat{o}, \hat{e}_1, \hat{e}_2$, any transformation that preserves the relations between $\hat{o}, \hat{e}_1, \hat{e}_2$ will NOT affect the relation between $\hat{p}, \hat{q}, \hat{o}, \hat{e}_1, \hat{e}_2$ (distances, angles)

Euclidean structure in 2D

Euclidean

$\hat{p} = p_1 \hat{e}_1 + p_2 \hat{e}_2 + p_3 \hat{o}$

$\hat{q} = q_1 \hat{e}_1 + q_2 \hat{e}_2 + q_3 \hat{o}$



$\hat{o}, \hat{e}_1, \hat{e}_2$ define a Euclidean frame of reference

Euclidean structure:

An assignment of coordinates relative to a Euclidean frame of reference

Represented by

p_1, p_2, p_3 $\quad (p_1+p_2+p_3=1)$

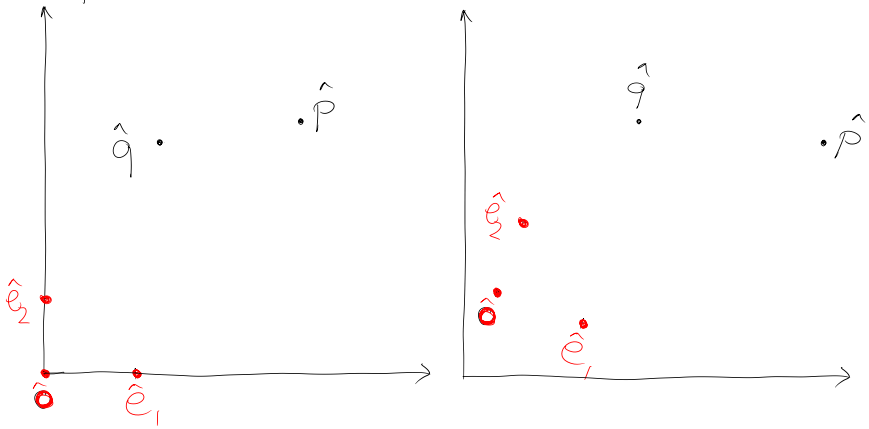
q_1, q_2, q_3 $\quad (q_1+q_2+q_3=1)$

Euclidean reference frames in 2D

Euclidean

$\hat{p} = p_1 \hat{e}_1 + p_2 \hat{e}_2 + p_3 \hat{o}$

$\hat{q} = q_1 \hat{e}_1 + q_2 \hat{e}_2 + q_3 \hat{o}$



$\hat{o}, \hat{e}_1, \hat{e}_2$ define a Euclidean frame of reference

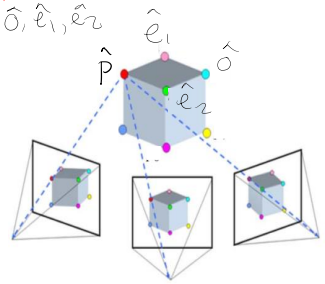
affine reference frame in 2D

Affine

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{e}_2 + P_3 \hat{O}$

$\hat{q} = q_1 \hat{e}_1 + q_2 \hat{e}_2 + q_3 \hat{O}$

e.g. if we dont know the 3D relation between $\hat{O}, \hat{e}_1, \hat{e}_2$



An affine frame of reference is defined by any non-collinear points $\hat{O}, \hat{e}_1, \hat{e}_2$ on the plane

Affine Structure

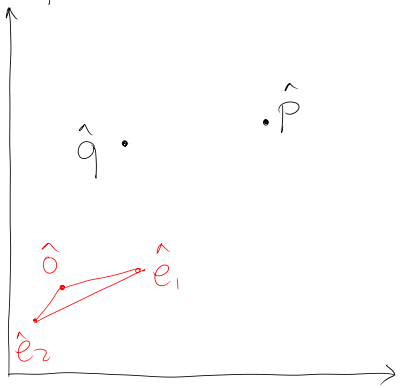
An assignment of coordinates relative to an affine frame of reference

affine reference frame in 2D

Affine

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{e}_2 + P_3 \hat{O}$

$\hat{q} = q_1 \hat{e}_1 + q_2 \hat{e}_2 + q_3 \hat{O}$



An affine frame of reference is defined by any non-collinear points $\hat{O}, \hat{e}_1, \hat{e}_2$ on the plane

Affine Structure

An assignment of coordinates relative to an affine frame of reference

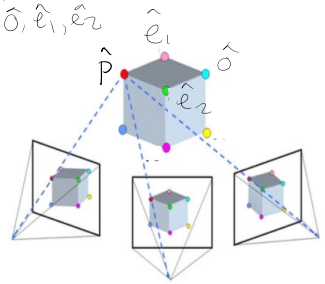
The Euclidean (aka metric) upgrade

Affine

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{e}_2 + P_3 \hat{O}$

$\hat{q} = q_1 \hat{e}_1 + q_2 \hat{e}_2 + q_3 \hat{O}$

e.g. if we dont know the 3D relation between $\hat{O}, \hat{e}_1, \hat{e}_2$



Euclidean upgrade

(*) An assignment of Euclidean 3D coordinates to the points $\hat{O}, \hat{e}_1, \hat{e}_2$ that define the affine reference frame

Once these coordinates are known, we can compute the Euclidean coords of $\hat{P}, \hat{q}$ using Eq(*)

affine structure in 2D

Affine

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{e}_2 + P_3 \hat{O}$

$\hat{q} = q_1 \hat{e}_1 + q_2 \hat{e}_2 + q_3 \hat{O}$

An affine frame of reference is defined by any non-collinear points $\hat{O}, \hat{e}_1, \hat{e}_2$ on the plane

Affine Structure

Whether structure is affine or Euclidean depends on the frame of reference, not the coords of $\hat{P}, \hat{q}$

Represented by

P_1, P_2, P_3 $\quad (P_1+P_2+P_3=1)$

q_1, q_2, q_3 $\quad (q_1+q_2+q_3=1)$

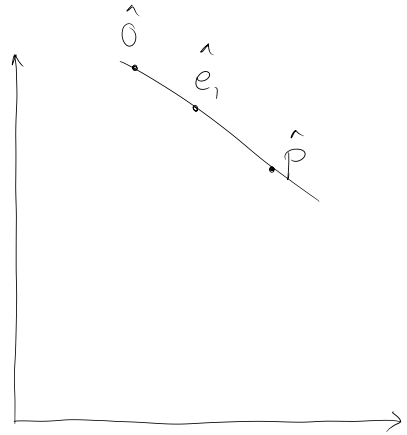
Topic 04:

Structure from Motion (SFM)

- big picture
- Euclidean & affine structure: definitions
- computing affine structure from image projections
- the factorization algorithm
- projective structure
- projective factorization
- SFM in Photo Tourism
  - projective matrices & Euclidean upgrade from F
  - incremental SFM

computing 1D affine structure from 1D images

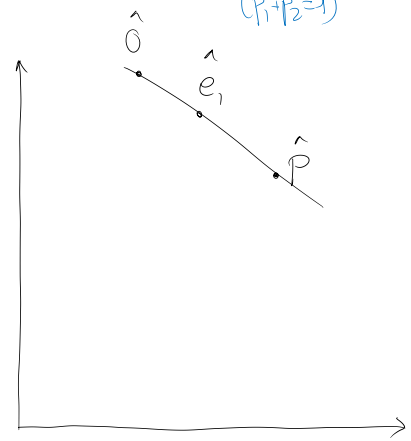
$\hat{P} = p_1 \hat{e}_1 + p_2 \hat{O}$



Goal: Compute (p_1, p_2) given the projection of $\hat{O}, \hat{e}_1, \hat{P}$ in ≥ 1 images under parallel (affine) projection

computing 1D affine structure from 1D images

$\hat{P} = p_1 \hat{e}_1 + p_2 \hat{O}$



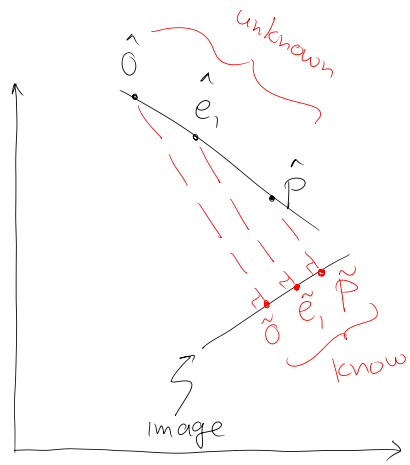
$(p_1 + p_2 = 1)$

Goal: Compute (p_1, p_2) given the projection of $\hat{O}, \hat{e}_1, \hat{P}$ in ≥ 1 images under parallel (affine) projection

$$\hat{P} = p_1 \hat{e}_1 + (1 - p_1)\hat{O}$$
$$= p_1 \hat{e}_1 + \hat{O} - p_1 \hat{O}$$
$$= p_1 (\hat{e}_1 - \hat{O}) + \hat{O}$$
$$\Rightarrow p_1 = \frac{\text{length}(\hat{O}\hat{P})}{\text{length}(\hat{O}\hat{e}_1)}$$

computing 1D affine structure from 1D images

$\hat{P} = p_1 \hat{e}_1 + p_2 \hat{O}$



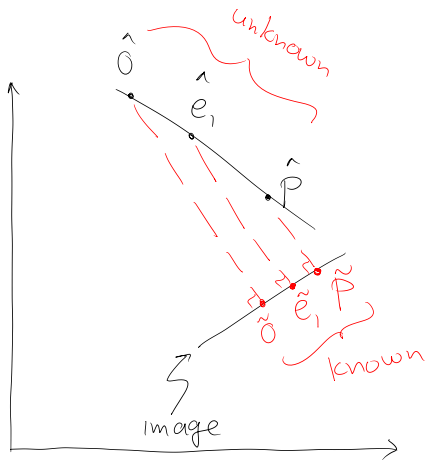
Goal: Compute (p_1, p_2) given the projection of $\hat{O}, \hat{e}_1, \hat{P}$ in ≥ 1 images under parallel (affine) projection

Question: is one image sufficient to compute p_1?

computing 1D affine structure from 1 image

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{O}$

Goal: Compute (P_1, P_2)

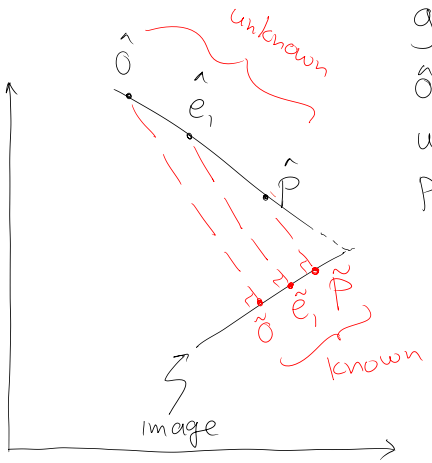


Yes, 1 orthographic view of ≥ 2 collinear points is sufficient to compute affine 1D structure

computing 1D affine structure from 1D images

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{O}$

Goal: Compute (P_1, P_2) given the projection of $\hat{O}, \hat{e}_1, \hat{P}$ in ≥ 1 images under parallel (affine) projection



From similar triangles:

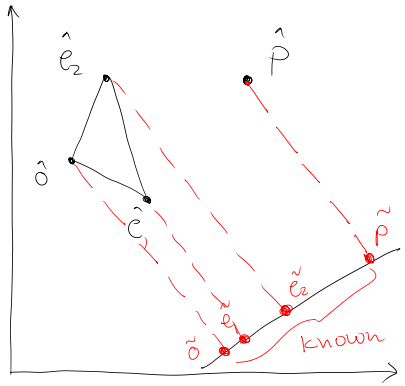
$$\frac{\text{len}(\hat{O}\hat{P})}{\text{len}(\hat{O}\hat{e}_1)} = \frac{\text{len}(\tilde{O}\tilde{P})}{\text{len}(\tilde{O}\tilde{e}_1)} = P_1$$

We also know that $P_1 + P_2 = 1$

computing 2D affine structure from 1D images

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{e}_2 + P_3 \hat{O}$

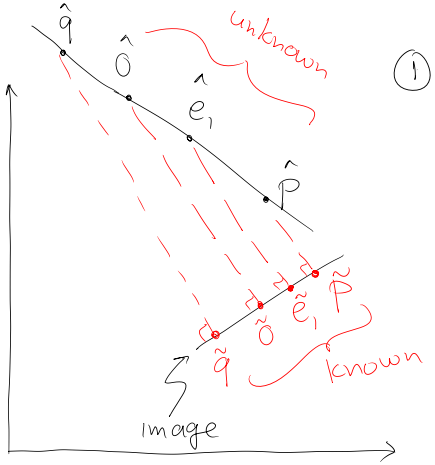
Goal: Compute (P_1, P_2, P_3) given the projection of $\hat{O}, \hat{e}_1, \hat{e}_2, \hat{P}$ in ≥ 1 images under parallel (affine) projection



computing 1D affine structure from 1 image

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{O}$

Goal: Compute (P_1, P_2)

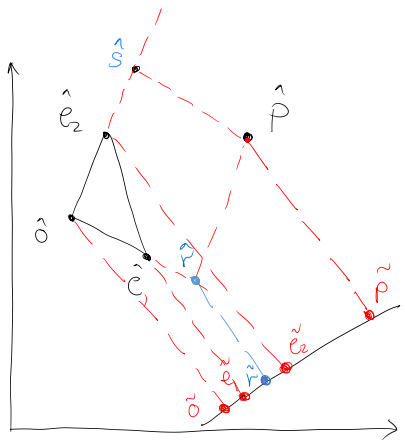


① Yes, 1 orthographic view of ≥ 2 collinear points is sufficient to compute affine 1D structure

② distances between any points $\hat{P}, \hat{q}$ known up to a scale factor determined by $\text{len}(\tilde{O}\tilde{e}_1)$

computing 2D affine structure from 1D images

$\hat{P} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{O}$



Goal: Compute (P_1, P_2, P_3) given the projection of $\hat{O}, \hat{e}_1, \hat{e}_2, \hat{P}$ in ≥ 1 images under parallel (affine) projection

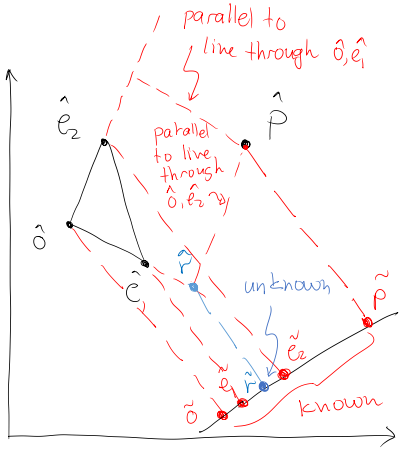
We have

$len(\tilde{O}\tilde{P}) = len(\tilde{O}\tilde{r}) + len(\tilde{r}\tilde{P})$

$P_1 \cdot len(\tilde{O}\tilde{e}_1)$ $\quad$ $len(\hat{O}\hat{s})$ // $P_2 \cdot len(\hat{O}\hat{e}_2)$

computing 2D affine structure from 1D images

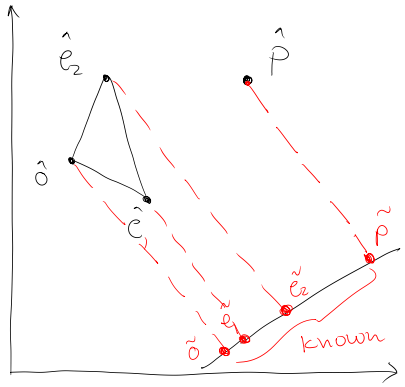
$\hat{P} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{O}$



Goal: Compute (P_1, P_2, P_3) given the projection of $\hat{O}, \hat{e}_1, \hat{e}_2, \hat{P}$ in ≥ 1 images under parallel (affine) projection

computing 2D affine structure from 1D images

$\hat{P} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{O}$

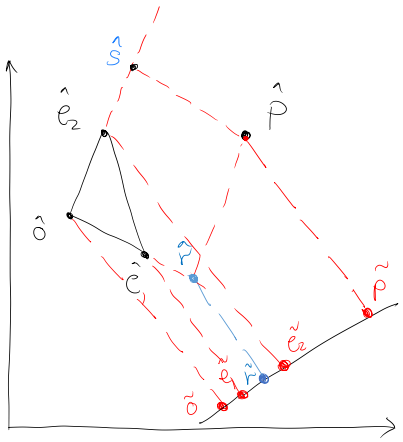


Goal: Compute (P_1, P_2, P_3) given the projection of $\hat{O}, \hat{e}_1, \hat{e}_2, \hat{P}$ in ≥ 1 images under parallel (affine) projection

- Each image gives one linear constraint on the unknowns P_1, P_2, P_3
- we also know $P_1 + P_2 + P_3 = 1$

computing 2D affine structure from 1D images

$\hat{P} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{O}$



Goal: Compute (P_1, P_2, P_3) given the projection of $\hat{O}, \hat{e}_1, \hat{e}_2, \hat{P}$ in ≥ 1 images under parallel (affine) projection

We have

$len(\tilde{O}\tilde{P}) = len(\tilde{O}\tilde{r}) + len(\tilde{r}\tilde{P})$

$\Leftrightarrow$

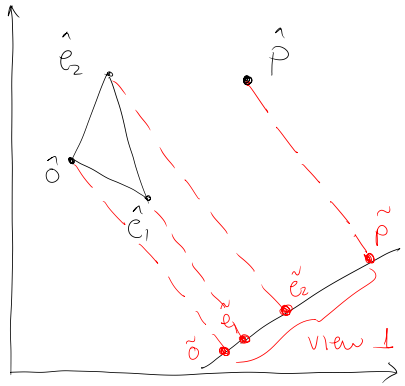
$len(\tilde{O}\tilde{P}) = P_1 len(\tilde{O}\tilde{e}_1) + P_2 len(\tilde{O}\tilde{e}_2)$

known: $len(\tilde{O}\tilde{P})$, $len(\tilde{O}\tilde{e}_1)$, $len(\tilde{O}\tilde{e}_2)$

unknown: P_1, P_2

computing 2D affine structure: 2 views, 4 points

$\hat{P} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{O}$



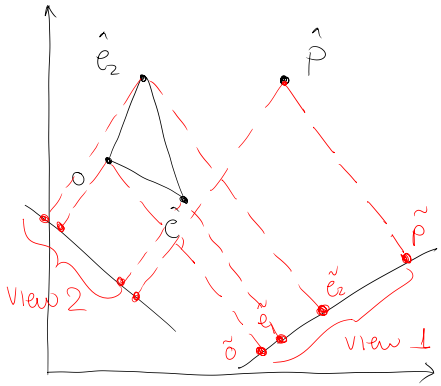
Note that the second view can be acquired under any arbitrary rearrangement of the reference

$\Uparrow$

structure recovery is possible for any 2 views of an affinely-distorted 2D pointset

computing 2D affine structure: 2 views, 4 points

$\hat{P} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{O}$



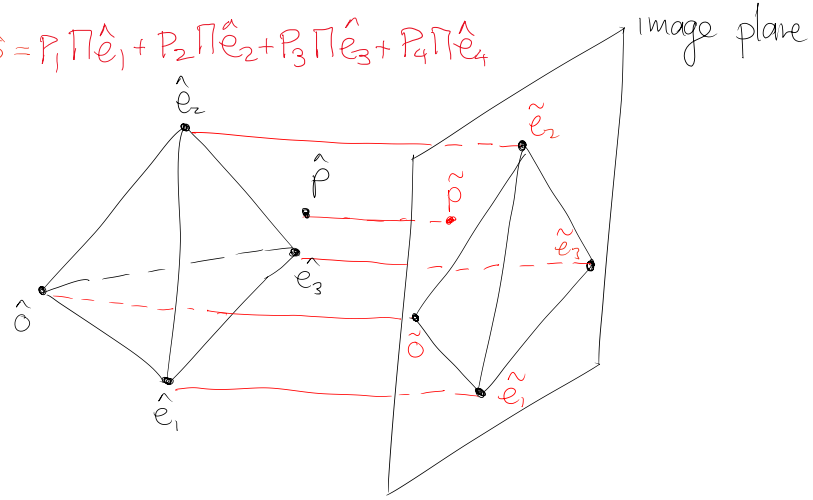
Goal: Compute (P_1, P_2, P_3) given the projection of $\hat{O}, \hat{e}_1, \hat{e}_2, \hat{P}$ in ≥ 1 images under parallel (affine) projection

$\Rightarrow$ 2 views of $N \geq 4$ points necessary & sufficient to recover affine 2D structure

computing 3D affine structure

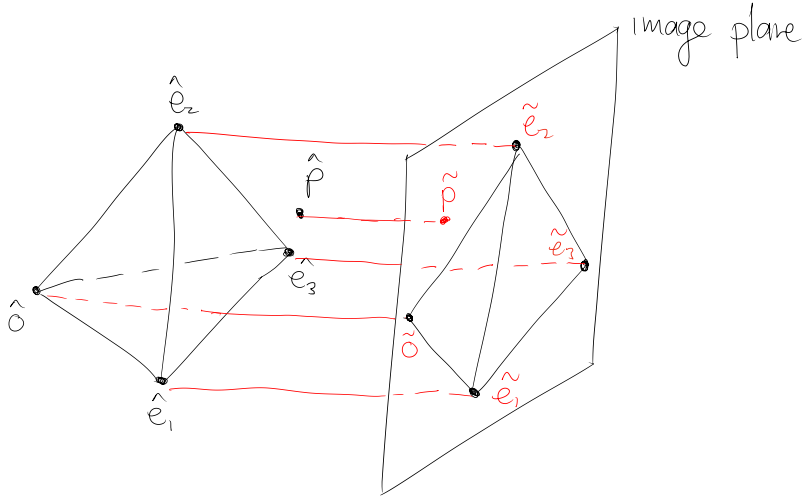
$\hat{P} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{e}_3 + P_4\hat{O} \quad \sum_{i=1}^{4} P_i = 1$

$\Pi\hat{P} = P_1\Pi\hat{e}_1 + P_2\Pi\hat{e}_2 + P_3\Pi\hat{e}_3 + P_4\Pi\hat{e}_4$



computing 3D affine structure

$\hat{P} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{e}_3 + P_4\hat{O} \quad \sum_{i=1}^{4} P_i = 1$



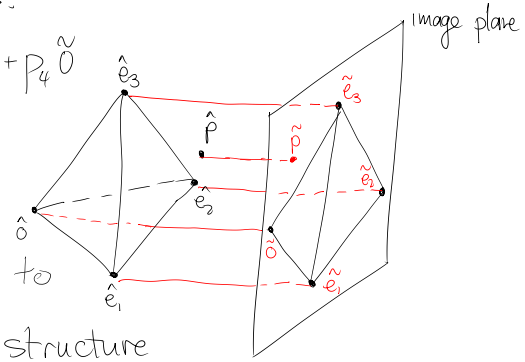
3D affine structure: constraints from 1 view

$\hat{p} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{e}_3 + P_4\hat{o} \quad \sum_{i=1}^{4} P_i = 1$

Goal: compute $P_1, \ldots P_4$ from $N \geq 1$ views

One view gives us:

$\tilde{p} = P_1\tilde{e}_1 + P_2\tilde{e}_2 + P_3\tilde{e}_3 + P_4\tilde{o}$



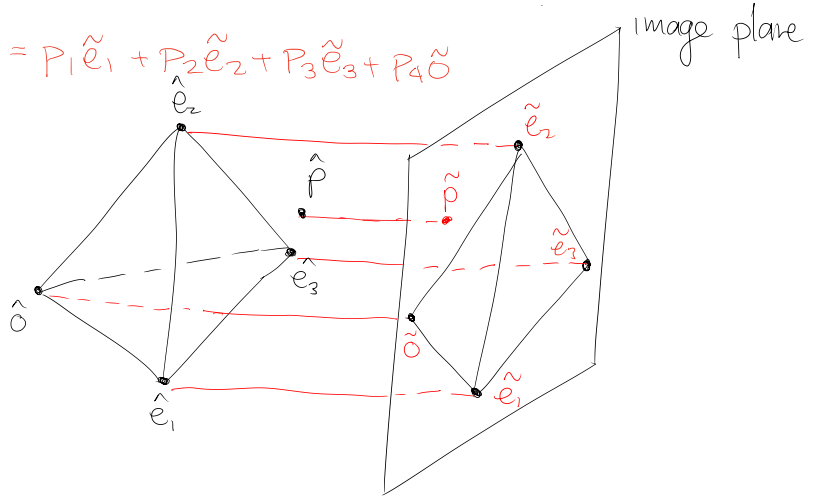
$\Rightarrow$ 2 constraints in 4 unknowns

$\Rightarrow$ need $N \geq 2$ views to get 3D affine structure

computing 3D affine structure

$\hat{p} = P_1\hat{e}_1 + P_2\hat{e}_2 + P_3\hat{e}_3 + P_4\hat{o} \quad \sum_{i=1}^{4} P_i = 1$

$\tilde{p} = P_1\tilde{e}_1 + P_2\tilde{e}_2 + P_3\tilde{e}_3 + P_4\tilde{o}$



Topic 04:

Structure from Motion (SFM)

- big picture
- Euclidean & affine structure
- the factorization algorithm
- projective structure
- projective factorization
- SFM in Photo Tourism
  - projective matrices & Euclidean upgrade from F
  - incremental SFM

3D affine structure: to read further

J. J. Koenderink and A. J. van Doorn,
"Affine Structure from Motion,"
J. Opt. Soc. Am. A, vol. 8, no 2, 1991

computing affine 3D structure from N views

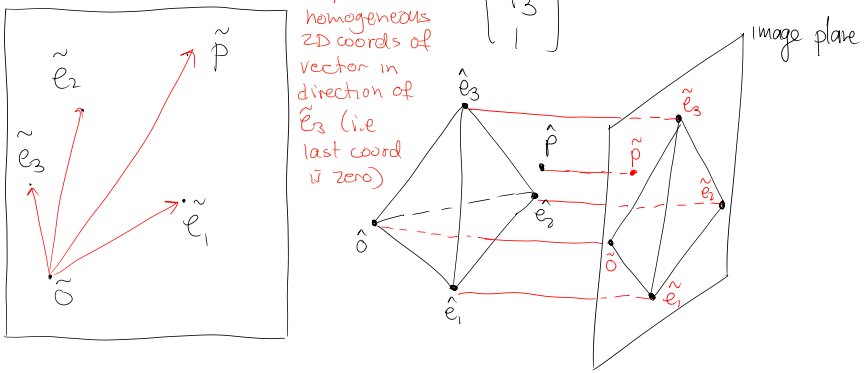
$$\tilde{P} = \begin{bmatrix} \tilde{e}_1 - \tilde{O} & \tilde{e}_2 - \tilde{O} & \tilde{e}_3 - \tilde{O} & \tilde{O} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ 1 \end{bmatrix}$$

3x4 projection matrix

homogeneous 2D coords of origin

homogeneous 3D coords of $\tilde{P}$

homogeneous 2D coords of vector in direction of $\tilde{e}_3$ (i.e last coord is zero)

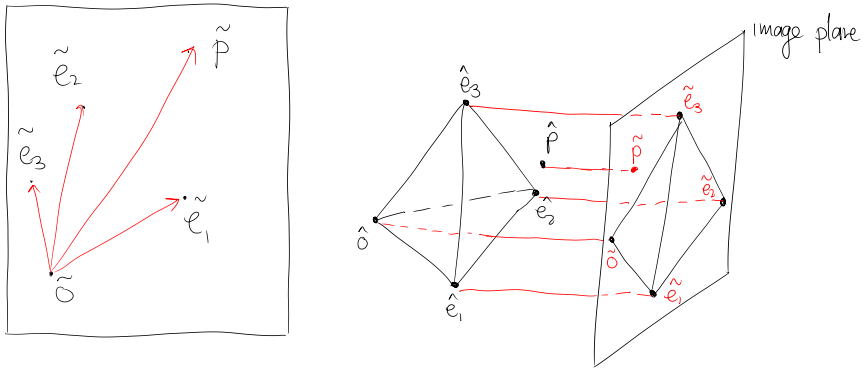


computing affine 3D structure from N views

$$\hat{P} = P_1(\hat{e}_1 - \hat{O}) + P_2(\hat{e}_2 - \hat{O}) + P_3(\hat{e}_3 - \hat{O}) + \hat{O}$$

$$\Longleftrightarrow$$

$$\tilde{P} = P_1(\tilde{e}_1 - \tilde{O}) + P_2(\tilde{e}_2 - \tilde{O}) + P_3(\tilde{e}_3 - \tilde{O}) + \tilde{O}$$



computing affine 3D structure from N views

- For the i-th input image

$$\tilde{P}_i = \begin{bmatrix} \tilde{e}_{i1} - \tilde{O} & \tilde{e}_{i2} - \tilde{O} & \tilde{e}_{i3} - \tilde{O} & \tilde{O} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ 1 \end{bmatrix}$$

last row is always [0 0 0 1] so redundant

- Equation becomes (in non-homogeneous coords):

$$\begin{bmatrix} \tilde{P}_i^x \\ \tilde{P}_i^y \end{bmatrix} - \tilde{O} = \begin{bmatrix} \tilde{e}_{i1} - \tilde{O} & \tilde{e}_{i2} - \tilde{O} & \tilde{e}_{i3} - \tilde{O} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

2D vector

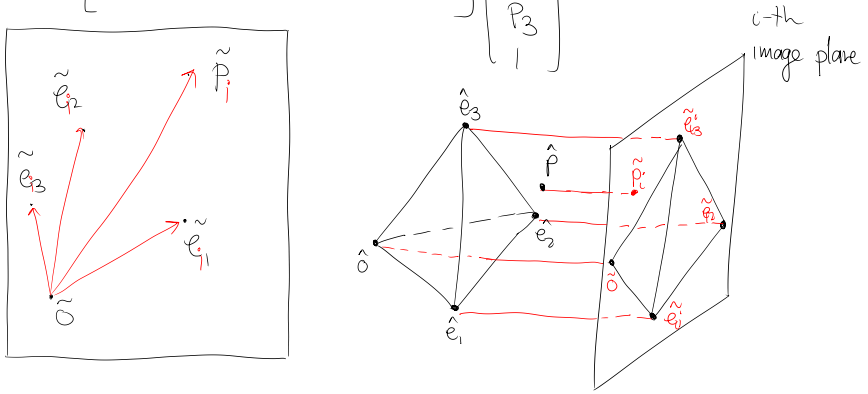
2D vectors

3D vector of affine coords

computing affine 3D structure from N views

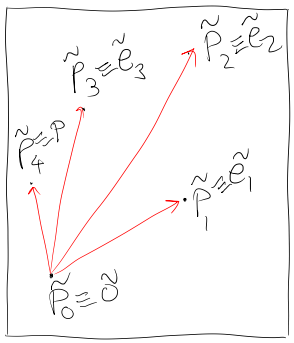
For the i-th input image

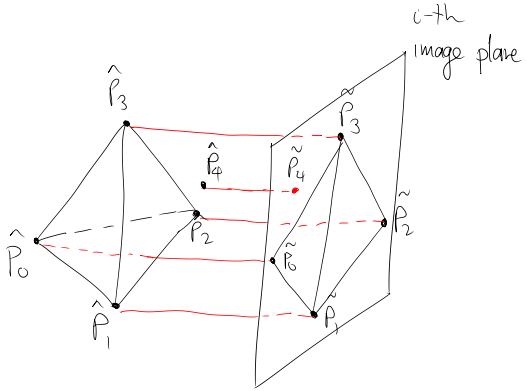
$$\tilde{P}_i = \begin{bmatrix} \tilde{e}_{i1} - \tilde{O} & \tilde{e}_{i2} - \tilde{O} & \tilde{e}_{i3} - \tilde{O} & \tilde{O} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ 1 \end{bmatrix}$$



computing affine 3D structure for M+1 points

For M+1 points in a frame, 4 act as reference frame and the rest play the role of point $\hat{P}$





computing affine 3D structure from N views

- For N frames

$$\begin{bmatrix} \bar{P}_1^x \\ \bar{P}_1^y \\ \bar{P}_2^x \\ \bar{P}_2^y \\ \vdots \\ \bar{P}_N^x \\ \bar{P}_N^y \end{bmatrix} = \begin{bmatrix} \bar{e}_{11}^x & \bar{e}_{12}^x & \bar{e}_{13}^x \\ \bar{e}_{11}^y & \bar{e}_{12}^y & \bar{e}_{13}^y \\ & \vdots & \\ \bar{e}_{N1}^x & \bar{e}_{N2}^x & \bar{e}_{N3}^x \\ \bar{e}_{N1}^y & \bar{e}_{N2}^y & \bar{e}_{N3}^y \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

where $\bar{P}_i = \tilde{P}_i - \tilde{o}_i$

$\bar{e}_{ij} = \tilde{e}_{ij} - \tilde{o}_j$

are inhomogeneous coordinates relative to projection $\tilde{o}$ of origin $\hat{o}$

affine 3D structure for N views, M+1 points ✳

- For N frames and M+1 points

$$\begin{bmatrix} \bar{P}_{11}^x & \cdots & \bar{P}_{M1}^x \\ \bar{P}_{11}^y & & \bar{P}_{M1}^y \\ \bar{P}_{12}^x & & \bar{P}_{M2}^x \\ \bar{P}_{12}^y & & \bar{P}_{M2}^y \\ \vdots & & \\ \bar{P}_{1N}^x & & \bar{P}_{MN}^x \\ \bar{P}_{1N}^y & & \bar{P}_{MN}^y \end{bmatrix} = \begin{bmatrix} \bar{P}_{11}^x & \bar{P}_{21}^x & \bar{P}_{31}^x \\ \bar{P}_{11}^y & \bar{P}_{21}^y & \bar{P}_{31}^y \\ & \vdots & \\ \bar{P}_{1N}^x & \bar{P}_{2N}^x & \bar{P}_{3N}^x \\ \bar{P}_{1N}^y & \bar{P}_{2N}^y & \bar{P}_{3N}^y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & P_M^x \\ 0 & 1 & 0 & P_M^y \\ 0 & 0 & 1 & P_M^z \end{bmatrix}$$

↑ all N centered views of a single point

↑ projection of $\hat{e}_1$ in first image

projection of $\hat{e}_3$ in N-th image

$\hat{e}_1$ $\hat{e}_3$

if the first 3 points we chosen as reference points $\hat{e}_1, \hat{e}_2, \hat{e}_3$ their affine coords are as shown

affine 3D structure for N views, M+1 points ✳

- For N frames and M+1 points

$$\begin{bmatrix} \bar{P}_{11}^x & \cdots & \bar{P}_{M1}^x \\ \bar{P}_{11}^y & & \bar{P}_{M1}^y \\ \bar{P}_{12}^x & & \bar{P}_{M2}^x \\ \bar{P}_{12}^y & & \bar{P}_{M2}^y \\ \vdots & & \\ \bar{P}_{1N}^x & & \bar{P}_{MN}^x \\ \bar{P}_{1N}^y & & \bar{P}_{MN}^y \end{bmatrix} = \begin{bmatrix} \bar{P}_{11}^x & \bar{P}_{21}^x & \bar{P}_{31}^x \\ \bar{P}_{11}^y & \bar{P}_{21}^y & \bar{P}_{31}^y \\ & \vdots & \\ \bar{P}_{1N}^x & \bar{P}_{2N}^x & \bar{P}_{3N}^x \\ \bar{P}_{1N}^y & \bar{P}_{2N}^y & \bar{P}_{3N}^y \end{bmatrix} \begin{bmatrix} P_1^x & \cdots & P_M^x \\ P_1^y & \cdots & P_M^y \\ P_1^z & & P_M^z \end{bmatrix}$$

↑ all N centered views of a single point

a single view of all points

projection matrix of N-th view

3D affine coordinates of M-th point

factorization algorithm: the rank-3 constraint

Data matrix C

$$\begin{bmatrix} p_{11}^x & \cdots & p_{M1}^x \\ p_{11}^y & & p_{M1}^y \\ p_{12}^x & & p_{M2}^x \\ p_{12}^y & & p_{M2}^y \\ \vdots & & \\ p_{1N}^x & & p_{MN}^x \\ p_{1N}^y & & p_{MN}^y \end{bmatrix} = \begin{bmatrix} P_{11}^x & P_{21}^x & P_{31}^x \\ P_{11}^y & P_{21}^y & P_{31}^y \\ & & \\ & & \\ & & \\ P_{1N}^x & P_{2N}^x & P_{3N}^x \\ P_{1N}^y & P_{2N}^y & P_{3N}^y \end{bmatrix} \begin{bmatrix} P_1^x & \cdots & P_M^x \\ P_1^y & \cdots & P_M^y \\ P_1^z & \cdots & P_M^z \end{bmatrix}$$

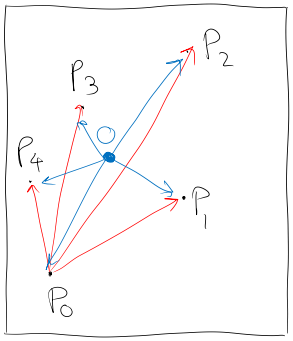
shape matrix D

camera motion matrix M

* matrix has rank 3 for $M+1 \geq 4$ centered points

factorization algorithm: data centering

Idea: No need to commit to a specific set of points for reference frame



① Under orthography, the projection of the centroid of N+1 points is the centroid of their projection

⇒ choose centroid for origin

factorization algorithm: affine structure from SVD

We seek a decomposition of C

$$\begin{bmatrix} \text{Data matrix} \\ C_{2N \times M} \\ \text{(known)} \end{bmatrix} = \begin{bmatrix} \text{Camera matrix} \\ M_{2N \times 3} \\ \text{(unknown)} \end{bmatrix} \begin{bmatrix} \text{Shape matrix} \\ D_{3 \times N} \\ \text{(unknown)} \end{bmatrix}$$

factorization algorithm: basic steps

Idea: No need to commit to a specific set of points for reference frame

② Define the basis via SVD

① Under orthography, the projection of the centroid of N+1 points is the centroid of their projection

⇒ choose centroid for origin

factorization algorithm: affine structure from SVD

Suppose $C = W^* \Sigma^* (V^*)^T$ via SVD

$$\begin{bmatrix} C_{2N\times M} \end{bmatrix} = \begin{bmatrix} W \\ 2N\times 3 \end{bmatrix} \begin{bmatrix} \Sigma_{3\times 3} \end{bmatrix} \begin{bmatrix} V_{3\times M} \end{bmatrix}$$

Since rank is at most 3, we keep first 3 colums of W^*, first 3×3 block of diagonal matrix (assuming the largest singular values are in the first 3 elements of Σ)

factorization algorithm: affine structure from SVD

Suppose $C = W^* \Sigma^* (V^*)^T$ via SVD

$$\begin{bmatrix} C_{2N\times M} \end{bmatrix} = \begin{bmatrix} W^* \\ 2N\times 2N \end{bmatrix} \begin{bmatrix} \Sigma \\ 2N\times M \end{bmatrix} \begin{bmatrix} V^{*T} \\ M\times M \end{bmatrix}$$

✳

Since rank is at most 3, we keep first 3 colums of W^*, first 3×3 block of diagonal matrix (assuming the largest singular values are in the first 3 elements of Σ)

structure & motion ambiguities

Every non-singular $A_{3\times 3}$ yields a consistent solution

$$\begin{bmatrix} \text{Data matrix} \\ C_{2N\times M} \\ \text{(known)} \end{bmatrix} = \begin{bmatrix} \text{Camera matrix} \\ M_{2N\times 3} \\ \text{11 def} \\ W \quad A \end{bmatrix} \begin{bmatrix} \text{Shape matrix} \\ D_{3\times N} \\ \text{11 def} \\ A^{-1} \Sigma \; V^T \end{bmatrix}$$

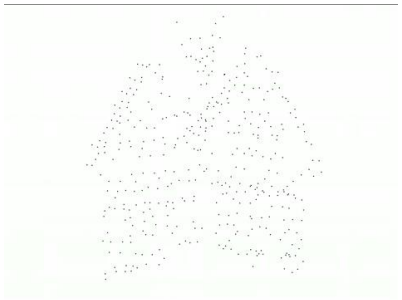
✳ $\Rightarrow$ the camera & shape matrices defined up to an invertible 3×3 linear transform

factorization algorithm: affine structure from SVD

The solution provided by factorization

$$\begin{bmatrix} \text{Data matrix} \\ C_{2N\times M} \\ \text{(known)} \end{bmatrix} = \begin{bmatrix} \text{Camera matrix} \\ M_{2N\times 3} \\ \text{11 define} \\ W \end{bmatrix} \begin{bmatrix} \text{Shape matrix} \\ D_{3\times N} \\ \text{11 define} \\ \Sigma \; V^T \end{bmatrix}$$

factorization algorithm: basic steps



factorization algorithm: basic steps

$$\begin{bmatrix} \text{Data matrix} \\ C_{2N \times M} \\ \text{(known)} \end{bmatrix}$$

1. Subtract centroid from all point projections $\tilde{p}_{ij}$
2. Form matrix C
3. Compute SVD of C: $C = W^{*} \Sigma^{*} V^{*T}$
4. Keep first 3 singular values of Σ^{*}, first 3 columns of U^{*}, first 3 rows of $(V^{*})^{T}$
5. Return $M = W$ $\quad D = \Sigma V^{T}$

factorization algorithm: basic steps

6. Optional: recover shape in a Euclidean frame of reference

↑

"Metric/Euclidean Upgrade"

1. Subtract centroid from all point projections $\tilde{p}_{ij}$
2. Form matrix C
3. Compute SVD of C: $C = W^{*} \Sigma^{*} V^{*T}$
4. Keep first 3 singular values of Σ^{*}, first 3 columns of U^{*}, first 3 rows of $(V^{*})^{T}$
5. Return $M = W$ $\quad D = \Sigma V^{T}$

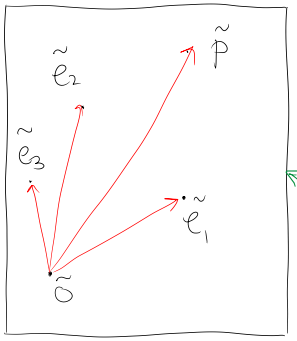
computing affine 3D structure from N views

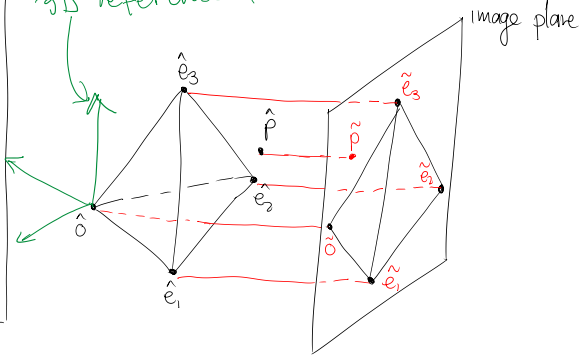
$$\hat{p} = P_1(\hat{e}_1 - \hat{o}) + P_2(\hat{e}_2 - \hat{o}) + P_3(\hat{e}_3 - \hat{o}) + \hat{o}$$

3D coordinates in Euclidean reference frame with the same origin → $\hat{p}^{e} = A\hat{p}$ ← 3D coordinates in affine reference frame

3×3 "Euclidean upgrade" matrix

orthonormal 3D reference frame





factorization algorithm: Euclidean upgrade

① Let us assume a scaled orthographic model

$$s_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R_{3\times3} = \boxed{W_1}_{2\times3} \; [A]_{3\times3} \quad *$$

- s_1: arbitrary scale factor (unknown)
- $R_{3\times3}$: 3×3 rotation matrix (unknown)
- $A_{3\times3}$: Euclidean upgrade matrix (unknown)

factorization algorithm: Euclidean upgrade

Every non-singular $A_{3\times3}$ yields a consistent solution

$$\begin{bmatrix} \text{Data matrix} \\ C_{2N\times M} \\ \text{(known)} \end{bmatrix} = \begin{bmatrix} \boxed{W_1}_{2\times3} \\ \boxed{W_2}_{2\times3} \\ \boxed{W_3}_{2\times3} \\ \boxed{W_N}_{2\times3} \end{bmatrix} [A]_{3\times3} [A^{-1}]_{3\times3} \begin{bmatrix} D \\ \cdots \\ 3\times M \end{bmatrix}$$

To compute Euclidean coords we seek A so that the W_i's represent Euclidean projection matrices

factorization algorithm: Euclidean upgrade

① Let us assume a scaled orthographic model

$$s_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R_{3\times3} = \boxed{W_1}_{2\times3} \; [A]_{3\times3}$$

③ * Define $Q = AA^T$ (= symmetric positive definite so 6 unknown elements)
for $N \geq 3$ we can solve for Q up to scale r

② Then $W_i A A^T W_i^T = s_i \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R R^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} s_i$

$$= \begin{bmatrix} s_i^2 & 0 \\ 0 & s_i^2 \end{bmatrix}$$

$\Rightarrow$ each W_i gives 2 linear constraints on AA^T homogeneous

factorization algorithm: Euclidean upgrade

① Let us assume a scaled orthographic model

$$s_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R_{3\times3} = \boxed{W_1}_{2\times3} \; [A]_{3\times3}$$

- s_1: arbitrary scale factor (unknown)
- $R_{3\times3}$: 3×3 rotation matrix (unknown)
- $A_{3\times3}$: Euclidean upgrade matrix (unknown)

② Then * $W_i A A^T W_i^T = s_i \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R R^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} s_i$

$$= \begin{bmatrix} s_i^2 & 0 \\ 0 & s_i^2 \end{bmatrix}$$

$\Rightarrow$ * each W_i gives 2 linear constraints on AA^T

factorization algorithm: Euclidean upgrade

Putting it all together: ✳

$$M = W \boxed{U_Q \Lambda_Q^{1/2}} \cdot r \cdot R$$

affine camera matrices ← Euclidean upgrade ← unknown global scale; unknown global rotation

$$D = \boxed{R^T \tfrac{1}{r}} \boxed{\Lambda_Q^{1/2} U_Q^T} \boxed{\Sigma V^T}$$

unknown rotation & scale; Euclidean upgrade inverse; affine shape matrix

factorization algorithm: Euclidean upgrade

④ We now perform an eigenvalue decomposition of Q to get $Q = U_Q \Lambda_Q U_Q^T$

✳ $\Rightarrow \boxed{A = U_Q \Lambda_Q^{1/2}}$ (assuming non-negative eigenvalues)

③ Define $Q = AA^T$ (= symmetric positive definite so 6 unknown elements)

for $N \geq 3$ we can solve for Q up to scale

② Then $W_i A A^T W_i^T = s_i \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R R^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} s_i$

$$= \begin{bmatrix} s_i^2 & 0 \\ 0 & s_i^2 \end{bmatrix}$$

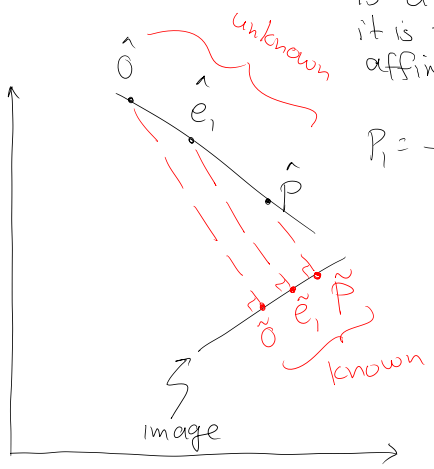
$\Rightarrow$ each W_i gives 2 linear constraints on AA^T homogeneous

affine-invariant representations

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{O}$

The affine coordinate P_1 is an affine invariant because it is not affected by affine transformations & projections

$$P_1 = \frac{\text{len}(\hat{P}\hat{O})}{\text{len}(\hat{e}_1 \hat{O})} = \frac{\text{len}(\tilde{P}\tilde{O})}{\text{len}(\tilde{e}_1 \tilde{O})}$$



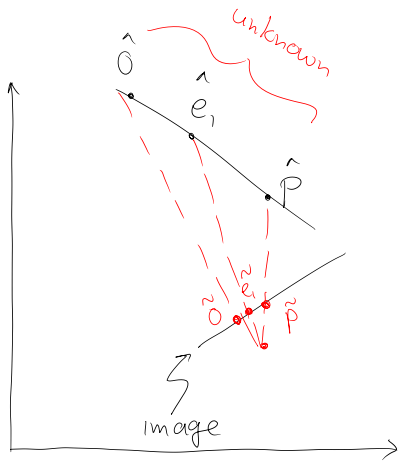
Topic 04:

Structure from Motion (SFM)

- big picture
- Euclidean & affine structure
- the factorization algorithm
- projective structure
- projective factorization
- SFM in Photo Tourism
  - projective matrices & Euclidean upgrade from F
  - incremental SFM

projectively-invariant representations

78

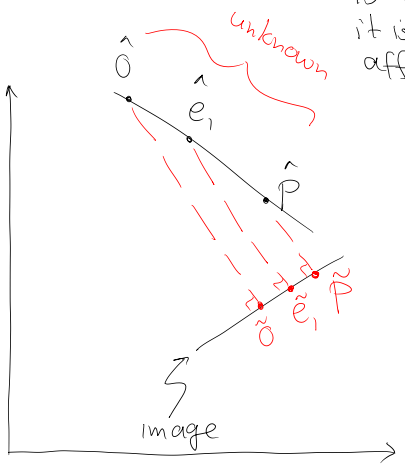


Goal: Represent $\hat{p}$ in a way that is NOT affected by perspective projection

affine-invariant representations

77

$\hat{P} = P_1 \hat{e}_1 + P_2 \hat{O}$



The affine coordinate P_1 is an affine invariant because it is not affected by affine transformations & projections

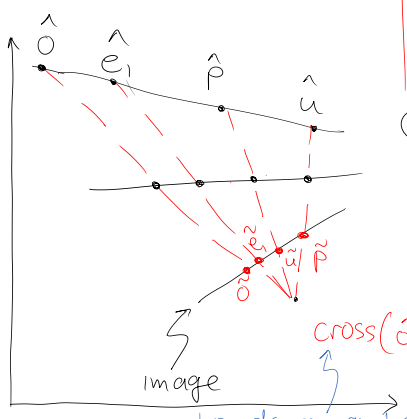
Under orthography, given the reference frame $\tilde{O}, \tilde{e}_1$ we represented $\tilde{p}$ using a single invariant ratio

$$\frac{\text{len}(\tilde{p}\,\tilde{o})}{\text{len}(\tilde{e}_1\,\tilde{o})}$$

the cross-ratio is a projective invariant

80

① Need a 3rd point, $\hat{u}$, to define "unit" on reference frame



Goal: Represent $\hat{p}$ in a way that is NOT affected by perspective projection

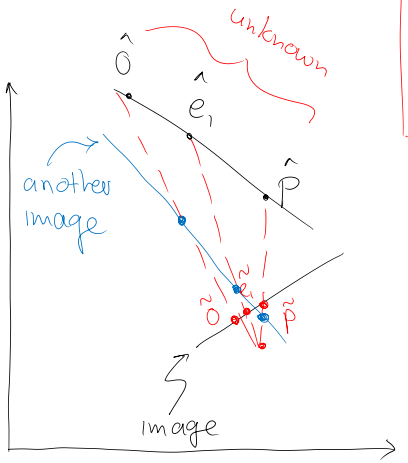
② Then the cross-ratio is invariant:

$$\text{cross}(\hat{o},\hat{e}_1,\hat{p},\hat{u}) = \frac{\frac{\text{len}(\tilde{p}\,\tilde{o})}{\text{len}(\tilde{p}\,\tilde{e}_1)}}{\frac{\text{len}(\tilde{u}\,\tilde{o})}{\text{len}(\tilde{u}\,\tilde{e}_1)}}$$

depends on angles between rays not point projections

projectively-invariant representations

79



Goal: Represent $\hat{p}$ in a way that is NOT affected by perspective projection

But ratios of distances are NOT preserved under perspective projection!

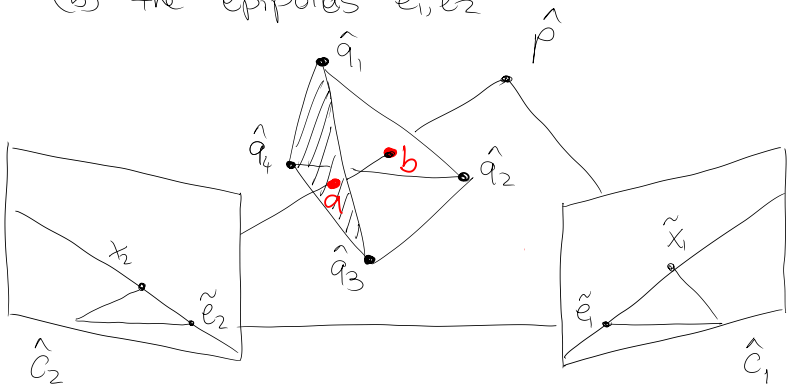
Are there projectively invariant quantities?

feasibility of computing projective structure from 2 views: a geometric proof

Let us reconstruct a point $\hat{p}$ given

(a) the projection of 4 non-coplanar points in 2 views: $\hat{a}_1 \hat{a}_2 \hat{a}_3 \hat{a}_4$

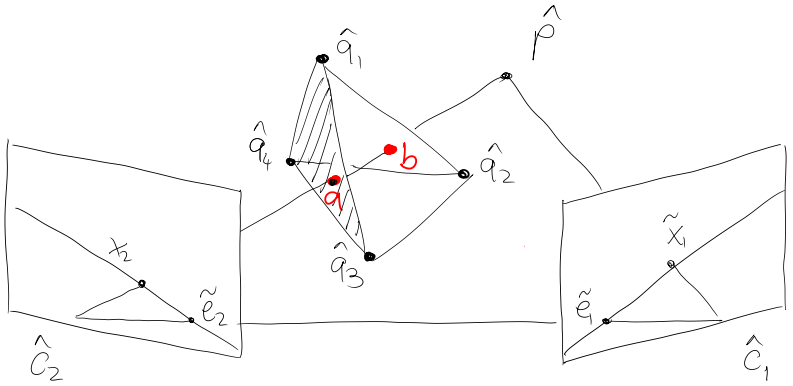
(b) the epipoles $\tilde{e}_1, \tilde{e}_2$



feasibility of computing projective structure from 2 views: a geometric proof

Let us reconstruct a point $\hat{p}$ given

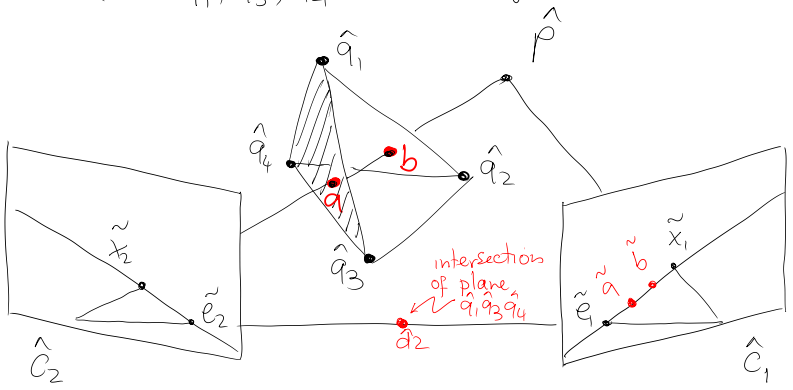
(a) the projection of 4 non-coplanar points in 2 views: $\hat{a}_1 \hat{a}_2 \hat{a}_3 \hat{a}_4$



feasibility of computing projective structure from 2 views: a geometric proof

Localizing $\tilde{a}$:

- a lies on the plane containing $\hat{q}_1 \hat{q}_3 \hat{q}_4 \hat{a}_2$
- these points project to $\tilde{q}_1, \tilde{q}_3, \tilde{q}_4, \tilde{e}_1$ in image 1 and $\tilde{q}_1', \tilde{q}_3', \tilde{q}_4', \tilde{e}_2$ in image 2

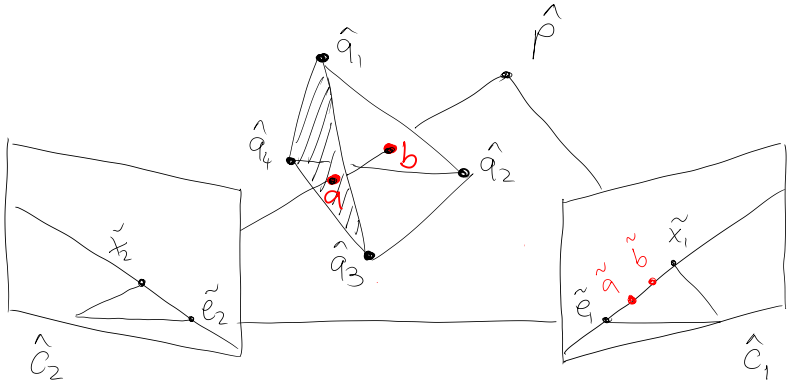


feasibility of computing projective structure from 2 views: a geometric proof

We will do this by computing a cross-ratio that localizes $\hat{p}$ uniquely:

$$\text{cross}(\hat{C}_2, a, b, \hat{p}) = \text{cross}(\tilde{e}_1, \tilde{a}, \tilde{b}, \tilde{x}_1)$$

can compute this as long as we can locate $\tilde{a}, \tilde{b}$



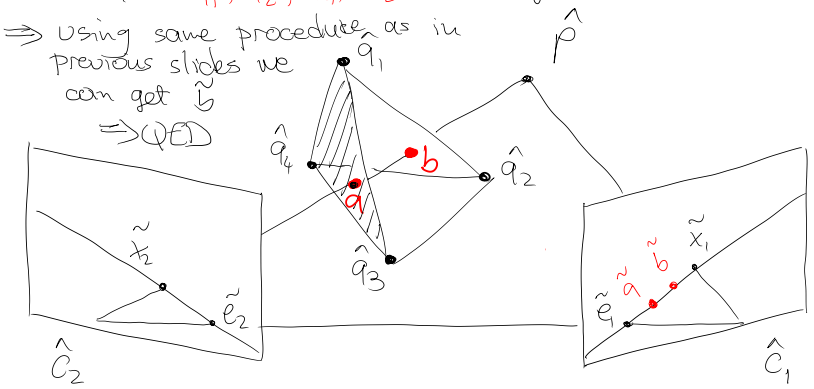
feasibility of computing projective structure from 2 views: a geometric proof

Localizing $\tilde{b}$

- b lies on the plane containing $\hat{q}_1 \hat{q}_2 \hat{q}_4 \hat{d}_1$
- these points project to $\tilde{q}_1, \tilde{q}_2, \tilde{q}_4, \tilde{e}_1$ in image 1 and $\tilde{q}_1', \tilde{q}_2', \tilde{q}_4', \tilde{e}_2$ in image 2

$\Rightarrow$ using same procedure as in previous slides we can get $\tilde{b}$

$\Rightarrow$ QED



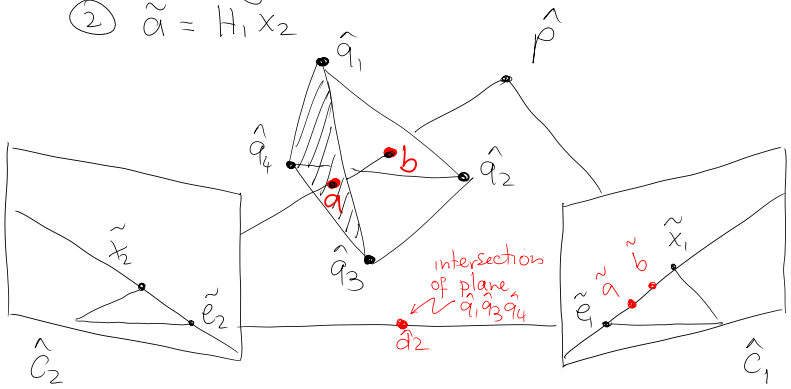
feasibility of computing projective structure from 2 views: a geometric proof

Localizing $\tilde{a}$:

- a lies on the plane containing $\hat{q}_1 \hat{q}_3 \hat{q}_4 \hat{d}_2$

$\Rightarrow$ ① compute the homography H_1 mapping image 2 to image 1

② $\tilde{a} = H_1 x_2$



Topic 04:

Structure from Motion (SFM)

- big picture
- Euclidean & affine structure
- the factorization algorithm
- projective structure
- projective factorization
- SFM in Photo Tourism
  - projective matrices & Euclidean upgrade from F
  - incremental SFM

projective structure from 2 views: to read further

For more details on this construction:

A. Shashua, "Projective Depth: A geometric invariant for 3D reconstruction from two perspective/orthographic views and for visual recognition," Proc. ICCV 1993

Topic 04:

Structure from Motion (SFM)

- big picture
- Euclidean & affine structure
- the factorization algorithm
- projective structure
- projective factorization
- SFM in Photo Tourism
  - projective matrices & Euclidean upgrade from F
  - incremental SFM

projective factorization (see notes)

Suppose we knew the projective depths $z_{j,n}$. We could then build a new data matrix C in terms of the 3D points $z_{j,n}\vec{p}_{j,n}$. Accordingly, we can form $C = [z_{j,n}\,\vec{p}_{j,n}]$, by stacking the depth-scaled image points just as we did in the orthographic case. Now C is $3J \times N$.

The perspective projection of the scene point $\vec{P}_n$ onto the j^{th} image plane (1) can be written as

$$z_{j,n}\,\vec{p}_{j,n} \;=\; M_j\,\vec{P}_n\,,$$

where M_j is now 3×4, and $\vec{P}_n$ is 4×1.

By stacking up the camera matrices, M_j, to form the $3J \times 4$ matrix M, and letting $P = (\vec{P}_1, \ldots, \vec{P}_N)$ be the $4 \times N$ shape matrix, we obtain a factorization of the data matrix:

$$C \;=\; MP\,. \qquad (19)$$

Of course, this assumes we knew the correct depths $z_{j,n}$ that we used to form C. And as above, from the form of (19) it is clear that C is, at most, rank 4 (ignoring noise).

SfM in Photo Tourism: correspondence-finding

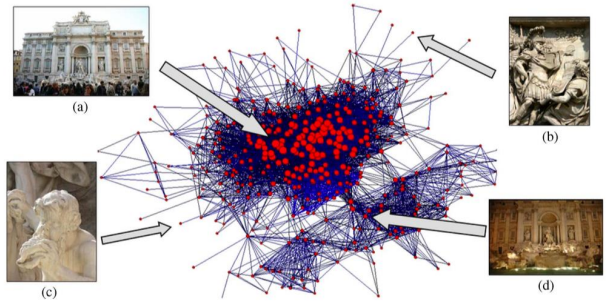


Fig. 6. *Image connectivity graph for the Trevi Fountain. This graph contains a node (red dot) for each image in a set of photos of the Trevi Fountain, and an edge between each pair of photos with matching points. The size of a node is proportional to its degree. There are two dominant clusters corresponding to daytime photos [e.g., image (a)] and nighttime photos [image (d)]. Similar views of the facade are clustered together in the center of the graph, while nodes in the periphery of the graph, e.g., (b) and (c), are more unusual (often closeup) views. An interactive version of this graph can be found at http://phototour.cs.washington.edu/imagegraphs/Trevi/.*

SfM in Photo Tourism: correspondence-finding

1. Pairwise image matching
   - SIFT for keypoint detection
   - for every image pair
     1. retrieve SIFT detector matches
     2. RANSAC + 8-point algorithm for F-matrix estimation
     3. followed by non-linear refinement

SfM in Photo Tourism: 2-view shape & motion

3. Processing I_1, I_2:

- compute projection matrices W_1, W_2
- compute the Euclidean projection matrices and the reconstructed point coordinates

SfM in Photo Tourism: initial image pair selection

2. Initial pair selection
   - find pair I_1, I_2 with
     1. large # of successful matches
     2. matches not consistent with a plane homography

SfM in Photo Tourism: incremental SfM

4. Current state:
   - M points with already-estimated 3D coords
   - Euclidean projection matrices for images $I_1, \ldots, I_k$

Adding a new image I_{k+1}:

1. initialize the projection matrix from the projection of previously-reconstructed points that also project to I_{k+1} (using DLT algorithm)

SfM in Photo Tourism: 2-view shape & motion

Fig. 7. *Incremental SfM. Our incremental SfM approach reconstructs the scene a few cameras at a time. This sequence of images shows the Trevi data set at three different stages of incremental reconstruction. Left: the initial two-frame reconstruction. Middle: an intermediate stage, after 60 images have been added. Right: the final reconstruction with 360 photos.*

Topic 04:

Structure from Motion (SFM)

- big picture
- Euclidean & affine structure
- the factorization algorithm
- projective structure
- projective factorization
- SFM in Photo Tourism
  - projective matrices & Euclidean upgrade from F
  - incremental SFM

SfM in Photo Tourism: incremental SfM

4. Adding a new image I_{k+1}:
   1. initialize the projection matrix from the projection of previously-reconstructed points that also project to I_{k+1} (using DLT algorithm)
   2. Add new matches to the reconstruction
   3. Run a non-linear optimizer to optimize projection matrices & point positions

projection matrices from the F-matrix

- compute projection matrices W_1, W_2 from fundamental matrix F

2. It is possible to show that W_2 can be chosen as (see Hartley & Zisserman, 2000, p. 237)

* $W_2 = [[e_2]_\times F \mid e_2]$

left epipole, ie. $e_2^T F = 0$

projection matrices from the F-matrix

- compute projection matrices W_1, W_2 from fundamental matrix F

1. Choose W_1 so that camera 1 defines the projective frame of reference

* $W_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Euclidean upgrade

- compute projection matrices W_1, W_2
- compute the Euclidean projection matrices and the reconstructed point coordinates

① make simplifying assumption that pixels are square, image has no skew and image intersection with optical axis is known

Euclidean upgrade

- compute projection matrices W_1, W_2
- compute the Euclidean projection matrices and the reconstructed point coordinates

Euclidean upgrade: constraints from rotations

So

$$W_i \Omega = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix} R_i$$

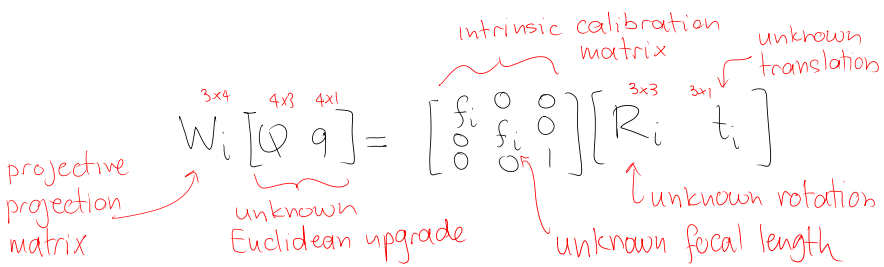
$$W_i \Omega \Omega^T W_i^T = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix} R_i R_i^T \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(↑ def)

$$W_i \Omega_\infty W_i^T = \begin{bmatrix} f_i^2 & 0 & 0 \\ 0 & f_i^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow$$

5 equations per view (6 elements − 1 uknown focal length)

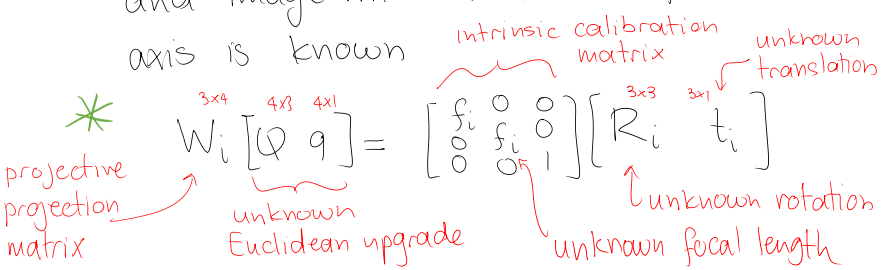
(Ω_∞: 4×4 symmetric matrix ⇒ 10 unknowns)



Euclidean upgrade: Euclidean projection matrix

- compute the Euclidean projection matrices and the reconstructed point coordinates

① make simplifying assumption that pixels are square, image has no skew and image intersection with optical axis is known



Topic 04:

Structure from Motion (SFM)

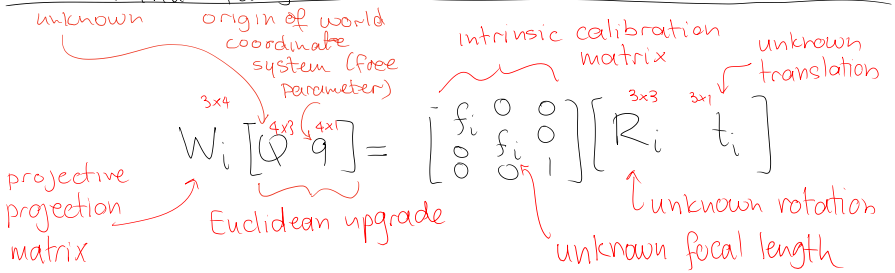
- big picture
- Euclidean & affine structure
- the factorization algorithm
- projective structure
- projective factorization
- SFM in Photo Tourism
  - projective matrices & Euclidean upgrade from F
  - incremental SFM

Euclidean upgrade: constraints from rotations

⇒ We can compute ω^* from 2 views

⇒ Once ω^* is known, we can compute the upgrade matrix using diagonalization of ω^* (see notes p. 23 and Forsyth & Ponce p. 293)

⇒ Once Q is known, we can choose an arbitrary q and thus fully determine R_i and t_i



adding cameras: exterior orientation & intrinsics

4. Adding a new image I_{k+1}:
   1. initialize the projection matrix from the projection of previously-reconstructed points that also project to I_{k+1} (using DLT algorithm)
   2. Add new matches to the reconstruction
   3. Run a non-linear optimizer to optimize projection matrices & point positions

SfM in Photo Tourism: 2-view shape & motion

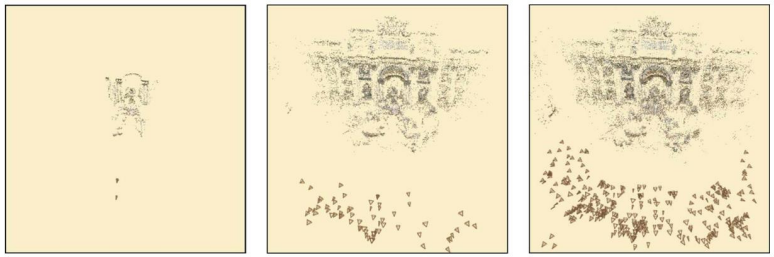


Fig. 7. *Incremental SfM. Our incremental SfM approach reconstructs the scene a few cameras at a time. This sequence of images shows the Trevi data set at three different stages of incremental reconstruction. Left: the initial two-frame reconstruction. Middle: an intermediate stage, after 60 images have been added. Right: the final reconstruction with 360 photos.*

DLT algorithm for projection matrix estimation

Given $M \geq 6$ correspondences between world points and image points, compute projection matrix Π

Basic relation:

$$x_i \cong \Pi X_i$$

- x_i: known homogeneous 2D coords
- $\cong$: equality up to scale
- X_i: known homogeneous 3D coords

After computing Π, we express it in the form $\begin{bmatrix} f & 0 \\ 0 & f \\ & & 1 \end{bmatrix} [R \; t]$ for some f and rotation matrix R

(see Hartley & Zisserman p. 143, p. 170)

DLT algorithm for projection matrix estimation

Given $M \geq 6$ correspondences between world points and image points, compute projection matrix Π

Basic relation:

$$x_i \cong \Pi X_i$$

- x_i: known homogeneous 2D coords
- $\cong$: equality up to scale
- X_i: known homogeneous 3D coords

$$\Rightarrow \quad x_i \times \Pi X_i = 0$$ (linear in the 12 elements of Π)

$\Rightarrow$ 2 linearly-independent eqs per point so ≥ 6 points are needed to get Π

bundle adjustment: objective function

objective: minimize total reprojection error

$$\sum_{i=1}^{N} \sum_{j=1}^{M} w_{ij} \| x_{ij} - P(\Theta_i, X_{ij}) \|^2$$

- P: projection operation
- w_{ij}: 1 iff point j appears in image i
- x_{ij}: 2D image coordinates of point j in image i
- Θ_i: params of camera i
- X_{ij}: 3D coordinates of point j in image i

(ω_i, f_i, c_i)

- ω_i: 3 unknown rotation angles
- f_i: unknown focal length
- c_i: known image center

bundle adjustment

1. initialize the projection matrix from the projection of previously-reconstructed points that also project to I_{k+1} (using DLT algorithm)
2. Add new matches to the reconstruction
3. Run a non-linear optimizer to optimize projection matrices & point positions

bundle adjustment: focal length estimation

two sources of information:

① image EXIF data focal length entry

② result of DLT algorithm

- prefer ① unless there is a big difference between the two (then use ②)

$(\vec{\omega}_i, f_i, \vec{c}_i)$

- $\vec{\omega}_i$: 3 unknown rotation angles
- f_i: unknown focal length
- $\vec{c}_i$: unknown camera center

bundle adjustment: rotation matrix representation

we already have an initial estimate R_j
so we optimize an incremental adjustment

$$R(\theta_i, \vec{n}_i) = \left(I + \sin\theta [\vec{n}]_x + (1-\cos\theta)[n]_x^2\right) R_j$$

with $\vec{\omega}_i = \theta_i \vec{n}_i$ and $\vec{n}_i$ unit vector

(ω_i, f_i, c_i)

- ω_i: 3 unknown rotation angles
- f_i: unknown focal length
- c_i: known image center

bundle adjustment

objective: minimize total reprojection error

$$\sum_{i=1}^{N} \sum_{j=1}^{M} w_{ij} \| x_{ij} - P(\Theta_i, X_{ij}) \|^2$$

- projection operation
- w_{ij}: 1 iff point j appears in image i
- x_{ij}: 2D image coordinates of point j in image i
- Θ_i: params of camera i
- X_{ij}: 3D coordinates of point j in image i

** the objective typically optimized using sparse methods.

bundle adjustment: projection function

objective: minimize total reprojection error

$$\sum_{i=1}^{N} \sum_{j=1}^{M} w_{ij} \| x_{ij} - P(\Theta_i, X_{ij}) \|^2$$

- projection operation
- w_{ij}: 1 iff point j appears in image i
- x_{ij}: 2D image coordinates of point j in image i
- Θ_i: params of camera i
- X_{ij}: 3D coordinates of point j in image i

$$P'(\Theta_i, X_{ij}) = \begin{bmatrix} f_i & 0 \\ 0 & f_i \end{bmatrix} R_i (X_{ij} - c_i) \qquad P = \begin{bmatrix} P_x'/P_z' & P_y'/P_z' \end{bmatrix}$$

- $\begin{bmatrix} f_i & 0 \\ 0 & f_i \end{bmatrix}$: intrinsics
- R_i: rotation matrix
- c_i: camera center
- perspective division

to read further

Hartley & Zisserman, *Multiple View Geometry in Computer Vision*, 2000

Triggs et al, "Bundle Adjustment – A modern Synthesis," *Vision Algorithms: Theory and Practice*, LNCS 1883, 2000

Snavely et al, "Photo Tourism: Exploring Photo Collections in 3D," ACM Siggraph 2006

Agrawal et a, "Building Rome in a Day," *Proc. ICCV 2009*

Pollefeys et al, Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters, *IJCV* 1999

Recent results: Flow-based methods for SFM