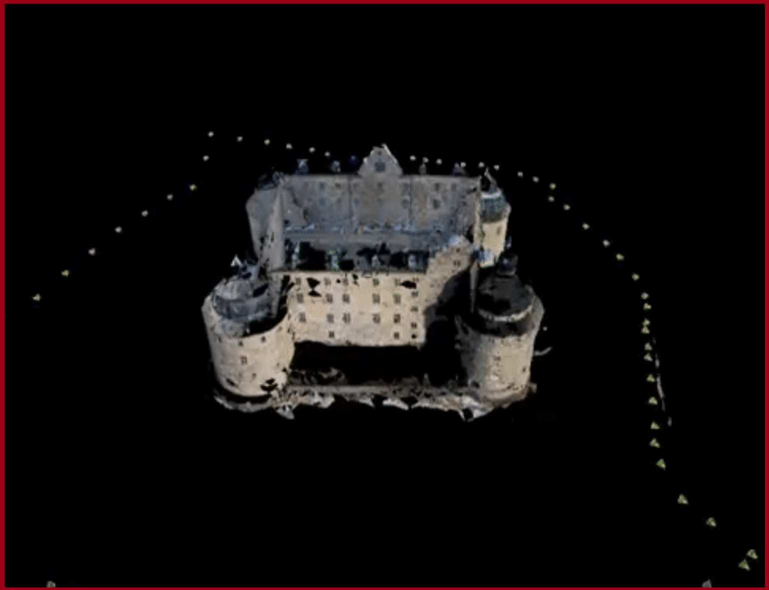
UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE



3D AUGMENTED REALITY

A.A. 2020/2021

LECTURE 15 – SFM

Simone Milani
Room 216
DEI A

Phone: 049 827 7641
E-mail:
simone.milani@dei.unipd.it

David Hockney's "Place Furstenberg", Paris, 1985.

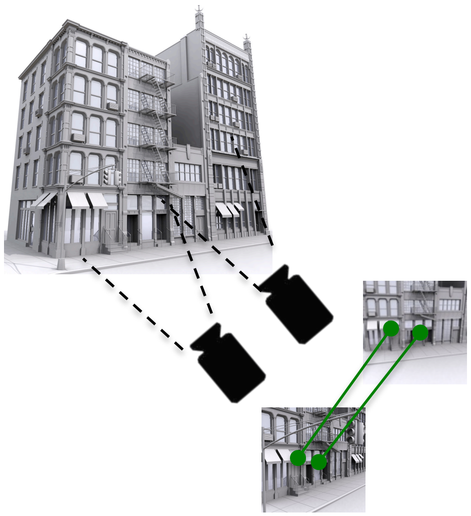
"Perspective starts from one viewpoint and never gets away from it.
But the viewpoint is quite unimportant.
It is though someone were to draw profiles all his life, leading people to think that a man has only one eye."
George Braque

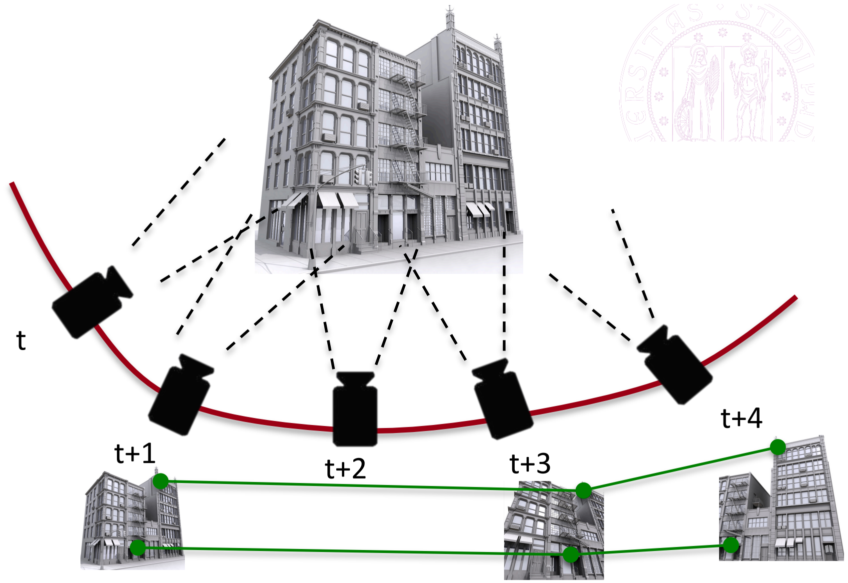
Structure from Motion

Assume that a camera is moving around a scene, intrinsics are known, and conjugate points are available (e.g. via SIFT matching)

Relative orientation

Compute 3D structure of the scene + orientation and location of the camera at the different instants





Stereopsis:
- 2 cameras
- Calibrated
- Dense reconstruction

Structure from Motion (SfM):
- 1 camera
- Not calibrated (only intrinsic are known) and moving
- Sparse matched conjugate points

SfM strategies

Differential: analyze speed of conjugate points
[Tian et al '96, Soatto et al '96, Soatto and Brockett '98]

Discrete: analyze positions of points into different views [Tomasi Kanade '92, Poelman Kanade '93, Snavely '06]

Widely-used: discrete strategy using perspective projection model of camera $\rightarrow$ essential matrix

Longuet-Higgins equation

$$\mathbf{m}'^T F \mathbf{m} = 0$$

Essential matrix is the fundamental matrix where intrinsics have been reduced to the normalized case, i.e.

K=[I | 0]

Structure-from-motion

Discrete approach

Recap: Longuet-Higgins equation

Conjugate points are linked by a bilinear equation. Let's show it!

It is possible to write

$$\underbrace{[\mathbf{e}']_\times \mathbf{m}'}_{\mathbf{e}' \times \mathbf{m}'\text{: equation of the line containing } \boldsymbol{m}' \text{ and } \boldsymbol{e}'} \simeq \lambda [\mathbf{e}']_\times Q' Q^{-1} \mathbf{m}$$

Multiply left and right for ***m'***

$$\underbrace{\mathbf{m}'^T [\mathbf{e}']_\times \mathbf{m}'}_{\text{From the definition of line containing two points in homogeneous coordinates}} = 0 \simeq \lambda \mathbf{m}'^T [\mathbf{e}']_\times Q' Q^{-1} \mathbf{m}$$

$$\boxed{0 = \mathbf{m}'^T [\mathbf{e}']_\times Q' Q^{-1} \mathbf{m}}$$

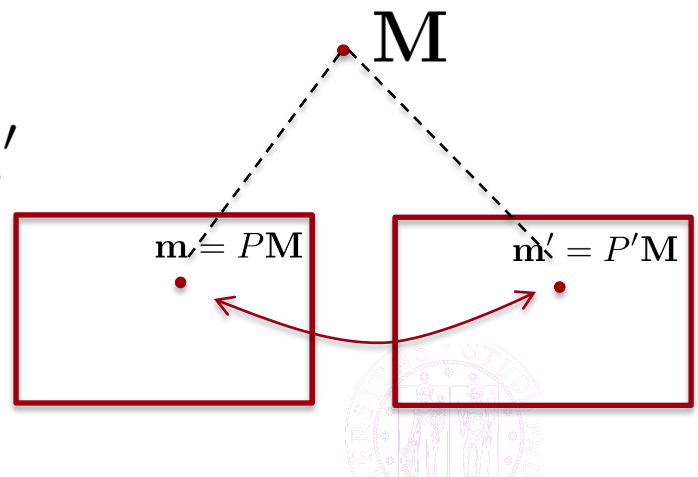
Longuet-Higgins equation

Bilinear relation between ***m*** and ***m'***

Essential matrix

Assume that intrinsic parameters (K) are known: it is possible to normalize the coordinates of conjugate points, i.e.

$$\mathbf{p} = K^{-1}\mathbf{m} \qquad \mathbf{p}' = K^{-1}\mathbf{m}'$$



Then, the corresponding camera matrices becomes

$$P = K\,[I|\mathbf{0}]$$
$$P' = K\,[I|\mathbf{0}]\,G = K\,[R|\mathbf{t}]$$

$$P = K^{-1}K\,[I|\mathbf{0}] = [I|\mathbf{0}]$$
$$P = K^{-1}K\,[R|\mathbf{t}] = [R|\mathbf{t}]$$

Longuet-Higgins becomes

$$\mathbf{p}'^{T}\,[\mathbf{t}]_{\times} R\,\mathbf{p} = 0$$

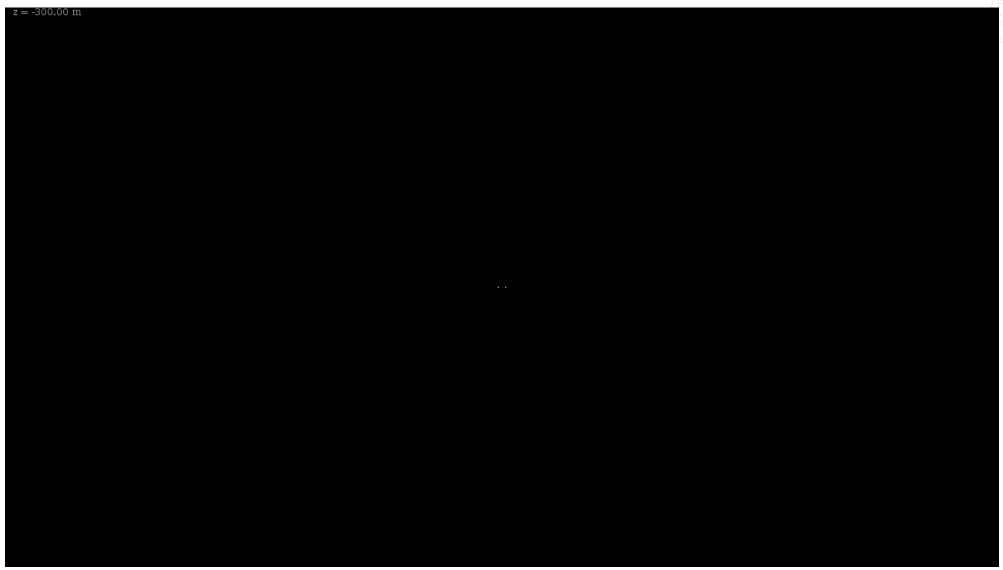
which leads to **essential matrix**

$$E \triangleq [\mathbf{t}]_{\times} R$$

COMMENTS OF ESSENTIAL MATRIX

- Since $det([\mathbf{t}]_\times) = 0$ E has rank 2.
- Changing scale factor in E does not change matrix.
- Ambiguity speed-depth (more details in the following lessons)

$$u = -f\frac{x}{z} \qquad x = v_x \cdot T + \mathbf{M}_x \qquad \Rightarrow u' = -f\frac{v_x T}{z} + u$$



E has 5 d.o.f. = 3 d.o.f. rotation + 2 d.o.f. translation (w.r.t. scale factor)

Decomposing essential matrix

Huang Faugeras1989

Theorem: a 3 x 3 real matrix E can be decomposed into the product of a non null anti-simmetric matrix S and a rotation R

$$E=S\,R$$

if and only if

E has two equal singular values and the other one equal to zero, i.e.

$\sigma_1=\sigma_2\neq 0$

$\sigma_3=0$

PROOF:

Remember $E \triangleq [\mathbf{t}]_\times R$

Pose $E = SR$ with S antisymmetric and R rotation matrix

$S = [\mathbf{t}]_\times$ $\quad ||\mathbf{t}|| = 1$ without loss of generality (since E is defined w.r.t. a scale factor)

Define a rotation U s.t. $U\mathbf{t} = [0,0,1]^T \triangleq \mathbf{a}$ which leads to

$$S = [\mathbf{t}]_\times = \left[U^T\mathbf{a}\right]_\times = U^T[\mathbf{a}]_\times U$$

where we used the property 2.2
(see linear algebra notes)

$$\left[A^{-1}\mathbf{u}\right]_\times = A^T[\mathbf{u}]_\times A$$

valid if A is 3x3 and $det(A) = 1$

$$EE^T = SRR^TS^T = SS^T = U^T[\mathbf{a}]_\times UU^T[\mathbf{a}]_\times^T U = U^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} U$$

Note that singular values of E are eigenvalues of $E^T E$ □

PROOF: (INV.)

W. l.o.g. it is possible to write using SVD

$$E = UDV^T \qquad D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \underbrace{\overbrace{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}^{S'}}_{\text{Anti-symmetric}} \underbrace{\overbrace{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\mathrm{R}'}}_{\text{rotation}} \triangleq S'R'$$

$$E = UDV^T = US'R'V^T = \underbrace{(US'U^T)}_{\text{antisymmetric}}\underbrace{(UR'V^T)}_{\text{Rotation matrix (product of rotations)}}$$

Note that antisymmetric matrices S' have only imaginary eigenvalues; by construction, S has real eigenvalues too.

$$S' = Q^T D_{Im} Q \quad \Longrightarrow \quad \mathrm{S} = US'U^T = U(Q^T D_{Im} Q)U^T = (QU^T)^T D_{Im} (QU^T)$$

ON DECOMPOSITION

4 possible decompositions changing the sign of R and S

$$S = U(\pm S')U^T$$

R= $UR'U^T$ or R= $UR'^T U^T$

where

$$S' \triangleq \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad R' \triangleq \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Choice can be done considering that points must be lying in front of the cameras (3rd coordinate must be positive)

MATLAB function: `sr`

HOW TO FIND E? 8 POINTS ALGORITHM

Given two images, it is possible to obtain a sequence of couples

$$(\mathbf{m}_i', \mathbf{m}_i)$$

which, after normalization, must satisfy the equation

$$\mathbf{p}_i'^T E \mathbf{p}_i = 0$$

8 POINTS ALGORITHM

Which can be decomposed

$$\mathbf{p}_i'^T E \mathbf{p}_i = 0 \Leftrightarrow \mathrm{vec}\left(\mathbf{p}_i'^T E \mathbf{p}_i\right) = 0 \Leftrightarrow \left(\mathbf{p}_i^T \otimes \mathbf{p}_i'^T\right) \mathrm{vec}(E) = 0$$

9 unknowns

Given n couples of points, we have a set of n linear equations

8 points algorithm

(similar to DLT)

$$\underbrace{\begin{bmatrix} \mathbf{p}_1^T \otimes \mathbf{p}_1'^T \\ \mathbf{p}_2^T \otimes \mathbf{p}_2'^T \\ \vdots \\ \mathbf{p}_n^T \otimes \mathbf{p}_n'^T \end{bmatrix}}_{U_n} \mathrm{vec}(E) = 0$$

Kernel of U_n is the solution

If **n=8** $dim(U_n) = 1$

E depends on 5 parameters: it is possible to use 5 equations + polynomial bounds derived from the previous theorem.

Faugeras and Maybank 1990
Nister 2003
Li and Hartley 2006

ON THE 8 POINTS ALGORITHM

The problem can be solved as a linear least square problem

More than 8 couples are usually available

Solution: minimum eigenvalue of $U^T_n U_n$ (normalized to 1) found using SVD.

In general E does not satisfy previous theorem $\sigma_1 = \sigma_2 = 1 \quad \sigma_3 = 0$

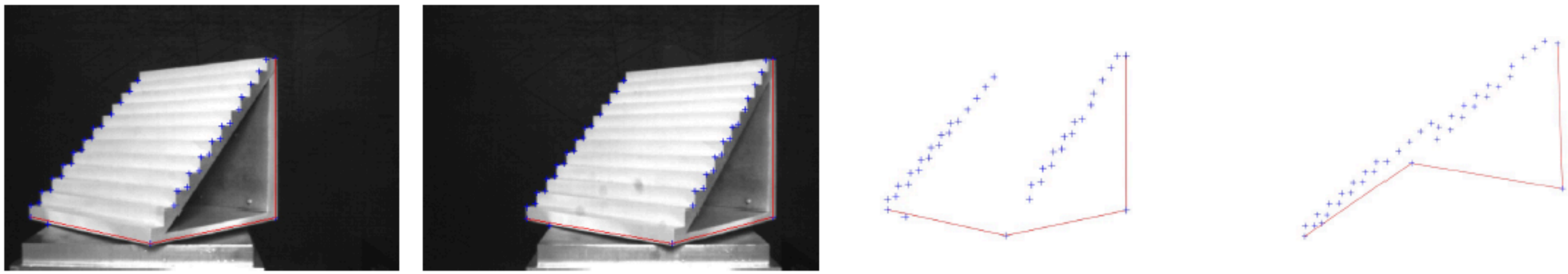
Replace E with Ê which is the closest matrix (in Frobenius norm) that satisfy the conditions, i.e., given

$$E_{3\times 3} = UDV^T \qquad D = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \qquad \sigma_1 \geq \sigma_2 \geq \sigma_3$$

It is possible to use

$$\hat{E} = U\hat{D}V^T \qquad \hat{D} = \begin{bmatrix} \frac{\sigma_1+\sigma_2}{2} & 0 & 0 \\ 0 & \frac{\sigma_1+\sigma_2}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

ON THE 8 POINTS ALGORITHM



Summary:

Input: given two images, we have the intrinsic matrix K and the couples $(\mathbf{m}'_i, \mathbf{m}_i)$

Output: points $\mathbf{M}_i$ and parameters $R, \mathbf{t}$

i) Find couples $(\mathbf{p}'_i, \mathbf{p}_i)$

ii) Compute E with the 8 points algorithm

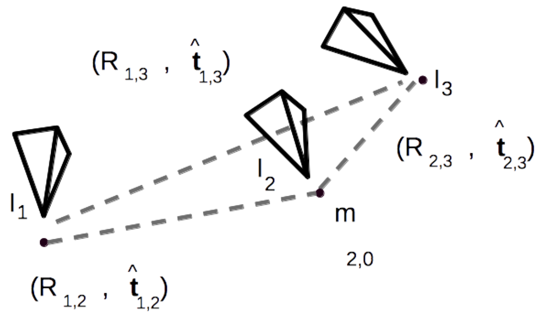
iii) Decompose E into $E = SR = [\mathbf{t}]_\times R$

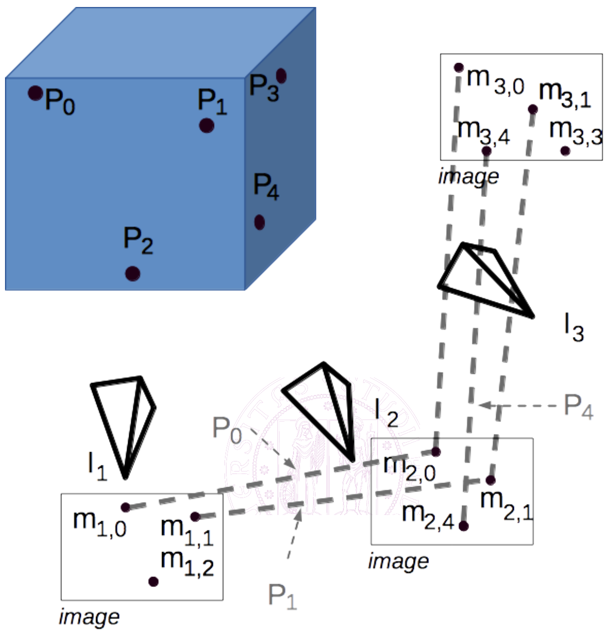
iv) Compute projection matrix $P = [I|\mathbf{0}]$ $P' = [R|\mathbf{t}]$

v) Compute $\mathbf{M}_i$ with triangulation.

Multiview

Let us assume that more than one view/image is available





^ denotes that each t is defined w.r.t. a scale factor

$$\mathbf{t}_{13} = R_{23}\mathbf{t}_{12} + \mathbf{t}_{23}$$

which depends on

$$\hat{\mathbf{t}}_{13} = \mu_1 R_{23}\hat{\mathbf{t}}_{12} + \mu_2\hat{\mathbf{t}}_{23}$$

Equation with 2 unknown

$$\mu_1 = \frac{\|\mathbf{t}_{12}\|}{\|\mathbf{t}_{13}\|} \qquad \mu_2 = \frac{\|\mathbf{t}_{23}\|}{\|\mathbf{t}_{13}\|}$$

It can be solved as

$$\mu_1 = \frac{\|\mathbf{t}_{12}\|}{\|\mathbf{t}_{13}\|} = \frac{\left(\hat{\mathbf{t}}_{13} \times \hat{\mathbf{t}}_{23}\right)^T \left(R_{23}\hat{\mathbf{t}}_{12} \times \hat{\mathbf{t}}_{23}\right)}{\|R_{23}\hat{\mathbf{t}}_{12} \times \hat{\mathbf{t}}_{23}\|^2}$$

Only relative scale is found; global scale is unknown

Similar for μ_2

BUNDLE ADJUSTMENT

Estimated points are highly noisy

It is possible to refine the estimated accuracy using Bundle adjustment, i.e., minimizing the target function

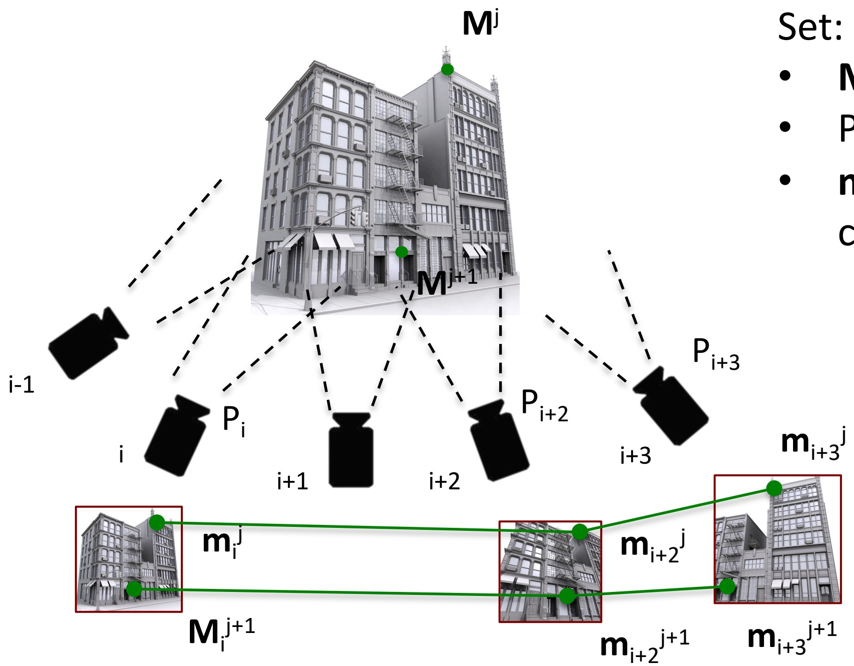
$$\min_{R_i, \mathbf{t}_i, \mathbf{M}_j} \sum_{i=1}^{N} \sum_{j=1}^{n} \left\| \mathbf{m}_j^i - K_i [R_i | \mathbf{t}_i] \mathbf{M}_j \right\|^2$$

Two stage process:

- Fix points $\mathbf{M}_j$ and compute $R_{i,}$ $\mathbf{t}_i$
- Fix $R_{i,}$ $\mathbf{t}_i$, estimate $\mathbf{M}_j$

UNCALIBRATED RECONSTRUCTION

SET-UP



Set:

- $\mathbf{M}^j$: 3D points
- P_i: camera projection matrix
- $\mathbf{m}_i^j$: projection of point $\mathbf{M}^j$ on the i-th camera, i.e.,

$$\mathbf{m}_i^j \simeq P_i \mathbf{M}^j$$

Suppose that we know $\mathbf{m}_i^j$: reconstruct $\mathbf{M}^j$ and P_i

The solution is found w.r.t. an arbitrary projection T_{4x4}, i.e.

if $\{P_i\}$ $\{M^j\}$ solution, then $\{P_i\ T\}$ $\{T^{-1}\ M^j\}$ as well.

We want a Euclidean reconstruction, i.e., it equals real 3D points w.r.t. a similarity: a rigid transform + scale change (ambiguity speed-depth).

PROJECTIVE RECONSTRUCTION (1/3)

Given h cameras and n points,

$$\zeta_i^j \mathbf{m}_i^j = P_i \mathbf{M}^j, \qquad i = 1, \ldots, h \qquad j = 1, \ldots, n$$

or in matrix form,

$$\underbrace{\begin{bmatrix} \zeta_1^1 \mathbf{m}_1^1 & \zeta_1^2 \mathbf{m}_1^2 & \ldots & \zeta_1^n \mathbf{m}_1^n \\ \zeta_2^1 \mathbf{m}_2^1 & \zeta_2^2 \mathbf{m}_2^2 & \ldots & \zeta_2^n \mathbf{m}_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_h^1 \mathbf{m}_h^1 & \zeta_h^2 \mathbf{m}_h^2 & \ldots & \zeta_h^n \mathbf{m}_h^n \end{bmatrix}}_{\text{Matrix W: } \mathbf{m}^j_j \text{ known}} = \underbrace{\begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_h \end{bmatrix}}_{\text{Matrix P: unknown}} \underbrace{\left[\mathbf{M}^1 \ \mathbf{M}^2 \ \ldots \mathbf{M}^n\right]}_{\text{Matrix M: unknown}}$$

The matrix W_{mxn} is factorized into matrices P_{3mx4} and M_{4xn}.

If ζ^j_i is known, W is known and the problem is similar to Tomasi-Kanade '92, i.e., with SVD

$$W = UDV^T$$

$$D = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Rank(W)=4; then only the first 4 columns of U and V are useful.

$$W = U_{3m\times 4} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \end{bmatrix} V^T_{n\times 4}$$

PROJECTIVE RECONSTRUCTION (2/3)

$$W = U_{3m\times 4}\begin{bmatrix}\sigma_1 & 0 & 0 & 0\\ 0 & \sigma_2 & 0 & 0\\ 0 & 0 & \sigma_3 & 0\\ 0 & 0 & 0 & \sigma_4\end{bmatrix} V^T_{n\times 4}$$

Given the SVD decomposition, and taking the 4 most significant cols, we have the factorization

$$P = U_{3m\times 4}\begin{bmatrix}\sigma_1 & 0 & 0 & 0\\ 0 & \sigma_2 & 0 & 0\\ 0 & 0 & \sigma_3 & 0\\ 0 & 0 & 0 & \sigma_4\end{bmatrix} \qquad M = V^T_{n\times 4}$$

Note: rank of W is 4 if there are no errors; commonly, rank ≠ 4.

Forcing $D = \begin{bmatrix}\sigma_1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \sigma_2 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & \sigma_3 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & \sigma_4 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$, i.e., put to zero elements in D, we find the

solution that minimizes Frobenius norm

$$\|W - PM\|_F^2 = \sum_{i,j} \|\zeta_i^j \mathbf{m}_i^j - P_i \mathbf{M}^j\|^2$$

PROJECTIVE RECONSTRUCTION (3/3)

But scales still remain unknown….

$$\begin{bmatrix} \zeta_1^j \mathbf{m}_1^j \\ \zeta_2^j \mathbf{m}_2^j \\ \vdots \\ \zeta_h^j \mathbf{m}_h^j \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{m}_1^j & 0 & \dots & 0 \\ 0 & \mathbf{m}_1^j & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \mathbf{m}_m^j \end{bmatrix}}_{Q^j} \underbrace{\begin{bmatrix} \zeta_1^j \\ \zeta_2^j \\ \vdots \\ \zeta_h^j \end{bmatrix}}_{\zeta^j} = PM^j$$

- Knowing P and M, find ζ_i^j
- Knowing ζ_i^j and $\mathbf{m}_i^j$ find P and M

Chicken-egg problem

Iterative solution: alternate both optimizations

Set $\zeta_i^j=1$, generate W from $\mathbf{m}_i^j$

i. Normalize W s.t. $\|W\|_F = 1$;

Needed to avoid trivial solution $\zeta_i^j = 0$

ii. Find P and M from SVD of W;

iii. **If** $\|W - PM\|_F^2$ small enough, **end;**

iv. Find ζ_i^j from $Q^j \zeta^j = PM^j, \quad \forall j = 1, \dots, n$

v. Update W;

vi. Go to step (i)

Note: no grants on convergence.

EUCLIDEAN RECONSTRUCTION (1/2)

- differs from perspective one w.r.t. to transformation T (change of basis)

- How to compute T?

Assumptions:

- Take a sufficient number of images (with enough points)
- K_i are constants! (in a video camera, pay attention to autofocus)

Compute $\{P_i^p\}$ from projective reconstruction. $P_0^p = [I|\mathbf{0}]$ $P_i^p = [Q_i|\mathbf{q}_i]$

We are looking for T s.t. $\mathbf{m}_i^j = P_i^p T T^{-1} \mathbf{M}^j$

i.e.

$$P_i^e \simeq P_i^p T$$

Let us assume $P_0^e = K\,[I|\mathbf{0}]$ $P_i^e = K\,[R_i|\mathbf{t}_i]\;\; i > 0$

Therefore, since $P_0^e = P_0^p T$, we have that $T = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{r}^T & s \end{bmatrix}$ $\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$

arbitrary

8 parameters = 5 + 3

s=1 since it is defined w.r.t. scale

Using previous eq. we have

$$P_i^e \simeq P_i^p T = \left[Q_i K + \mathbf{q}_i \mathbf{r}^T | \mathbf{q}_i\right]$$

EUCLIDEAN RECONSTRUCTION (2/2)

Comparing we can write $P_i^e \simeq P_i^p T = \left[Q_i K + \mathbf{q}_i \mathbf{r}^T | \mathbf{q}_i \right]$ with the eq. $P_i^e = K \left[R_i | \mathbf{t}_i \right] = \left[K R_i | K \mathbf{t}_i \right]$

$$Q_i K + \mathbf{q}_i \mathbf{r}^T \simeq K R_i$$

known

unknown

Unknown, it must be a rotation matrix

Keydem-Anstrom '96

$$Q_i K + \mathbf{q}_i \mathbf{r}^T \simeq K R_i \longrightarrow P_i^p \begin{bmatrix} K \\ \mathbf{r}^T \end{bmatrix} \simeq K R_i$$

that allows writing

$$P_i^p \begin{bmatrix} K \\ \mathbf{r}^T \end{bmatrix} \begin{bmatrix} K \\ \mathbf{r}^T \end{bmatrix}^T P_i^{pT} = P_i^p \begin{bmatrix} K K^T & K \mathbf{r} \\ \mathbf{r}^T K^T & \mathbf{r}^T \mathbf{r} \end{bmatrix} P_i^{pT} \simeq K R_i R_i^T K^T = K K^T$$

Kruppa's bounds

Note that this generates 5 equations with unknown $\alpha_u, \alpha_v, \gamma, u_0, v_0, r_1, r_2, r_3$

We have 1 solution whenever 3 cameras are available.

Considering the scale factor $$0 = f_i(K, \mathbf{r}, \lambda_i) = \lambda_i^2 K K^T - P_i^p \begin{bmatrix} K K^T & K \mathbf{r} \\ \mathbf{r}^T K^T & \mathbf{r}^T \mathbf{r} \end{bmatrix} P_i^{pT}$$

We can see that 3 cameras are still enough: 12 equations in 10 unknowns (8 + two λscale factors).

LS solution with numerical method (Gauss-Newton)

SELF-CALIBRATION

From Longuet-Higgins equation we can define the Fundamental matrix s.t.

$$\mathbf{m}'^T[\mathbf{e}']_\times\mathbf{m}' = 0 \simeq \lambda\mathbf{m}'^T[\mathbf{e}']_\times Q'Q^{-1}\mathbf{m}$$

$$\mathbf{m}'^T \underbrace{F\mathbf{m}}_{\text{Epipolar line}} = 0$$

with

$$F = [\mathbf{e}']_\times Q'Q^{-1}$$

Note that:

- F is defined w.r.t. a scale factor
- $det([e]_\times) = 0 \quad \Rightarrow \quad det(F) = 0$

7 d.o.f.

$$\mathbf{m}^T F^T \mathbf{m}' = 0$$

but epipole belongs to all the epipolar lines, then

$$\mathbf{e}'F = 0 \qquad \mathbf{e}^T F^T = 0$$

Moreover, Fundamental and essential matrix are related via the equation

$$F = K'^{-T} E K^{-1}$$

Given a set of conjugate points, it is possible to find F from LH equation.

8 points algorithm

8 POINTS ALGORITHM: IMPORTANT STEPS

Standardization:most of the problems arise because of bad conditioning; variance of u and v is not the same of the 3rd coord.

$$\overline{\mathbf{m}} = T\mathbf{m} \qquad \overline{\mathbf{m}}' = T'\mathbf{m}'$$

Geometric residual: most of the problems arise because of bad conditioning; variance of u and v is not the same of the 3rd coord.

$$\min_F \sum_j d(F\mathbf{m}_i, \mathbf{m}_i')^2 + d(F^\top \mathbf{m}_i', \mathbf{m}_i)^2$$

Additional approaches:

- Mendonça-Cipolla method
- Huang-Faugeras method

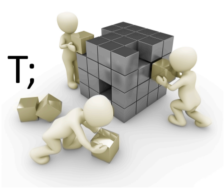
SfM PIPELINE

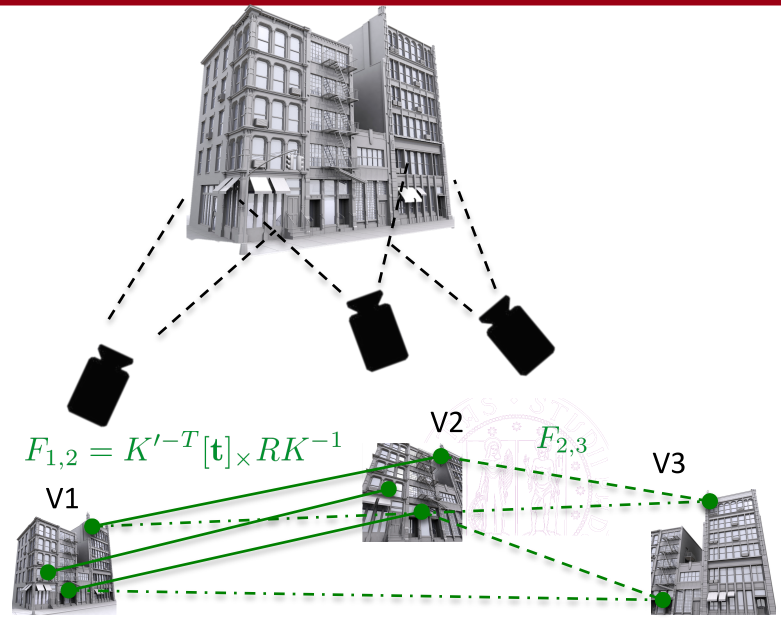
① Images collected "in-the-wild"

② Corresponding points in the different views are found (SIFT, SURF). Generate C

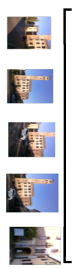
③ Epipolar geometry between couples of cameras is computed and integrated progressively (order matters)

④ Matrix can be decomposed into R and T; intrinsics can be estimated





⑤ Refinement using some bundle–adjustment strategy.



$$C = \begin{bmatrix} - & 345 & 234 & 123 & 12 \\ 345 & - & 267 & 145 & 6 \\ 234 & 267 & - & 143 & 23 \\ 123 & 145 & 143 & - & 45 \\ 12 & 6 & 23 & 45 & - \end{bmatrix}$$

Some of the most recent approaches uses also (in place of SIFT-matched points):

- Line segments [Micusik, Widenauer , JVCI 2017]
- Shadows [Taneja et al., ACCV 2010]
- Silouettes [Ben-Artzi *et al.*, CVPR 2016]
- Intensity values [Engel, 2014]
- Objects [Crocco *et al.,* CVPR 2016]
- DNN-based registration [Liu *et al.*, NIPS 2015]

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

KEYPOINTS MATCHING

The first steps of the pipeline consists in matching keypoints on different frames

Image 1 Image 2 Image 3

Same procedure used for:

- image stitching in panorama
- object localization
- orientation
- Object search
- …

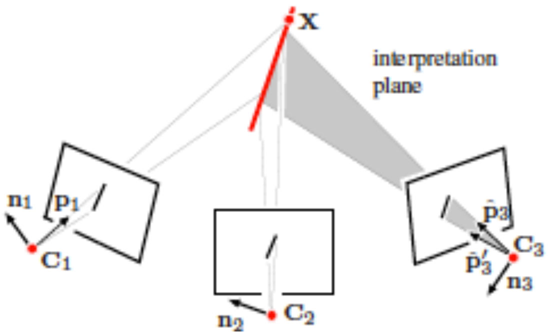
} Augmented Reality

Different descriptors/keypoint detectors can be used (SIFT, SURF, ORB, …) , depending on

- Complexity
- Size
- Speed
- Robustness to noise, viewpoints, compress
- patents

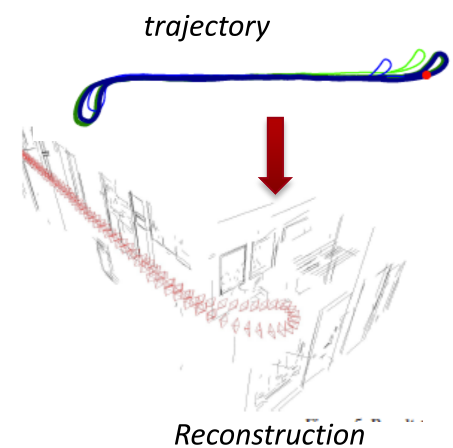
Harris SURF

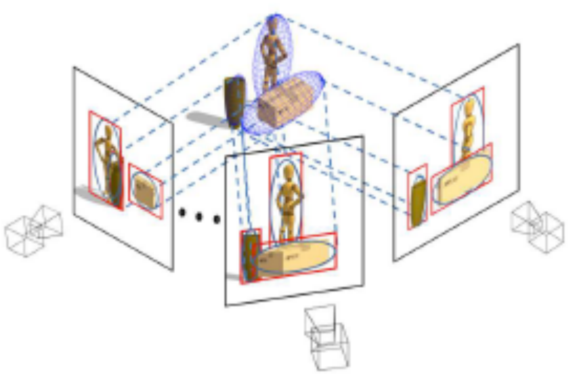
MATCHING DETAILS (1/3)



Line segments *[Micusik, Widenauer , JVCI 2017]*

- More robust
- Denser point clouds
- Applied to traditional SfM
  (what about wrong matches?)



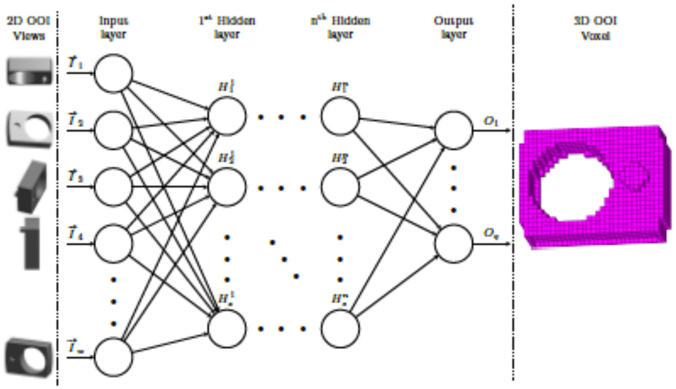


Objects encapsulatd in bounding boxes/ellipsoids
[Crocco et al.*, CVPR 2016]*

- Requires object detector
- Find ellipsoids, building conic relations in the dual space
- Jointly solve calibration and 3D reconstruction
- Robust
- Avoid oscillation behaviours

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

MATCHING DETAILS (2/3)

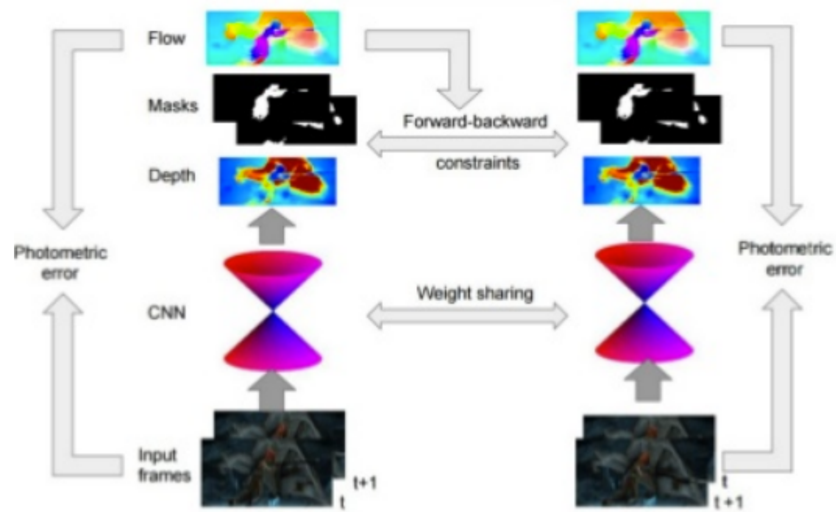


ANN-based SfM [*Schöning* et al.*, 2014*]

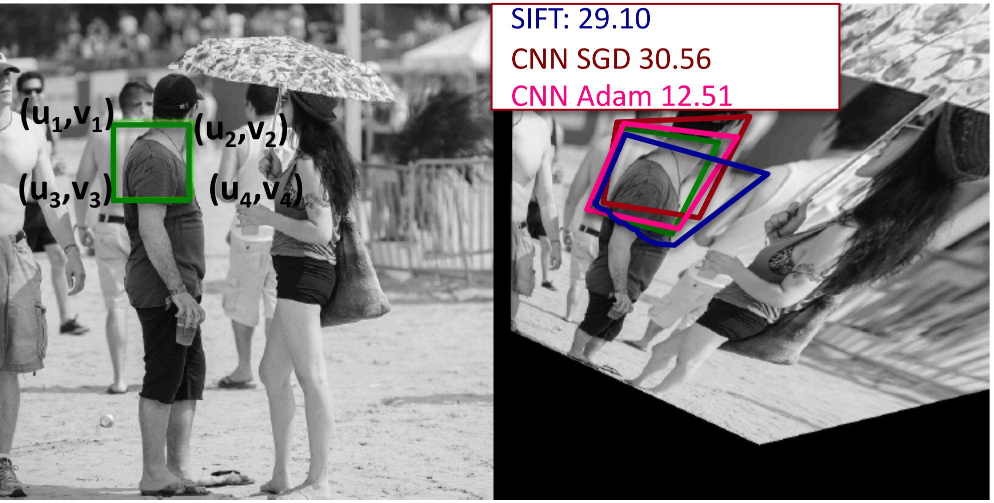
- Targets single simple objects
- Output voxels
- More complex scenes not analyzed

SfM-Net [*Vijayanarasimhan* et al.*, 2017*]

- Estimate depth between couple of images via CNN
- Transfer weights to following couples
- Self-supervised (reprojections) or supervised (depth from RGB, ego-motion)



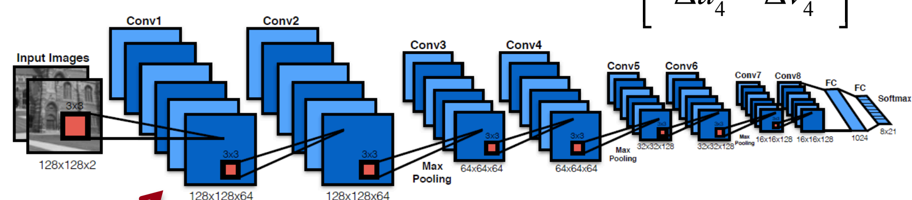
Matching details (3/3)



HomographyNet [*De Tone* et al.*, 2016*]

- Compute homography matrix
- Replace H with H_4 (matrix from perturbation of 4 corners in an image)
- Are considered all possible perturbations?
- Overfitting …

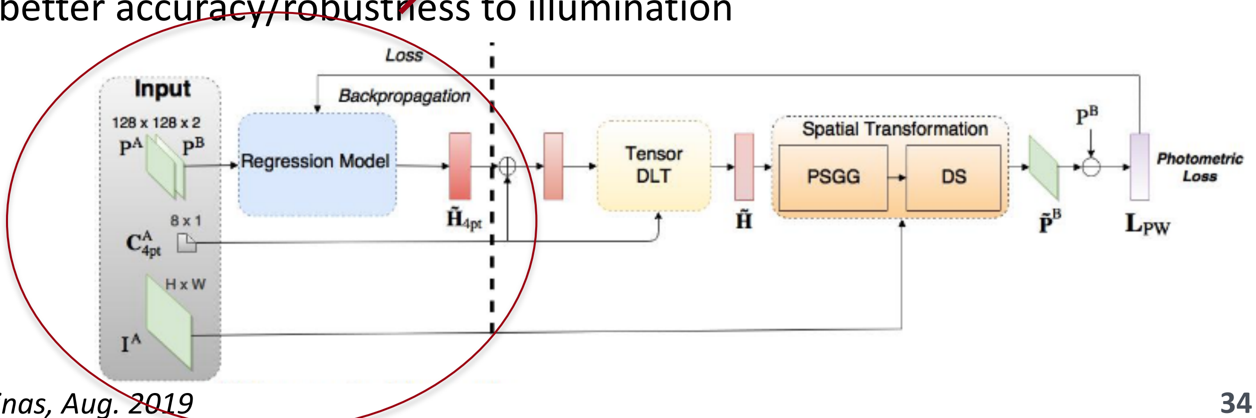
$$H_4 = \begin{bmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{bmatrix}$$



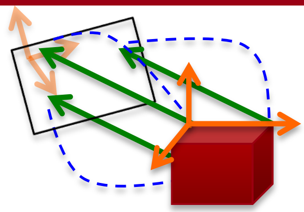
Unsupervised Deep Homography [*Nguyen* et al.*, 2017*]

- Compute homography inline for robotic applications
- Learn/adapt to different perturbation
- faster inference speed, same or better accuracy/robustness to illumination
- Better adaptability

VGGNet



RELOCALIZING IMAGES (1/2)



Generally speaking a PnP problem:

*"**Perspective-n-Point** is the problem of estimating the pose of a calibrated camera given a set of n 3D points in the world and their corresponding 2D projections in the image." Wikipedia*

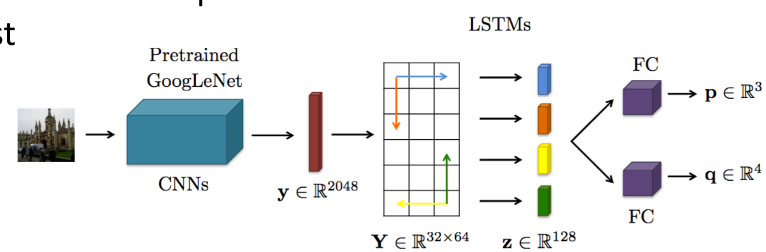
Match the new image with those in the dataset, find 3D correspondences

P3P: minimum amount of matches is 3;
EPnP: iterative method to solve a non-linear LS problem [Lepetit *et al.*, 2008]
RANSAC: increase the robustness [Li *et al.*, 2012]

Keypoint matching is computationally a bottle-neck!

PoseNet *[Kendall et al., ICCV 2015]*

- End-to-end solution
- Convolutional Network (GoogLeNet) that ouputs $\mathbf{p} = [\mathbf{x}, \mathbf{q}]$
- Maps images to a high dimensional space linear in pose and robust
- Performs regression



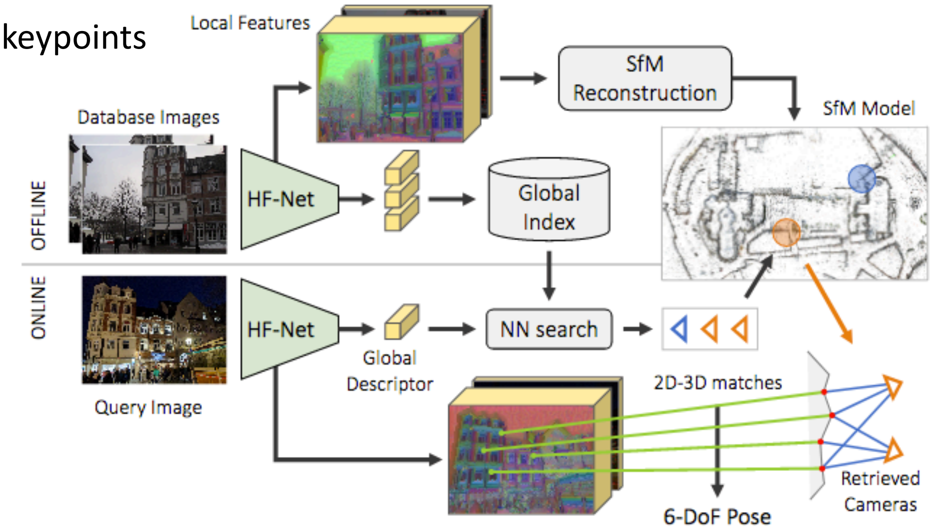
CNN + LSTM
[Kendall et al., ICCV 2015]

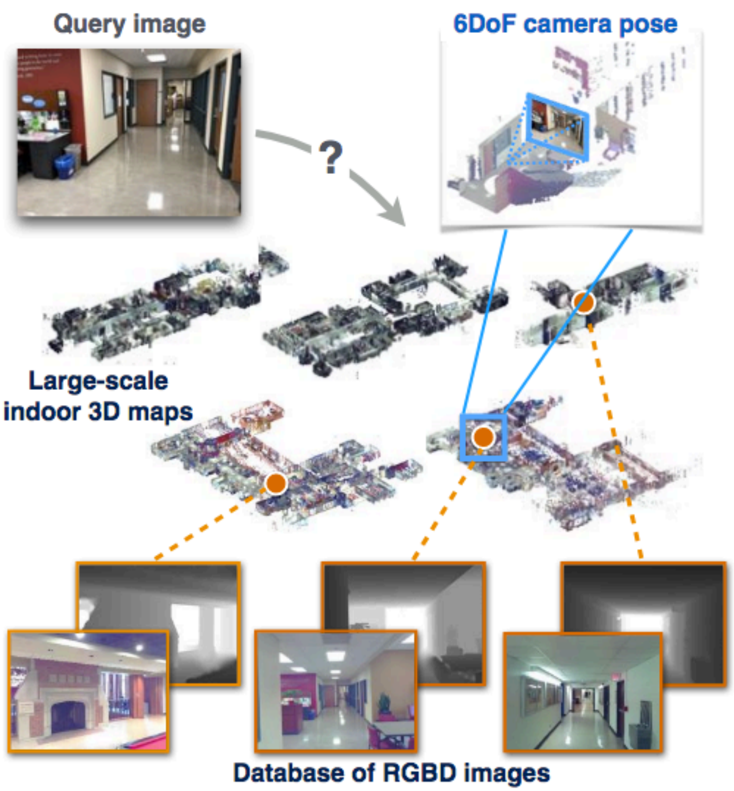
- Computes features using CNN
- Regress the pose via LSTM

Relocalizing images (2/2)

Robust Hierarchical Localization at Large Scale *[Sarlin et al., CVPR 2019]*

- HF-Net generates a global desc., local desc, a map of keypoints for each image
- Hierarchical localization
- Faster matching speed





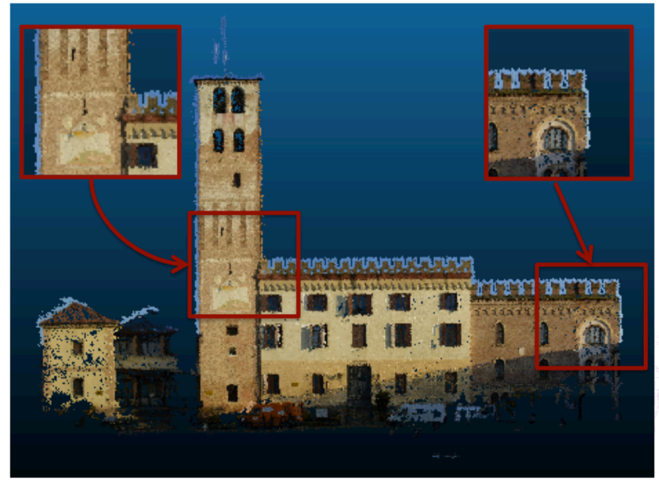
InLoc *[Taira et al., CVPR 2018]*

- RGBD images in the database, RGB images to localize
- Match query to dataset
- P3P-RANSAC to localize minimizing some proj. metrics

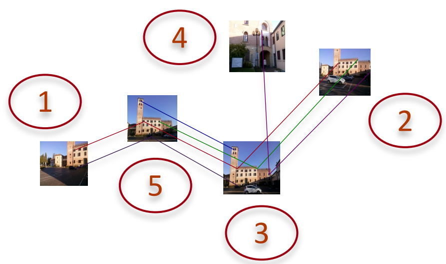
Next-best view problems

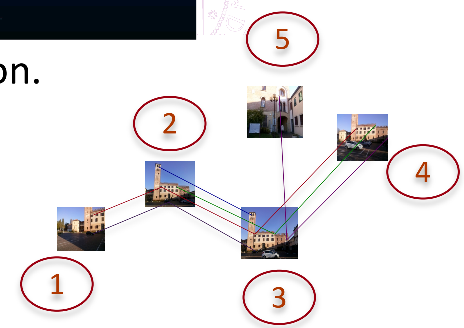
No image order is pre-defined. The inclusion order makes a difference.

Random inclusion.



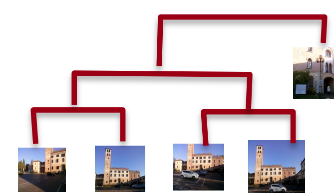
Position based inclusion.





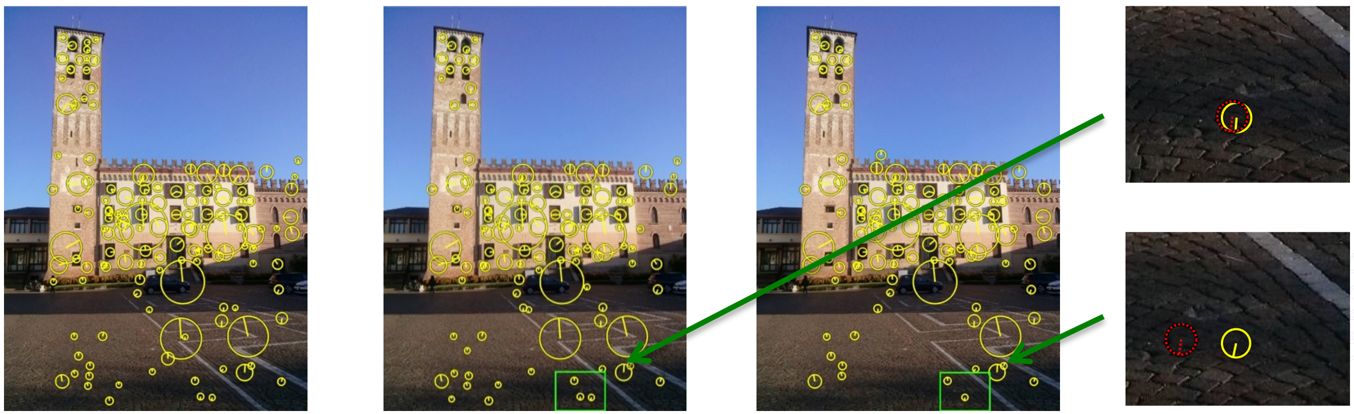
Different solutions:

- Incremental [Snavely et al. , SIGGRAPH 2006]
- Hierarchical [Gherardi et al. , CVPR 2010]
- Global estimation [Jiang et al., CVPR 2013]



ISSUES WITH HETEROGENEOUS COLLECTIONS

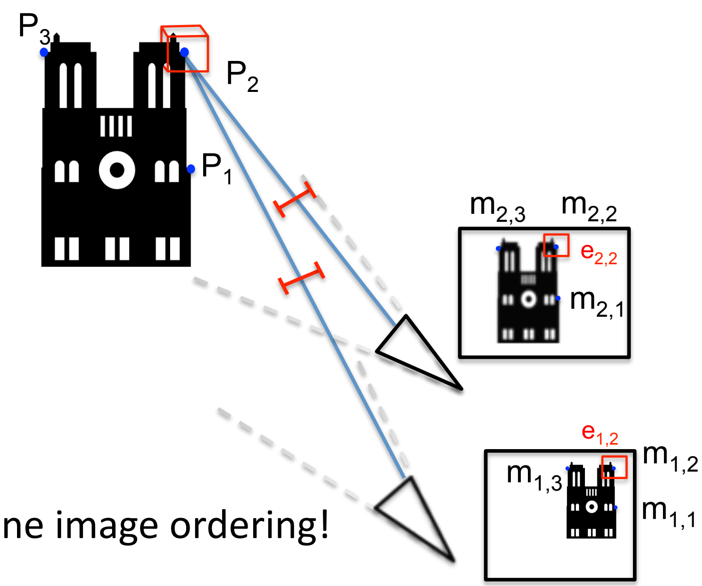
Online images may have experienced several editing steps! Local features could have been compromised.



Original image | 2 editing steps | 4 editing steps

Web-harvested images can be quite noisy: many editing steps operated.

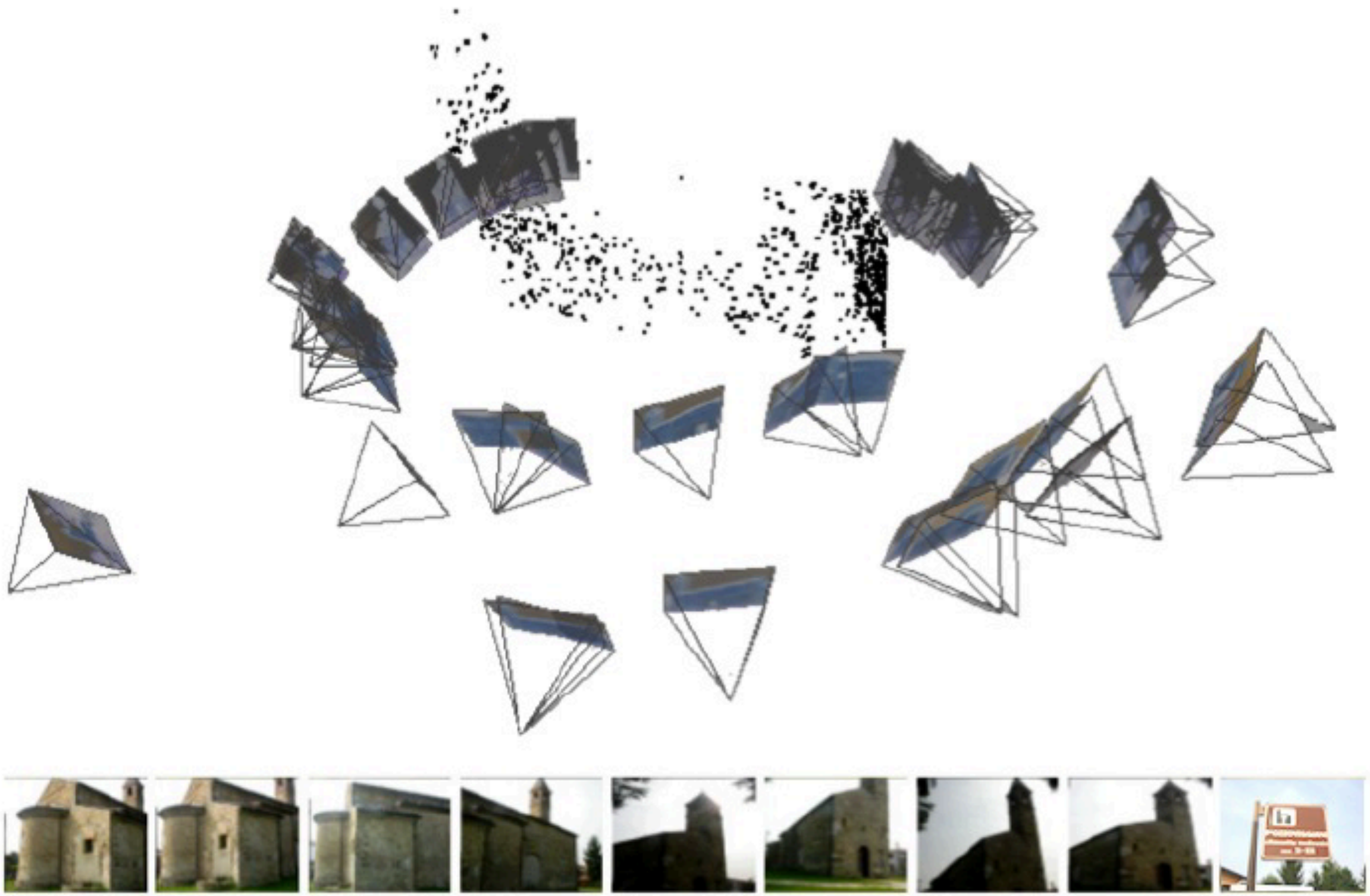
- Reduced number of features
- Wrong matches
- Keypoint displacements



Additional noise that increases the uncertainty on 3D points.

This affects the correspondence matrix, which is used to determine image ordering!

INCREMENTAL RECONSTRUCTION



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

ORDER MATTERS: BUNDLER

ORDER MATTERS: EX SAMANTHA