

# Structure from Motion (SfM)

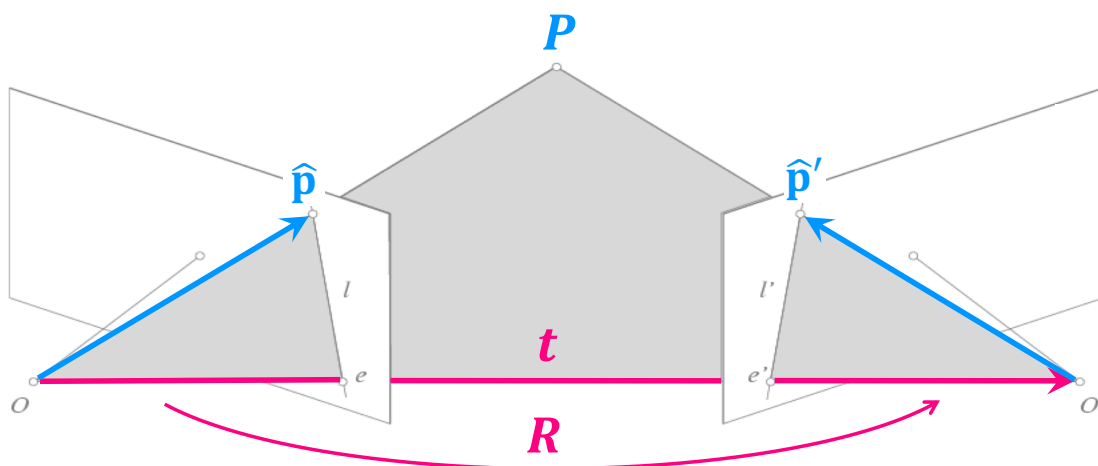
## Active 3D Vision

- ❖ **SfM (camere intrinsecamente calibrate)**
  - caso binoculare – 8-point algorithm
  - camere multiple
- ❖ **SfM (camere non calibrate)**
  - camere multiple
- ❖ **Active 3D vision**
  - Laser scanning, Structured light, Shape from silhouettes

(Forsyth/Ponce: Capitolo 8)

Slide credits:  
materiale rielaborato a partire da slide di Luigi Cinque (Univ. La Sapienza, Roma) e altre sorgenti (citare)

## Geometria binoculare – vincolo epipolare



### Camera 1:

Matrice di proiezione:  $\mathbf{M}_1 = [I_3 \mid \mathbf{0}_{[3 \times 1]}]$

$$\mathbf{P} = (x_P, y_P, z_P, 1)^T$$

$$\hat{\mathbf{p}} = (x, y, 1)^T, \quad \hat{\mathbf{p}}' = \mathbf{R} (x', y', 1)^T$$

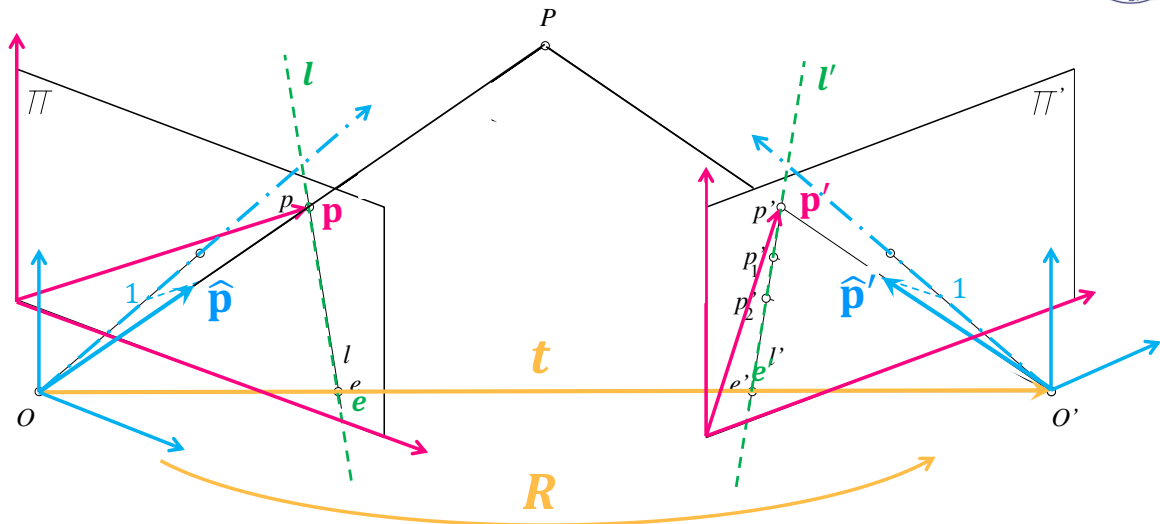
### Camera 2:

Matrice di proiezione:  $\mathbf{M}_2 = [\mathbf{R}^T \mid -\mathbf{R}^T \mathbf{t}]$

il vettore  $\hat{\mathbf{p}}'$  nel s.d.r. di camera 2 ha coordinate  $\mathbf{R}\hat{\mathbf{p}}'$  nel s.d.r. di camera 1

**Vincolo epipolare:**  $\hat{\mathbf{p}}, \mathbf{t}$  e  $\hat{\mathbf{p}}'$  sono **coplanari**

➔ nel riferimento di camera 1:  $\hat{\mathbf{p}} \cdot \mathbf{t} \times \mathbf{R}\hat{\mathbf{p}}' = \hat{\mathbf{p}}^T (\mathbf{t} \times \mathbf{R}) \hat{\mathbf{p}}' = \hat{\mathbf{p}}^T \mathbf{E} \hat{\mathbf{p}}' = 0$



coordinate **immaginate**

In coordinate **omogenee**  
→ matrice **essenziale**

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

$$\begin{cases} \mathbf{p} = \mathbf{K} \hat{\mathbf{p}} \\ \mathbf{p}' = \mathbf{K}' \hat{\mathbf{p}}' \end{cases} ; \quad \mathbf{K} = \begin{bmatrix} f & 0 & c_X \\ 0 & f & c_Y \\ 0 & 0 & 1 \end{bmatrix}$$

In coordinate **immaginate**  
→ matrice **fondamentale**

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

$$\mathbf{F} = (\mathbf{K}^{-1})^T \mathbf{E} \mathbf{K}'^{-1}$$

## Eight-point algorithm



### 8-Points Algorithm:

stima della matrice fondamentale  $\mathbf{F}$ , a partire da corrispondenze binoculari

Date le coordinate-immagine  $\mathbf{p}$  e  $\mathbf{p}'$  dello stesso punto nelle due immagini:

$$\mathbf{p} = (x, y, 1)^T, \quad \mathbf{p}' = (x', y', 1)^T$$

Pongo il **vincolo epipolare** in coord. immagine (→ matrice fondamentale):

$$[x \ y \ 1] \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = 0 \quad \longrightarrow \quad [xx' \ xy' \ x \ yx' \ yy' \ y \ x' \ y' \ 1] \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

❖  $\mathbf{F}$  è definita a meno di un fattore di scala:

→ pongo  $F_{33} = 1$ :

$$\longrightarrow [xx' \ xy' \ x \ yx' \ yy' \ y \ x' \ y'] \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{bmatrix} = -1$$

→ **1 equazione** lineare, **8 incognite**  
(8 coefficienti di  $\mathbf{F}$ )



Considero **8 punti immagine**:

$$\mathbf{p}_i = (x_i, y_i, 1)^T, \quad \mathbf{p}'_i = (x'_i, y'_i, 1)^T, \quad i = 1..8$$

$$[x, y, 1] \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = 0$$

→ **8 equazioni lineari in 8 incognite**  
(8 coefficienti di F):

$$\mathbf{A}_{[8 \times 8]} \mathbf{F} = -\mathbf{1}_{[8 \times 1]} \quad : \quad \begin{bmatrix} x_1 x'_1 & x_1 y'_1 & x_1 & y_1 x'_1 & y_1 y'_1 & y_1 & x'_1 & y'_1 \\ x_2 x'_2 & x_2 y'_2 & x_2 & y_2 x'_2 & y_2 y'_2 & y_2 & x'_2 & y'_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_8 x'_8 & x_8 y'_8 & x_8 & y_8 x'_8 & y_8 y'_8 & y_8 & x'_8 & y'_8 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ \dots \\ -1 \end{bmatrix}$$

- ❖ A causa degli errori di localizzazione conviene considerare più di 8 punti: **N > 8**
- ❖ In generale, per **N > 8**, il sistema non avrà soluzione → **errore non nullo**
- ❖ Per **N > 8** ho un sistema lineare **sovradeterminato, non omogeneo**:  
→ soluzione ai minimi quadrati: **pseudoinversa**

$$E = \sum_{i=1}^N (\mathbf{p}_i^T \mathbf{F} \mathbf{p}'_i)^2 > 0$$

$$\mathbf{A}_{[N \times 8]} \mathbf{F}_{[8 \times 1]} = -\mathbf{1}_{[N \times 1]}$$



Soluzione **lineare**: pseudo-inversa:

$$\mathbf{A}_{[N \times 8]} \mathbf{F}_8 = -\mathbf{1}_N \quad \rightarrow \quad \mathbf{F} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T (-\mathbf{1}_N), \quad F_{33} = 1$$

oppure soluzione **non lineare**: **F** (non nullo) che minimizza l'errore **E**:

$$E = \sum_{i=1}^N (\mathbf{p}_i^T \mathbf{F} \mathbf{p}'_i)^2 \quad \rightarrow \quad \hat{\mathbf{F}} = \underset{\mathbf{F}}{\operatorname{argmin}} \left( \sum_{i=1}^N (\mathbf{p}_i^T \mathbf{F} \mathbf{p}'_i)^2 \right), \quad \|\mathbf{F}\| = 1$$

Soluzione **ottima**:

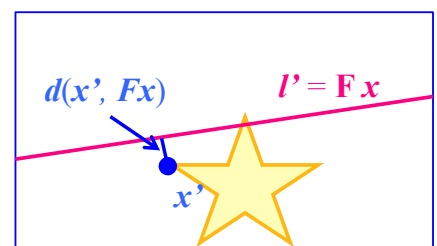
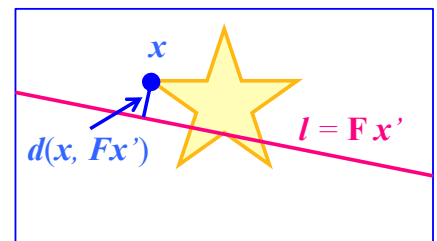
il minimo dell'errore **E non** corrisponde al minimo dell'errore di osservazione (errore sul piano immagine)

→ La minimizzazione non lineare dell'errore di localizzazione dei punti immagine **E<sub>IM</sub>** (distanza dall'epipolare) fornisce risultati migliori:

$$E_{IM} = \sum_{i=1}^N [d^2(x_i, \mathbf{F}x'_i) + d^2(x'_i, \mathbf{F}^{-1}x_i)]$$

→ soluzione ottima (ricerca **non lineare**):

$$\hat{\mathbf{F}} = \underset{\mathbf{F}}{\operatorname{argmin}} (E_{IM}) : \quad \|\mathbf{F}\| = 1$$





Problema dell'approccio **lineare**:  
la matrice **A** è **mal condizionata!**



$$\begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48

$$\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Soluzione: **normalizzazione dei dati** (Hartley, 1995)

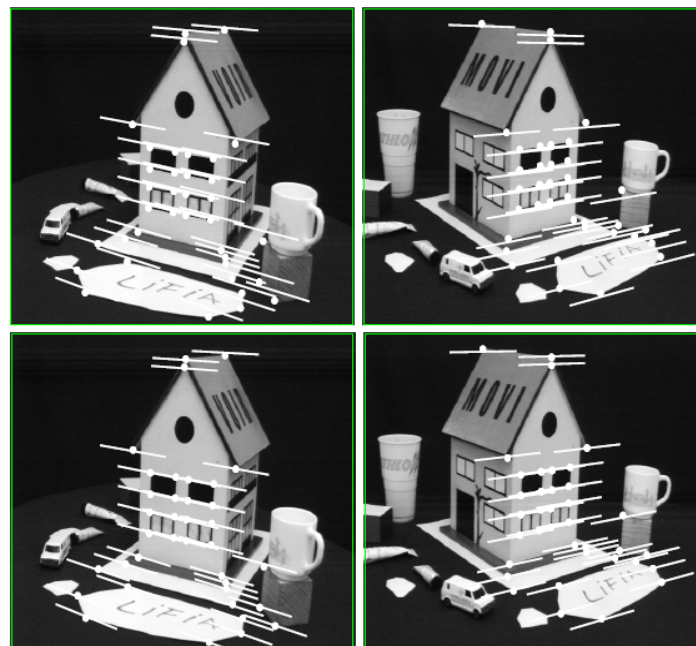
❖ **OFFSET**: impongo origine nel centro ottico

$$\langle x_{i,n}, y_{i,n} \rangle = \frac{1}{D} (\langle x_i, y_i \rangle - \langle x_C, y_C \rangle)$$

❖ **SCALA**: distanza media dall'origine:  $\bar{r} = \sqrt{2}$ :

$$\bar{r} = \frac{1}{N} \sum_{i=2}^N \sqrt{x_{i,n}^2 + y_{i,n}^2} = \sqrt{2}$$

## Comparison of estimation algorithms



3D reconstruction  
test images:  
[HAL - inria.fr](http://HAL-inria.fr)

	8-point	normalized 8-point	nonlinear least squares
Distanza media $d(x, Fx')$	2.33 pixels	0.92 pixel	0.86 pixel
Distanza media $d(x', Fx)$	2.18 pixels	0.85 pixel	0.80 pixel



## Normalized eight-point algorithm (Hartley, 1995):

1. Normalizzazione delle coordinate immagine:

- Origine nel centro ottico dell'immagine
- Scalatura → media quadratica delle distanze dall'origine = 2

$$\mathbf{p}_n = \mathbf{T}\mathbf{p} ; \mathbf{p}'_n = \mathbf{T}'\mathbf{p}'$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & D \end{bmatrix}$$

2. Calcolo di  $\mathbf{F}_n$  (coordinate normalizzate), con l'algoritmo a 8 punti

3. Imposizione del vincolo:  $\text{rank}(\mathbf{F}_n) = 2$

- Calcolo  $\text{SVD}(\mathbf{F}_n)$  e azzero il valore singolare più piccolo

$$\mathbf{F}_n \xrightarrow{\text{SVD}} \mathbf{F}_n = \mathbf{U} \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & t \end{bmatrix} \mathbf{V}^T \rightarrow \tilde{\mathbf{F}}_n = \mathbf{U} \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^T$$

4. Mediante  $\mathbf{T}$  e  $\mathbf{T}'$ , si ritrasforma la matrice fondamentale normalizzata  $\mathbf{F}_n$  nella matrice fondamentale  $\mathbf{F}$  (riferita alle coordinate immagine):

$$\mathbf{p}'^T \mathbf{F} \mathbf{p}' = \mathbf{p}_n^T \underbrace{(\mathbf{T}^{-1})^T \mathbf{F} \mathbf{T}'^{-1}}_{\mathbf{F}_n} \mathbf{p}_n = 0 \rightarrow \mathbf{F}_n = (\mathbf{T}^{-1})^T \mathbf{F} \mathbf{T}'^{-1} \rightarrow$$

$$\mathbf{F} = \mathbf{T}^T \tilde{\mathbf{F}}_n \mathbf{T}'$$

## Da $\mathbf{F}$ alla calibrazione estrinseca ( $\mathbf{R}, \mathbf{t}$ )



da  $\mathbf{F}$  alla calibrazione estrinseca:

1. Stima di  $\mathbf{F}$  a partire dalle corrispondenze (**weak calibration**)

2. Se le camere sono calibrate (calibrazione intrinseca:  $f, \langle x_c, y_c \rangle$  noti), conosco  $\mathbf{K}$

➔ da  $\mathbf{F}$ , posso ricavare  $\mathbf{E}$ :

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad \longrightarrow \quad \tilde{\mathbf{E}} = \mathbf{K}^T \mathbf{F} \mathbf{K}'$$

Per migliorare la stima impongo  $\text{rank}(\mathbf{E}) = 2$ , con i 2 valori singolari uguali (=1):

$$\tilde{\mathbf{E}} \xrightarrow{\text{SVD}} \tilde{\mathbf{E}} = \mathbf{U} \mathbf{W} \mathbf{V}^T \rightarrow \mathbf{E} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^T$$

3. Da  $\mathbf{E}$ , posso ricavare  $\mathbf{R}$  e  $\mathbf{t}$ :

$$\tilde{\mathbf{E}} \xrightarrow{\text{SVD}} \mathbf{E} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^T \rightarrow \begin{cases} \mathbf{t} = \pm \mathbf{u}_3 \quad (\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3]) \\ \mathbf{R} = \mathbf{U} \begin{bmatrix} 0 & \mp 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^T \end{cases}$$



**Ambiguità di segno**  
nella stima di  $R$  e  $t$  da  $E$ :

$$\tilde{E} \xrightarrow{SVD} U, W, V^T \rightarrow$$

$$\begin{cases} t = \pm u_3 \quad (U = [u_1 \ u_2 \ u_3]) \\ R = U \begin{bmatrix} 0 & \mp 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T \end{cases}$$

ho ambiguità di segno  
su  $R$  e su  $t \rightarrow 4$  casi possibili

**Soluzione corretta:**  
punti 3D di scena  
**davanti** a entrambe le camere

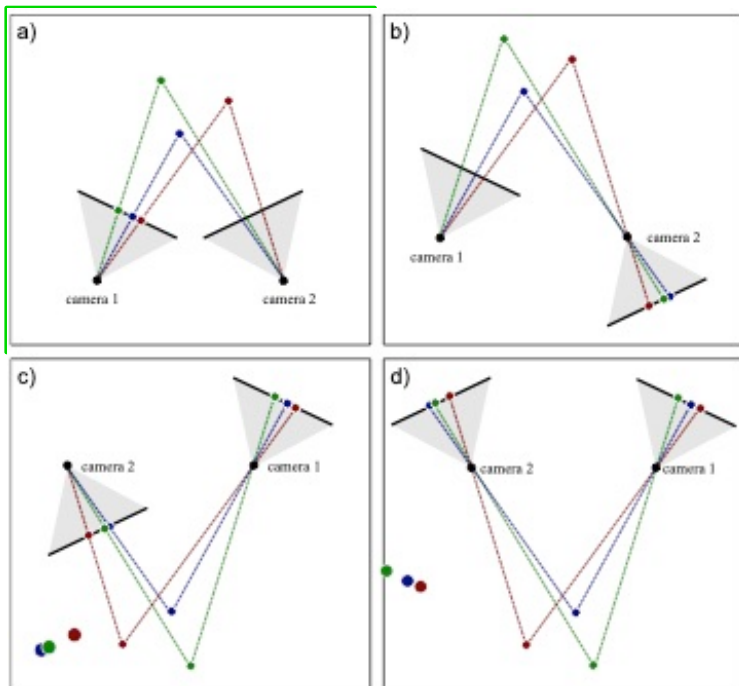


figura: S. Prince, Computer Vision

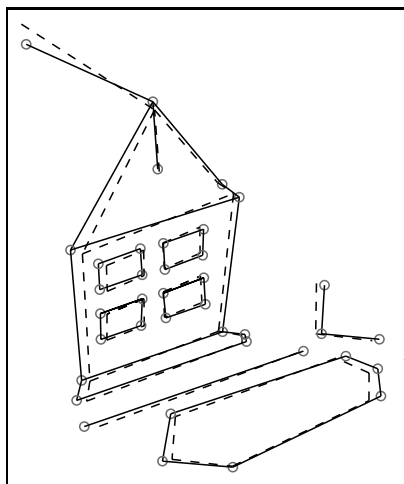
## SfM binoculare – camere calibrate



### 8-Point Algorithm (Longuet-Higgins, 1981):

Risultati:

- ❖ Calibrazione estrinseca ( $R, t$ )
- ❖ Coordinate 3D dei punti corrispondenti



— : punti ricostruiti  
- - - : ground truth

#### 1. Estimate $\mathcal{F}$ .

- Compute Hartley's normalization transformation  $\mathcal{T}$  and  $\mathcal{T}'$ , and the corresponding points  $\tilde{p}_i$  and  $\tilde{p}'_i$ .
- Use homogeneous linear least squares to estimate the matrix  $\tilde{\mathcal{F}}$  minimizing  $\frac{1}{n} \sum_{i=1}^n (\tilde{p}_i^T \tilde{\mathcal{F}} \tilde{p}'_i)^2$  under the constraint  $\|\tilde{\mathcal{F}}\|_{\mathcal{F}}^2 = 1$ .
- Compute the singular value decomposition  $U \text{diag}(r, s, t) V^T$  of  $\tilde{\mathcal{F}}$ , and set  $\tilde{\mathcal{F}} = U \text{diag}(r, s, 0) V^T$ .
- Output the fundamental matrix  $\mathcal{F} = \mathcal{T}^T \tilde{\mathcal{F}} \mathcal{T}'$ .

#### 2. Estimate $\mathcal{E}$ .

- Compute the matrix  $\tilde{\mathcal{E}} = \mathcal{K}^T \mathcal{F} \mathcal{K}'$ .
- Set  $\mathcal{E} = U \text{diag}(1, 1, 0) V^T$ , where  $U W V^T$  is the singular value decomposition of the matrix  $\tilde{\mathcal{E}}$ .

#### 3. Compute $\mathcal{R}$ and $t$ .

- Compute the rotation matrices  $\mathcal{R}' = U W V^T$  and  $\mathcal{R}'' = U W'^T V'^T$ , and the translation vectors  $t' = u_3$  and  $t'' = -u_3$ , where  $u_3$  is the third column of the matrix  $U$ .
- Output the combination of the rotation matrices  $\mathcal{R}'$ ,  $\mathcal{R}''$ , and the translation vectors  $t'$ ,  $t''$  such that the reconstructed points lie in front of both cameras.

...a meno di un **fattore di scala!**

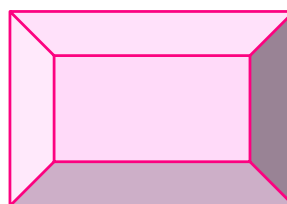
## Esercitazione MATLAB®:

- ❖ **8-point Algorithm**  
stima della matrice fondamentale e calibrazione estrinseca di una coppia binoculare con calibrazione intrinseca (parametri intrinseci noti)
- ❖ MATLAB LIVE script: `Fundamental.mlx`
- >> `edit Fundamental.mlx`

## SfM a camere multiple

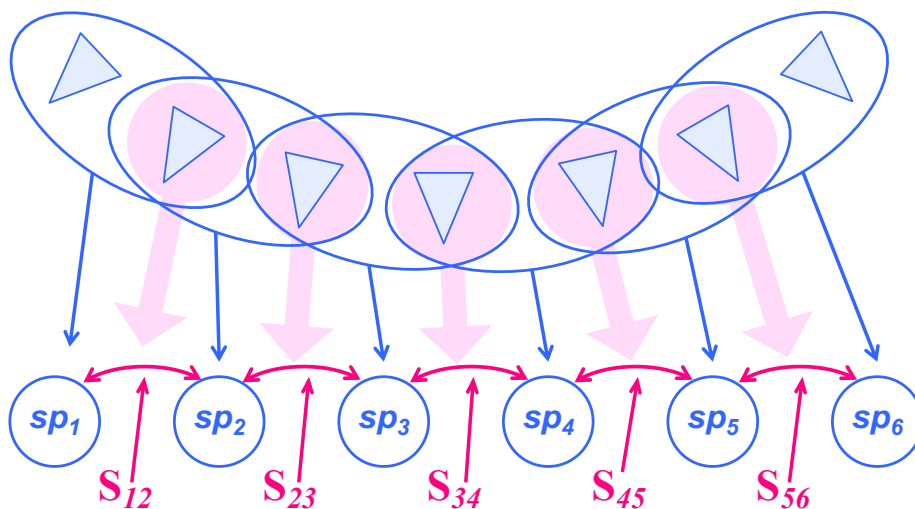
### SfM con più di 2 camere (calibrate):

- ❖ risolvo a coppie
- ❖ come fondo le ricostruzioni parziali?



### Grafo $G(V, e)$

- ❖ **V**: coppia di camere ricostruite (*stereo pair*)
- ❖ **e**: arco tra SP aventi almeno **3 punti in comune**
- ❖ La relazione tra i 2 sist. di rif.  $h$  e  $k$  è una **similarità**:  $S_{hk}(R, t, \lambda)$   
(rototraslazione + scala)

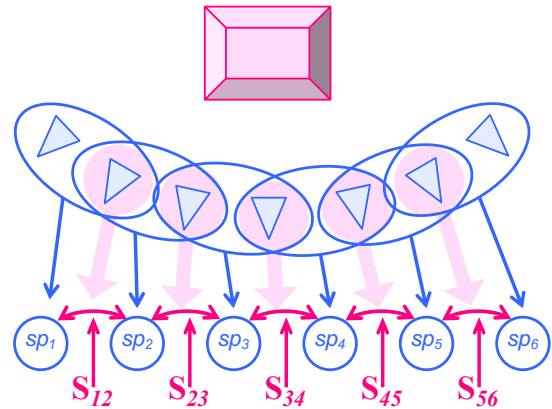


## SfM con più di 2 camere (calibrate):

- ❖ risolvo a coppie
- ❖ come fondo le ricostruzioni parziali?

## Grafo di ricostruzione $G(V, e)$

- ❖ **V**: coppia di camere ricostruite
- ❖ **e**: congiunge 2 coppie aventi almeno **3 punti in comune,  $P_j$**



Arco  $e_{hk}$ : collega rif. camera  $h$  a rif. camera  $k$

- ❖ La relazione tra 2 sist. di riferimento  $h$  e  $k$  è una **similarità:  $S_{hk}$**  ( $R, t, \lambda$ )

$$e_{hk} \rightarrow S_{hk} = \langle R_{hk}, t_{hk}, \lambda_{hk} \rangle$$

- ❖ Posso calcolare la similarità  $S_{hk}$  tra  $h$  e  $k$ :

$$\langle {}^h P_j, {}^k P_j \rangle, \quad j = 1..n, n \geq 3$$

$${}^h P_j = S_{hk} {}^k P_j, \quad \forall j = 1..n, n \geq 3$$

$$\hat{S}_{hk} = \operatorname{argmin}_{S_{hk}} \left\| {}^h P_j - S_{hk} {}^k P_j \right\|^2$$

- ❖ Se il grafo è connesso  $\rightarrow$  **full 3D reconstruction**

# SfM a camere multiple

## SfM con $N$ camere:

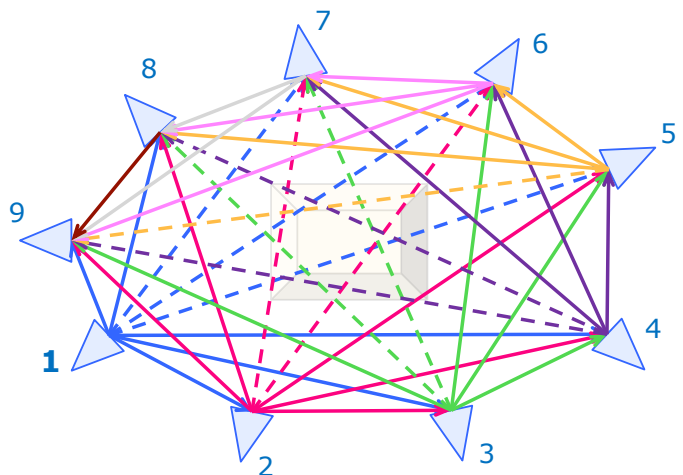
- ❖ risolvo  $[Rt\lambda]$  a coppie, per tutte le coppie possibili:  $\frac{1}{2}(N-1)^2$

Approccio:  
**grafo completamente connesso**

Normalizzando tutte le viste a una scala comune  $\lambda_0$  e sfruttando il vincolo di composizione delle rototraslazioni:

$$[Rt]_{ik} = [Rt]_{ij} \cdot [Rt]_{jk}$$

$\rightarrow$  determino  $N - 1$  rototraslazioni che rendono consistente la semi-matrice



8	$[Rt]_{89}$							
7	$[Rt]_{79}$	$[Rt]_{78}$						
6	$[Rt]_{69}$	$[Rt]_{68}$	$[Rt]_{67}$					
5	---	$[Rt]_{58}$	$[Rt]_{57}$	$[Rt]_{56}$				
4	---	---	$[Rt]_{47}$	$[Rt]_{46}$	$[Rt]_{45}$			
3	$[Rt]_{39}$	---	---	$[Rt]_{36}$	$[Rt]_{35}$	$[Rt]_{34}$		
2	$[Rt]_{29}$	$[Rt]_{28}$	---	---	$[Rt]_{25}$	$[Rt]_{24}$	$[Rt]_{23}$	
1	$[Rt]_{19}$	$[Rt]_{18}$	---	---	---	$[Rt]_{14}$	$[Rt]_{13}$	$[Rt]_{12}$
view	9	8	7	6	5	4	3	2



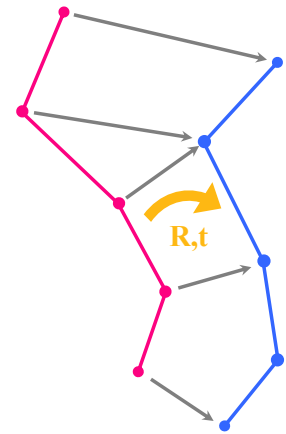
Approccio alternativo:

## Registrazione di point clouds

- ❖ Ricostruzioni binoculari → 3D *point cloud* parziali
- ❖ Allineamento (registrazione) delle diverse ricostruzioni nello stesso sistema di riferimento → **full 3D reconstruction**

Definizione del problema:

- ❖ Date 2 point cloud (Model, Scene),
- ❖ determinare  $(R, t)$  che sovrappone esattamente **Scene** → **Model**



## ICP (Iterative Closest Point) algorithm:

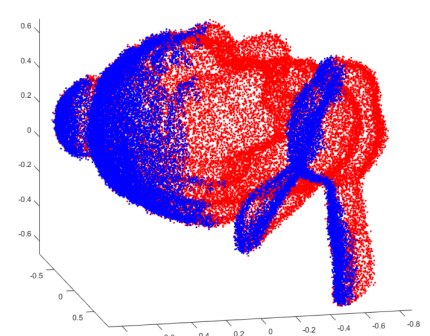
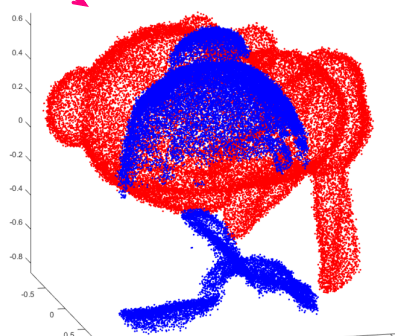
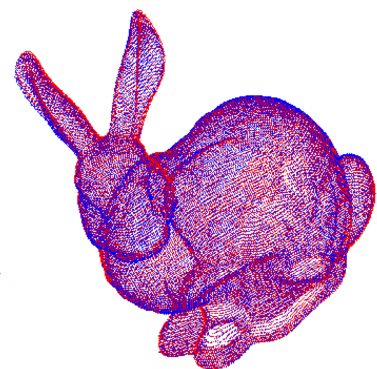
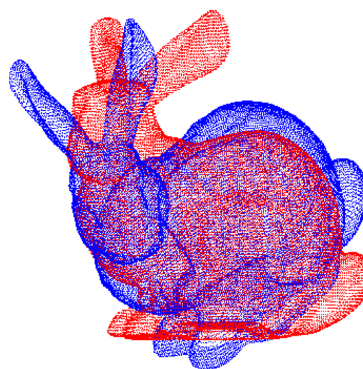
```

Err' ← Inf; R ← RInit; t ← tInit;
repeat
  Err ← Err' ;
  RegScene ← RotoTrasla( Scene, R, t );
  Pairs ← ClosestPairs( RegScene, Model );
  (R, t, Err') ← Compute_Rt(Scene, Model, Pairs);
until | Err-Err' | < TolleranzaErrore
return( R, t );
    
```

# SfM a camere multiple

## ICP (Iterative Closest Point)

- ❖ Garantisce **decrecita monotona** di  $E$ , ma non la correttezza!
  - posso incorrere in **minimi locali**
- ❖ Implementazioni ottimizzate per ridurre il tempo di ricerca del punto più vicino (**closest point**)
- ❖ Funziona anche per superfici **solo parzialmente sovrapposte** (partial overlapping)



## Structure from Motion (SfM) Active 3D Vision

- ❖ **SfM (camere intrinsecamente calibrate)**
  - caso binoculare – 8-point algorithm
  - camere multiple
- ❖ **SfM (camere non calibrate)**
  - camere multiple
- ❖ **Active 3D vision**
  - Laser scanning, Structured light, Shape from silhouettes

## Structure from motion – camere non calibrate



### SfM da camere non calibrate – definizione del problema

Date:

- ❖  $m$  immagini di  $n$  punti 3D nella scena

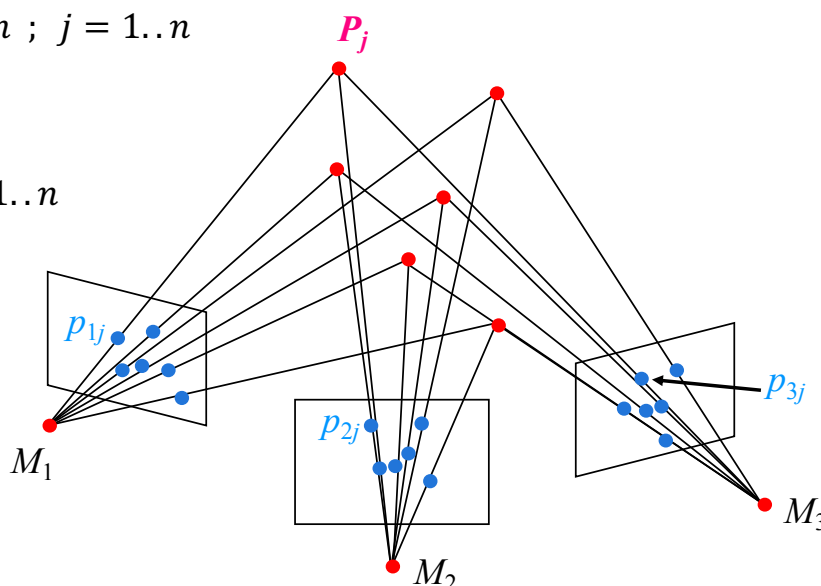
$$z_{ij} \mathbf{p}_{ij} = \mathbf{M}_i \mathbf{P}_j, \quad i = 1..m ; j = 1..n$$

- ❖ le  $n$   $m$ -ple di corrispondenza  $\mathbf{p}_{ij}$  (punto  $j$  in camera  $i$ )

$$\langle \mathbf{p}_{1j}, \mathbf{p}_{2j}, \dots, \mathbf{p}_{mj} \rangle, \quad j = 1..n$$

Stimare:

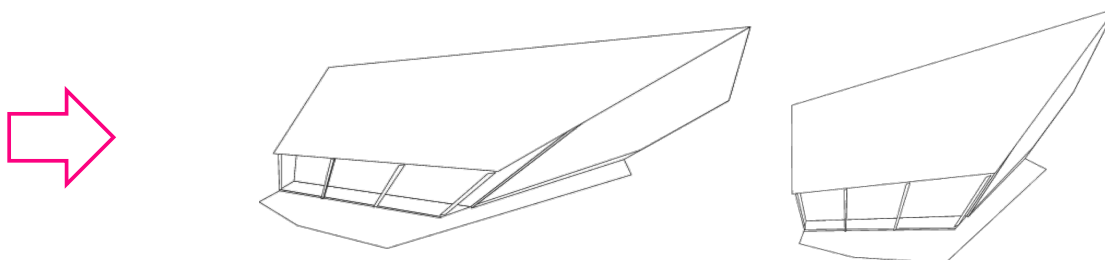
- ❖  $m$  matrici di proiezione  $\mathbf{M}_i$  e
- ❖  $n$  punti 3D di scena  $\mathbf{P}_j$



## Ambiguità di ricostruzione prospettica:

Entrambe le ricostruzioni soddisfano i vincoli proiettivi

→ entrambe soluzioni valide



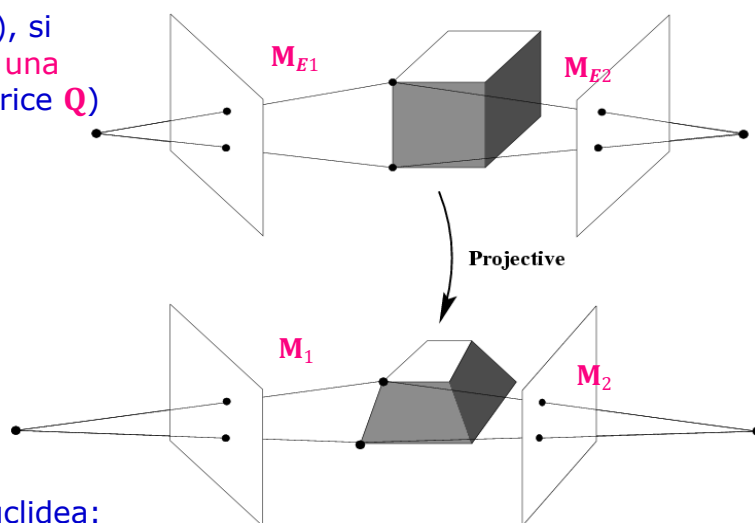
# Projective SfM: reconstruction ambiguity

## Ambiguità di ricostruzione prospettica:

- Se le camere non sono note ( $M_i$ ), si ottiene una soluzione a meno di una trasformazione prospettica (matrice  $Q$ )

$$\mathbf{p} = \mathbf{M}_E \mathbf{P} = (\mathbf{M} \mathbf{Q}^{-1})(\mathbf{Q} \mathbf{P})$$

$$\mathbf{Q}_{[4 \times 4]} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix}$$



## Soluzione:

Detta  $M_E$  la matrice di proiezione Euclidea:

- si determina  $M = Q M_E$  imponendo  $\mathbf{p} = M \mathbf{P}$
- si determina  $Q$  sfruttando altri vincoli →  $M_E = M Q^{-1} \rightarrow \mathbf{P}_j$  (ricostruzione)

- $2mn$  equazioni
- $3n + 11m - 15$  incognite

$$2mn \geq 3n + 11m - 15$$

$$\Rightarrow m = 2 \rightarrow n \geq 7, m = 3 \rightarrow n \geq 6$$



## Caso binoculare: calcolo matrice fondamentale $F$ tra 2 viste

Scelgo come sistema di riferimento di scena: camera 2

- ❖ camera 1:  $M = [A|b] \rightarrow M_E = [A|b] Q^{-1}$
- ❖ camera 2:  $M' = [I_3|0] \rightarrow M'_E = [I_3|0] Q^{-1}$

Dato:  $\tilde{P} = QP$  ( $\tilde{P}$ : con distorsione prospettica)

Si ha che:  $Z'p' = [I_3 | 0]\tilde{P}$ ,  $Zp = [A | b]\tilde{P}$

$$\rightarrow Zp = A\tilde{P} + b = A[I_3 | 0]\tilde{P} + b = Z'Ap' + b$$

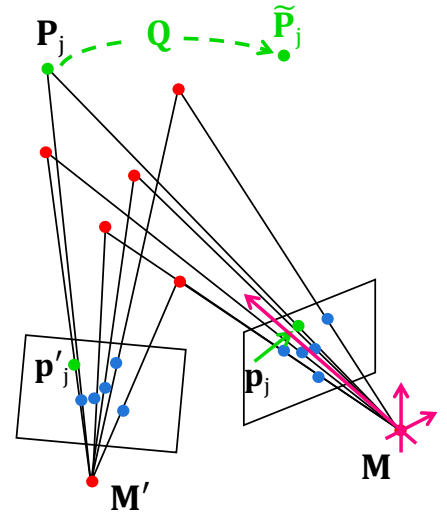
Vettori  $a, b, a$  coplanari  
 $\rightarrow$  doppio prod. misto=0

$$Zp \times b = Z'Ap' \times b + b \times b$$

$$(\lambda a \times b) \cdot a = 0 \rightarrow (Zp \times b) \cdot p = (Z'Ap' \times b) \cdot p = 0$$

$$\rightarrow (Ap' \times b) \cdot p = p \cdot (b \times Ap') = p \cdot (b \times A)p' = p^T (b \times A) p' = 0$$

$$\text{Ma: } p^T F p' = 0 \rightarrow F = b \times A, \quad F^T b = 0 \rightarrow b: \text{epipolo}$$



## Caso binoculare: calcolo matrice fondamentale $F$ tra 2 viste

$$F = b \times A, \quad F^T b = 0 \rightarrow b: \text{epipolo}$$

Da  $F$ , ottengo  $A$  e  $b$ :

$$F^T b = 0 \rightarrow b = \min_b [F^T b]; \quad \|b\| = 1$$

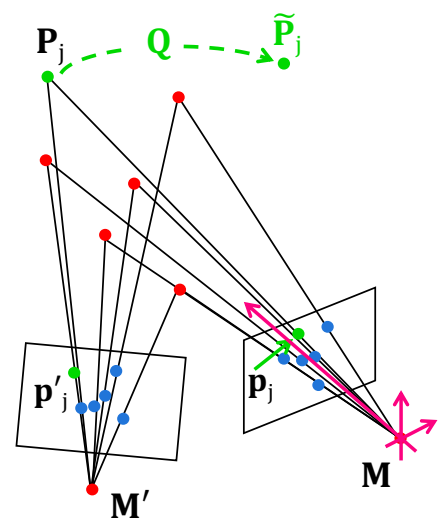
$$\rightarrow A = -b \times F$$

Noti  $A$  e  $b$ , posso ricostruire il punto  $P$  determinando la sua distanza  $Z$  ( $Z'$ )

$$Zp = Z'Ap' + b \rightarrow Z, Z'$$

Coordinate euclidee di  $P$ :  
 nel sistema di riferimento  
 di camera 2

$$P = Z p$$





## Bundle Adjustment

Metodo tradizionale (fotogrammetria) non lineare per determinare **structure & motion**:

❖ Imposizione **vincoli di proiezione**:

- $n$  punti  $\rightarrow n$  fasci (*bundles*) di rette (raggi) uscenti da ogni centro camera  $O_i$
- $m$  camere  $\rightarrow m$  fasci (*bundles*) di rette (raggi) uscenti da ogni punto  $P_j$

$$x_{ij} = \frac{\mathbf{m}_{i1} \mathbf{P}_j}{\mathbf{m}_{i3} \mathbf{P}_j}, \quad y_{ij} = \frac{\mathbf{m}_{i2} \mathbf{P}_j}{\mathbf{m}_{i3} \mathbf{P}_j}$$

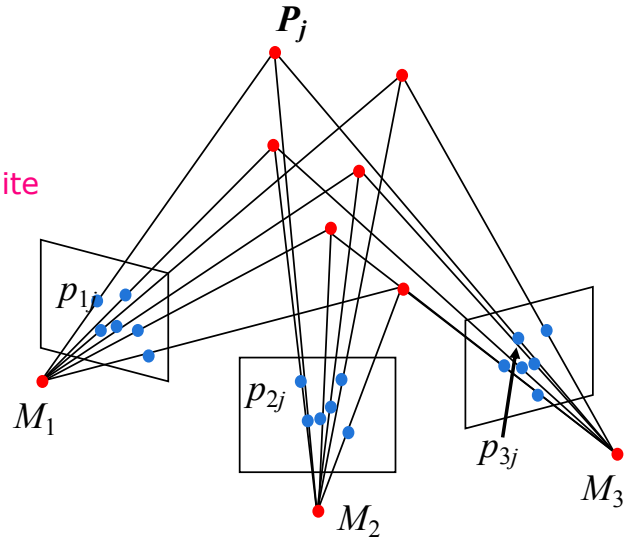
- $mn$  equazioni,  $9 + 11m + 3n - 15$  incognite

❖ **Soluzione**:

minimizzazione dell'errore di riproiezione:

$$E = \sum_{i,j} d^2(\mathbf{p}_{ij}, \mathbf{M}_i \mathbf{P}_j)$$

$$= \sum_{i,j} \left[ \left( x_{ij} - \frac{\mathbf{m}_{i1} \mathbf{P}_j}{\mathbf{m}_{i3} \mathbf{P}_j} \right)^2 + \left( y_{ij} - \frac{\mathbf{m}_{i2} \mathbf{P}_j}{\mathbf{m}_{i3} \mathbf{P}_j} \right)^2 \right]$$



## Bundle adjustment



### Bundle Adjustment

- ❖ risolvo mediante minimizzazione non lineare (Levenberg-Marquardt)

$$\langle \mathbf{M}, \mathbf{P} \rangle = \underset{\mathbf{M}, \mathbf{P}}{\operatorname{argmin}} \left( E = \sum_{i,j} d^2(\mathbf{p}_{ij}, \mathbf{M}_i \mathbf{P}_j) \right)$$

### Ricostruzione Euclidea

I risultati:  $\mathbf{M}_i, i=1..m$ ;  $\mathbf{P}_j, j=1..n$  sono corretti a meno di una trasformazione prospettica  $\mathbf{Q}$   $[4 \times 4]$

$$\mathbf{M}_i^E = \mathbf{M}_i \mathbf{Q}, \quad i = 1..m$$

$$\mathbf{P}_j^E = \mathbf{Q}^{-1} \mathbf{P}_j, \quad j = 1..n$$

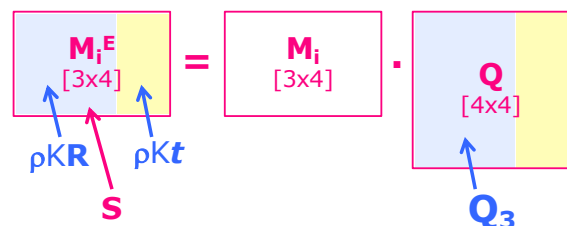
$\mathbf{Q}$  - upgrade matrix:

$$\mathbf{M}_i^E = \mathbf{M}_i \mathbf{Q} = \rho_i \mathbf{K}_i \left[ \mathbf{R}_i | \mathbf{t}_i \right]$$

$$\rightarrow \mathbf{S}_i = \mathbf{M}_i \mathbf{Q}_3 = \rho_i \mathbf{K}_i \mathbf{R}_i$$

$\mathbf{S}$  è una matrice di rotazione:

- ➔ vettori riga  $\mathbf{s}_i$  ( $i = 1,2,3$ ) **ortogonali**
- ➔ vettori riga hanno la **stessa norma**



$$\mathbf{s}_i \cdot \mathbf{s}_j^T = \mathbf{m}_i \mathbf{Q}_3 \mathbf{Q}_3^T \mathbf{m}_j^T = 0, \quad i, j = 1..3, \quad i \neq j \quad \longrightarrow \quad 3 \text{ equazioni/camera}$$

$$\mathbf{s}_1 \cdot \mathbf{s}_1^T = \mathbf{s}_2 \cdot \mathbf{s}_2^T = \mathbf{s}_3 \cdot \mathbf{s}_3^T \quad \longrightarrow \quad 2 \text{ equazioni/camera}$$

## Ricostruzione Euclidea

❖ Per ogni camera  $M_i$ :

$$\begin{cases} \mathbf{m}_1 \mathbf{A} \mathbf{m}_2^T = 0 \\ \mathbf{m}_1 \mathbf{A} \mathbf{m}_3^T = 0 \\ \mathbf{m}_2 \mathbf{A} \mathbf{m}_3^T = 0 \\ \mathbf{m}_1 \mathbf{A} \mathbf{m}_1^T - \mathbf{m}_2 \mathbf{A} \mathbf{m}_2^T = 0 \\ \mathbf{m}_2 \mathbf{A} \mathbf{m}_2^T - \mathbf{m}_3 \mathbf{A} \mathbf{m}_3^T = 0 \end{cases}$$

$$s_i \cdot s_j = 0$$

$$s_i \cdot s_i = 1$$

$$\mathbf{A} = \mathbf{Q}_3 \mathbf{Q}_3^T \quad [4 \times 4]$$

$$a_{ij} = q_{i1}q_{j1} + q_{i2}q_{j2} + q_{i3}q_{j3}$$

$$\begin{bmatrix} \mathbf{M}_i^E \\ [3 \times 4] \end{bmatrix} = \begin{bmatrix} \mathbf{M}_i \\ [3 \times 4] \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Q} \\ [4 \times 4] \end{bmatrix}$$

$\rho \mathbf{K} \mathbf{R}$       $\rho \mathbf{K} \mathbf{t}$   
 $\mathbf{S}$

$\mathbf{Q}_3$       $\mathbf{q}_4$

❖ **5 equazioni/camera**, **11 incognite quadratiche** ( $\mathbf{Q}_3$ :  $\mathbf{q}_{ij}$ )  
 o **16 incognite lineari** ( $\mathbf{A}$ :  $\mathbf{a}_{ij}$ )

$$\mathbf{A} = \mathbf{Q}_3 \mathbf{Q}_3^T \quad [4 \times 4]$$

$$\text{SVD}[\mathbf{A}] \rightarrow \mathbf{A} = \mathbf{U}_3 \mathbf{D}_3 \mathbf{V}_3^T = \left( \mathbf{U}_3 \sqrt{\mathbf{D}_3} \right) \left( \sqrt{\mathbf{D}_3}^{-T} \mathbf{V}_3^T \right) = \mathbf{Q}_3 \mathbf{Q}_3^T \rightarrow \mathbf{Q}_3 = \mathbf{U}_3 \sqrt{\mathbf{D}_3}$$

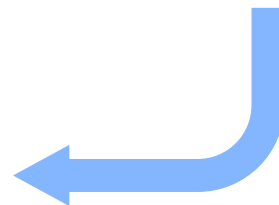
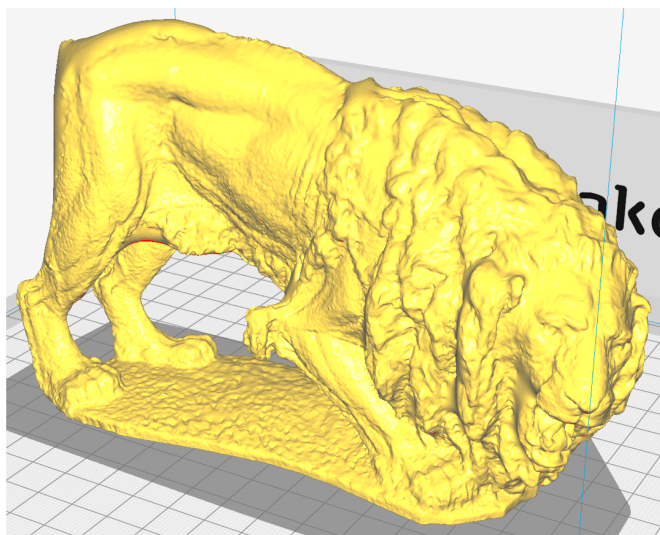
❖  $\mathbf{q}_4$ : scelgo come origine il Centro Ottico di Camera 1  $\rightarrow \mathbf{t}_1 = [0, 0, 0]$

$$\mathbf{M}_1 \mathbf{q}_4 = \rho_1 \mathbf{K}_1 \mathbf{t}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \mathbf{Q} = [\mathbf{Q}_3 | \mathbf{q}_4] \rightarrow \mathbf{M}_i^E = \mathbf{M}_i \mathbf{Q} = \rho_i \mathbf{K}_i \begin{bmatrix} \mathbf{R}_i | \mathbf{t}_i \end{bmatrix} \quad \text{ricostruzione euclidea}$$

## Esempi: uncalibrated, multiple-view 3D reconstruction

### 3D reconstruction pipeline:

1. Feature Detection (SIFT)
2. Binocular self-calibration / reconstruction
3. Global calibration / reconstruction
4. Dense depth maps  $\rightarrow$  point clouds
5. Points  $\rightarrow$  Surface

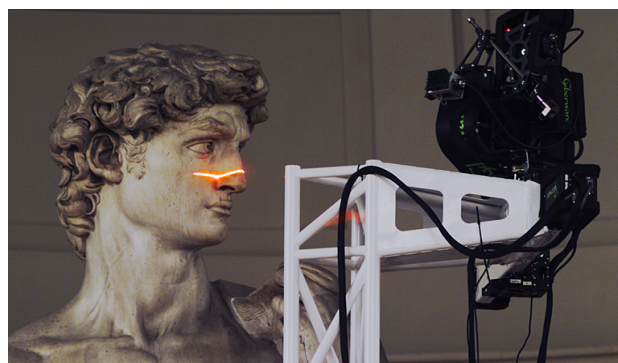
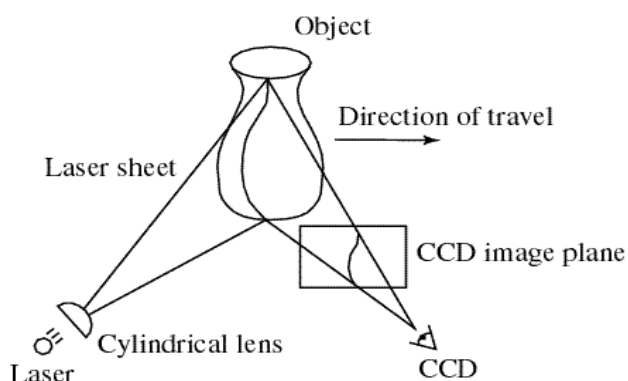


## Structure from Motion (SfM)

### Active 3D Vision

- ❖ **SfM (camere intrinsecamente calibrate)**
  - caso binoculare – 8-point algorithm
  - camere multiple
- ❖ **SfM (camere non calibrate)**
  - caso binoculare
  - camere multiple
- ❖ **Active 3D vision**
  - Laser scanning, Structured light, Shape from silhouettes

## Laser scanning



Digital Michelangelo Project  
<http://graphics.stanford.edu/projects/mich/>

- ❖ **Laser scanning**
  - Proiezione di una singola **linea laser** che scandisce tutta la superficie
  - immagine della linea acquisita dalla camera (sistema **laser/camera calibrato**)
  - Posizione 3D della linea ottenuta per **triangolazione**
- ❖ **Prestazioni:**
  - Apparato di acquisizione complesso
  - Precisione e affidabilità **elevatissime**



## **The Digital Michelangelo Project**

Levoy et al. (1999)

Scansione laser e  
ricostruzione 3D del  
Davide di Michelangelo  
Firenze,  
Galleria dell'Accademia

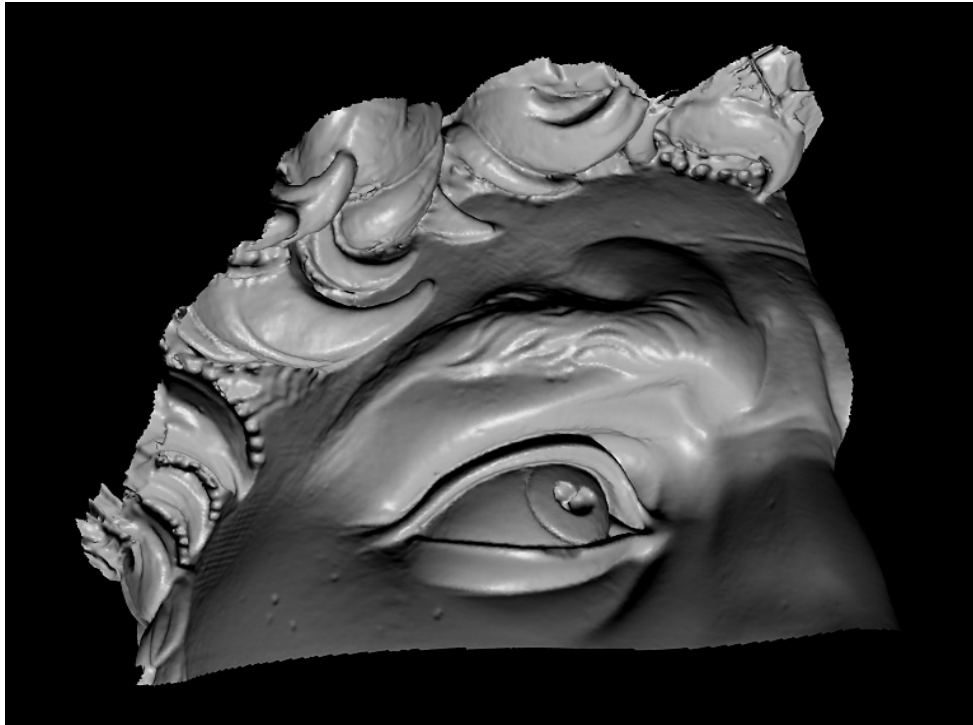
Source: S. Seitz



*The Digital Michelangelo Project, Levoy et al.*

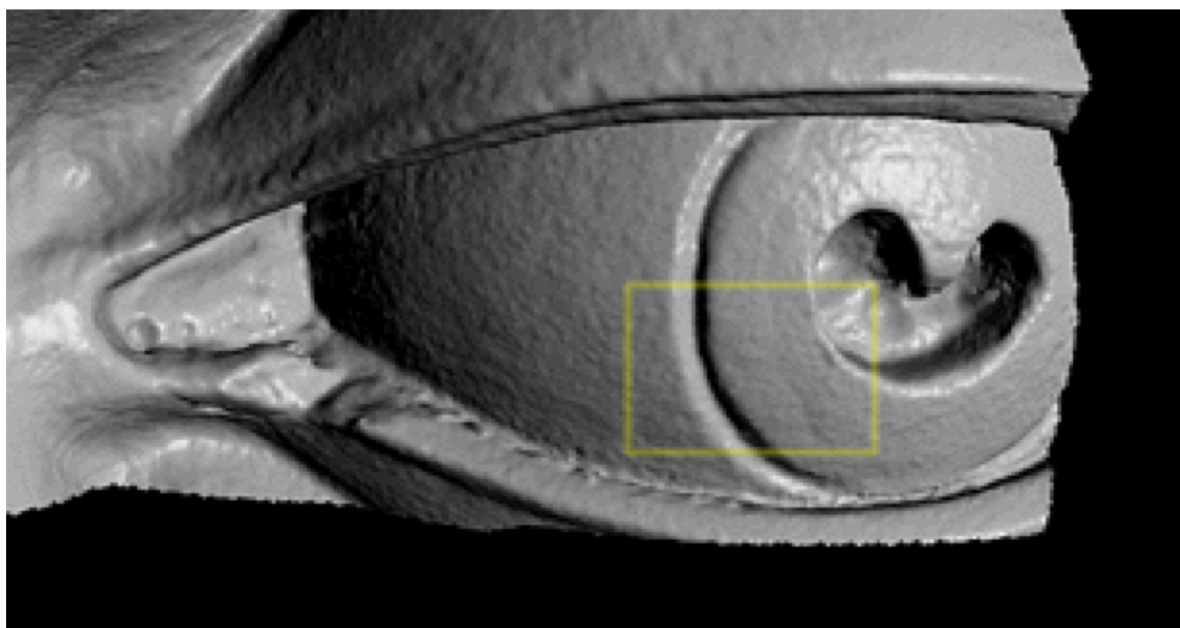
Source: S. Seitz





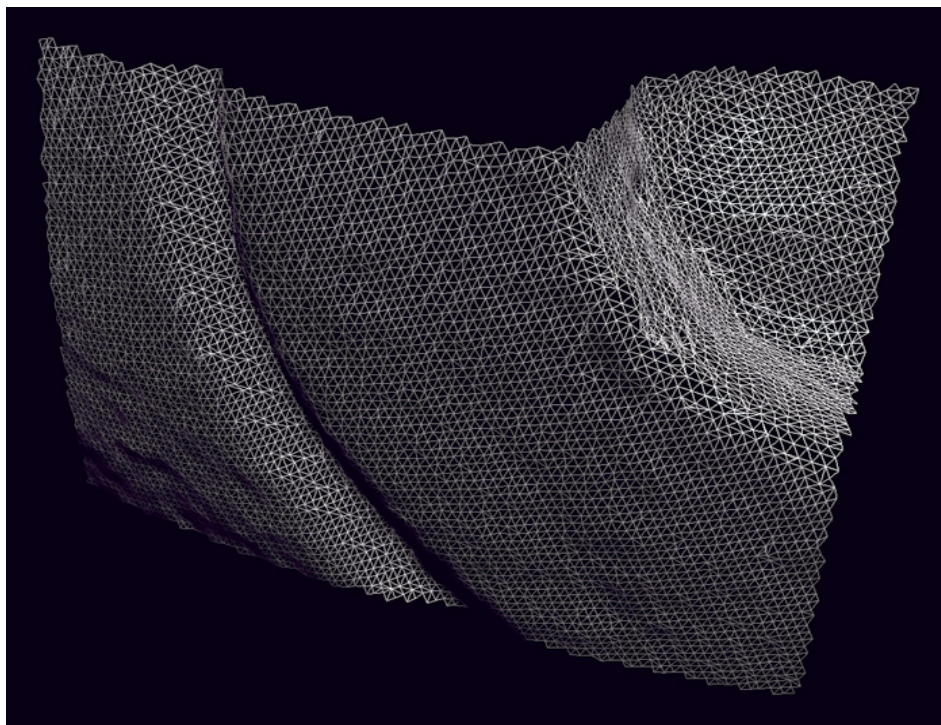
*The Digital Michelangelo Project, Levoy et al.*

Source: S. Seitz



*The Digital Michelangelo Project, Levoy et al.*

Source: S. Seitz



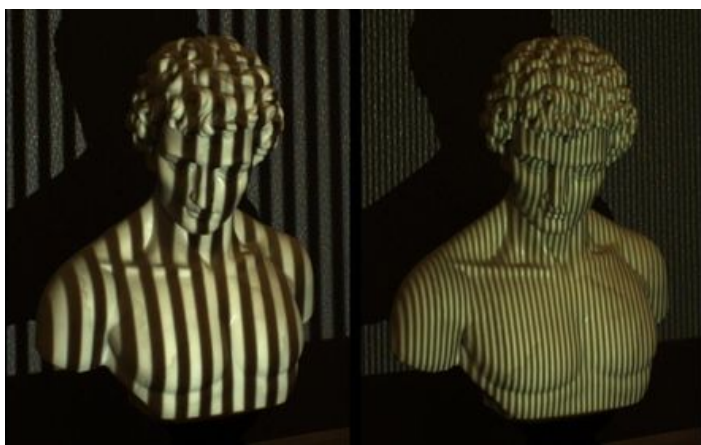
*The Digital Michelangelo Project, Levoy et al.*

Source: S. Seitz

## Luce strutturata: codici a barre, codici di Gray

### Luce strutturata: **codici binari**

- ❖ Geometria proiettore/camera nota (sistema calibrato)
- ❖ Proiezione multipla di pattern differenti:
  - **coarse**: matching initialization
  - **fine**: max 3D resolution



## Luce strutturata: **codici binari**

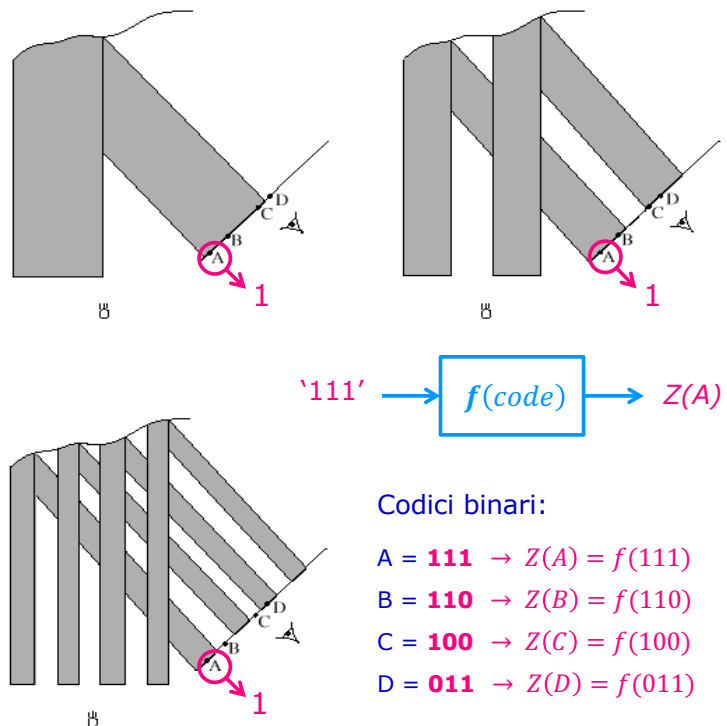
- ❖ Geometria proiettore/camera: nota a priori con precisione
- ❖ Proietta immagini multiple (pattern a 'risoluzione' crescente)

## Proprietà

- ❖ Complessità sistema di acquisizione
  - sistema camera/proiettore calibrato

## Semplicità elaborazione:

- ❖ **Corrispondenza diretta** tra
  - codice letto in un punto e la sua **posizione 3D**
- ❖ Altissima velocità di calcolo
  - ➔ applicazioni industriali (controllo qualità on-line)



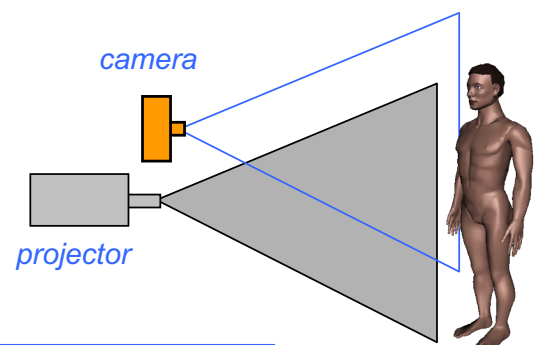
## Codici binari:

- $A = 111 \rightarrow Z(A) = f(111)$
- $B = 110 \rightarrow Z(B) = f(110)$
- $C = 100 \rightarrow Z(C) = f(100)$
- $D = 011 \rightarrow Z(D) = f(011)$

# Active stereo – luce strutturata

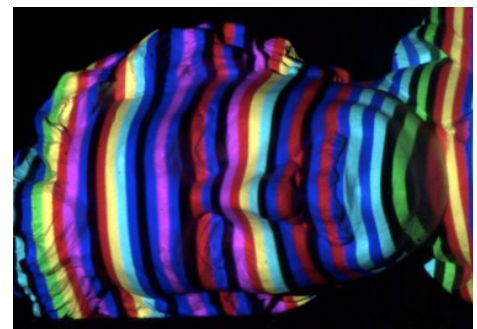
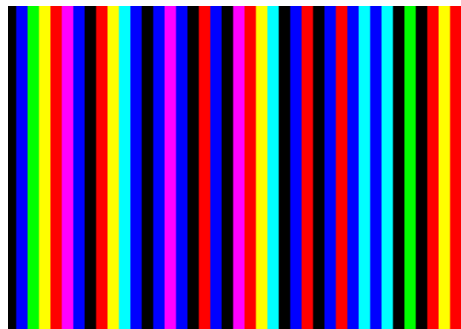
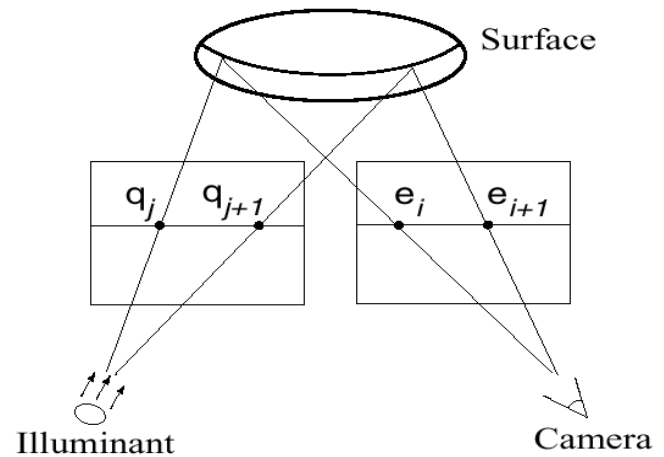


- ❖ Progetto sofisticato del pattern proiettato
  - massimizzazione della robustezza (minima ambiguità) di matching
    - colori 'massimamente' diversi
    - unicità sequenze di colori adiacenti
- ❖ Posso utilizzare **una sola immagine** ricostruzione 3D di **oggetti in movimento**



L. Zhang, B. Curless, S. M. Seitz.  
*Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming*  
 3DPVT 2002

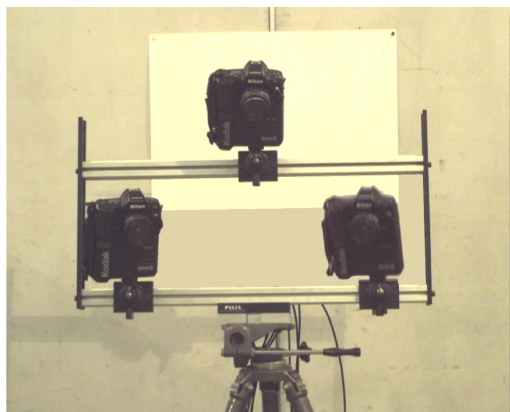
## Visione stereo attiva Camera/projector setup



L. Zhang, B. Curless, S.M. Seitz  
*Rapid Shape Acquisition Using  
Color Structured Light and Multi-  
pass Dynamic Programming.*  
3DPVT 2002

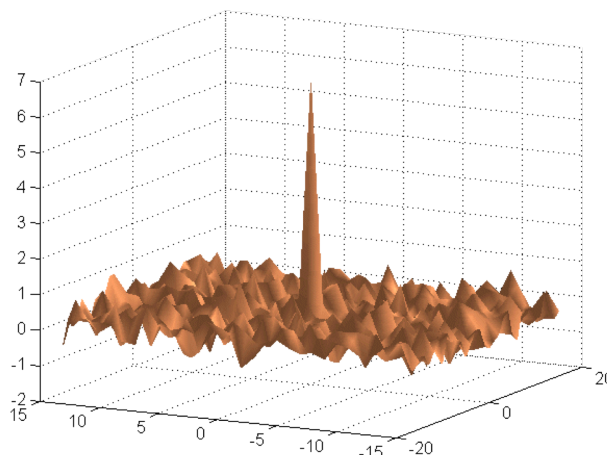
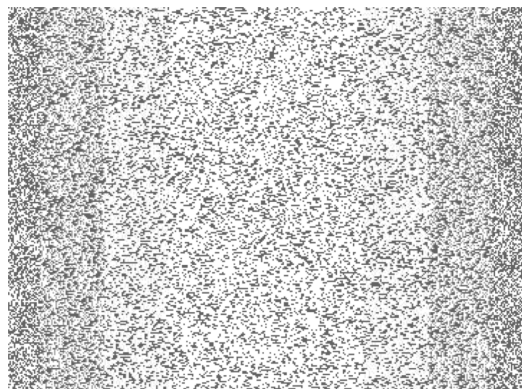
# Luce strutturata: pattern pseudocasuale

- Visione stereo attiva:  
Proiezione di pattern pseudocasuale
- ❖ Acquisizione immagini con scena illuminata da pattern pseudo-casuale

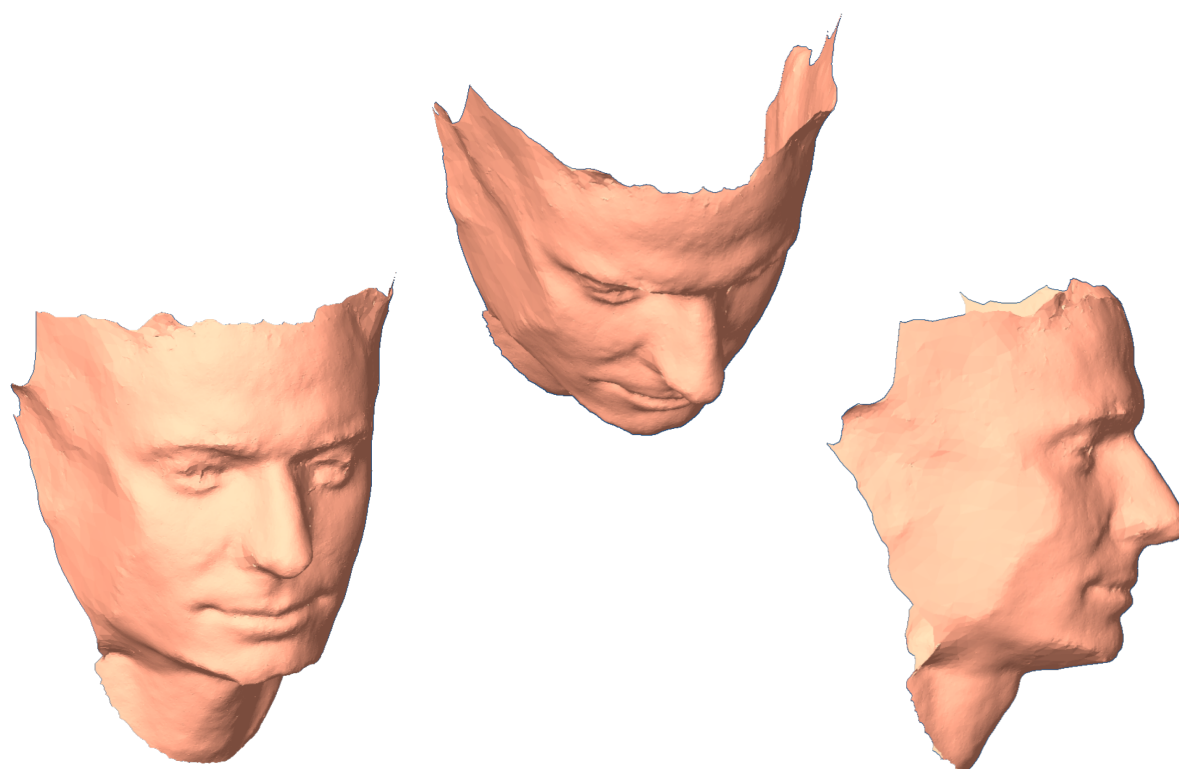


### Proiezione di pattern pseudocasuale

- ❖ L' autocorrelazione del pattern ha un unico massimo ben definito
- ❖ Autocorrelazione robusta a:
  - deformazioni prospettiche del pattern
  - differenti condizioni radiometriche



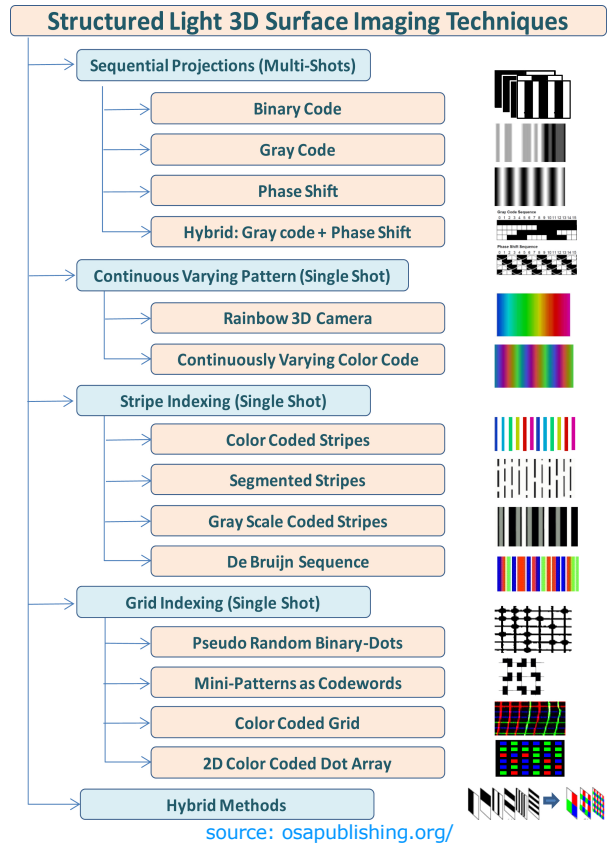
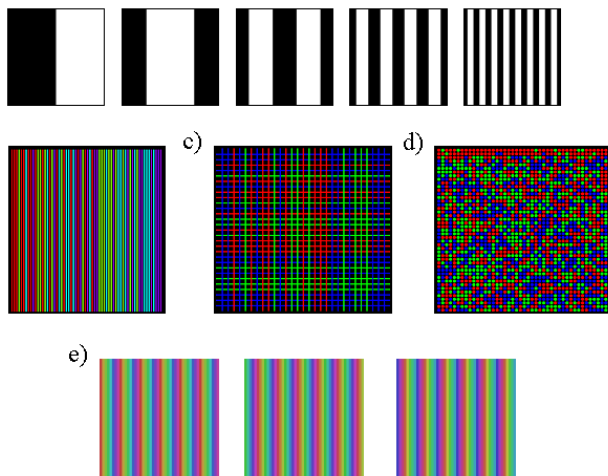
### Esempio di rilievo 3D facciale



## Active stereo con luce strutturata:

Panoramica dello stato dell'arte

- ❖ Time-multiplexing (pattern multipli)
- ❖ Sfruttamento informazioni adiacenti
- ❖ Codifica diretta



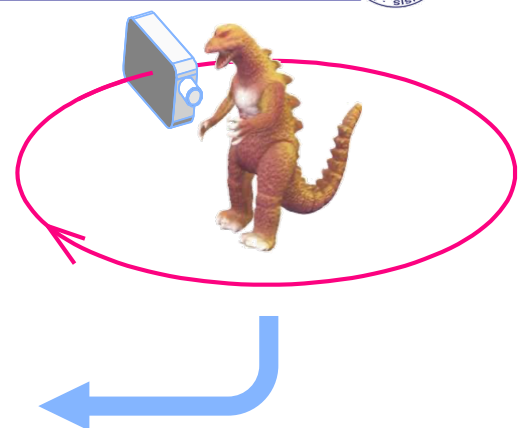
# Ricostruzione da silhouettes

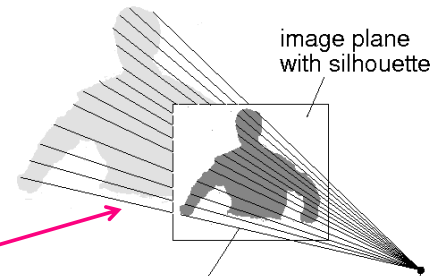
## ❖ Singola camera calibrata (int.+ ext.)

- calibrazione precedente o contestuale (markers in scena)

## ❖ Soggetto inquadrato su sfondo uniforme

→ chroma-keying





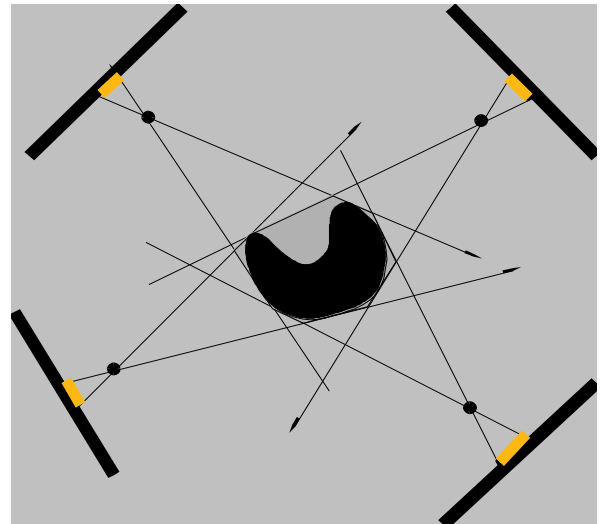
- ❖ Ogni silhouette è un **cono** con:
  - **vertice** nel centro ottico di camera
  - **sezione** normale = silhouette
- ❖ Ricostruzione: **intersezione dei coni** di tutte le viste

## Limitazioni

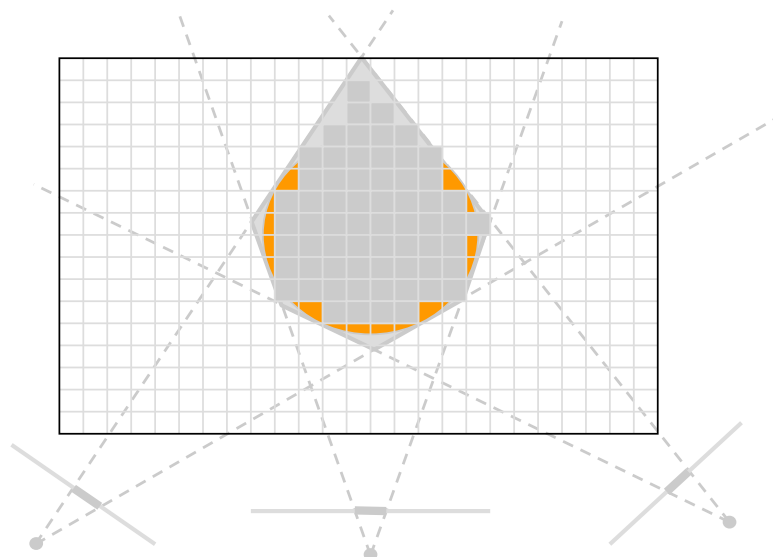
- **Approssimazione convessa** dell'oggetto (*convex hull*)

## Vantaggi

- Alta affidabilità
- Texture originale già a disposizione

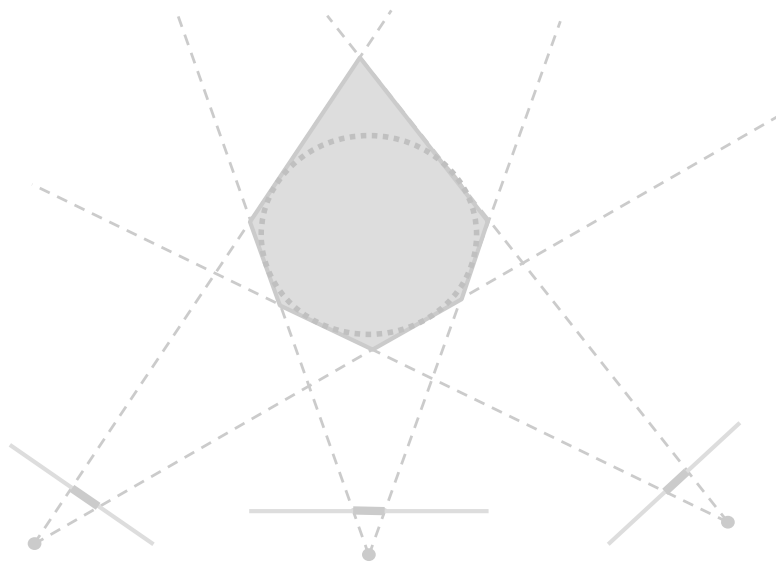


# Voxel algorithm for volume intersection



- ❖ Color voxel black if on silhouette in every image

- ❖ Il volume è un'approssimazione **convessa, per eccesso** dell'oggetto osservato
  - tende all'**inviluppo convesso (convex hull)** della forma dell'oggetto per infinite viste



## Ricostruzione da silhouettes

