


Dal 2D al 3D

Alberto Borghese
Department of Computer Science
University of Milano
http://homes.dsi.unimi.it/MAVR.html
borghese@dsi.unimi.it


Ricostruzione 3D

E' il problema fondamentale della Visione e della Grafica 3D

Argomenti:
· *Processo di formazione di un' immagine su fotocamera/videocamera.*
· *Descrizione del processo mediante trasformazione prospettica.*
· *Ricostruzione della posizione di 3D di punti della scena a partire dalla loro immagine su fotocamere/videocamere.*
· Calibrazione delle camere.
· Errori sistematici introdotti dall'ottica (distorsioni).
· Correzione delle distorsioni (linearizzazione delle camere).

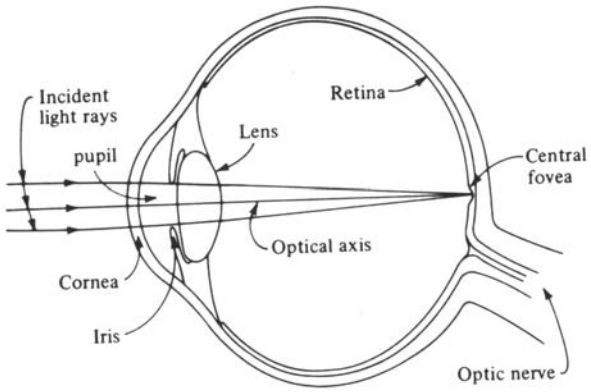

Dal 3D al 2D al 3D

Come si forma un'immagine?

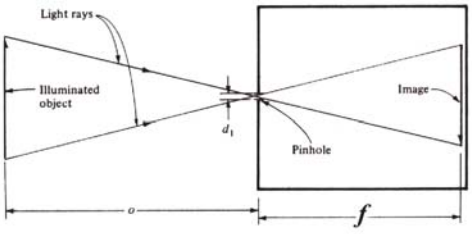
- Scena con oggetti riflettenti.
- Sorgente di illuminazione
- Piano di rilevazione della luce riflessa.

Il motore di questa trasformazione è la **proiezione prospettica**.

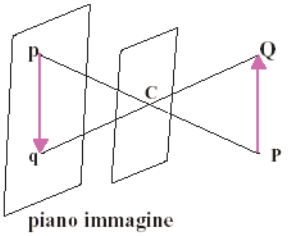

L'occhio umano



La pin-hole camera

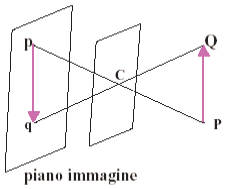


Proiezione prospettica: tutti i raggi di proiezione passano per un unico punto, detto **centro di proiezione**.

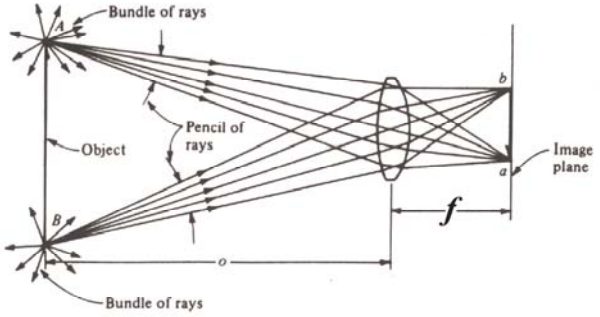


Pinhole camera

La lente

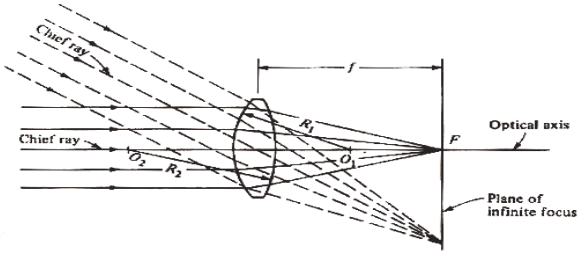


Pinhole camera



Lente convergente

Geometria dell'ottica

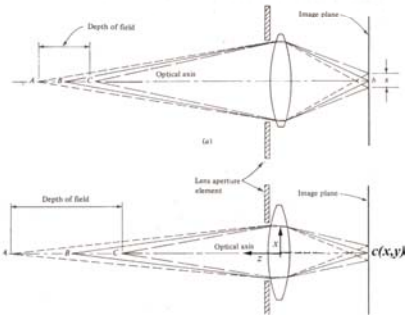


Oggetti all'infinito

•Distanza focale: distanza del piano immagine quando un oggetto si trova all'infinito.
•Asse ottico: raggio che non viene deviato dalla lente.

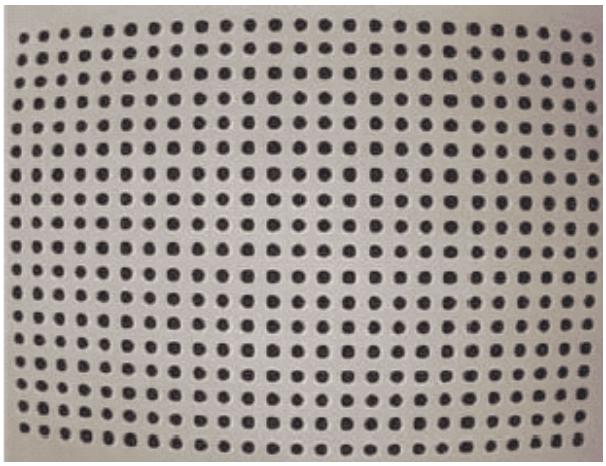
Messa a fuoco

Problema della messa a fuoco



Parametri di camera (o intrinseci):
•Punto principale c(x,y) + lunghezza focale, f (3 parametri).
•Occorre conoscere anche il fattore di forma dei pixel nel caso di immagini digitali (è una costante, non un parametro).
•(Distorsioni).

Distorsioni



Ottime per effetti speciali, un po' meno per delle misure…..

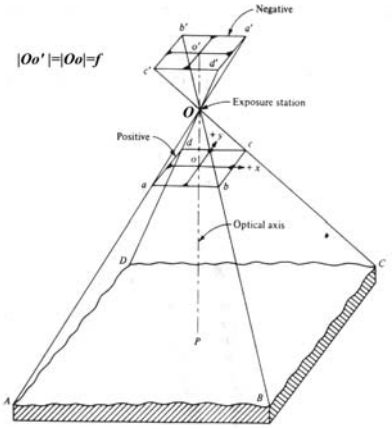
Trasformazioni

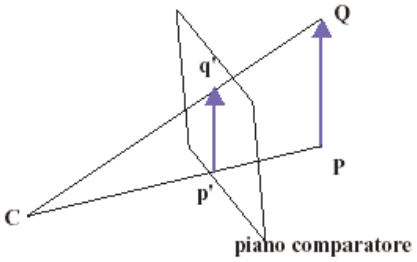
Trasformo **P** in **P'**

Cambia il sistema di riferimento

Cambia la posizione del punto

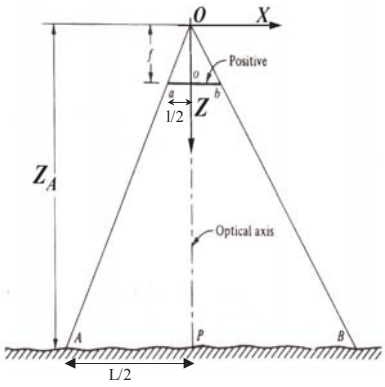
Raddrizzamento dell'immagine

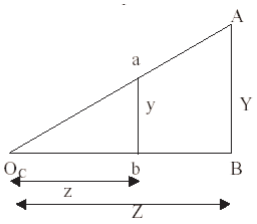




Si considera un piano posto davanti al centro di proiezione (si ottiene scandendo l'immagine dall'angolo dx in basso).

Raddrizzamento dell'immagine



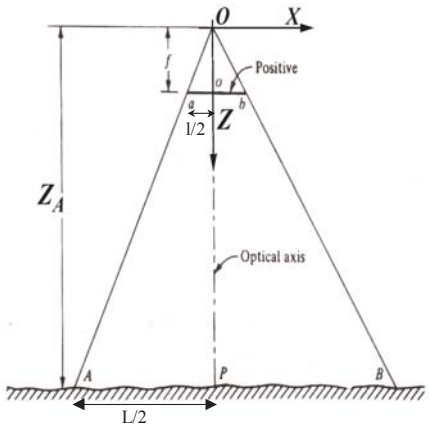


Per similitudine fra i triangoli aO_cb e AO_cB: $z : Z = y : Y$

da cui: $$y = z \cdot \frac{Y}{Z} = -f \cdot \frac{Y}{Z}$$

e analogamente: $$x = z \cdot \frac{X}{Z} = -f \cdot \frac{X}{Z}$$

Proiezione semplice

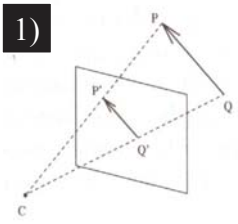


$x_a - x_o = X_A f / Z_A$

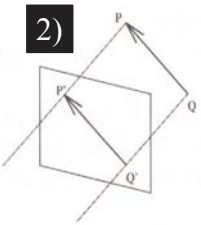
$a(x_a; y_a ; z_a) \rightarrow y_a - y_o = Y_A f / Z_A$

$z_a = f$

1)



2)



ABO e abO sono triangoli simili:

ab = AB f /Z_A

f : Z_A = x_a : X_A

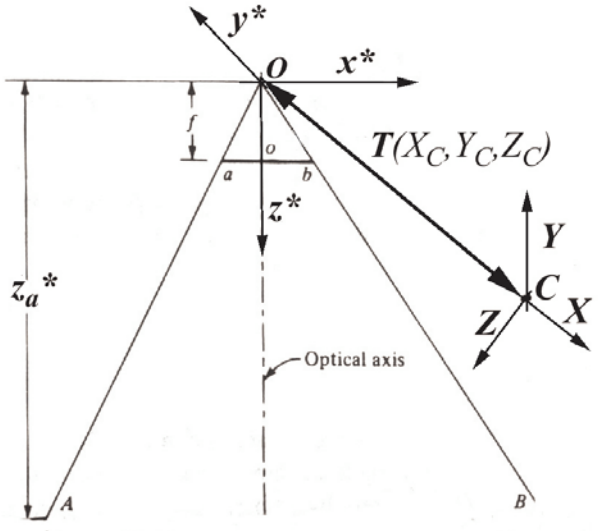
1) x_a dipende da X_a **e** Z_a, e da f e x_p.

2) x_a non dipende da Z_a, ma solo da X_a

I parametri esterni

- ***Traslazione:***
3 componenti:
T(X_C, Y_C, Z_C).

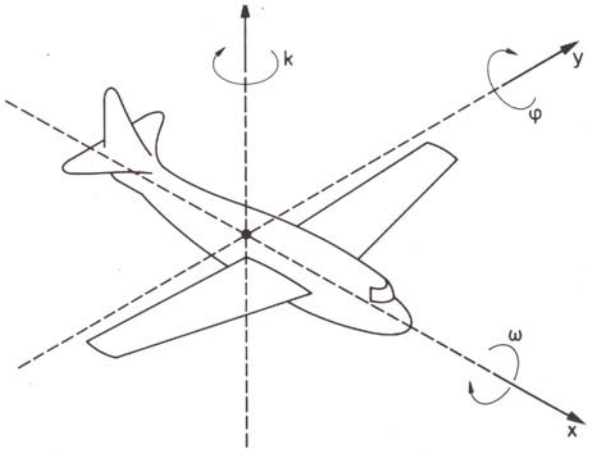
- ***Rotazione***
$R_{3x3}(\omega, \phi, k)$



Angoli di orientamento nello spazio 3D

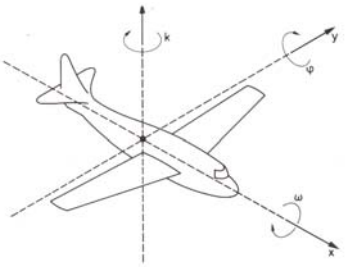
Modo generale: roll, pitch, e yaw.
(ω, ϕ, k): rollio, beccheggio e deriva.

Sono 3 rotazioni sequenziali, non commutative.



Rotazioni

Modo generale: roll, pitch, e yaw.
(ω, ϕ, k): rollio, beccheggio e deriva.

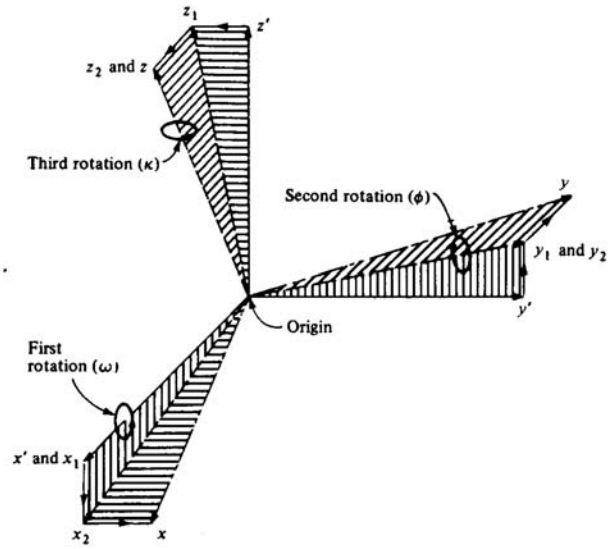


$$R_\omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix}$$

$$R_\phi = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

$$R_\kappa = \begin{bmatrix} \cos k & \sin k & 0 \\ -\sin k & \cos k & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotazioni sequenziali



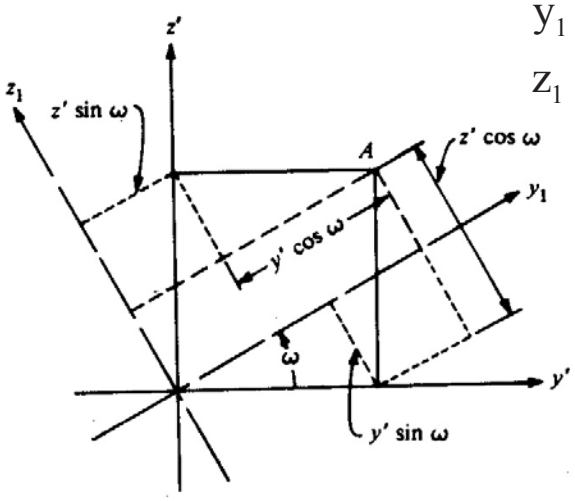
Ciascuna rotazione avviene su uno dei piani coordinati.

I) Rotazione attorno all'asse x (roll)

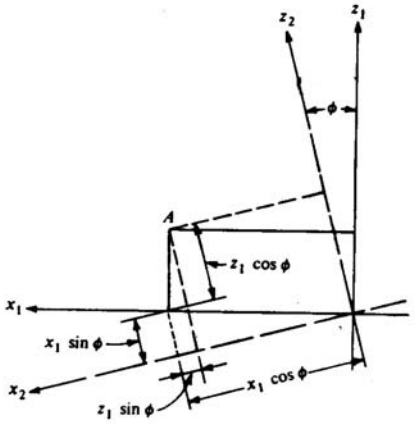
$$x_1 = x$$

$$y_1 = y'\cos\omega + z'\sin\omega$$

$$z_1 = -y'\sin\omega + z'\cos\omega$$



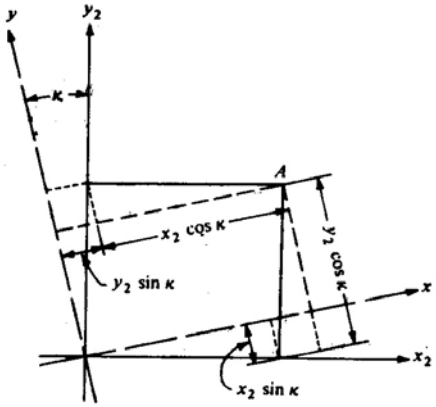
II) Rotazione attorno all'asse y (pitch)



$$x_2 = x_1 \cos\phi + z_1 \sin\phi$$
$$y_2 = y_1$$
$$z_2 = -x_1 \sin\phi + z_1 \cos\phi$$

$$x_2 = x'\cos\phi + (-y'\sin\omega + z'\cos\omega)\sin\phi$$
$$y_2 = y'\cos\omega + z'\sin\omega$$
$$z_2 = -x'\sin\phi + (-y'\sin\omega + z'\cos\omega)\cos\phi$$

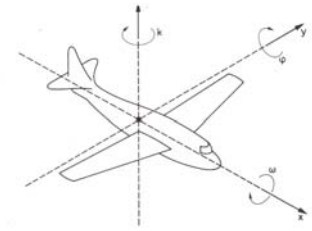
III) Rotazione attorno all'asse z (yaw)



$$x_3 = x_2 \cos k + y_2 \sin k$$
$$y_3 = -x_2 \sin k + z_2 \cos k$$
$$z_3 = z_2$$

$$x_3 = [x'\cos\phi + (-y'\sin\omega + z'\cos\omega)\sin\phi]\cos k + [y'\cos\omega + z'\sin\omega]\sin k$$
$$y_3 = -[x'\cos\phi + (-y'\sin\omega + z'\cos\omega)\sin\phi]\sin k + [y'\cos\omega + z'\sin\omega]\cos k$$
$$z_3 = -x'\sin\phi + (-y'\sin\omega + z'\cos\omega)\cos\phi$$

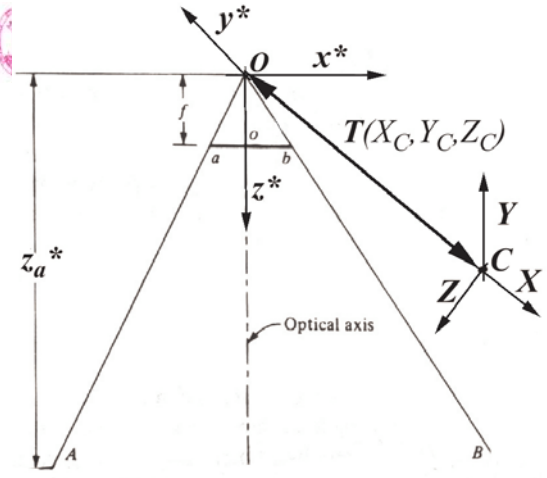
Dalle rotazioni alla matrice di rotazione



Come è legata R alle tre rotazioni indipendenti?

$$\mathbf{R} = \begin{bmatrix} \cos(\varphi)\cos(k) & -\cos(\varphi)\sin(k) & -\sin(k) \\ \cos(w)\sin(k)-\sin(w)\sin(\varphi)\cos(k) & \cos(w)\cos(k)+\sin(w)\sin(\varphi)\sin(k) & -\sin(w)\cos(\varphi) \\ \sin(w)\sin(k)+\cos(w)\sin(\varphi)\cos(k) & \sin(w)\cos(k)-\cos(w)\sin(\varphi)\sin(k) & \cos(w)\cos(\varphi) \end{bmatrix}$$

Si ricava eseguendo le rotazioni sequenziali. Ogni rotazione tiene fermo un asse e agisce sul piano perpendicolare.
Rotazioni "*semplici*" utilizzate dai programmi di animazione, gestione matriciale *efficiente* del calcolo.



La Rotazione 3D

$x^*_A = X'_A \cos(xx^*) + Y'_A \cos(yx^*) + Z'_A \cos(zx^*)$
$y^*_A = X'_A \cos(xy^*) + Y'_A \cos(yy^*) + Z'_A \cos(xy^*)$
$z^*_A = X'_A \cos(xz^*) + Y'_A \cos(yz^*) + Z'_A \cos(zz^*)$

In forma compatta: $\mathbf{p^* = R\,P'}$


I quaternioni

- la rotazione di un vettore ***r*** di un angolo si può esprimere con un operatore chiamato quaternione, caratterizzato da 4 numeri reali
- abbiamo 4 gradi di libertà invece dei 9 elementi della matrice
- useremo quaternioni *unitari*
- i quaternioni possono essere considerati come una generalizzazione dei numeri complessi, con uno scalare *s* come parte reale e un vettore ***v*** come parte immaginaria


- denotiamo un quaternione con:

$$q = s + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

dove ***i,j,k*** sono i quaternioni unitari ed equivalgono ai vettori unitari degli assi in un sistema vettoriale e hanno le proprietà:

$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$; $\mathbf{ij} = \mathbf{k}$; $\mathbf{ji} = -\mathbf{k}$

- da queste proprietà ricaviamo le operazioni somma e moltiplicazione

Operazioni sui quaternioni

- somma:

$q+q'=(s+s', \mathbf{v}+\mathbf{v}')$

- moltiplicazione:

$qq'=(ss'-\mathbf{v}\cdot\mathbf{v}', \mathbf{v}\times\mathbf{v}' + s\mathbf{v}' + s'\mathbf{v})$

- coniugato:

$q=(s,\mathbf{v})\quad q^*=(s,-\mathbf{v})$

- il prodotto di un quaternione con il suo coniugato dà il modulo del quaternione:

$qq^*=(ss-|\mathbf{v}^2|)=q^2$

- quaternioni della forma: $q=(s,(0,0,0))$ sono associati ai numeri reali

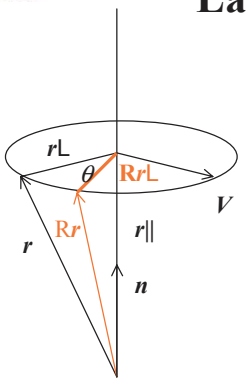
- quaternioni della forma: $q=(s,(a,0,0))$ sono associati ai numeri complessi

- negazione:

dato $q=(s,\mathbf{v})$ si ha $-q=(-s,-\mathbf{v})$

- Se $|q|=1$ il quaternione è detto *unitario*
- L'insieme dei quaternioni unitari forma una sfera in uno spazio a 4 dimensioni
- Si può dimostrare che se $q=(s,\mathbf{v})$ allora esiste un vettore $\mathbf{v'}$ e un numero $-\pi<=\theta<=\pi$ tale che: $q=(\cos\theta, \mathbf{v'}\sin\theta)$
- Se q è unitario allora $q=(\cos\theta, \sin\theta\,\mathbf{n})$ con $\mathbf{n}$ unitario.
- i quaternioni non sono commutativi rispetto al prodotto

La rotazione con quaternioni



- $\mathbf{r}$ è definito dal quaternione $p=(0,\mathbf{r})$
- definiamo l'operatore $R_q=q(.)q^{-1}$ con q quaternione unitario $(s,\mathbf{v})$
- applicato a p l'operatore dà: qpq^{-1}
- in forma esplicita:
- $R_q(p)=(0,(s^2-\mathbf{v}.\mathbf{v})\mathbf{r}+2\mathbf{v}(\mathbf{v}.\mathbf{r})+2s(\mathbf{v}\times\mathbf{r})$
- ricordando che: se q è unitario allora $q=(\cos\theta, \sin\theta\,\mathbf{n})$ con $\mathbf{n}$ unitario e sostituendo si ha:

$R_q(p)=(0,(\cos^2\theta-\sin^2\theta)\mathbf{r}+2\sin^2\theta\,\mathbf{n}(\mathbf{n}.\mathbf{r})+2\cos\theta\sin\theta(\mathbf{n}\times\mathbf{r}))=$

$(0, \mathbf{r}\cos2\theta+(1-\cos2\theta\,\mathbf{n}(\mathbf{n}.\mathbf{r})+\sin2\theta(\mathbf{n}\times\mathbf{r}))$


- confrontiamo la:

*(0, **r**cos2θ +(1- cos2θ **n**(**n.r**)+sin2θ(**n**×**r**))*

- con l'equazione ricavata prima:

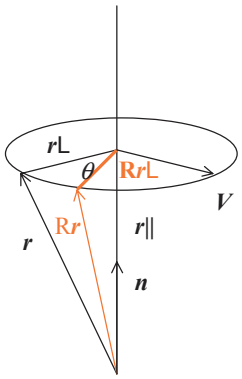
$$(\cos\theta)\mathbf{r} + (1-\cos\theta)\mathbf{n}(\mathbf{n}.\mathbf{r}) + (\sin\theta)\mathbf{n}\times\mathbf{r}$$

- a meno del coefficiente 2 sono identiche
- la rotazione di un vettore ***r*** di *(θ,**n**)* si può quindi attuare:
  - passando allo spazio dei quaternioni
  - rappresentando la rotazione con un quaternione unitario *q=(cos θ/2, sin θ/2 **n**)*
  - applicando l'operatore $q(.)q^{-1}$ al quaternione *(0,**r**)*
- la rotazione si parametrizza quindi con i 4 parametri: *cos θ/2, sin θ/2* $\mathbf{n}_x$, *sin θ/2* $\mathbf{n}_y$, *sin θ/2* $\mathbf{n}_z$

continua ...

un po' di link

- http://www.3dgamedev.com/articles/eulers_are_evil.htm
- http://www.gamedev.net/reference/articles/article1095.asp
- keyword per ricerca in rete: quaternion, euler angle



il vettore ***r*** può essere scomposto in una componente parallela a ***n*** e in una ortogonale:
r*||=(*n*.*r*) ×*n
r*⊥=*r* - (*n*.*r*) ×*n
la componente || resta invariata nella rotazione, varia solo la componente ⊥ (rossa). ***V*** sia ortogonale a ***r*⊥**:
V*=*n*× *r*⊥ = *n*×*r da cui il vettore ruotato (rosso) espresso in funzione di ***V***:

$$R\mathbf{r}\perp = (\cos\theta)\mathbf{r}\perp + (\sin\theta)\mathbf{V}$$

da cui:

$$R\mathbf{r} = R\mathbf{r}\| + R\mathbf{r}\perp$$
$$= R\mathbf{r}\| + (\cos\theta)\mathbf{r}\perp + (\sin\theta)\mathbf{V}$$
$$= \mathbf{n}.\mathbf{r}\times\mathbf{n} + (\cos\theta)(\mathbf{r} - \mathbf{n}.\mathbf{r}\times\mathbf{n}) + (\sin\theta)\mathbf{n}\times\mathbf{r}$$
$$= (\cos\theta)\mathbf{r} + (1-\cos\theta)\mathbf{n}(\mathbf{n}.\mathbf{r}) + (\sin\theta)\mathbf{n}\times\mathbf{r}$$

ancora un esempio

- ruotiamo un oggetto di 180° attorno all'asse *x* con la sequenza di rotazioni R(0,0,0), ... R(πt,0,0), ..., R(π,0,0) con 0<=t<=1
- la seconda sequenza ruota attorno *y, z* : R(0,0,0), ... R(0, πt, πt), ..., R(0, π, π)
- la posizione finale e' identica, ma l'oggetto "twista" nella seconda
- occorre controllare i 3 angoli di Eulero per governare la sequenza desiderata
- da qui l'uso dei quaternioni

con i quaternioni

- la rotazione ottenuta con la sequenza R(0,0,0), ... R(πt,0,0), ..., R(π,0,0) è rappresentata dal quaternione $(cos(\pi/2), sin(\pi/2)(1,0,0)) = (0,(1,0,0))$
- la rotazione ottenuta con la sequenza R(0,0,0), ... R(0, πt, πt), ..., R(0, π, π) è rappresentata dal prodotto dei due quaternioni $(0,(0,1,0))(0,(0,0,1) = (0,(1,0,0)$
- **Il risultato è uguale**

La Rototraslazione 3D

Traslazione: 3 componenti: $\mathbf{T}(X_C, Y_C, Z_C)$.
Rotazione: 3 componenti: R(ω, ϕ, k).

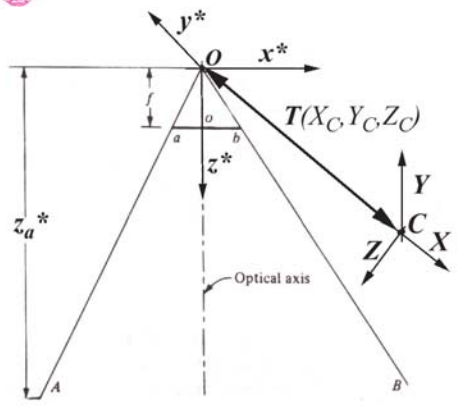
Matrice di rotazione con : ω, ϕ, k, è:

$$R\begin{bmatrix} \cos(\varphi)\cos(k) & -\cos(\varphi)sin(k) & -sin(k) \\ \cos(w)sin(k) - sin(w)sin(\varphi)\cos(k) & \cos(w)\cos(k) + sin(w)sin(\varphi)sin(k) & -sin(w)\cos(\varphi) \\ sin(w)sin(k) + \cos(w)sin(\varphi)\cos(k) & sin(w)\cos(k) - \cos(w)sin(\varphi)sin(k) & \cos(w)\cos(\varphi) \end{bmatrix}$$

In forma compatta: **p* = R (P' - T**)

I parametri esterni sono perciò 6.

Dal 3D al 2D



$x_a - x_p = x^*_A \, f / z^*_A$

$a(x_a; y_a ; z_a) \rightarrow y_a - y_p = y^*_A \, f / z^*_A$

$z_a = f$

$\mathbf{p_A^* = R\,(P_A - T)}$

$A(X_A,Y_A,Z_A) \Rightarrow A(x^*_a, y^*_a, z^*_a) \Rightarrow a(x_a,y_a,f).$

Equazioni di collinearità

$$x^*_A = r_{11}(X_A-X_C) + r_{12}(Y_A-Y_C) + r_{13}(Z_A-Z_C)$$
$$y^*_A = r_{21}(X_A-X_C) + r_{22}(Y_A-Y_C) + r_{23}(Z_A-Z_C)$$
$$z^*_A = r_{31}(X_A-X_C) + r_{32}(Y_A-Y_C) + r_{33}(Z_A-Z_C)$$

$$x_a - x_p = x^*_A \, f / z^*_A = f \, \frac{r_{11}(X_A-X_C) + r_{12}(Y_A-Y_C) + r_{13}(Z_A-Z_C)}{r_{31}(X_A-X_C) + r_{32}(Y_A-Y_C) + r_{33}(Z_A-Z_C)}$$

$$y_a - y_p = y^*_A \, f / z^*_A = f \, \frac{r_{21}(X_A-X_C) + r_{22}(Y_A-Y_C) + r_{23}(Z_A-Z_C)}{r_{31}(X_A-X_C) + r_{32}(Y_A-Y_C) + r_{33}(Z_A-Z_C)}$$

Complessivamente 9 parametri.

Correzione delle distorsioni

Shrinkage – fattore di forma. Diversa scala sugli assi. E' un dato di fabbrica del dispositivo.

Distorsioni ottiche (principalmente a forma radiale).

$$x = x_d(1 + k_1 r^2 + k_2 r^4)$$
$$y = y_d(1 + k_1 r^2 + k_2 r^4)$$

Equazioni di collinearità

Proiezione prospettica completa

Parametri intrinseci → focale, centro ottico

Parametri estrinseci → traslazione, rotazione

Parametri di distorsione

La calibrazione

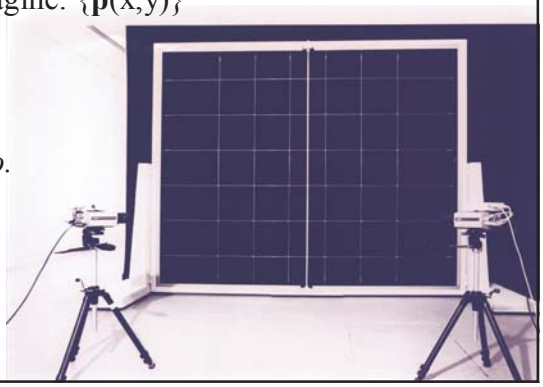
Determinazione dei parametri della trasformazione prospettica:

- Parametri interni: x_o, y_o, f.
- Parametri esterni: X_C, Y_C, Z_C, $m_{ij}(\omega,\phi, k)$.

A partire dalle coppie:

- Punti 3D: {**P**(X,Y,Z)}
- Proiezioni sul piano immagine: {**p**(x,y)}

I punti devono essere disposti all'interno del volume di lavoro.



Calibrazione con punti di coordinate note (I)

Per ogni punto, $[p_k(x_k,y_k)\ P_k(X_k,Y_k,Z_k)]$*:*

$$x_k = x_o + f\ \frac{m_{11}(X_k - X_C)+m_{12}(Y_k - Y_C)+m_{13}(Z_k - Z_C)}{m_{31}(X_k - X_C)+m_{32}(Y_k - Y_C)+m_{33}(Z_k - Z_C)}$$

$$y_k = y_o + f\ \frac{m_{21}(X_k - X_C)+m_{22}(Y_k - Y_C)+m_{23}(Z_k - Z_C)}{m_{31}(X_k - X_C)+m_{32}(Y_k - Y_C)+m_{33}(Z_k - Z_C)}$$

• *Riscrivo le equazioni di collinearità:*

$x_k = F(X_k,Y_k,Z_k \mid X_C, Y_C, Z_C, m_{ij}(\omega,\phi, k), f, x_o)$

$y_k = G(X_k,Y_k,Z_k \mid X_C, Y_C, Z_C, m_{ij}(\omega,\phi, k), f, y_o)$

Definisco residuo la quantità:

$\Delta x_k = F(X_k,Y_k,Z_k \mid X_C, Y_C, Z_C, m_{ij}(\omega,\phi, k), f, x_o) - x_k$

$\Delta y_k = G(X_k,Y_k,Z_k \mid X_C, Y_C, Z_C, m_{ij}(\omega,\phi, k), f, y_o) - y_k$

Chiamo: $S(X_C, Y_C, Z_C, m_{ij}(\omega,\phi, k), f, x_o, y_o)$ *l'insieme dei parametri:*

Minimizzo: $$\min_S \sum_{k=1}^{M} [(x_k - F(X_k, Y_k, Z_k \mid ...)^2 + (y_k - G(X_k, Y_k, Z_k \mid ...)^2] = \min_S \sum_{k=1}^{M} [\Delta x_k^{\,2} + \Delta y_k^{\,2}]$$

Calibrazione con punti di coordinate note (II)

Stima iterativa (gradiente discendente):
Sviluppo in serie di Taylor + Stima ai minimi quadrati.

1) Sviluppo in serie di Taylor.
Partiamo da un valore iniziale dei parametri:
$S^I(X^I_C, Y^I_C, Z^I_C, m_{ij}(\omega^I, \phi^I, k^I), f^I, x_o^I, y_o^I)$.

Per ogni punto p_k: $F(X_k,Y_k,Z_k \mid S^I) = x_k+\Delta x_k$; $G(X_k,Y_k,Z_k \mid S^I) = y_k+\Delta y_k$

Δx_k; Δy_k *indicano che quando proietto il punto* $P(X_k,Y_k,Z_k)$ *di coordinate note non ottengo* $p(x_k,y_k)$ *ma un punto spostato di* Δx_k, Δy_k *se i parametri non sono ancora quello corretti.*
Quando i parametri sono stati determinati correttamente otterrò:
$dX_C = dY_C = dZ_C = d\omega_C = d\phi_C = d\kappa_C = df = = dx_o = dy_o = 0.$
$\Delta x_k = \Delta y_k = 0.$ $\forall k$

Calibrazione con punti di coordinate note (III)

Caso Reale. *Non misuriamo esattamente x, y, ma* $x+\nu_x$; $y+\nu_y$:

$$x + \nu_x = x_o + f \frac{m_{11}(X-X_C)+m_{12}(Y-Y_C)+m_{13}(Z-Z_C)}{m_{31}(X-X_C)+m_{32}(Y-Y_C)+m_{33}(Z-Z_C)}$$

$$y + \nu_y = y_o + f \frac{m_{21}(X-X_C)+m_{22}(Y-Y_C)+m_{23}(Z-Z_C)}{m_{31}(X-X_C)+m_{32}(Y-Y_C)+m_{33}(Z-Z_C)}$$

E quindi lo sviluppo in serie di Taylor sarà:
$\Delta x_k = F(X_k,Y_k,Z_k \mid X_C, Y_C, Z_C, m_{ij}(\omega,\phi, k), f, x_o) - x_k + \nu_{xk}$
$\Delta y_k = G(X_k,Y_k,Z_k \mid X_C, Y_C, Z_C, m_{ij}(\omega,\phi, k), f, y_o) - y_k + \nu_{yk}$

Quando i parametri sono stati determinati correttamente otterrò:
$dX_C = dY_C = dZ_C = d\omega_C = d\phi_C = d\kappa_C = df = = dx_o = dy_o = 0.$
$\Delta x_k = \nu_{xk}$ e $\Delta y_k = \nu_{yk}$ $\forall k$

Stima ai minimi quadrati.

$$\min_S \sum_{k=1}^{M} [\Delta x_k^2 + \Delta y_k^2]$$

Calibrazione con punti di coordinate note (IV)

Al passo j della stima risulta per ogni punto, p_k:

$$\nu_{xk} + F(X_k,Y_k,Z_k|S_j) = a_{1k}dX_c + a_{2k}dY_c + a_{3k}dZ_c + a_{4k}d\omega_c + a_{5k}d\phi_c + a_{6k}d\kappa_c + a_{7k}df + dx_o$$

$$\nu_{yk} + G(X_k,Y_k,Z_k|S_j) = b_{1k}dX_c + b_{2k}dY_c + b_{3k}dZ_c + b_{4k}d\omega_c + b_{5k}d\phi_c + b_{6k}d\kappa_c + b_{7k}df + dy_o$$

In forma matriciale: $\boldsymbol{R + B = AX}$

Dimensioni:

B = [2P x 1] R = [2P x 1]

A = [2P x 9]

X = [9 x 1]

Soluzione mediante stima ai minimi quadrati:

$X = (A^TA)^{-1}A^TB$ (svd)

Soluzione iterativa: $S^{j+1} = S^j + dS$ fino a che $dS \approx 0$

Bundle adjustment

Regolazione del fascio (rette retro-proiettate) via minimizzazione dell'errore di retro-proiezione.

Approccio più generale nel quale si considerano anche punti in posizione non nota.

Ciascuna proiezione genera 2 equazioni, 2 video-camere: 4 equazioni, M camere, 2*M equazioni.
Ogni punto 3D di coordinate non note sono 3 incognite aggiuntive.

Perché il sistema sia risolubile occorre avere almeno N punti 3D:

2*M*N = 3*N+9 ➔ N = 9 / (2M-3), almeno 9 punti con 2 camere.

NB Si tralasciano le distorsioni.

Le matrici affini

$$\mathbf{P' = RP + T \Rightarrow P' = AP}$$

$$\mathbf{R^T RP = R^T P' - R^T T \Rightarrow P = A^{-1} P'}$$

Proiezione di **T** sugli assi di arrivo: $\mathbf{r_i \bullet T}$

$$\begin{bmatrix} X_P \\ Y_P \\ Z_P \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & r_{11}T_x + r_{21}T_y + r_{31}T_z \\ r_{12} & r_{22} & r_{32} & r_{12}T_x + r_{22}T_y + r_{32}T_z \\ r_{13} & r_{23} & r_{33} & r_{13}T_x + r_{23}T_y + r_{33}T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X'_P \\ Y'_P \\ Z'_P \\ 1 \end{bmatrix}$$

Matrice di rotazione (inversa)

Vettore di traslazione (inverso)

Formulazione matriciale della proiezione

$$\mathbf{K} = \begin{bmatrix} -f & 0 & x_c \\ 0 & -f & y_c \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} \mathbf{R} & \mathbf{-RT} \\ \mathbf{0} & 1 \end{bmatrix}$$

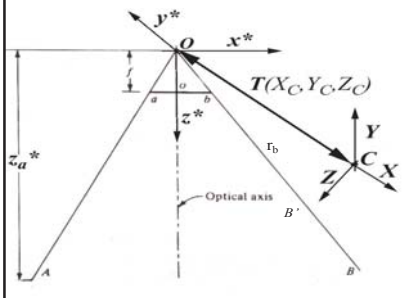
$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{p} \begin{bmatrix} x_a / f \\ y_a / f \end{bmatrix} \Rightarrow \begin{bmatrix} x_a \\ y_a \\ f \end{bmatrix} = \mathbf{KHDP} \begin{bmatrix} X_A \\ Y_A \\ Z_A \\ 1 \end{bmatrix}$$

Fattorizzazione dei parametri interni ed esterni.
Coordinate omogenee per la rappresentazione dei punti.

Dal 2D al 3D

ray - intersection

Dal 2D al 3D



Nello spazio 3D esistono ∞^1 punti compatibili con p_b

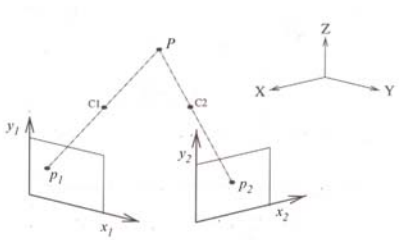
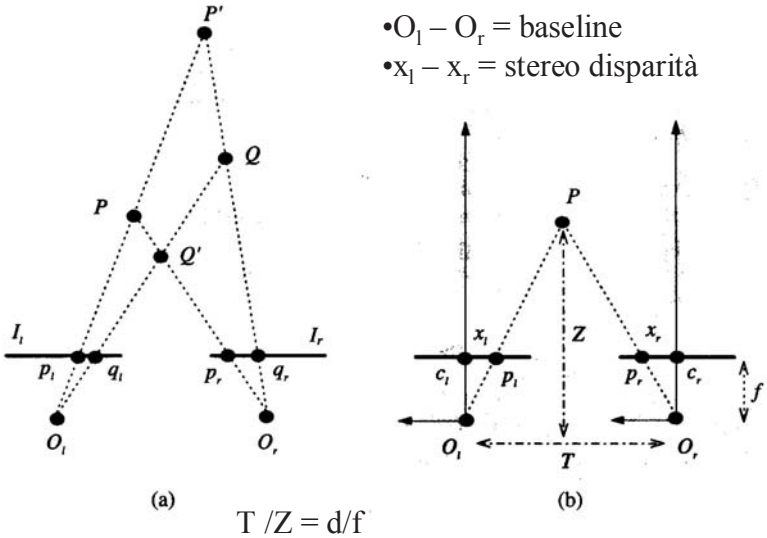


Figura 2.6 Il problema della ricostruzione 3D.

Soluzione: Stereoscopia

3D basato su stereo-disparità

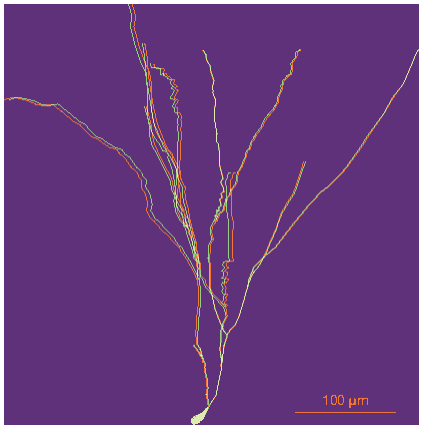


- $O_l - O_r$ = baseline
- $x_l - x_r$ = stereo disparità

$T / Z = d / f$

Il neurone (stereo)

Neurone piramidale CA1
Ippocampo



Neurone granulare
Ippocampo

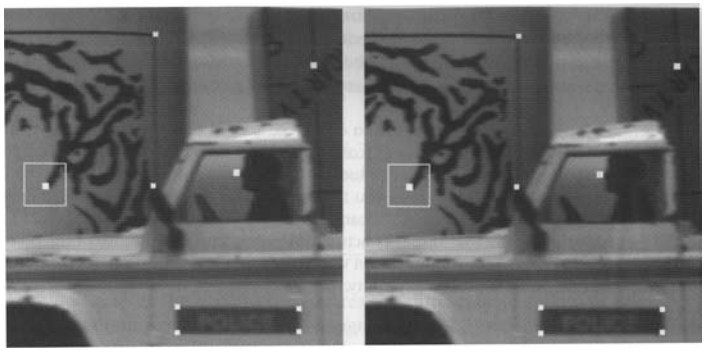
Problemi

- Corrispondenza.
- Ricostruzione posizione 3D.

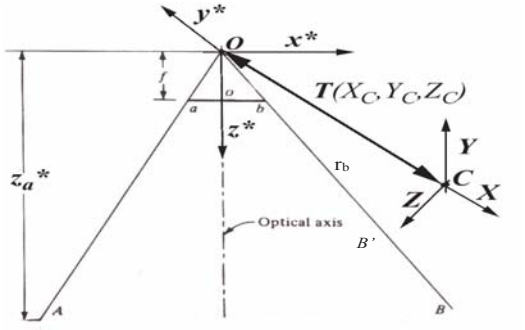
Low-level processing

- Classificazione.

High-level processing



Dal 2D al 3D



Invertiamo le
Equazioni di collinearità
P(X, Y, Z) => p(x y, f)
---->
p(x, y, f) => P(X, Y, Z)

$$x_a = x_o + f * \frac{m_{11}(X - X_C) + m_{12}(Y - Y_C) + m_{13}(Z - Z_C)}{m_{31}(X - X_C) + m_{32}(Y - Y_C) + m_{33}(Z - Z_C)}$$

$$y_a = y_o + f * \frac{m_{21}(X - X_C) + m_{22}(Y - Y_C) + m_{23}(Z - Z_C)}{m_{31}(X - X_C) + m_{32}(Y - Y_C) + m_{33}(Z - Z_C)}$$

$$z_a = f$$

Dal 2D al 3D

- Equazioni inverse p(x, y, f) => P(X, Y, Z)

$$[m_{31}(X-X_C)+m_{32}(Y-Y_C)+m_{33}(Z-Z_C)]*(x_a-x_o) = f[m_{11}(X-X_C)+m_{12}(Y-Y_C)+m_{13}(Z-Z_C)]$$
$$[m_{31}(X-X_C)+m_{32}(Y-Y_C)+m_{33}(Z-Z_C)]*(y_a-y_o) = f[m_{21}(X-X_C)+m_{22}(Y-Y_C)+m_{23}(Z-Z_C)]$$

Sono equazioni lineari in X, Y, Z:

- $[m_{31}(x_a-x_o) - f m_{11}]\,\mathbf{X} + [m_{32}(x_a-x_o) - f m_{12}]\,\mathbf{Y} + [m_{33}(x_a-x_o) - f m_{13}]\,\mathbf{Z} =$
  $[m_{31}(x_a-x_o) - f m_{11}]\,X_C + [m_{32}(x_a-x_o) - f m_{12}]\,Y_C + [m_{33}(x_a-x_o) - f m_{13}]\,Z_C$
- $[m_{31}(y_a-y_o) - f m_{21}]\,\mathbf{X} + [m_{32}(y_a-y_o) - f m_{22}]\,\mathbf{Y} + [m_{33}(y_a-y_o) - f m_{23}]\,\mathbf{Z} =$
  $[m_{31}(y_a-y_o) - f m_{21}]\,X_C + [m_{32}(y_a-y_o) - f m_{22}]\,Y_C + [m_{33}(y_a-y_o) - f m_{23}]\,Z_C$

Identificano due piani:

- $a_1\,\mathbf{X} + b_1\,\mathbf{Y} + c_1\mathbf{Z} = d_1$ // asse y
- $a_2\,\mathbf{X} + b_2\,\mathbf{Y} + c_2\,\mathbf{Z} = d_2$ // asse x

Interesezione di due piani individua una retta nello spazio 3D (raggio della video-camera, camera ray). Retta per il punto sul piano immagine ed il centro di prospettiva.

Ricostruzione stereoscopica

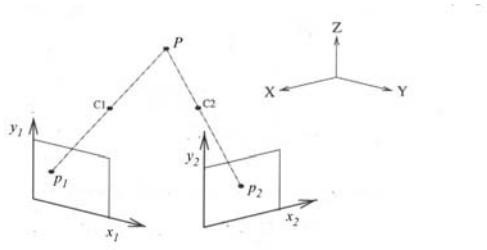


Figura 2.6 Il problema della ricostruzione 3D.

$\{p_1(x,y) <==> p_2(x,y)\}$

$[p_1(xy) \mid p_2(xy)] => P(XYZ).$

Equazioni di collinearità per due video-camere:

- $a_{1TV1}\,\mathbf{X} + b_{1TV1}\,\mathbf{Y} + c_{1TV1}\,\mathbf{Z} = d_{1TV1}$
- $a_{2TV1}\,\mathbf{X} + b_{2TV1}\,\mathbf{Y} + c_{2TV1}\,\mathbf{Z} = d_{2TV1}$

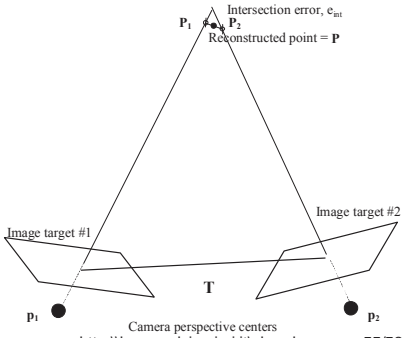
- $a_{1TV2}\,\mathbf{X} + b_{1TV2}\,\mathbf{Y} + c_{1TV2}\,\mathbf{Z} = d_{1TV2}$
- $a_{2TV2}\,\mathbf{X} + b_{2TV2}\,\mathbf{Y} + c_{2TV2}\,\mathbf{Z} = d_{2TV2}$

Ricostruzione - matrici

$$A = \begin{bmatrix} a_{1TV1} & b_{1TV1} & c_{1TV1} \\ a_{2TV1} & a1_{TV1} & a1_{TV1} \\ a_{1TV2} & b_{1TV2} & c_{1TV2} \\ a_{2TV2} & b_{2TV2} & c_{2TV2} \end{bmatrix} \quad B = \begin{bmatrix} d_{1TV1} \\ d_{2TV1} \\ d_{1TV2} \\ d_{2TV2} \end{bmatrix} \quad P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

In forma matriciale: A P = B => $P = (A^T A)^{-1} A^T B$

- Effetto dell'errore di misura.
- Quando non ammette soluzioni?



Ricostruzione - Ray intersection

- *Altra soluzione:* equazioni in forma parametriche delle due rette:

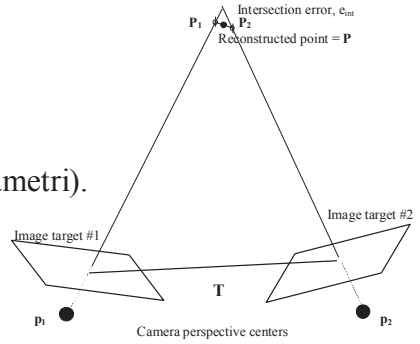
$X_1 = Xc_1 + a_1 k$ $\quad X_2 = Xc_2 + a_2 h$

$Y_1 = Yc_1 + b_1 k$ $\quad Y_2 = Yc_2 + b_2 h$

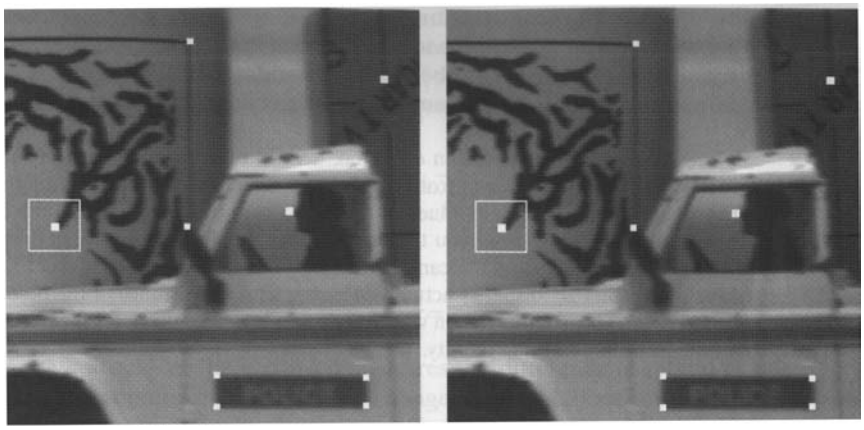
$Z_1 = Zc_1 + c_1 k$ $\quad Z_2 = Zc_2 + c_2 h$

$\min_{h,k}[(X_1-X_2)^2+(Y_1-Y_2)^2+(Z_1-Z_2)^2]$

- Calcolo della distanza (funzione dei due parametri, h e k);
- Minimizzazione (funzione dei due parametri).



Vincolo di intersezione



C.N. per la corrispondenza di punti

Riassunto

•La trasformazione prospettica (3D -> 2D) è funzione di nove parametri.

• Lunghezza e punto principale definiscono internamente la camera (parametri interni).

• Rotazione e traslazione definiscono esternamente (rispetto ad un sistema di riferimento preferenziale) la camera (parametri esterni).

•Ricostruzione 3D a partire dall'immagine dello stesso punto ripresa da due punti di vista.

•Calibrazione di una video-camera.