

# Trasformazioni Geometriche 2D e 3D

*Giulio Agostini*

31 Luglio, 1997

## **Sommario**

*Trasformazioni geometriche del piano e dello spazio. Trasformazioni lineari elementari: traslazione, scalamento, rotazione, riflessione, deformazione di taglio. Rappresentazioni matriciali classiche e nello spazio omogeneo. Proprietà delle trasformazioni lineari. Composizione di trasformazioni lineari. Rotazioni 3D. Trasformazioni non lineari. Trasformazioni polinomiali. Applicazione delle trasformazioni polinomiali al problema della compensazione della distorsione di ottiche non ideali. Immagini digitali e relative elaborazioni al calcolatore. Ricampionamento e relativi metodi di interpolazione. Metodi convoluzionali per l'ingrandimento di fattori interi. Cenni sull'ottimizzazione della complessità degli algoritmi per le trasformazioni geometriche sul calcolatore: separabilità delle trasformazioni e approssimazione per piccoli angoli.*

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Trasformazioni lineari</b>	<b>3</b>
2.1	Trasformazioni 2D . . . . .	3
2.1.1	Una rappresentazione elementare . . . . .	3
2.1.2	Una rappresentazione unificata: coordinate omogenee e matrici di trasformazione . . . . .	6
2.1.3	Composizione di trasformazioni . . . . .	7
2.2	Trasformazioni 3D . . . . .	9
2.2.1	Rappresentazione matriciale . . . . .	9
<b>3</b>	<b>Trasformazioni non lineari del piano</b>	<b>12</b>
<b>4</b>	<b>Dall'analogico al digitale: elaborazione di immagini al calcolatore</b>	<b>14</b>
4.1	Ricampionamento . . . . .	15
4.1.1	Metodi di interpolazione . . . . .	17
4.1.2	Metodi convoluzionali . . . . .	18
4.2	Implementazione di algoritmi efficienti . . . . .	18
4.2.1	Separabilità delle trasformazioni elementari . . . . .	20
4.2.2	Approssimazione per piccoli angoli . . . . .	20

# 1 Introduzione

Una delle più comuni pratiche di elaborazione delle immagini è la *trasformazione geometrica* ([1], [2]), di cui traslazione, rotazione, e scalamento sono semplici esempi.

Il sempre crescente numero di applicazioni di *computer graphics* e realtà virtuale più o meno spinte (dai CAD, a VRML, all'ultimo videogioco Doom-like) richiedono dapprima una elaborazione dei dati tridimensionali (si pensi per esempio una rotazione 3D per un semplice cambio di punto di vista), e poi una proiezione da 3D a 2D per visualizzare l'immagine sullo schermo. Chiaramente, soprattutto per le applicazioni in tempo reale, esistono vincoli di tempo molto stringenti, che si traducono nella necessità di algoritmi di trasformazione efficienti.

Un'altra applicazione di notevole importanza è la compensazione delle distorsioni introdotte dalle non-idealità dell'ottica del sistema di acquisizione, attraverso una post-elaborazione di tipo non lineare, come si vedrà meglio in seguito.

La trattazione si apre con le trasformazioni lineari (sezione 2). Nel paragrafo 2.1 vengono dapprima presentate le trasformazioni del piano "analogico", incominciando con la rappresentazione classica dei cambi di variabile per trasformazioni lineari. Si definiscono poi le coordinate omogenee, introducendo una rappresentazione matriciale più generale e compatta.

Si estenderanno questi risultati allo spazio 3D nel paragrafo 2.2, ponendo particolare attenzione al caso delle rotazioni 3D.

Nella sezione 3 si introdurranno brevemente le trasformazioni non lineari con riferimento al problema della compensazione delle distorsioni della telecamera.

La sezione 4 tratterà l'argomento dal punto di vista operativo: si introdurranno le immagini digitali e si spiegherà come concretamente vengano effettuate le trasformazioni geometriche sull'elaboratore. Seguirà un breve cenno sulle tecniche di ottimizzazione degli algoritmi.

## 2 Trasformazioni lineari

### 2.1 Trasformazioni 2D

#### 2.1.1 Una rappresentazione elementare

Si definisce *trasformazione lineare 2D* una generica funzione  $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$  lineare nelle sue variabili indipendenti, esprimibile cioè dalla seguente familiare forma generale

$$\begin{cases} x' = ax + by + e \\ y' = cx + dy + f \end{cases} \quad (1)$$

o dalla più elegante forma matriciale

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (2)$$

Studiamone qualche caso particolare notevole e le relative proprietà, riferendoci ad entrambe le forme (1) e (2).

**Traslazione.** Le seguenti equazioni effettuano una *traslazione* del piano di  $d_x$  e di  $d_y$  lungo i due assi

$$\begin{cases} x' = x + d_x \\ y' = y + d_y \end{cases} \quad (3)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix} \quad (4)$$

La traslazione è una *trasformazione rigida* (mantiene inalterate le distanze tra i punti) additiva.

Banalmente, la composizione di due traslazioni equivale ad una traslazione. In simboli, se rappresentiamo con  $\mathbf{p}$  il vettore  $[x \ y]^T$ , con  $\mathbf{p}'$  il vettore  $[x' \ y']^T$  e con  $\mathbf{d}_1$  e  $\mathbf{d}_2$  due generici vettori di traslazione, è:  $\mathbf{p}' = (\mathbf{p} + \mathbf{d}_1) + \mathbf{d}_2 = \mathbf{p} + \mathbf{d}$ , con  $\mathbf{d} = \mathbf{d}_1 + \mathbf{d}_2$ .

**Scalamento.** Le seguenti equazioni effettuano uno *scalamento* del piano di fattori  $s_x$  ed  $s_y$  lungo i due assi

$$\begin{cases} x' = s_x x & s_x > 0 \\ y' = s_y y & s_y > 0 \end{cases} \quad (5)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad \begin{matrix} s_x > 0 \\ s_y > 0 \end{matrix} \quad (6)$$

Lo scalamento è una trasformazione moltiplicativa. Va da sé che se  $s_x$  ( $s_y$ ) è maggiore di uno, ne risulterà un ingrandimento lungo l'asse  $x$  ( $y$ ), e viceversa. Inoltre si noti che lo scalamento è compiuto rispetto all'origine, e quindi i singoli oggetti non verranno solo rimpiccioliti od ingranditi, ma anche spostati rispetto all'origine. Vedremo in un esempio del paragrafo 2.1.3 come ovviare in parte a questo effetto collaterale.

Se  $s_x = s_y$  si parla di scalamento *uniforme*, od *omogeneo*. Se  $s_x \neq s_y$  lo scalamento si dice *differenziale*, o *non omogeneo*.

La composizione di due scalamenti equivale ad uno scalamento. In simboli, se rappresentiamo con  $\mathbf{S}_1$  e  $\mathbf{S}_2$  due generiche matrici di scalamento, è:  $\mathbf{p}' = \mathbf{S}_1(\mathbf{S}_2\mathbf{p}) = \mathbf{S}\mathbf{p}$ , con  $\mathbf{S} = \mathbf{S}_1\mathbf{S}_2$ .

**Rotazione.** Le seguenti equazioni effettuano una *rotazione* del piano di  $\vartheta$  gradi in senso antiorario

$$\begin{cases} x' = x \cos \vartheta - y \sin \vartheta \\ y' = y \sin \vartheta + x \cos \vartheta \end{cases} \quad (7)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (8)$$

La rotazione è una trasformazione rigida moltiplicativa. Come nel caso dello scalamento, la rotazione viene effettuata rispetto all'origine.

La composizione di due rotazioni di angoli  $\vartheta$  e  $\psi$  equivale alla rotazione di un angolo pari a  $\vartheta + \psi$ . In simboli, se si rappresenta con  $\mathbf{R}(\cdot)$  una generica matrice di rotazione funzione dell'angolo di rotazione, è:  $\mathbf{p}' = \mathbf{R}(\vartheta)(\mathbf{R}(\psi)\mathbf{p}) = \mathbf{R}(\vartheta + \psi)\mathbf{p}$ . Le matrici di rotazione sono caratterizzate dall'essere ortonormali ( $\mathbf{R}^{-1}(\cdot) = \mathbf{R}^T(\cdot)$ ) ed hanno determinante unitario ( $\det(\mathbf{R}(\cdot)) = 1$ ). La sola ortogonalità non è sufficiente a caratterizzare una rotazione; per escludere riflessioni speculari è infatti necessario anche che il determinante sia unitario.

**Riflessione.** Le seguenti equazioni (esprese per semplicità nella sola rappresentazione matriciale (2)) effettuano una *riflessione* del piano rispetto agli assi  $x$  ed  $y$  rispettivamente

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (10)$$

La riflessione è una trasformazione rigida moltiplicativa.

Evidentemente, una riflessione rispetto ad entrambi gli assi equivale ad una rotazione di 180 gradi, e la (10) può essere ottenuta componendo la (9) con una rotazione di 180 gradi.

**Deformazione di taglio (*shear*).** Le seguenti equazioni effettuano una *deformazione di taglio* (*shear*) del piano, rispettivamente lungo l'asse  $x$  e lungo l'asse  $y$  di fattori  $t_x$  e  $t_y$  (si veda la figura 1)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & t_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ t_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (12)$$

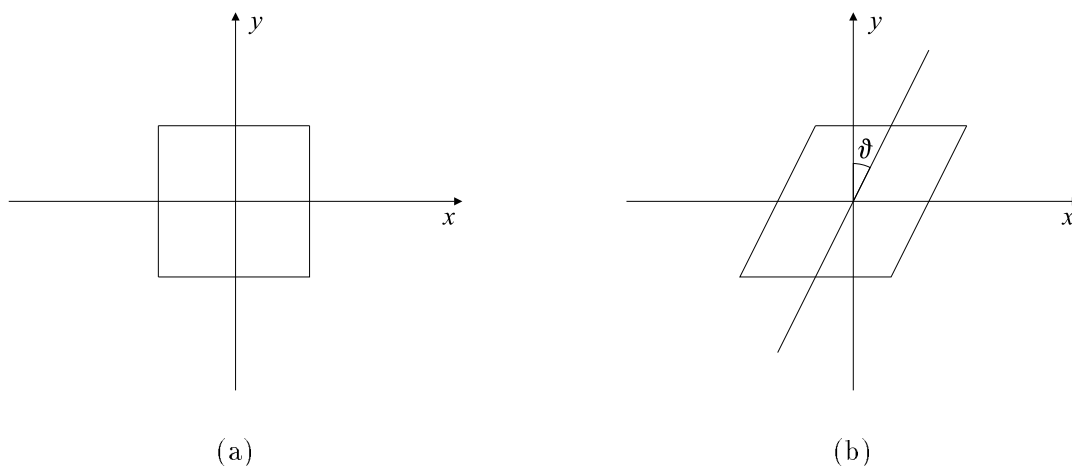


Figura 1: Esempio di deformazione di taglio. (a) Immagine originale; (b) Immagine deformata lungo l'asse  $x$ .

La deformazione di taglio è una trasformazione moltiplicativa. Esiste un interessante legame tra queste trasformazioni e le rotazioni, e di questo parleremo nel paragrafo 4.2.1.

La composizione di due deformazioni di taglio *lungo lo stesso asse* di coefficienti  $t_1$  e  $t_2$  equivale ad una deformazione di taglio di coefficiente  $t = t_1 + t_2$ .

Il lettore è invitato a verificare, come utile esercizio, che il parametro  $t_x$  ( $t_y$ ) è legato all'angolo  $\vartheta$  compreso tra l'asse  $y$  ( $x$ ) originario e l'asse  $y$  ( $x$ ) trasformato (si vedano le figure 1 e 2) attraverso la relazione:  $t_x = 1/\tan \vartheta$ .

### 2.1.2 Una rappresentazione unificata: coordinate omogenee e matrici di trasformazione

La notazione introdotta nel paragrafo precedente prevede trasformazioni elementari additive e moltiplicative. Per poter disporre di una notazione omogenea, interamente basata su trasformazioni moltiplicative, possiamo estendere la trattazione con le cosiddette *coordinate omogenee* ([1], [2], [3], [4]).

Si definisce terna di coordinate omogenee bidimensionale una terna ordinata non nulla di reali

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (13)$$

Si definiscono *equivalenti* due terne di coordinate omogenee con componenti proporzionali tra loro

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \equiv \begin{bmatrix} k \cdot x \\ k \cdot y \\ k \cdot w \end{bmatrix} \iff k \neq 0 \quad (14)$$

Per semplificare la notazione, scriveremo il vettore normalizzato in modo tale che la terza componente risulti unitaria. Moltiplicheremo cioè, quando si renderà necessario, il vettore delle coordinate  $\mathbf{p} \triangleq [x \ y \ w]^T$  per  $1/w$ . Ciò è equivalente a considerare, dello spazio tridimensionale definito da  $x$ ,  $y$  e  $w$ , l'intersezione delle rette definite dai vettori col piano  $w = 1$ .

Questo modo di procedere, che può apparire inutilmente complicato, ci permette di rappresentare tutte le trasformazioni lineari elementari in forma moltiplicativa attraverso delle più generali *matrici di trasformazione*. Si può verificare immediatamente che le trasformazioni precedentemente citate sono rappresentabili in coordinate omogenee dalle matrici di seguito riportate.

**Traslazione.**

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (15)$$

**Scalamento.**

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (16)$$

**Rotazione.**

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \vartheta & -\sin \vartheta & 0 \\ \sin \vartheta & \cos \vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (17)$$

**Riflessione (rispetto all'asse  $x$ ).**

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (18)$$

**Deformazione di taglio (*shear*).**

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & t_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (19)$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ t_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (20)$$

Si noti che anche la traslazione è, in questo contesto, una trasformazione moltiplicativa. Inoltre continua a valere, ed è un facile esercizio verificarlo, la proprietà per cui, se una trasformazione viene composta con una trasformazione *dello stesso tipo*, la trasformazione risultante rimane di quel tipo ed i suoi parametri sono la somma (o il prodotto, nel caso dello scalamento) dei parametri originari.

Può essere interessante notare che le matrici viste sono non singolari. Esse ammettono quindi una matrice inversa che viene definita, per ovvi motivi, *matrice di trasformazione inversa*. Naturalmente le trasformazioni inverse applicate alle immagini trasformate forniscono come risultato l'immagine originaria. Ricavare le inverse e confrontarle con le matrici originarie può aiutare a capire meglio quanto detto. Per quanto riguarda la rotazione, in particolare, si tenga conto delle relazioni trigonometriche  $\cos \vartheta = \cos(-\vartheta)$  e  $-\sin \vartheta = \sin(-\vartheta)$ .

### 2.1.3 Composizione di trasformazioni

Applicando successivamente un numero arbitrario di trasformazioni lineari qualsiasi, la trasformazione equivalente è una trasformazione lineare, la cui matrice è data dal prodotto delle matrici delle trasformazioni coinvolte. Nella produttoria, le matrici delle trasformazioni si troveranno nell'ordine inverso rispetto a quello di applicazione. Si ricordi, infatti, che il prodotto matriciale *non* gode della proprietà commutativa.

Si dimostra che la più generale delle trasformazioni lineari, o *trasformazione affine*, è rappresentata dalla seguente matrice

$$\mathbf{T} \triangleq \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \quad (21)$$

Facciamo qualche esempio.

**Esempio 1.** Si desidera ruotare il piano di  $\varphi$  gradi rispetto ad un punto di coordinate  $(\bar{x}, \bar{y})$  diverso dall'origine degli assi. Si può pensare a questa trasformazione come la composizione di tre trasformazioni elementari:

1. Traslazione del piano di  $(-\bar{x}, -\bar{y})$ ;
2. Rotazione del piano di  $\varphi$  gradi rispetto all'origine;
3. Traslazione del piano di  $(\bar{x}, \bar{y})$ .

La matrice di trasformazione da applicare ad ogni punto  $\mathbf{p}$  dello spazio omogeneo è data dal prodotto delle tre trasformazioni *nel seguente ordine*

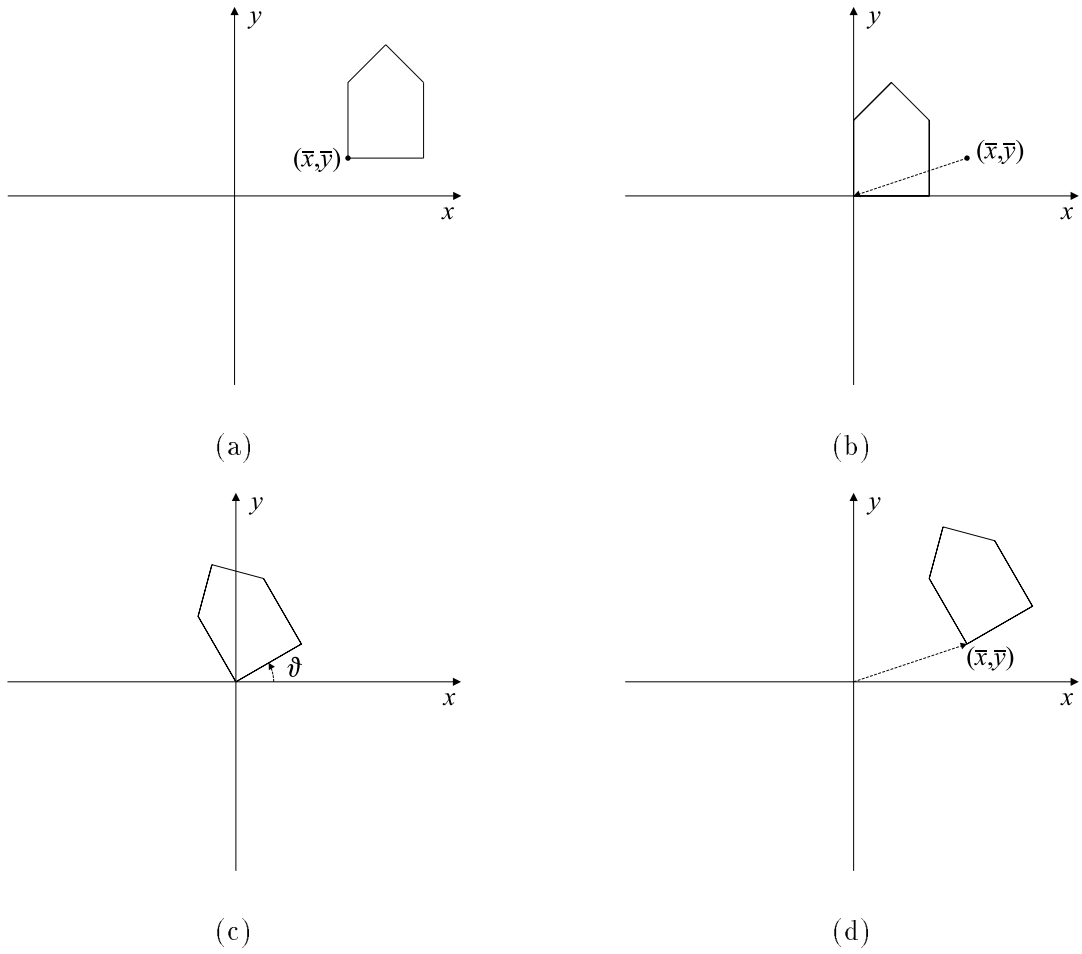


Figura 2: Composizione di trasformazioni. (a) Immagine originale; (b) traslazione di  $(-\bar{x}, -\bar{y})$ ; (c) rotazione rispetto all'origine; (d) traslazione di  $(\bar{x}, \bar{y})$ .

$$\begin{aligned}
 & \begin{bmatrix} 1 & 0 & \bar{x} \\ 0 & 1 & \bar{y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\bar{x} \\ 0 & 1 & -\bar{y} \\ 0 & 0 & 1 \end{bmatrix} = \\
 & = \begin{bmatrix} \cos \varphi & -\sin \varphi & \bar{x}(1 - \cos \varphi) + \bar{y} \sin \varphi \\ \sin \varphi & \cos \varphi & \bar{y}(1 - \cos \varphi) - \bar{x} \sin \varphi \\ 0 & 0 & 1 \end{bmatrix} \quad (22)
 \end{aligned}$$

Il risultato lascia intendere che esiste una soluzione più diretta al problema: la matrice (22) è infatti fattorizzabile in una matrice di rotazione di  $\varphi$  gradi ed una traslazione di  $[d_x \ d_y]^T = [\bar{x}(1 - \cos \varphi) + \bar{y} \sin \varphi \ \bar{y}(1 - \cos \varphi) - \bar{x} \sin \varphi]^T$ .

Più in generale, le matrici di trasformazione del piano sono costituite da un blocco di traslazione  $1 \times 2$  in alto a destra, completamente disaccoppiato dal blocco  $2 \times 2$  in alto a sinistra che accorpa tutte le altre trasformazioni.  $\square$

**Esempio 2.** Siamo ora in grado di effettuare uno scalamento del piano di fattori  $s_x$  ed  $s_y$  rispetto ad un punto di coordinate  $(\bar{x}, \bar{y})$  diverso dall'origine degli assi.



Attraverso ragionamenti analoghi a quelli del precedente Esempio 1, si ottiene la matrice

$$\begin{bmatrix} s_x & 0 & \bar{x}(1-s_x) \\ 0 & s_y & \bar{y}(1-s_y) \\ 0 & 0 & 1 \end{bmatrix} \quad (23)$$

□

Date due matrici  $\mathbf{M}_1$  ed  $\mathbf{M}_2$ , entrambe rappresentanti una trasformazione lineare elementare, è interessante chiedersi quando il loro prodotto commuti, cioè quando accada che  $\mathbf{M}_1 \cdot \mathbf{M}_2 = \mathbf{M}_2 \cdot \mathbf{M}_1$ . Si può verificare che questo accade nei seguenti casi particolari

- Traslazione·Traslazione
- Scalamento·Scalamento
- Rotazione·Rotazione
- Scalamento\_uniforme·Rotazione

Sussiste inoltre il seguente notevole risultato: il determinante di una qualsiasi matrice di trasformazione è pari al rapporto tra le aree dell'immagine trasformata e quelle dell'immagine originaria

$$\det(\mathbf{T}) = \frac{A'}{A} \quad (24)$$

Poco sorprendentemente, risulta che il determinante di una generica matrice di rototraslazione è unitario.

## 2.2 Trasformazioni 3D

Parlando di sistemi di coordinate cartesiane a tre dimensioni è necessario, per fugare ogni ambiguità, specificare a che tipo di sistema ci si riferisca. Nel presente paragrafo si tratteranno solamente *terne destrorse* (si veda la figura 3), con la convenzione che rotazioni di angoli positivi corrispondano, guardando un asse positivo verso l'origine, ad una rotazione in senso *antiorario*. Inoltre, i nomi degli assi saranno tali per cui una rotazione positiva con asse di rotazione l'asse  $x$  ruota il sistema (in senso antiorario, come già precisato) in modo che l'asse  $y$  "inseguia" l'asse  $z$ .

### 2.2.1 Rappresentazione matriciale

Le definizioni e i concetti presentati nella sezione 2.1 per le trasformazioni del piano possono essere estese alle trasformazioni dello spazio 3D. In particolare, lavorando in un iperspazio tetradiimensionale di coordinate omogenee, si perviene a matrici di trasformazione dello spazio 3D di semplice interpretazione.

**Traslazione.**

$$\begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

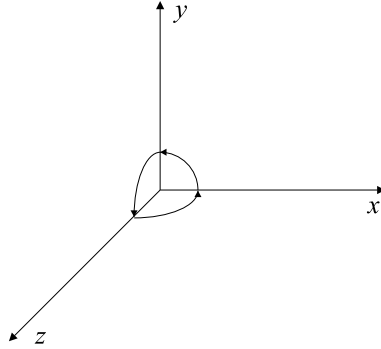


Figura 3: Sistema di riferimento destrorso.

**Scalamento.**

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

**Riflessione (rispetto al piano  $x = 0$ ).**

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

**Deformazione di taglio (*shear*).** Per quanto riguarda questa trasformazione, la terza dimensione ha triplicato i gradi di libertà dei parametri (un caso analogo, ma più interessante si verificherà, come vedremo, nel caso della rotazione, trattata separatamente). Riportiamo qui sotto due delle sei possibili deformazioni di taglio elementari, illustrate nella figura 4: quelle relative alla coordinata  $x$ .

$$\begin{bmatrix} 1 & 0 & t_x & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & t_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

Le relazioni che legano i parametri  $t_x$  e  $t_y$  all'angolo  $\vartheta$  della deformazione sono identiche a quelle del caso bidimensionale.

**Rotazioni.** Come accennato, nello spazio tridimensionale le rotazioni elementari diventano tre, una per ogni asse ordinato. Nella terminologia aeronautica, assumendo che la direzione del

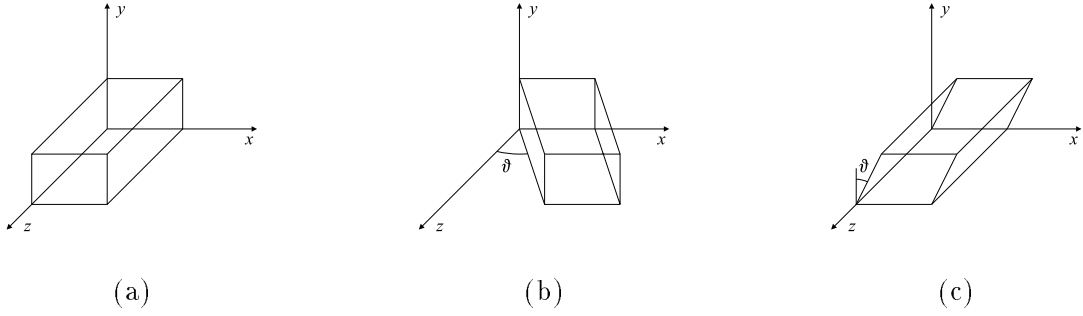


Figura 4: Esempi di deformazione di taglio 3D. (a) Oggetto originale; (b) oggetto deformato lungo il piano  $z = 0$ ; (c) oggetto deformato lungo il piano  $y = 0$ .

moto sia l'asse  $x$ , le rotazioni attorno agli assi  $x$ ,  $y$ ,  $z$  si dicono, rispettivamente, *roll* (*rollio*), *yaw* (*virata*) e *pitch* (*beccheggio*); esse sono rappresentate, nello stesso ordine, dalle seguenti matrici.

$$\mathbf{R}_x \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \vartheta & -\sin \vartheta & 0 \\ 0 & \sin \vartheta & \cos \vartheta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (30)$$

$$\mathbf{R}_y \triangleq \begin{bmatrix} \cos \vartheta & 0 & \sin \vartheta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \vartheta & 0 & \cos \vartheta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (31)$$

$$\mathbf{R}_z \triangleq \begin{bmatrix} \cos \vartheta & -\sin \vartheta & 0 & 0 \\ \sin \vartheta & \cos \vartheta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (32)$$

Componendo le tre rotazioni elementari, si ottiene la più generica matrice di rotazione

$$\mathbf{R} \triangleq \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (33)$$

avente, analogamente al caso bidimensionale, le proprietà di ortogonalità ( $\mathbf{R}^{-1} = \mathbf{R}^T$ ) e di determinante unitario. Si dimostra che la trasformazione (33) può essere effettuata avvalendosi di due sole delle tre rotazioni elementari, applicando comunque tre rotazioni in totale.

Si dimostra inoltre (teorema di Eulero) che la stessa generica rotazione è equivalente ad un'unica rotazione attorno ad una retta dello spazio 3D.

Un'altra rappresentazione delle rotazioni 3D, molto più tecnica e articolata [5], permette una interessante interpretazione di questo risultato. Si supponga di dover effettuare una rotazione attorno all'asse definito dal vettore

$$\mathbf{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (34)$$

di un angolo  $\vartheta$  pari a  $\|\mathbf{x}\|_2$ . Si dimostra che il blocco  $\mathbf{r}$   $3 \times 3$  di rotazione della matrice (33) risulta essere

$$\mathbf{r} = \exp(\mathbf{X}) \triangleq \sum_{k=0}^{\infty} \frac{\mathbf{X}^k}{k!} \quad (35)$$

dove

$$\mathbf{X} \triangleq \mathbf{x} \times \triangleq \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad (36)$$

Questo risultato giustifica quanto già detto: è possibile ottenere una qualsiasi combinazione di rotazioni con una sola rotazione rispetto ad un asse opportuno.

Inoltre, si può scrivere  $\mathbf{x}$  come combinazione lineare dei versori  $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$

$$\mathbf{x} = x_1 \mathbf{u}_1 + x_2 \mathbf{u}_2 + x_3 \mathbf{u}_3 \quad (37)$$

e quindi

$$\mathbf{X} = x_1 \mathbf{X}_1 + x_2 \mathbf{X}_2 + x_3 \mathbf{X}_3 \quad (38)$$

avendo definito

$$\mathbf{X}_1 \triangleq \mathbf{u}_1 \times \triangleq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{X}_2 \triangleq \mathbf{u}_2 \times \triangleq \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad \mathbf{X}_3 \triangleq \mathbf{u}_3 \times \triangleq \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (39)$$

In altre parole lo spazio delle matrici emisimmetriche è uno spazio vettoriale.

Dalla (35) si ricava la cosiddetta rappresentazione *esponenziale*, o di *prima specie* delle rotazioni. Si osservi che le componenti  $x_1, x_2, x_3$  *non coincidono* con i valori di *roll, yaw e pitch* della rappresentazione (detta di *seconda specie*) (33), in quanto in generale

$$\exp(x_1 \mathbf{X}_1 + x_2 \mathbf{X}_2 + x_3 \mathbf{X}_3) \neq \exp(x_1 \mathbf{X}_1) \exp(x_2 \mathbf{X}_2) \exp(x_3 \mathbf{X}_3) \quad (40)$$

poiché il prodotto matriciale non commuta.

Ancora, le proposizioni

$$\begin{aligned} \forall \mathbf{R} \quad \exists \mathbf{X} : \quad e^{\mathbf{X}} &= \mathbf{R} \\ \forall \mathbf{X} \quad \exists \mathbf{R} : \quad e^{\mathbf{X}} &= \mathbf{R} \end{aligned} \quad (41)$$

sono vere solo per  $\mathbf{R} \in SO(3)$  (matrici di rotazione) e per  $\mathbf{X} \in so(3)$  (matrici emisimmetriche).

### 3 Trasformazioni non lineari del piano

Le trasformazioni non lineari che vengono considerate in questa sezione rappresentano una classe ristretta delle possibili non linearità.

Si definisce *trasformazione polinomiale del piano* una qualsiasi funzione polinomiale del vettore bidimensionale delle coordinate

$$x' = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + a_6x^3 + a_7x^2y + a_8xy^2 + a_9y^3 + \dots \quad (42)$$

$$y' = b_0 + b_1x + b_2y + b_3x^2 + b_4xy + b_5y^2 + b_6x^3 + b_7x^2y + b_8xy^2 + b_9y^3 + \dots \quad (43)$$

Ci si arresti per semplicità ai termini del second'ordine, ottenendo le *trasformazioni quadratiche*, qui espresse in forma matriciale

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix} \quad (44)$$

Trasformazioni di questo genere o di ordine superiore possono essere impiegate, naturalmente nella loro versione digitale (vedi la seguente sezione 4), per compensare la distorsione spaziale causata da un sistema di acquisizione di immagini non ideale. Tipicamente, una videocamera elettronica introduce una distorsione radiale a cuscinetto o a barilotto, come illustrato nella figura 5.

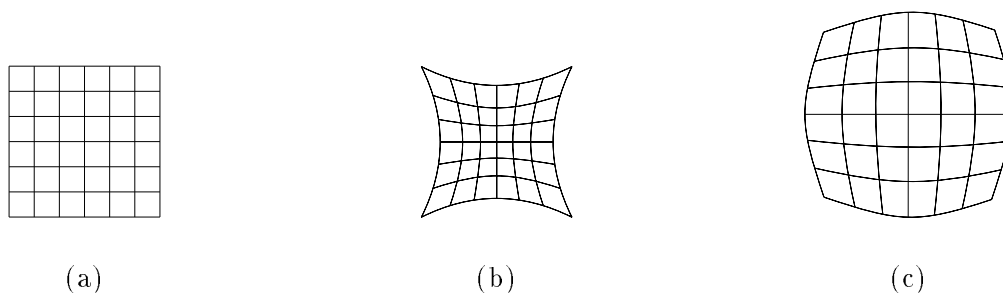


Figura 5: Tipiche distorsioni non lineari dell'ottica. (a) Griglia originale; (b) distorsione a cuscinetto; (c) distorsione a barilotto.

Ci si pone il problema di stimare i parametri  $\hat{a}_i$  e  $\hat{b}_i$  tali per cui la (44) meglio approssimi la distorsione non lineare  $f$  (ignota) che effettivamente l'immagine originale ha subito, in modo da effettuare la trasformazione inversa (trasformazione di *compensazione*) sull'immagine acquisita (si veda la figura 6).

Il criterio su cui si basa la scelta dei parametri ottimi è tipicamente la minimizzazione l'*errore quadratico medio* (*mean square error*) tra un insieme di coordinate di *punti di osservazione*

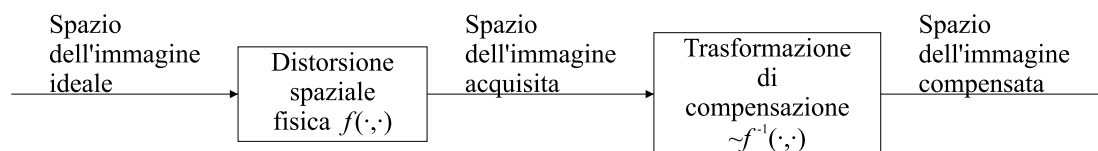


Figura 6: Concetto di compensazione delle distorsioni.

$(x'_m, y'_m)$  appartenenti all'immagine acquisita e le stime  $(x_m, y_m)$  delle coordinate dei punti corrispondenti nell'immagine originale (*punti di controllo*, o *control points*), con  $1 \leq m \leq M$ .

Definendo i vettori

$$\begin{aligned}\mathbf{x}' &\triangleq [x'_1 \quad x'_2 \quad \dots \quad x'_M]^T \\ \mathbf{y}' &\triangleq [y'_1 \quad y'_2 \quad \dots \quad y'_M]^T \\ \mathbf{a} &\triangleq [a_0 \quad a_1 \quad \dots \quad a_5]^T \\ \mathbf{b} &\triangleq [b_0 \quad b_1 \quad \dots \quad b_5]^T\end{aligned}$$

e la matrice

$$\mathbf{A} \triangleq \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2 y_2 & y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_M & y_M & x_M^2 & x_M y_M & y_M^2 \end{bmatrix}$$

si può esprimere l'errore quadratico medio nella compatta forma matriciale

$$\mathcal{MSE} \triangleq (\mathbf{x}' - \mathbf{A}\mathbf{a})^T (\mathbf{x}' - \mathbf{A}\mathbf{a}) + (\mathbf{y}' - \mathbf{A}\mathbf{b})^T (\mathbf{y}' - \mathbf{A}\mathbf{b}) \quad (45)$$

dove le uniche incognite sono proprio i vettori  $\mathbf{a}$  e  $\mathbf{b}$ . Minimizzare la (45) rispetto ai dodici parametri  $a_i$  e  $b_i$  è un classico problema di ottimizzazione: si dimostra che la sua soluzione è data dai vettori

$$\hat{\mathbf{a}} = \mathbf{A}^* \mathbf{x}' \quad (46)$$

$$\hat{\mathbf{b}} = \mathbf{A}^* \mathbf{y}' \quad (47)$$

avendo definito con  $\mathbf{A}^*$  la matrice *pseudoinversa* di  $\mathbf{A}$

$$\mathbf{A}^* \triangleq (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \quad (48)$$

Ottenuti i parametri della trasformazione diretta, siamo ora in grado di invertire la trasformazione (44) attraverso calcoli analitici o iterativi ed applicarla all'immagine distorta, correggendola.

## 4 Dall'analogico al digitale: elaborazione di immagini al calcolatore

Fino ad ora si è parlato di immagini identificandole implicitamente con funzioni continue delle coordinate continue ed illimitate del piano. Più propriamente, però, quando si parla di *immagine analogica* si intende in genere una funzione continua di due variabili continue e *limitate*. Similmente, per *immagine digitale* si intende in genere una funzione a valori finiti (a seguito della quantizzazione) di due variabili discrete (come effetto del campionamento) e limitate.

Il fatto di lavorare con un numero finito di coppie ordinate di variabili indipendenti (che sul monitor prendono il nome di *pel*, o *pixel* – acronimi di *picture element*, *elemento pittorico* –, e che in memoria RAM prendono il nome di *indirizzi di memoria*) genera delle ambiguità nelle

trasformazioni. Basti pensare ad una rotazione di 45 gradi di un'immagine digitale: i quattro angoli dell'immagine originaria non saranno mappati nella nuova immagine, e viceversa nessun valore dell'immagine originale si mapperà nell'area scoperta dalla trasformazione. Non sarà quindi così facile come nel caso di trasformazioni del piano di pagina 7 parlare di trasformazioni inverse...

Si introducono, con qualche abuso di notazione, l'immagine di ingresso  $F(p, q)$ , con  $p$  e  $q$  interi e tali che  $1 \leq p \leq P$  e  $1 \leq q \leq Q$ , e l'immagine di uscita  $G(j, k)$ , con  $j$  e  $k$  interi e tali che  $1 \leq j \leq J$  e  $1 \leq k \leq K$ . Esse sono di fatto due matrici, con dimensioni in generale diverse tra loro. Tornerà utile nel seguito associare *biunivocamente* ad ogni indice un valore reale, definito dalle equazioni seguenti

$$x_p \triangleq p - \frac{1}{2} \quad (49)$$

$$y_q \triangleq Q + \frac{1}{2} - q \quad (50)$$

$$u_j \triangleq j - \frac{1}{2} \quad (51)$$

$$v_k \triangleq K + \frac{1}{2} - k \quad (52)$$

Intuitivamente, questo corrisponde ad adagiare le immagini di ingresso e di uscita sul primo quadrante di un sistema di assi cartesiani, ed assegnare a ciascun *pixel* l'ascissa e l'ordinata che corrispondono al suo baricentro, come mostra la figura 7. Si osservi che lungo l'asse delle ordinate gli indici  $k$  e  $q$  crescono avvicinandosi all'origine.

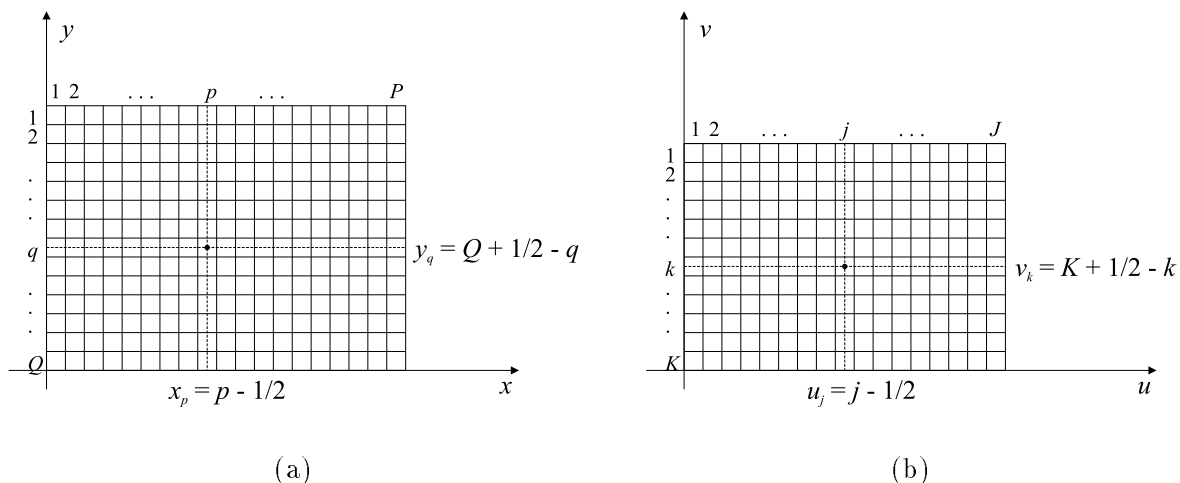


Figura 7: Relazione tra immagini discrete e coordinate cartesiane. (a) Immagine di ingresso; (b) immagine di uscita.

#### 4.1 Ricampionamento

Si può avere l'impressione di avere di nuovo complicato inutilmente cose semplici, ma è stato necessario. Si consideri, infatti una semplice traslazione: se ci si sposta di quantità intere in entrambe le dimensioni è facile far corrispondere, a meno di regioni scoperte, ogni *pixel* di

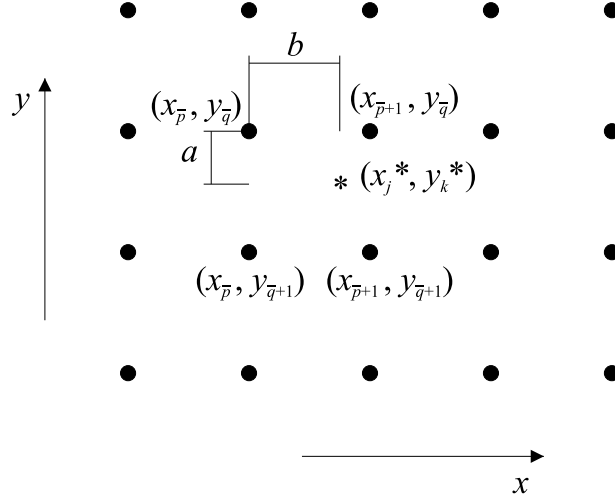


Figura 8: Posizione del punto  $(x_j^*, y_k^*)$  rispetto ai baricentri dei *pixel* adiacenti.

$G(j, k)$  ad uno ed un solo *pixel* di  $F(p, q)$ . Supponiamo invece di traslare l'immagine di un vettore  $[d_x \ d_y]^T$  di componenti non intere: che valore daremo al generico *pixel* di uscita?

Il modo piÙ semplice di risolvere la questione è proprio quello di calcolare il baricentro di ogni *pixel* di  $G(j, k)$  attraverso la (51) e la (52) ed applicarvi la trasformazione *inversa*. Nel nostro esempio si tratta di una traslazione di  $[-d_x \ -d_y]^T$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (53)$$

Come esemplificato nella figura 8, si otterrà un punto nel piano  $(x, y)$ , diciamo  $(x_j^*, y_k^*)$ , che giace all'interno del reticolo dei baricentri dei *pixel* di  $F(p, q)$ . Si sceglie quindi un'interpolazione, in modo che il valore assegnato a  $G(j, k)$  sia funzione dei valori assunti dai *pixel* di  $F(p, q)$  i cui baricentri sono circumvicini a  $(x_j^*, y_k^*)$ .

Sempre riferendosi alla figura 8, si definisce  $(\bar{p}, \bar{q})$  il *pixel* di ingresso con il baricentro piÙ vicino ed "in alto a sinistra" rispetto a  $(x_j^*, y_k^*)$ , e si definiscono  $a$  e  $b$  le proiezioni della distanza tra  $(x_j^*, y_k^*)$  e  $(x_{\bar{p}}, y_{\bar{q}})$  sull'asse  $x$  e  $y$ , rispettivamente. È necessario sottolineare che, per trasformazioni piÙ complesse della traslazione, non solo  $\bar{p}$  e  $\bar{q}$ , ma anche  $a$  e  $b$  dipendono dalla particolare coordinata  $(j, k)$  dell'immagine di uscita considerata.

Questa operazione, detta di *reverse address computation*, prende anche il nome di *ricampionamento*, in quanto è equivalente a ricampionare l'immagine originale su un nuovo reticolo, in modo che l'immagine di uscita vi si mappi biunivocamente. Si confronti la figura 9 per una visione dall'alto dell'intero metodo. Ovviamente, dal punto di vista operativo, non è necessario calcolarsi i baricentri di  $F(p, q)$ , in quanto esiste una relazione diretta tra  $(\bar{p}, \bar{q})$  e  $(x_j^*, y_k^*)$

$$\bar{p} = \left\lfloor x_j^* - \frac{1}{2} \right\rfloor + 1 \quad (54)$$



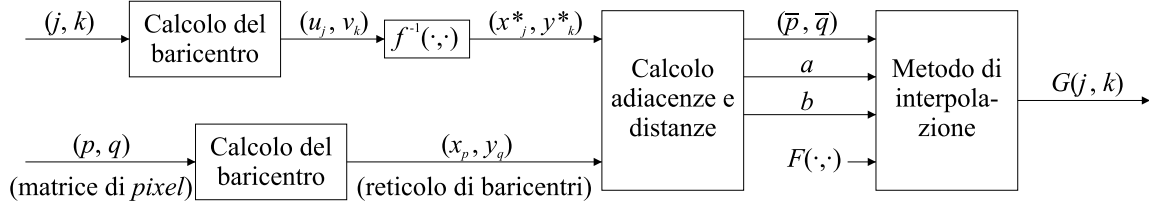


Figura 9: Schema a blocchi del processo di ricampionamento ed interpolazione per il generico *pixel*  $(j, k)$ .

$$\bar{q} = Q - \left\lceil y_k^* + \frac{1}{2} \right\rceil + 1 \quad (55)$$

da cui si ricavano anche  $a$  e  $b$ , attraverso la (49) e la (50).

Prima di considerare i più comuni metodi di interpolazione, un'osservazione: dal punto di vista computazionale, il collo di bottiglia del ricampionamento è, come spesso accade, l'inversione della matrice (nel caso di trasformazioni lineari). Si ricordi, però, che le matrici di trasformazione posseggono una particolarissima struttura, che ne agevola l'inversione. Inoltre le matrici inverse delle trasformazioni elementari esposte nel paragrafo 2.1 sono note in forma chiusa, e quindi immediatamente ricavabili.

#### 4.1.1 Metodi di interpolazione

La più semplice forma di interpolazione 2D consiste nell'assumere il valore del campione più vicino al punto  $(x_j^*, y_k^*)$  di interpolazione. Questa operazione, chiamata *nearest neighbor interpolation*, è ovviamente grossolana.

L'errore di interpolazione può essere significativamente ridotto utilizzando tutti e quattro i campioni che circondano  $(x_j^*, y_k^*)$ . Un approccio comune, chiamato *interpolazione bilineare*, è di interpolare linearmente lungo ogni riga dell'immagine, e poi interpolare linearmente il risultato lungo la colonna. Assumendo che il campione in alto a sinistra sia associato al valore  $F(\bar{p}, \bar{q})$ , si ha, in formule

$$G(j, k) \leftarrow (1-a)[(1-b)F(\bar{p}, \bar{q}) + bF(\bar{p}, \bar{q} + 1)] + a[(1-b)F(\bar{p} + 1, \bar{q}) + bF(\bar{p} + 1, \bar{q} + 1)] \quad (56)$$

dove si è indicato con il segno “ $\leftarrow$ ” l'operatore informatico di assegnamento.

Sebbene entrambe le interpolazioni verticale ed orizzontale siano operazioni lineari, la loro applicazione sequenziale risulta in generale una superficie non lineare tra i quattro campioni vicini.

L'interpolazione bilineare può essere generalizzata per ogni funzione  $W(\cdot)$  che sia nulla fuori dell'intervallo  $[-1, 1]$ . Ricordando che  $0 \leq a \leq 1$  e  $0 \leq b \leq 1$ , si ottiene la più generale interpolazione

$$G(j, k) \leftarrow F(\bar{p}, \bar{q})W(-a)W(b)$$

$$\begin{aligned}
& +F(\bar{p}, \bar{q} + 1)W(-a)W(-(1 - b)) \\
& +F(\bar{p} + 1, \bar{q})W(1 - a)W(b) \\
& +F(\bar{p} + 1, \bar{q} + 1)W(1 - a)W(-(1 - b))
\end{aligned} \tag{57}$$

di cui la (56) è un caso particolare, ottenuta ponendo

$$W(t) = \begin{cases} 1 + t & -1 \leq t \leq 0 \\ 1 - t & 0 \leq t \leq 1 \\ 0 & \text{altrove} \end{cases} \tag{58}$$

cioè la classica finestra triangolare, illustrata nella figura 10.

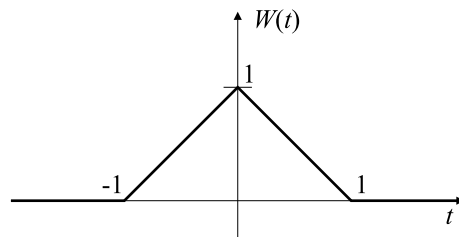


Figura 10:  $W(\cdot)$  per l'interpolazione bilineare.

Naturalmente si possono prendere in considerazione più di quattro campioni circostanti, pervenendo a metodi di interpolazione più sofisticati come l'*interpolazione bicubica*.

#### 4.1.2 Metodi convoluzionali

In molti casi il metodo di ricampionamento per interpolazione non raggiunge risultati soddisfacenti. Per la trasformazione di scalamento, in particolare, si sono studiati metodi *ad hoc* di tipo convoluzionale [6].

Si consideri, ad esempio, l'ingrandimento di un fattore 2 di un'immagine lungo entrambi gli assi. L'operazione può essere eseguita in due passi. Innanzitutto l'immagine originale viene trasferita su una matrice di dimensioni doppie, frapponendo ai valori originali righe e colonne di zeri (*zero interleaving*), come di seguito illustrato

$$\begin{array}{cc} A & B \\ C & D \end{array} \longrightarrow \begin{array}{ccc} A & 0 & B \\ 0 & 0 & 0 \\ C & 0 & D \end{array}$$

L'immagine così ottenuta viene convoluta con uno dei *nuclei di interpolazione discreta* riportati nella tabella 1.

Per fattori di ingrandimento interi  $N$  si opera uno  $(N - 1)$ -*zero interleaving*, utilizzando in genere un nucleo "Peg"  $N \times N$ . Per nuclei di grosse dimensioni, può essere computazionalmente conveniente filtrare nel dominio delle frequenze.

## 4.2 Implementazione di algoritmi efficienti

La seguente trattazione si riferisce, per semplicità, alle trasformazioni lineari del piano. I risultati sono naturalmente estendibili al 3D.

“Peg”	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$
Piramide	$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$
Campana	$\frac{1}{16} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 9 & 9 & 3 \\ 3 & 9 & 9 & 3 \\ 1 & 3 & 3 & 1 \end{bmatrix}$
B-spline cubica	$\frac{1}{64} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$

Tabella 1: Alcuni nuclei di interpolazione.

È bene non dare per scontato che se si desidera effettuare una trasformazione *lineare* di immagini sintetiche di grafica vettoriale, e cioè univocamente descritte da vettori di punti e dalle specifiche di interpolazione tra essi (si pensi ad esempio ad una applicazione di realtà virtuale), è perfettamente inutile (quanto dispendioso!) procedere *pixel per pixel*, poiché è facilmente dimostrabile che ogni trasformazione lineare mappa segmenti in segmenti, e superficie piane in superficie piane. Sarà quindi sufficiente trasformare i vettori dei vertici mantenendo invariate le mutue relazioni di interpolazione e tessitura, affidando il tracciamento ad un buon algoritmo di *rendering*.

Si è già fatto cenno ai notevoli vantaggi che la semplice ed invariante struttura (21) delle matrici di trasformazione presenta. La moltiplicazione di tale matrice per un vettore di coordinate omogenee può essere disaccoppiata nella moltiplicazione del vettore bidimensionale associato per la sottomatrice  $2 \times 2$  in alto a sinistra seguita dalla somma con il vettore di traslazione

$$x' \leftarrow xa_{11} + ya_{12} + a_{13} \quad (59)$$

$$y' \leftarrow xa_{21} + ya_{22} + a_{23} \quad (60)$$

In questo modo le operazioni elementari si riducono da nove moltiplicazioni e sei somme a quattro moltiplicazioni e quattro somme.

Come già evidenziato nel precedente paragrafo 4.1, per calcolare una matrice di trasformazione inversa non è necessario procedere con metodi tradizionali. Inoltre le inverse delle

trasformazioni elementari sono facilmente calcolabili simbolicamente, e quindi disponibili in forma chiusa.

#### 4.2.1 Separabilità delle trasformazioni elementari

Le trasformazioni di traslazione, scalamento, riflessione e deformazione di taglio sono separabili, nel senso che è possibile calcolare almeno una delle nuove coordinate  $(x', y')$  in funzione di una sola delle due “vecchie” coordinate  $(x, y)$ . Ciò consente di effettuare l’operazione in due passi. Prima si processa l’immagine per righe e poi per colonne, o viceversa, con conseguente risparmio di risorse di calcolo.

A prima vista, sembrerebbe che la rotazione non goda di questa proprietà, ma Catmull e Smith [7] hanno dimostrato il contrario.

Dalle equazioni della rotazione si ricavano le seguenti relazioni

$$x' = x \cos \vartheta - y \sin \vartheta \quad (61)$$

$$x = \frac{x' + y \sin \vartheta}{\cos \vartheta} \quad (62)$$

Ora, la trasformazione di rotazione inversa a quella considerata (nella forma classica) si scrive

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \vartheta & \sin \vartheta \\ -\sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (63)$$

da cui

$$y' = \frac{x' \sin \vartheta + y}{\cos \vartheta} \quad (64)$$

Sostituendo la (61) nella (64) e la (62) nella (61), si ottengono le equazioni cercate. È possibile mostrare che i due passi della rotazione equivalgono a due deformazioni di taglio, effettuate una lungo l’asse  $x$  e l’altra lungo l’asse  $y$ . Come corollario di tutto ciò si può quindi affermare che è possibile ottenere *qualsiasi* trasformazione lineare avvalendosi delle sole trasformazioni di *traslazione* (in entrambe le dimensioni), *scalamiento e/o riflessione* (entrambe rispetto ad un solo asse) e *deformazione di taglio* (lungo entrambi gli assi). Si dice che essi sono il *nucleo* delle trasformazioni lineari.

#### 4.2.2 Approssimazione per piccoli angoli

Una particolare applicazione in cui è importante l’efficacia degli algoritmi è la creazione di viste successive di un oggetto, distanti di pochi gradi tra loro. Se le figure possono essere calcolate e visualizzate in tempi sufficientemente brevi (dai 30 ai 60 millisecondi ognuna), l’oggetto apparirà in rotazione dinamica.

Le equazioni di rotazione (17) (si sta analizzando il più semplice caso bidimensionale) richiede quattro moltiplicazioni e quattro somme. Per piccoli angoli  $\vartheta$  di rotazione, però, le (17) possono essere approssimate, osservando che  $\cos \vartheta \approx 1$ , dalle

$$x' \leftarrow x - y \sin \vartheta \quad (65)$$

$$y' \leftarrow x \sin \vartheta + y \quad (66)$$

che richiedono solo due moltiplicazioni e due somme. Un'approssimazione migliore si ottiene facendo uso della  $x'$  appena calcolata nella seconda equazione, ottenendo, tra l'altro, una matrice di trasformazione dal determinante unitario, come dovrebbe essere

$$x' \leftarrow x - y \sin \vartheta \quad (67)$$

$$y' \leftarrow x' \sin \vartheta + y \quad (68)$$

Naturalmente, l'approssimazione introduce un errore, errore che si propaga reiterando l'approssimazione nelle immagini successive. È quindi buona norma conservare in memoria l'immagine originale, riutilizzandola dopo  $360^\circ$  di rotazione.

## Bibliografia

- [1]
- [2] J.D. Foley, A. van Dam: *Fundamentals of Interactive Computer Graphics*. Addison-Wesley, 1982.
- [3] L.G. Roberts: "Machine Perception of Three-Dimensional Solids". In: *Optical and Electro-Optical Information Processing*, J.T. Tippett *et al.* Eds., MIT Press, 1965.
- [4] D.F. Rogers, J.A. Adams: *Mathematical Elements for Computer Graphics*. Mc Graw-Hill, 1976.
- [5]
- [6] W.K. Pratt: "Image Processing and Analysis Using Primitive Computational Elements". In: *Selected Topics in Signal Processing*, S. Haykin, Ed., Prentice-Hall, 1989
- [7] E. Catmull, A.R. Smith: "3-D Transformation of Images in Scanline Order". In: *Computer Graphics, (SIGGRAPH '80 Proc.)*, Vol. 14, No. 3, July 1980, pp. 279-285.
- [8] R. Bernstein: "Digital Image Processing of Earth Observation Sensor Data". In: *IBM J. Res. Devel.*, Vol. 20, No. 1, 1976, pp. 40-56
- [9] D.A. O'Handley, W.B. Greene: "Recent Developments in Digital Image Processing at the Image Processing Laboratory of the Jet Propulsion Laboratory". In: *Proc. IEEE*, Vol. 60, No. 7, July 1972, pp. 821-828
- [10] K.S. Fu, R.C. Gonzalez, C.S.G. Lee: *Robotics: Control, Sensing, Vision and Intelligence*. Mc Graw-Hill, 1987.

## **Indirizzo dell'autore**

Giulio Agostini

Via Residenza al Bosco, 6

24030

Caprino Bergamasco

Bergamo

email: [piu2046@cdc8g5.cdc.polimi.it](mailto:piu2046@cdc8g5.cdc.polimi.it)