

# Capitolo 6

## L'implementazione in tempo reale

### 6.1 L'implementazione in tempo distribuito e in tempo reale

Come accennato nel capitolo introduttivo, i principali parametri attraverso cui valutare un sistema di scansione tridimensionale sono: velocità di acquisizione, precisione nella ricostruzione, facilità di utilizzo e costo del sistema.

E' già stato visto che il metodo a luce debolmente strutturata richiede un hardware esterno veramente essenziale:

- un computer (di prestazioni medie)
- una telecamera (non necessariamente professionale)
- un frame grabber<sup>1</sup>
- una lampada che funge da sorgente luminosa
- una scacchiera per la calibrazione della telecamera
- una matita per la calibrazione della sorgente luminosa
- una bacchetta per generare il piano d'ombra

Quindi il metodo proposto in questo lavoro, in termini di costo, risulta sicuramente valido. Per valutare invece sperimentalmente le prestazioni del sistema in termini di accuratezza della ricostruzione, ci si può riferire inizialmente ad una implementazione in tempo distribuito. Una prima versione dell'algoritmo è stata infatti realizzata in Matlab, linguaggio sicuramente idoneo nel trattare immagini e operazioni tra matrici e vettori.

---

<sup>1</sup> Un frame grabber è essenzialmente una memoria che conserva le immagini provenienti dalla telecamera per poi scaricarle sul bus principale del computer attraverso un'opportuna temporizzazione.

Senza entrare nel merito dell'implementazione in tempo distribuito, l'idea essenziale e' quella di valutare per ciascun punto dell'immagine il relativo tempo d'ombra, cioe' l'istante di tempo in cui uno dei due bordi dell'ombra sta passando per il punto in esame. Il secondo passo (invero le due procedure si possono invertire) e' quello di valutare la posizione delle ascisse di attraversamento per ciascun istante di tempo, ossia per ciascun frame. Mettendo insieme queste informazioni, si associa univocamente ad ogni pixel il relativo piano ombra, da cui, attraverso la triangolazione, si ricavano le corrispondenti coordinate spaziali.

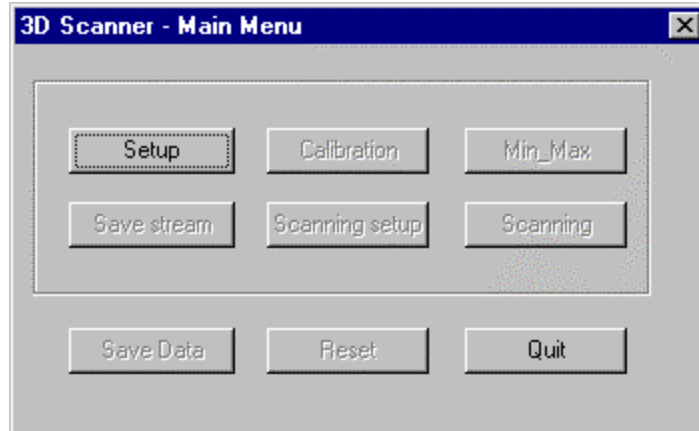
E' evidente che l'algoritmo richiede la memorizzazione integrale della sequenza dell'ombra che scorre sullo scenario. In particolare ciascuna immagine deve essere memorizzata in una matrice, nel senso che ogni elemento della suddetta matrice andra' a conservare il valore di luminosita' del corrispondente pixel dell'immagine. Si puo' intuire quindi che l'algoritmo richiede una considerevole quantita' di memoria. Non e' superfluo inoltre notare che il movimento della bacchetta, durante la sequenza da memorizzare, deve essere unidirezionale, non essendo lecita la possibilita', da parte dell'ombra, di tornare indietro, oscurando zone gia' scandite.

Per poter valutare quanto il metodo sia valido in termini di velocita' computazionale occorre necessariamente riferirsi ad un'implementazione in tempo reale. Inoltre, data la possibilita', in quest'ultimo caso, di avere disponibili i risultati subito dopo la scansione, si puo' valutare meglio l'affidabilita' della ricostruzione attraverso una massiccia fase di testing.

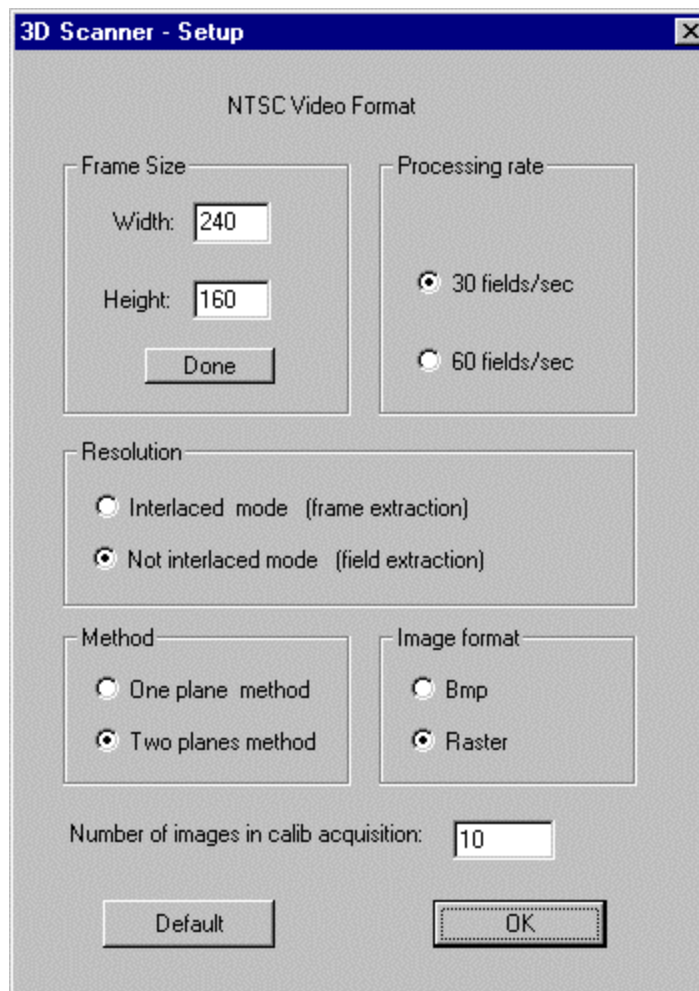
Infine un vantaggio sostanziale puo' essere quello di poter scandire anche piu' volte uno stesso pixel dell'immagine (facendovi passare ripetutamente uno dei due bordi dell'ombra). In tal modo si possono collezionare una successione di stime sulla posizione del punto nello spazio, mediando le quali, con opportuni pesi, si ottiene come risultato finale una stima senz'altro piu' affidabile (fase di *merging*).

Il codice in tempo reale e' stato realizzato in Visual C++. Tale linguaggio ha permesso di costruire una comoda interfaccia grafica a finestre in ambiente Microsoft Windows. Tramite un menu' principale (cfr. finestra 1) si puo' procedere attraverso le varie fasi preliminari della scansione:

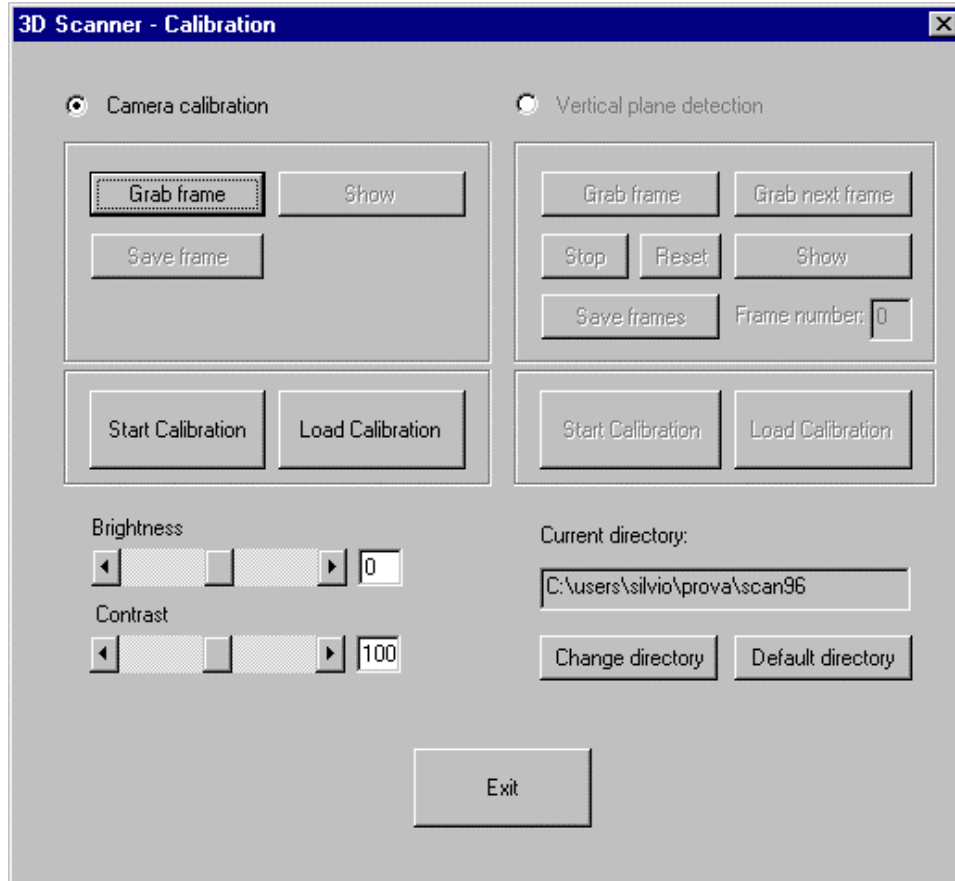
- Impostazione dei parametri preliminari (cfr. finestra 2):
  - Scelta del tipo di formato e della dimensione dell'immagine.
  - Utilizzo o meno del metodo con piano verticale
- Calibrazione (cfr. finestra 3):
  - Acquisizioni di immagini necessarie per la calibrazione della telecamera
  - Acquisizioni di immagini necessarie per la calibrazione del punto luce
  - Prelievo dei parametri di calibrazione da file esterno.
- Determinazione della matrice di contrasto (cfr. finestra 4):
  - Acquisizione della sequenza dell'ombra che scorre sullo scenario. Da tale sequenza vengono ricavate informazioni sui valori di massima e minima luminosità per ciascun pixel, necessarie per determinare la matrice di contrasto (cfr. § 2.3).
- Acquisizione di sequenze di immagini (cfr. finestra 5):
  - In tale menu' viene data la possibilità di memorizzare sequenze di immagini, utili sia come archivio che nella fase di testing per codici funzionanti in tempo distribuito.
- Impostazione per la scansione (cfr. finestra 6):
  - Scelta delle linee di localizzazione spaziale (cfr. § 2.6) per il piano orizzontale e verticale (nel caso eventuale non si conosca la posizione del punto luce nello spazio)
  - Scelta del limite destro e sinistro (in pixel) per delimitare la regione di scansione.
  - Scelta del bordo attivo (sinistro o destro) dell'ombra.
- Scansione (cfr. finestra 7):
  - Scansione in tempo reale dello scenario.
  - Possibilità di memorizzare i dati per una eventuale fase di *meshing* (cfr. capitolo 7 sui risultati sperimentali).
  - Analisi del profilo di una delle sezioni spaziali.
  - Misura delle coordinate spaziali di uno o più pixel spia controllabili da tastiera. Questa funzione è utile per misurare la distanza reale (in cm) tra due cursori mobili lampeggianti sull'immagine dello scenario



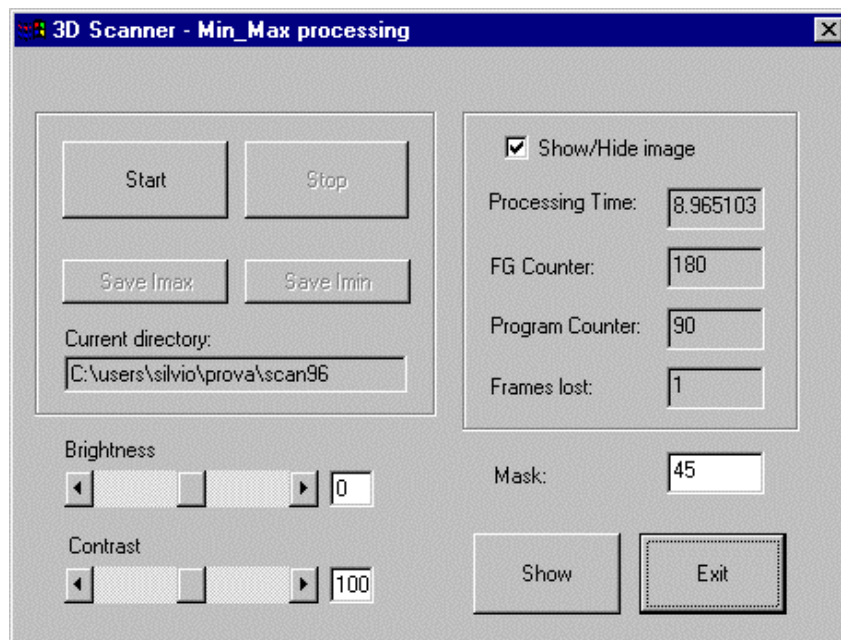
Finestra 1: Menu principale.



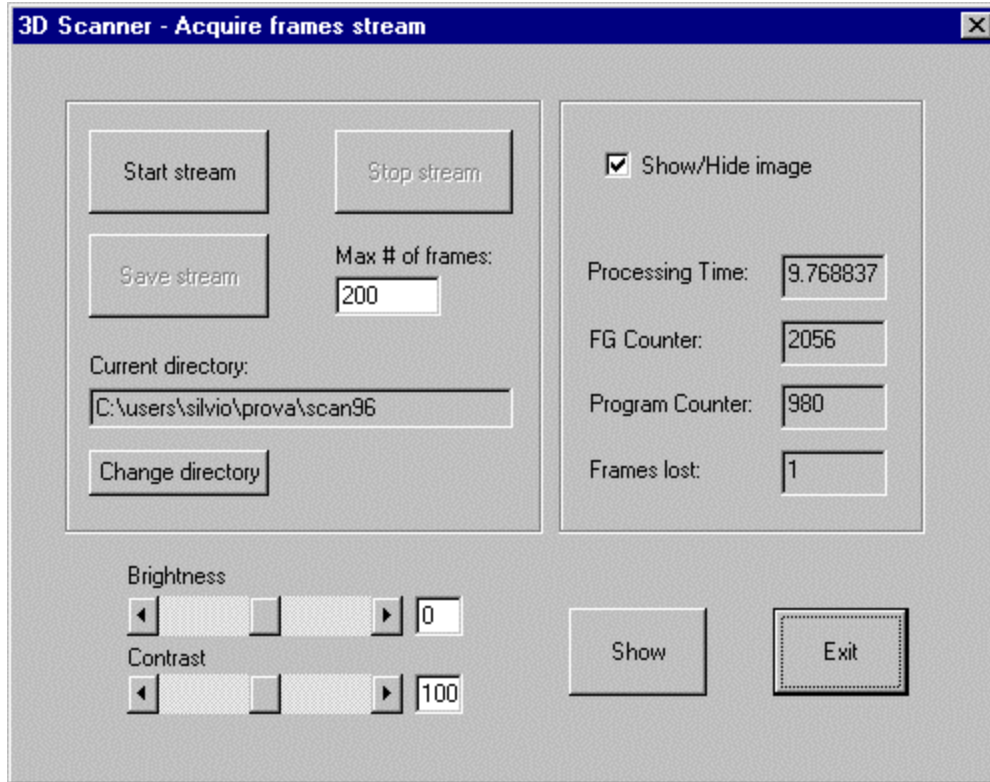
Finestra 1: Impostazione dei parametri preliminari.



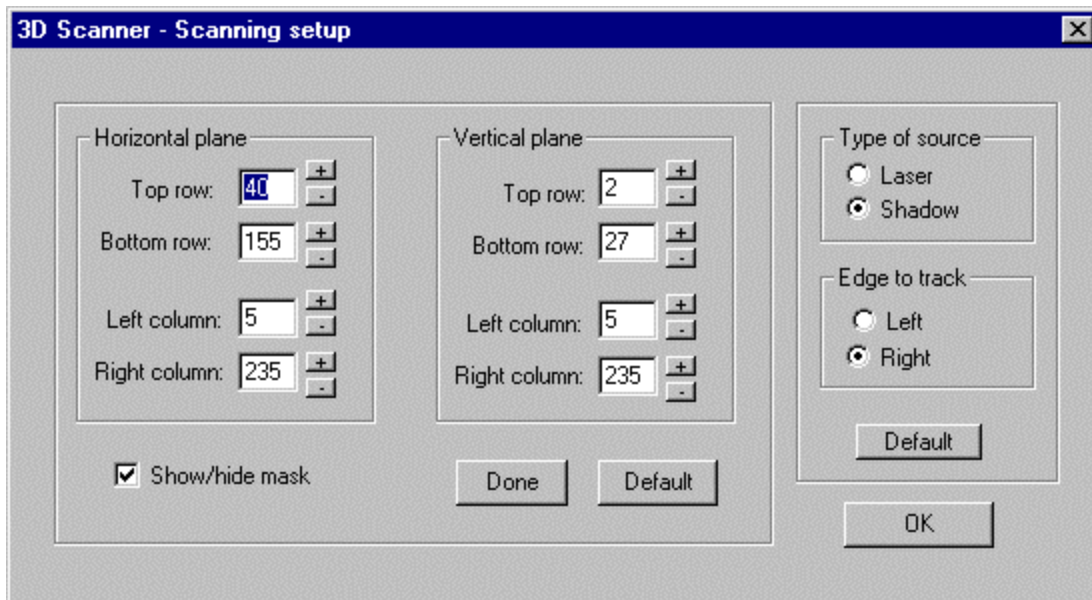
Finestra 3: Calibrazione.



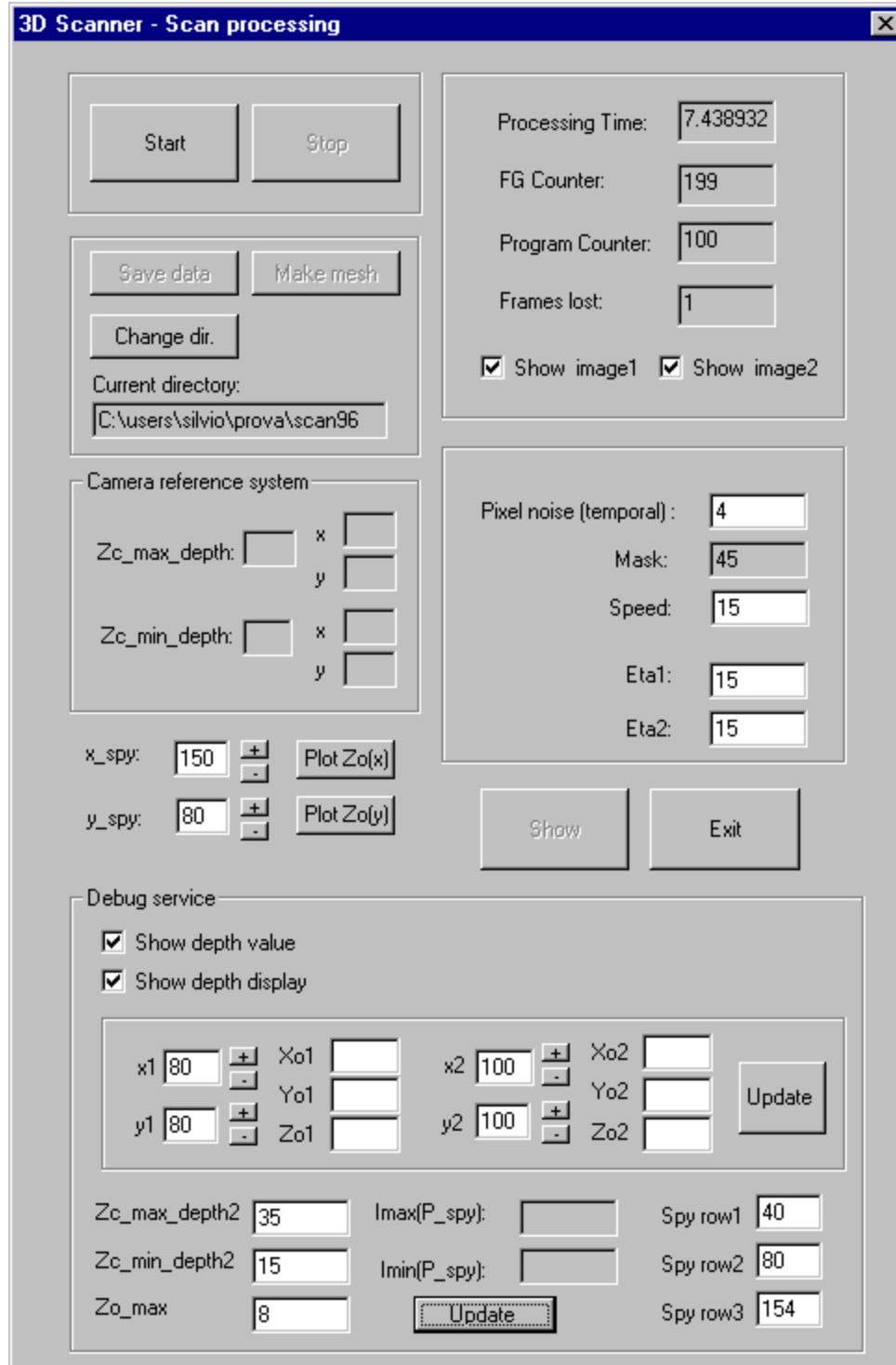
Finestra 4: Determinazione della matrice dei contrasti



Finestra 5: Aquisizione di sequenze d'immagini



Finestra 6: Impostazione per la scansione.



Finestra 7: Scansione dello scenario

Nel successivo paragrafo verranno presi in esame le principali caratteristiche dell'implementazione in tempo reale senza però mai scendere nei dettagli tecnici del codice. Verranno invece analizzati quegli aspetti originali che distinguono tale versione da quella in tempo distribuito.

## 6.2 Principali caratteristiche dell'implementazione in tempo reale

Il metodo a luce debolmente strutturata si adatta con naturalezza ad essere implementato in tempo reale, superando così le principali limitazioni presenti nella versione in tempo distribuito. I motivi vanno ricercati essenzialmente nelle due seguenti caratteristiche:

- L'elaborazione spaziale è locale. In altre parole l'elaborazione della singola immagine, istante per istante, coinvolge un intorno limitato del pixel di lavoro. Ciò rappresenta sicuramente un vantaggio in termini di velocità computazionale.
- L'elaborazione temporale richiede solo l'immagine corrente e quella all'istante (frame) precedente. Ciò evita di dover allocare una quantità considerevole di memoria.

I suddetti punti verranno giustificati richiamando e approfondendo quanto detto nel capitolo II. Per fissare le idee verrà presa in considerazione la versione del metodo con il solo piano orizzontale e verrà supposta la direzione del movimento dell'ombra da sinistra verso destra.

La localizzazione spaziale (cfr. § 2.6) consiste nel determinare istante per istante la posizione di uno dei due bordi (per esempio quello destro) dell'ombra sulle linee di localizzazione spaziale superiore e inferiore (individuate dalle ordinate  $y_{top}$  e  $y_{bot}$ , rispettivamente). Per fare ciò si localizzano le cosiddette ascisse di attraversamento  $x_{op}$  e  $x_{bot}$  definite dalle condizioni (2.6) del capitolo II:

$$\begin{cases} J(x-1, y_{top}, \mathbf{t}) < 0 \\ J(x, y_{top}, \mathbf{t}) \geq 0 \end{cases} \Rightarrow x_{top} \qquad \begin{cases} J(x-1, y_{bot}, \mathbf{t}) < 0 \\ J(x, y_{bot}, \mathbf{t}) \geq 0 \end{cases} \Rightarrow x_{bot} \qquad (2.6)$$

Pertanto, dal punto di vista algoritmico, trovare l'ascissa di attraversamento in una determinata immagine (quindi per un fissato istante  $\tau$ ) significa dover verificare per ogni pixel appartenente alle due linee di localizzazione spaziale se la condizione (2.6) e' soddisfatta. Tale verifica e' immediata essendo richiesti solo il pixel di lavoro e quello adiacente.

Lo scopo della localizzazione temporale (cfr. figura 6.1) e' quello di determinare il tempo d'ombra  $\tau_s$  ovvero l'istante di tempo in cui uno dei due lati dell'ombra (per esempio quello destro) sta passando attraverso un fissato pixel di lavoro  $\underline{x}_c=(x_c, y_c)$ . Per individuare  $\tau_s$  occorre verificare in quale istante di tempo  $\tau$  (ossia in quale frame) la condizione (2.4) e' verificata:

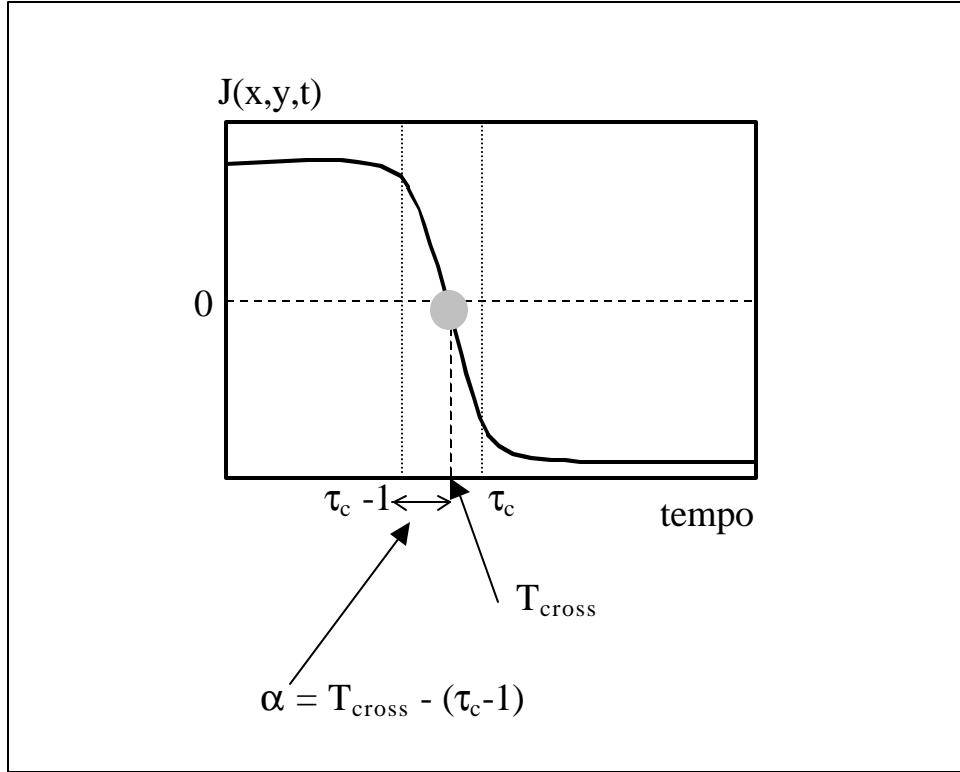
$$\begin{cases} J(x_c, y_c, \mathbf{t}-1) \geq 0 \\ J(x_c, y_c, \mathbf{t}) < 0 \end{cases} \Rightarrow \mathbf{t} \quad (2.4)$$

In realta', contrariamente alla versione in tempo distribuito, ora non e' piu' necessario memorizzare i tempi d'ombra per ciascun pixel dell'immagine. In altre parole, il tempo d'ombra perde di significato dal momento che non serve piu' come collegamento tra la condizione (2.6) e la (2.4) per la determinazione della corrispondente linea di terra  $\lambda_h(t)$ .

Si supponga infatti che ad un certo istante  $\tau_c$  il frame grabber fornisca un certa immagine. Le operazioni che l'algoritmo in tempo reale affettua sull'immagine consistono nel trovare le ascisse di attraversamento  $x_{top}$  e  $x_{bot}$  e localizzare quali sono i pixel appartenenti all'immagine che soddisfano la condizione temporale (si confronti la figura 6.1):

$$\begin{cases} J(x, y, \mathbf{t}-1) \geq 0 \\ J(x, y, \mathbf{t}) < 0 \end{cases} \quad (6.1)$$

Detto  $\underline{X}_{\tau_c} = \{\underline{x}_{\tau_c1}, \underline{x}_{\tau_c2} \dots, \underline{x}_{\tau_cn}\}$  l'insieme di tali pixel, si puo' calcolare subito qual'e la retta di terra  $\lambda_h(t)$  associata all'inseme  $\underline{X}_{\tau_c}$ , essendo noti i relativi  $x_{top}$  e  $x_{bot}$ . In definitiva immagine per immagine si e' cosi' in grado di valutare la posizione del piano d'ombra relativamente ai pixel che soddisfano la (6.1).



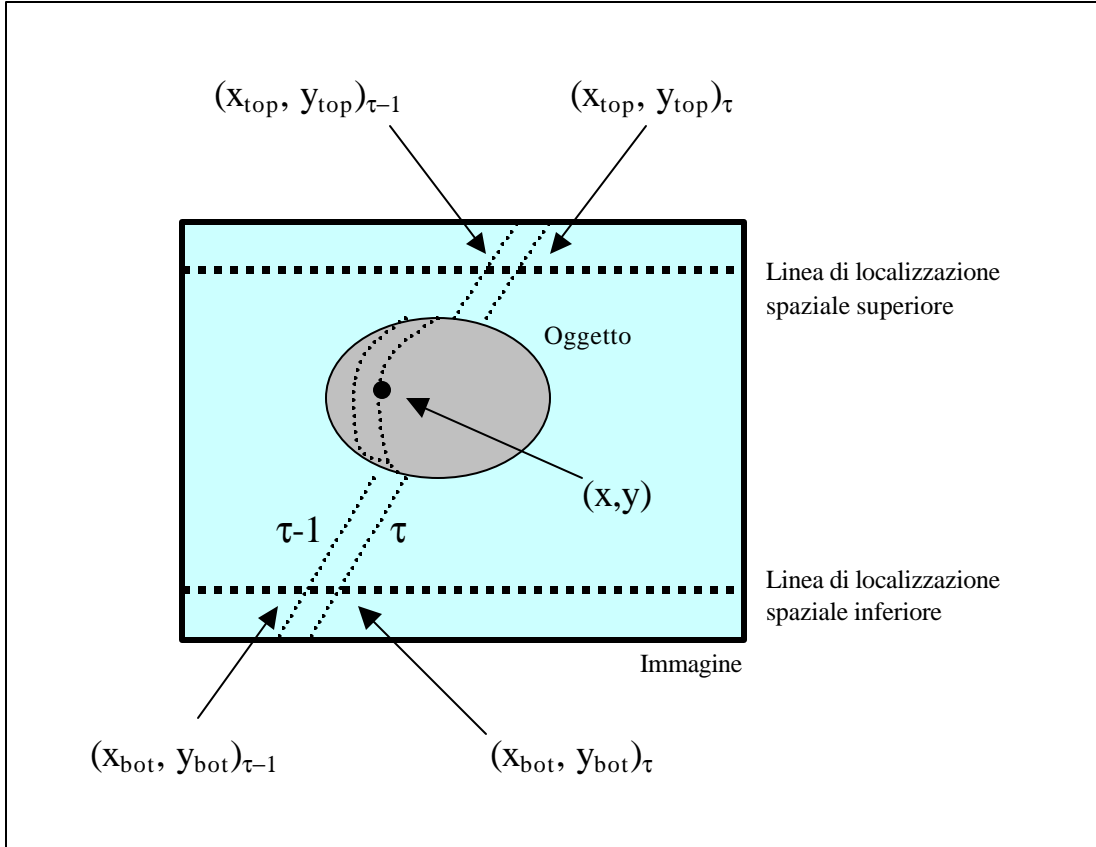
**Figura 6.1: Andamento temporale della luminosita' normalizzata  $J(x,y,t)$  per un determinato pixel.** Fissato un certo frame  $\tau_c$ , la localizzazione temporale consiste nell'individuare tutti quei pixel che soddisfano la (6.1).  $T_{\text{cross}}$  rappresenta l'istante di tempo in cui la luminosita' normalizzata  $J(x,y,t)$  attraversa realmente lo zero.

In effetti, per ciascuno dei pixel dell'insieme  $X_{\tau_c}$  si vuole determinare in quale istante di tempo la corrispondente funzione di luminosita' normalizzata  $J(x,y,t)$  attraversi lo zero. Detto  $T_{\text{cross}}$  tale tempo, esso sara' sicuramente compreso tra  $\tau_c - 1$  e  $\tau_c$ . Si confronti la figura 6.1 per visualizzare la grandezza in questione. Lo scopo infatti e' quello di valutare le ascisse di attraversamento  $x_{\text{top}}$  e  $x_{\text{bot}}$  in corrispondenza dell'istante  $T_{\text{cross}}$  (invece che  $\tau_c$ ). Con riferimento al paragrafo 2.6, si introduce la grandezza  $\alpha$  definita come:

$$\alpha = T_{\text{cross}} - (\tau_c - 1) \quad (6.2)$$

Indicando con  $x_{\text{top}}(\tau_c)$ ,  $x_{\text{bot}}(\tau_c)$  e  $x_{\text{top}}(\tau_c - 1)$ ,  $x_{\text{bot}}(\tau_c - 1)$  le ascisse di attraversamento valutate per l'immagine corrente e per l'immagine precedente (cfr. figura 6.2), le quantita'  $x_{\text{top}}(T_{\text{cross}})$  e  $x_{\text{bot}}(T_{\text{cross}})$  si calcolano tramite interpolazione lineare:

$$\begin{aligned} x_{\text{top}}(T_{\text{cross}}) &= (1-\alpha) x_{\text{top}}(\tau_c - 1) + \alpha x_{\text{top}}(\tau_c) \\ x_{\text{bot}}(T_{\text{cross}}) &= (1-\alpha) x_{\text{bot}}(\tau_c - 1) + \alpha x_{\text{bot}}(\tau_c) \end{aligned} \quad (6.3)$$



**Figura 6.2: Individuazione delle ascisse di attraversamento nell'istante corrente e precedente.** Per individuare le ascisse  $x_{top}$  e  $x_{bot}$  all'istante  $T_{cross}$  e' necessario conoscere  $x_{top}$  e  $x_{bot}$  relativi al frame corrente e a quello precedente.

Le ascisse  $x_{op}(T_{cross})$  e  $x_{op}(T_{cross})$  danno luogo ai pixel  $\underline{x}_{top}(T_{cross})=(x_{top}(T_{cross}),y_{top})$  e  $\underline{x}_{bot}(T_{cross})=(x_{bot}(T_{cross}),y_{bot})$ , che vengono indicati come *pixel di attraversamento*. Da questi e' immediato passare ai corrispondenti punti  $\bar{x}_{top}$  e  $\bar{x}_{bot}$  sul piano immagine (cfr. equazione 4.6) :

$$\bar{x}_{top}(T_{cross}) = \begin{bmatrix} \frac{x_{top}(T_{cross}) - c_x}{f} \\ \frac{y_{top} - c_y}{f} \\ 1 \end{bmatrix}, \quad \bar{x}_{bot}(T_{cross}) = \begin{bmatrix} \frac{x_{bot}(T_{cross}) - c_x}{f} \\ \frac{y_{bot} - c_y}{f} \\ 1 \end{bmatrix}$$

avendo ipotizzato trascurabile il fattore di distorsione radiale  $k$ . Se invece si considera presente tale fattore, non vi e' piu' una corrispondenza diretta tra le componenti delle due quantita'.

Noti  $\bar{x}_{top}$  e  $\bar{x}_{bot}$ , la retta di terra  $\bar{I}_h(T_{cross})$  e' calcolabile (cfr. proprieta' 3 del capitolo III) tramite la:

$$\bar{I}_h(T_{cross}) = \bar{x}_{top}(T_{cross}) \times \bar{x}_{bot}(T_{cross}) \quad (6.4)$$

da cui si ottiene il piano ombra (cfr. eq. 3.1):

$$\bar{w}_s(T_{cross}) = \bar{w}_h + \alpha_s \bar{I}_h(T_{cross}) \quad (3.1)$$

con  $\alpha_s$  pari a:

$$\alpha_s = \frac{1 - \langle \bar{w}_h, \bar{X}_s \rangle}{\langle \bar{I}_h, \bar{X}_s \rangle}$$

Non e' superfluo notare che i pixel appartenenti all'insieme  $\underline{X}_{\tau_c} = \{x_{\tau_c1}, x_{\tau_c2} \dots, x_{\tau_cn}\}$  pur essendo associati tutti alle medesime ascisse di attraversamento  $x_{top}(\tau_c)$  e  $x_{bot}(\tau_c)$ , presentano piani ombra differenti. Infatti il tempo  $T_{cross}$  cambia per i vari pixel di  $\underline{X}_{\tau_c}$  e di conseguenza anche i vettori  $\bar{x}_{top}(T_{cross})$  e  $\bar{x}_{bot}(T_{cross})$ , la linea di terra  $\bar{I}_h(T_{cross})$  e il piano ombra  $\bar{w}_s(T_{cross})$ .

Si consideri il pixel  $\underline{x}_{\tau_{ci}} = (x_{\tau_{ci}}, y_{\tau_{ci}})$ , per fissare le idee. Il corrispondente punto sul piano immagine e' dato dal vettore (cfr. eq. 4.6):

$$\bar{x}_{\tau_{ci}} = \begin{bmatrix} \frac{x_{\tau_{ci}} - c_x}{f} \\ \frac{y_{\tau_{ci}} - c_y}{f} \\ 1 \end{bmatrix}, \quad (4.6)$$

Quindi le coordinate del relativo punto sullo scenario saranno date da (cfr. eq 3.6):

$$\bar{X}_{tci} = \frac{\bar{X}_{tci}}{\langle \bar{\mathbf{w}}_s, \bar{X}_{tci} \rangle} \quad (3.6)$$

In definitiva data una certa immagine, l'algoritmo procede nel seguente modo:

- Valutazione delle ascisse di attraversamento  $x_{top}(\tau_c)$  e  $x_{bot}(\tau_c)$  nell'immagine corrente tramite la (4.2).
- Valutazione delle ascisse di attraversamento  $x_{top}(\tau_c - 1)$  e  $x_{bot}(\tau_c - 1)$  nell'immagine precedente tramite la (4.2).
- Determinazione dei pixel soddisfacenti la condizione temporale (6.1) adoperando l'immagine corrente e quella precedente. In altri termini si individua l'insieme  $\underline{X}_{\tau_c} = \{x_{\tau_c1}, x_{\tau_c2}, \dots, x_{\tau_cn}\}$ .
- Valutazione del tempo  $T_{cross}$  per ciascun pixel appartenente a  $\underline{X}_{\tau_c}$  tramite la (6.2).
- Valutazione dell'ascisse di attraversamento  $x_{top}(T_{cross})$  e  $x_{bot}(T_{cross})$  per ciascun pixel appartenente a  $\underline{X}_{\tau_c}$  facendo uso della (6.3).
- Valutazione della retta di terra  $\bar{I}_h(T_{cross})$  per ciascun pixel appartenente a  $\underline{X}_{\tau_c}$  facendo uso della (6.4).
- Valutazione del piano d'ombra  $\bar{\mathbf{w}}_s(T_{cross})$  per ciascun pixel appartenente a  $\underline{X}_{\tau_c}$  facendo uso della (3.1).
- Individuazione delle coordinate dei punti sul piano immagine corrispondenti a ciascun pixel appartenente a  $\underline{X}_{\tau_c}$  tramite la (4.6).
- Individuazione delle coordinate dei punti sullo scenario corrispondenti a ciascun pixel appartenente a  $\underline{X}_{\tau_c}$  tramite la (3.6).

Come risulta evidente dai passi di cui sopra, l'individuazione delle coordinate spaziali associate ai pixel scanditi richiede solo l'immagine corrente e quella precedente. Non e' quindi necessario memorizzare tutta la sequenza come nel caso a tempo distribuito. In secondo luogo l'elaborazione spaziale coinvolge un intorno del pixel di lavoro pari a due pixel solamente.

### 6.3 Obiettivi dell'implementazione in tempo reale

La versione in tempo reale deve garantire alcuni requisiti che nella versione in tempo distribuito non avevano senso oppure non potevano essere soddisfatti:

- Non si vuole aver il vincolo di unidirezionalita' dello spostamento dell'ombra proiettata. In tal modo si riducono sensibilmente le limitazioni che l'operatore ha nello spostare la bacchetta sullo scenario e si ha la possibilita' di far ripassare anche piu' volte l'ombra su punti gia'scanditi (si confronti fase di merging nel paragrafo 6.9).
- Si deve tenere in conto la possibilita' che l'operatore sposti la bacchetta con velocita' variabile. Tale parametro gioca un ruolo determinante nella risoluzione e nella qualita' della ricostruzione.
- Si deve tenere in conto la possibilita' che l'operatore, nello spostare la bacchetta, modifichi la sua altezza dal piano di terra. L'effetto e' quello di proiettare un'ombra il cui bordo e' piu' o meno smussato a seconda se la bacchetta e' piu' o meno vicina dalla sorgente luminosa. Anche il tipo di bordo dell'ombra e' un parametro da portare in conto per la corretta elaborazione dell'immagine.

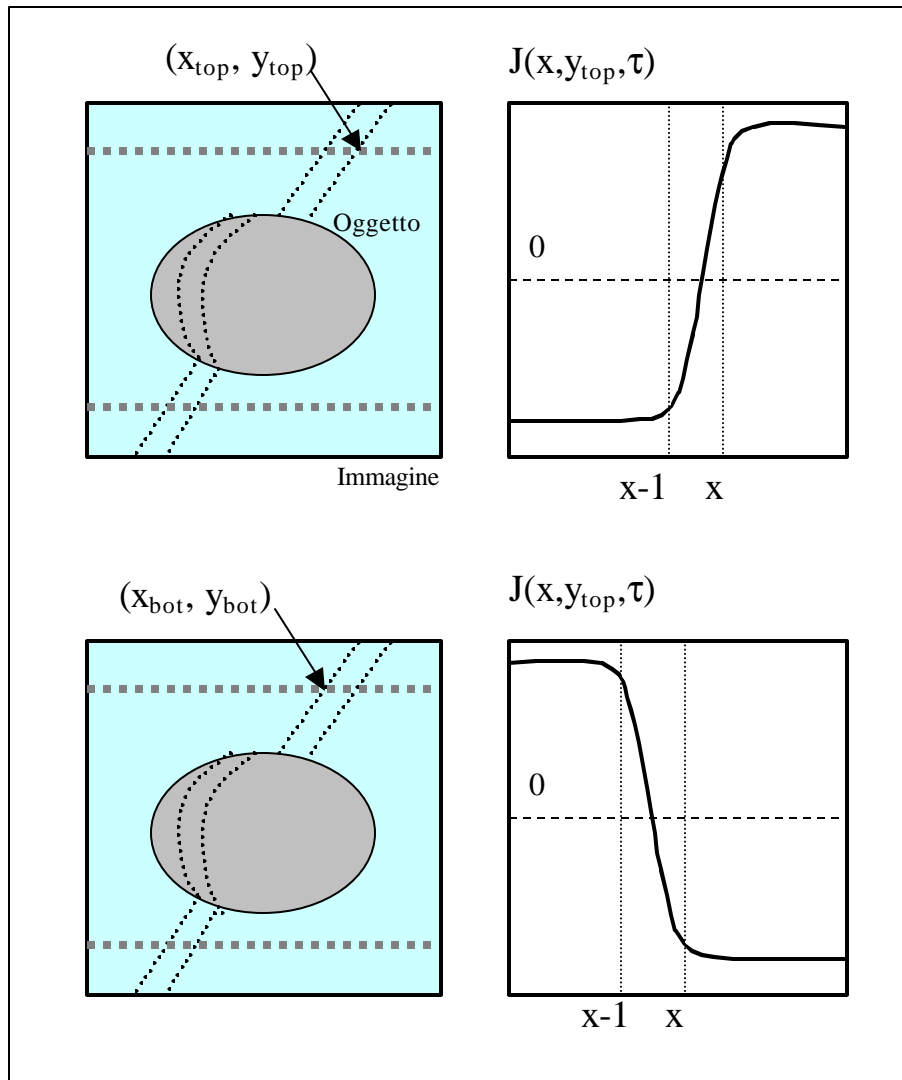
Nei prossimi capitoli verranno considerati in dettaglio tutti questi aspetti.

### 6.4 Condizioni per la direzione dell'ombra

E' noto dal capitolo II che la ascissa di attraversamento (per esempio superiore) e' ottenibile imponendo la condizione (con  $\tau$  fissato):

$$\begin{cases} J(x-1, y_{top}, \mathbf{t}) < 0, \\ J(x, y_{top}, \mathbf{t}) \geq 0 \end{cases} \quad (6.5)$$

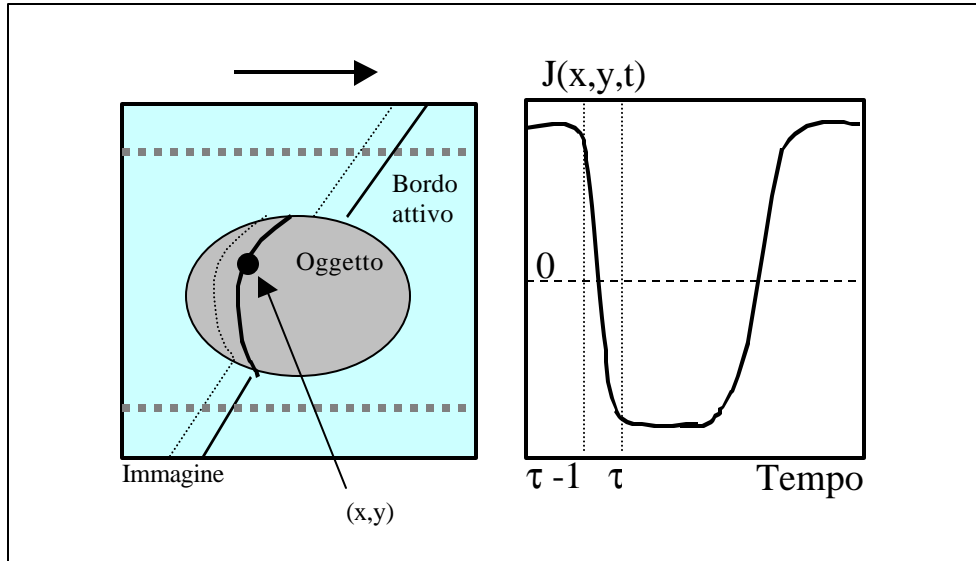
In tal caso si ottiene l'ascissa di un pixel collocato sul bordo destro dell'ombra. Imponendo invece la seguente condizione:



**Figura 6.3: Individuazione dell'ascissa di attraversamento  $x_{top}$ .** La condizione espressa nella (6.5) individua l'ascissa  $x_{op}$  sul bordo destro dell'ombra (figure in alto), laddove la condizione (6.5)' individua l'ascissa  $x_{top}$  sul bordo sinistro (figure in basso).

$$\begin{cases} J(x - 1, y_{top}, \mathbf{t}) \geq 0, \\ J(x, y_{top}, \mathbf{t}) < 0 \end{cases} \quad (6.5)'$$

si ottiene l'ascissa di un pixel collocato sul bordo sinistro dell'ombra. I due casi sono illustrati nella figura 6.3. Nella parte destra della figura compare l'andamento di  $J(x,y,t)$  lungo  $x$  nell'istante  $\tau$ . E' utile osservare che tali risultati non dipendono dalla direzione del movimento dell'ombra.



**Figura 6.4: Individuazione dei pixel che soddisfano la condizione temporale.** Ipotizzando che l'ombra all'istante  $\tau$ , si stia spostando da sinistra verso destra e che si adoperi la condizione temporale (6.6), il bordo attivo dell'ombra e' quello di destra.

Per valutare quali pixel dello scenario sono oscurati dall'ombra in un determinato istante  $\tau$  occorre imporre la solita condizione temporale (cfr. equazione (6.1)):

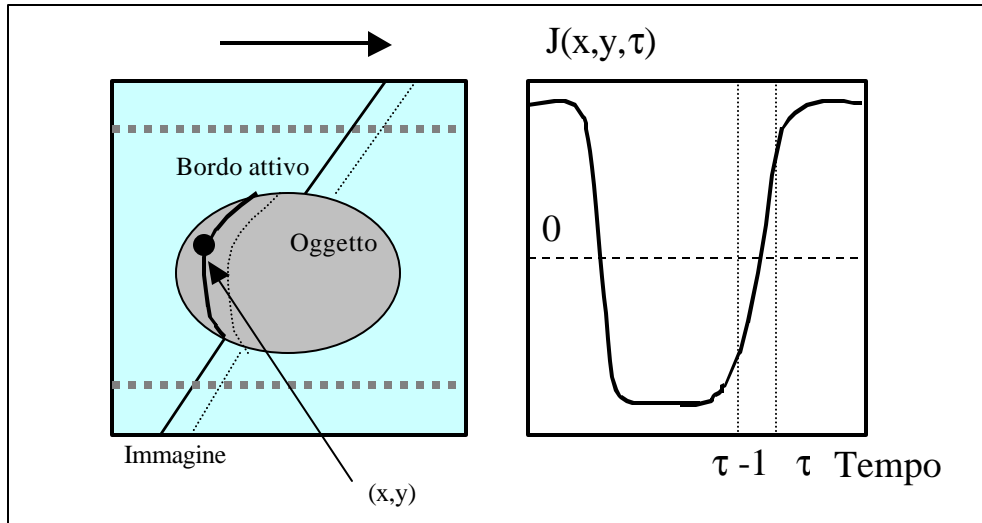
$$\begin{cases} J(x, y, \mathbf{t}-1) \geq 0 \\ J(x, y, \mathbf{t}) < 0 \end{cases} \quad (6.6)$$

Se si ipotizza che la direzione dell'ombra all'istante  $\tau$  e' da sinistra verso destra, la condizione (6.6) individua i pixel sul bordo destro dell'ombra (che viene detto *bordo attivo*).

La figura 6.4 di destra illustra il diagramma temporale della  $J(x,y,t)$  per un determinato pixel  $(x,y)$  appartenente al bordo attivo. Nella parte di sinistra viene mostrata la posizione del bordo attivo.

Se invece, a parita' di movimento dell'ombra, viene imposta la seguente condizione temporale:

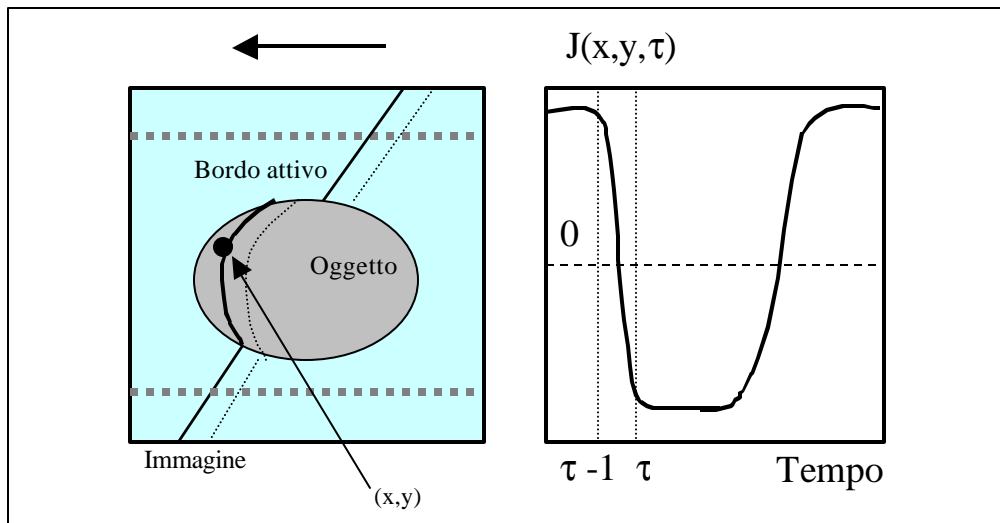
$$\begin{cases} J(x, y, \mathbf{t}-1) < 0 \\ J(x, y, \mathbf{t}) \geq 0 \end{cases} \quad (6.7)$$



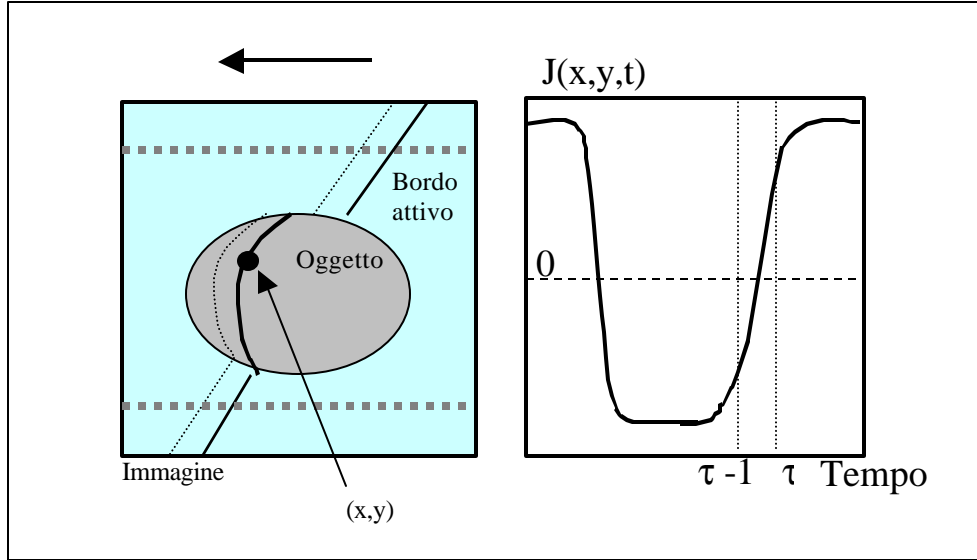
**Figura 6.5: Individuazione dei pixel che soddisfano la condizione temporale.** Ipotizzando che l'ombra, all'istante  $\tau$ , si stia spostando da sinistra verso destra e che si adoperi la condizione temporale (6.7), il bordo attivo dell'ombra e' quello di sinistra.

il lato attivo dell'ombra diventa quello di sinistra, come appare evidente nella figura 6.5.

Nel caso in cui l'ombra si muove nella direzione opposta, la situazione e' perfettamente duale. Imponendo la condizione (6.6) il bordo attivo e' quello di sinistra laddove imponendo la condizione (6.7) il bordo attivo diventa quello di destra. Le figure 6.6 e 6.7 illustrano quest'ultimo caso.



**Figura 6.6: Individuazione dei pixel che soddisfano la condizione temporale.** Ipotizzando che l'ombra, all'istante  $\tau$ , si stia spostando da destra verso sinistra e che si adoperi la condizione temporale (6.6), il bordo attivo dell'ombra e' quello di sinistra.



**Figura 6.7: Individuazione dei pixel che soddisfano la condizione temporale.** Ipotizzando che l'ombra all'istante  $\tau$ , si stia spostando da destra verso sinistra e che si adoperi la condizione temporale (6.7), il bordo attivo dell'ombra e' quello di destra.

E' possibile riepilogare nelle seguenti tabelle i risultati ottenuti:

Direzione dell'ombra	Condizione temporale	Bordo attivo
⊗	$\begin{cases} J(x, y, t-1) \geq 0 & (6.6) \\ J(x, y, t) < 0 \end{cases}$	Destro
⊗	$\begin{cases} J(x, y, t-1) < 0 & (6.7) \\ J(x, y, t) \geq 0 \end{cases}$	Sinistro
↖	$\begin{cases} J(x, y, t-1) \geq 0 & (6.6) \\ J(x, y, t) < 0 \end{cases}$	Sinistro
↖	$\begin{cases} J(x, y, t-1) < 0 & (6.7) \\ J(x, y, t) \geq 0 \end{cases}$	Destro

**Tabella 6.1: Individuazione del bordo attivo**

Direzione dell'ombra	Condizione spaziale	Posizione dell'ascissa
⊗ ↖	$\begin{cases} J(x-1, y_{top}, t) \geq 0 & (6.5) \\ J(x, y_{top}, t) < 0 \end{cases}$	A destra
⊗ ↖	$\begin{cases} J(x-1, y_{top}, t) < 0 & (6.5) \\ J(x, y_{top}, t) \geq 0 \end{cases}$	A sinistra

**Tabella 6.2: Individuazione dell'ascissa di attraversamento.**

Una condizione necessaria per il corretto funzionamento del metodo e' che vi sia, immagine per immagine, consistenza tra l'ascissa di attraversamento e il bordo attivo. In altri termini e' necessario che, se l'ascissa di attraversamento viene individuata sul bordo destro dell'ombra, il bordo attivo deve essere quello di destra. Analogo discorso vale per il lato sinistro. Se cio' non dovesse accadere verrebbe commesso un errore nella valutazione del piano ombra direttamente legato allo spessore dell'ombra proiettata. Nella maggior parte dei casi cio' si trasferisce in un errore assolutamente non accettabile nella ricostruzione della superficie di lavoro.

Supponendo un movimento unidirezionale dell'ombra e imponendo le condizioni (6.5) e (6.6), oppure (6.5)' e (6.7), l'ipotesi di cui sopra e' sicuramente soddisfatta. Il problema si pone rimuovendo il vincolo di unidirezionalita'. Nel momento in cui si impongono le condizioni (6.5) e (6.6), per esempio, dalle tabelle 6.1 e 6.2 si deduce che non e' affatto detto che la consistenza tra l'ascissa di attraversamento e il bordo attivo possa essere garantita. E' necessario quindi dover imporre delle condizioni supplementari.

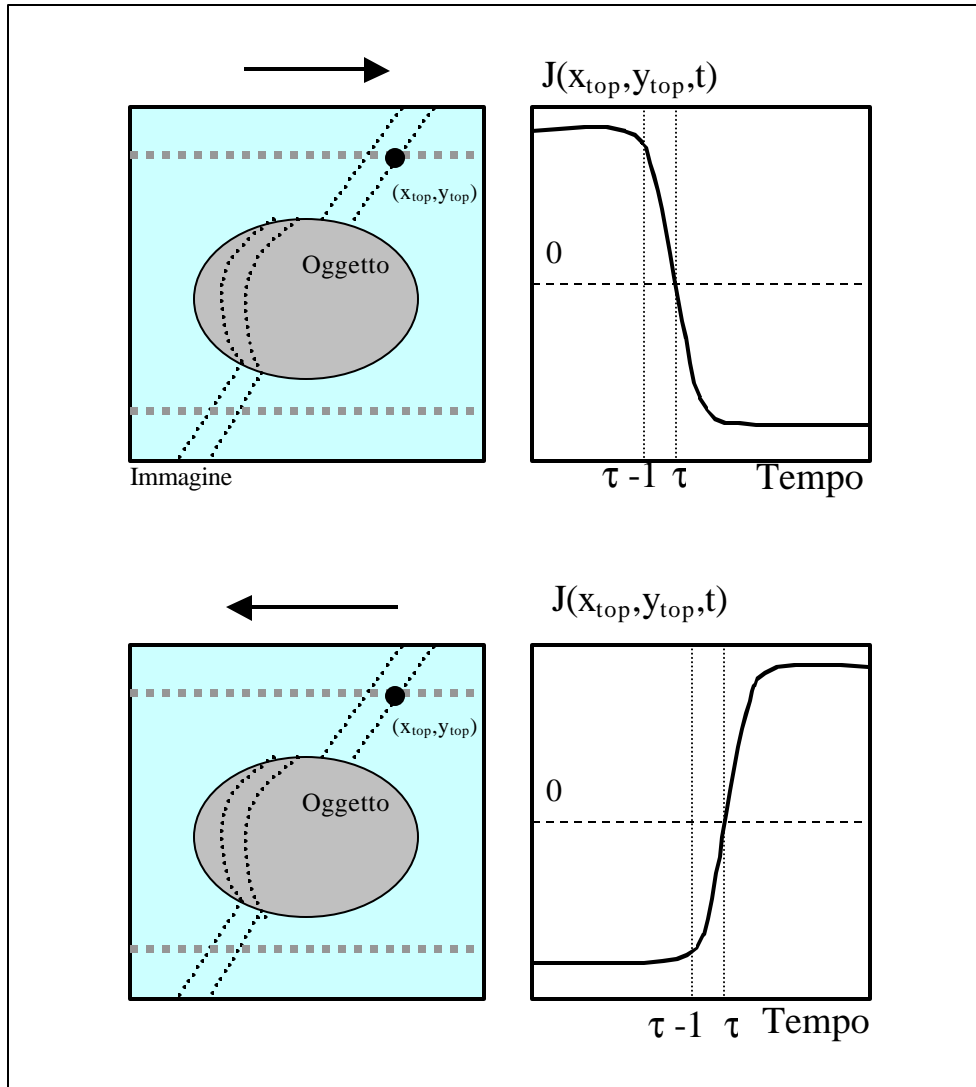
Un modo per affrontare il problema puo' essere quello di ricavare, per ciascun istante di tempo  $\tau$ , l'informazione sulla direzione del moto dell'ombra. Si supponga, per fissare le idee, di volere come bordo attivo quello di destra. Di conseguenza le ascisse di attraversamento  $x_{top}(\tau)$  e  $x_{bot}(\tau)$  dovranno anch'esse essere individuate a destra. Cio' impone di dover usare la condizione spaziale (6.5).

Una volta trovate  $x_{top}(\tau)$  e  $x_{bot}(\tau)$ , si verifica quali delle seguenti disuguaglianze sono soddisfatte (cfr. figura 6.8):

$$\left\{ \begin{array}{l} J(x_{top}, y_{top}, \mathbf{t}-1) > J(x_{top}, y_{top}, \mathbf{t}) \\ J(x_{bot}, y_{bot}, \mathbf{t}-1) > J(x_{bot}, y_{bot}, \mathbf{t}) \end{array} \right. \quad \begin{array}{l} (6.8) \\ (6.9) \end{array}$$

$$\left\{ \begin{array}{l} J(x_{top}, y_{top}, \mathbf{t}-1) < J(x_{top}, y_{top}, \mathbf{t}) \\ J(x_{bot}, y_{bot}, \mathbf{t}-1) < J(x_{bot}, y_{bot}, \mathbf{t}) \end{array} \right. \quad \begin{array}{l} (6.10) \\ (6.11) \end{array}$$

A partire da queste informazioni si puo' dedurre la direzione dell'ombra all'istante  $\tau$ . Se infatti vengono verificate la (6.8) e la (6.9) significa che la luminosita' dei pixel  $x_{top}$  e  $x_{bot}$  sta diminuendo, il che implica che l'ombra si sta spostando da sinistra verso destra.



**Figura 6.8: Individuazione della direzione dell'ombra.** Osservano la luminosita' dei pixel  $\underline{x}_{top}$  e  $\underline{x}_{bot}$  agli istanti  $\tau$  e  $\tau-1$  (diagrammi a destra) si vede quale delle disuguaglianze (6.8)+(6.11) sono verificate. A partire da queste informazioni si ricava la direzione dello spostamento dell'ombra all'istante  $\tau$

Se invece vengono verificate la (6.10) e la (6.11) significa che la luminosita' dei pixel  $\underline{x}_{top}$  e  $\underline{x}_{bot}$  sta aumentando, il che implica che l'ombra si sta spostando da destra verso sinistra. Se infine sono verificate la (6.8) e la (6.11) oppure la (6.9) e la (6.12), vengono fornite informazioni discordanti. Questo caso si puo' presentare per particolari movimenti della bacchetta in cui lo spessore dell'ombra proiettata si restringe in corrispondenza della linea superiore e si allarga in corrispondenza di quella inferiore. In presenza di informazioni discordanti, nulla si puo' dire sulla direzione dell'ombra e pertanto l'immagine non viene processata. La tabella 6.3 riassume questi ragionamenti.

Disuguaglianze verificate	Direzione dell'ombra
$J(x_{top}, y_{top}, t-1) > J(x_{top}, y_{top}, t)$ $J(x_{bot}, y_{bot}, t-1) > J(x_{bot}, y_{bot}, t)$	Ⓜ
$J(x_{top}, y_{top}, t-1) < J(x_{top}, y_{top}, t)$ $J(x_{bot}, y_{bot}, t-1) < J(x_{bot}, y_{bot}, t)$	¬
$J(x_{top}, y_{top}, t-1) > J(x_{top}, y_{top}, t)$ $J(x_{bot}, y_{bot}, t-1) < J(x_{bot}, y_{bot}, t)$	?
$J(x_{bot}, y_{bot}, t-1) < J(x_{bot}, y_{bot}, t)$ $J(x_{bot}, y_{bot}, t-1) > J(x_{bot}, y_{bot}, t)$	?

**Tabella 6.3:** Individuazione della direzione dell'ombra in funzione delle coppie di disuguaglianze verificate.

Nota la direzione del movimento dell'ombra, la consistenza tra l'ascissa di attraversamento e il bordo attivo puo' essere garantita combinando opportunamente le condizioni temporali (6.6) e (6.7) con le informazioni sulla direzione. Pertanto l'insieme dei pixel, che al generico istante  $\tau$  appartengono al bordo attivo di destra, viene selezionato imponendo le seguenti condizioni:

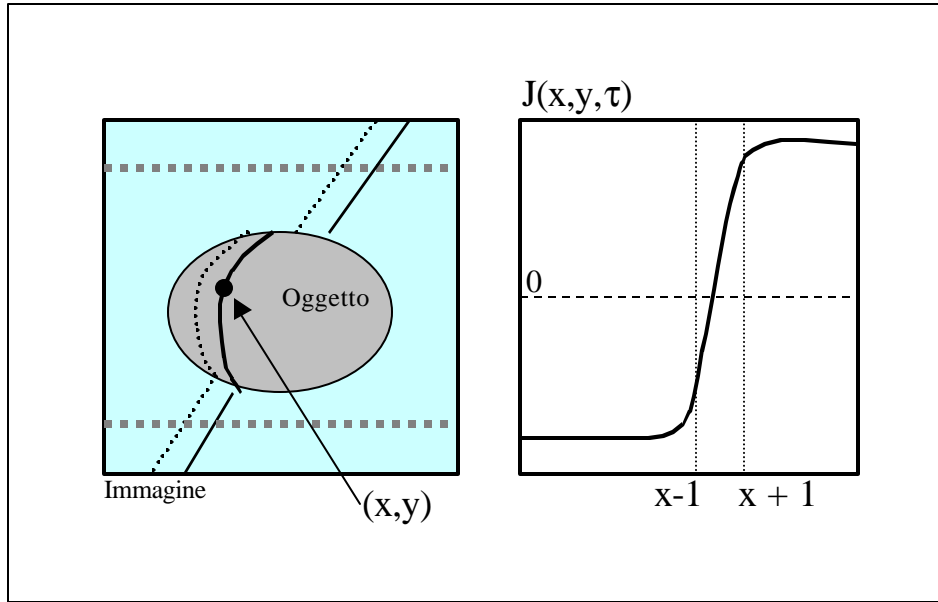
$$\begin{cases} J(x, y, t-1) \geq 0 \\ J(x, y, t) < 0 \end{cases} \quad \text{se la direzione dell'ombra va da sinistra a destra}$$

*oppure*

$$\begin{cases} J(x, y, t-1) < 0 \\ J(x, y, t) \geq 0 \end{cases} \quad \text{se la direzione dell'ombra va da destra a sinistra}$$

*per ogni  $x, y$  appartenenti all'immagine.*

Puo' essere utile introdurre un'ulteriore condizione per verificare se il pixel in esame appartiene o meno al bordo attivo di destra. A prescindere dalla direzione dell'ombra, la funzione di luminosita' normalizzata associata ai pixel situati sul bordo



**Figura 6.9: Condizione spaziale per il bordo destro dell'ombra.** Se il pixel in esame soddisfa la condizione (6.12) significa che nel suo intorno (lungo la direzione x) la luminosita' sta crescendo. Pertanto tale pixel potrebbe effettivamente essere collocato sul bordo destro dell'ombra.

destro dell'ombra deve, almeno in teoria, soddisfare la seguente condizione (si confronti la figura 6.9 che visualizza la grandezza in gioco):

$$J(x-1, y, \mathbf{t}) > J(x+1, y, \mathbf{t}) \quad (6.12)$$

Si noti tuttavia che la (6.12) costituisce una condizione necessaria ma non sufficiente. Infatti da sola non basta per individuare i pixel appartenenti al bordo attivo. Bisogna inoltre considerare che, a causa delle fluttuazioni della  $J(x, y, \mathbf{t})$  dovute al rumore sul pixel, tale condizione non sempre viene verificata in modo corretto quando invece dovrebbe esserlo. La situazione diventa particolarmente critica in corrispondenza di quei pixel caratterizzati da un basso valore di contrasto e che tuttavia non vengono scartati dalla maschera definita dalla  $I_{\text{mask}}(x, y)$  (crf. § 2.3).

Pertanto, per la selezione dell'insieme di pixel che al generico istante  $\tau$  appartengono al bordo attivo di destra, si impongono le seguenti condizioni:

$$\begin{cases} J(x, y, \mathbf{t}-1) \geq 0 \\ J(x, y, \mathbf{t}) < 0 \end{cases} \quad \begin{array}{l} \text{se la direzione dell'ombra va da sinistra a destra e} \\ \text{se vale } J(x-1, y, \mathbf{t}) > J(x+1, y, \mathbf{t}) \end{array}$$

*oppure*

$$\begin{cases} J(x, y, \mathbf{t}-1) < 0 \\ J(x, y, \mathbf{t}) \geq 0 \end{cases} \quad \begin{array}{l} \text{se la direzione dell'ombra va da destra a sinistra e} \\ \text{se vale } J(x-1, y, \mathbf{t}) > J(x+1, y, \mathbf{t}) \end{array}$$

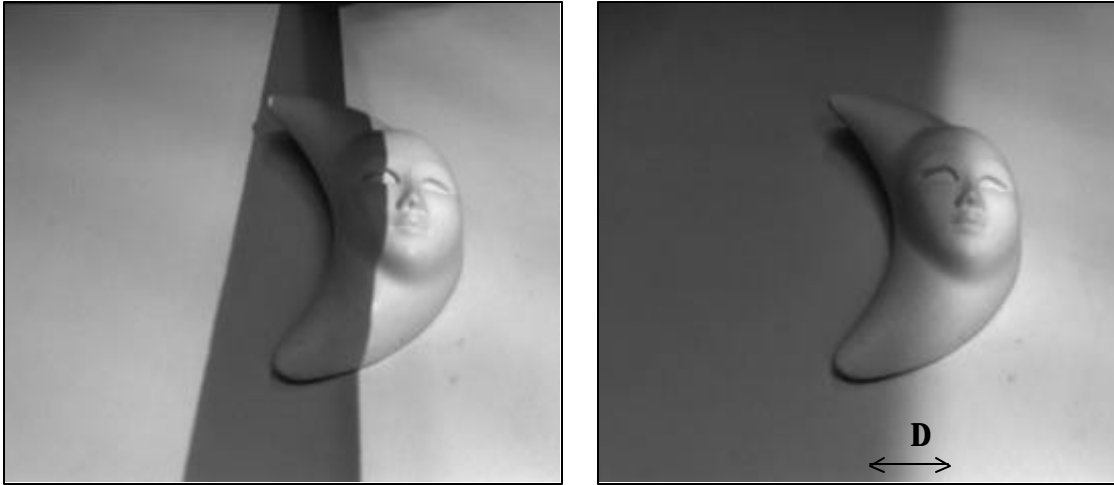
*per ogni  $x, y$  appartenenti all'immagine. (6.13)*

Condizioni analoghe possono essere scritte nel caso si scelga come bordo attivo quello di sinistra.

Prima di concludere questo paragrafo, puo' essere utile generalizzare la definizione dell'insieme  $\underline{X}_\tau$ . Esso puo' essere ridefinito come l'insieme di pixel che al tempo  $\tau$  vengono selezionati, tra tutti quelli appartenenti all'immagine, per la successiva elaborazione. In altre parole tali pixel sono quelli per i quali verranno calcolate le coordinate spaziali attraverso la procedura di triangolazione. I pixel che soddisfano le condizioni (6.13) sono dei buoni candidati da inserire nell'insieme  $\underline{X}_\tau$ . Come si vedra' tra poco, per potervici far parte occorre che soddisfino altre condizioni.

## 6.5 Condizioni sul bordo dell'ombra

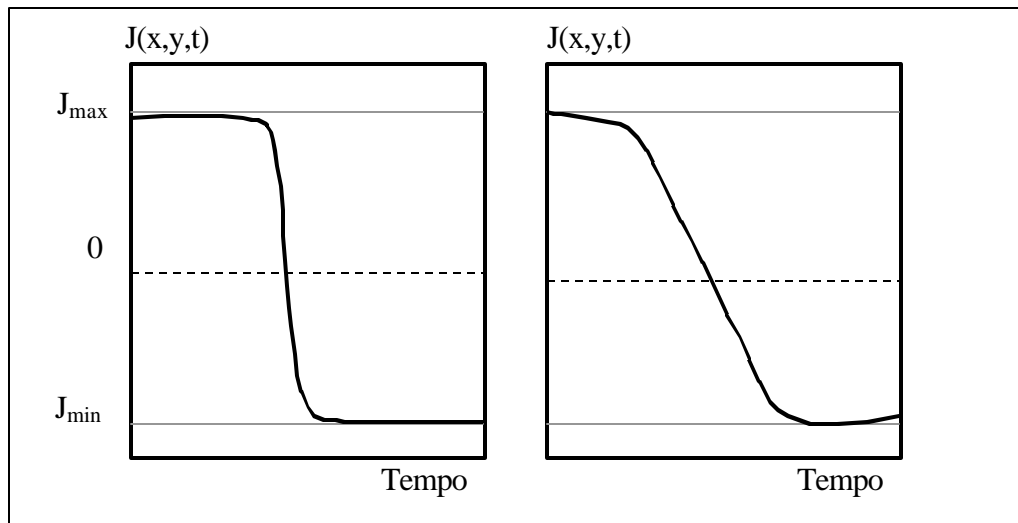
L'ombra proiettata sullo scenario si presenta come una striscia di un certo spessore, funzione della larghezza della bacchetta. Il passaggio dalla zona ombrata alla zona priva di ombra ovviamente non e' improvviso. In altri termini il bordo della striscia non e' perfettamente brusco ma esiste una certa zona di transizione. Tale zona di transizione e' funzione della posizione relativa della bacchetta tra il piano di terra e la sorgente luminosa. Quanto piu' la bacchetta sara' vicina alla sorgente luminosa tanto piu'



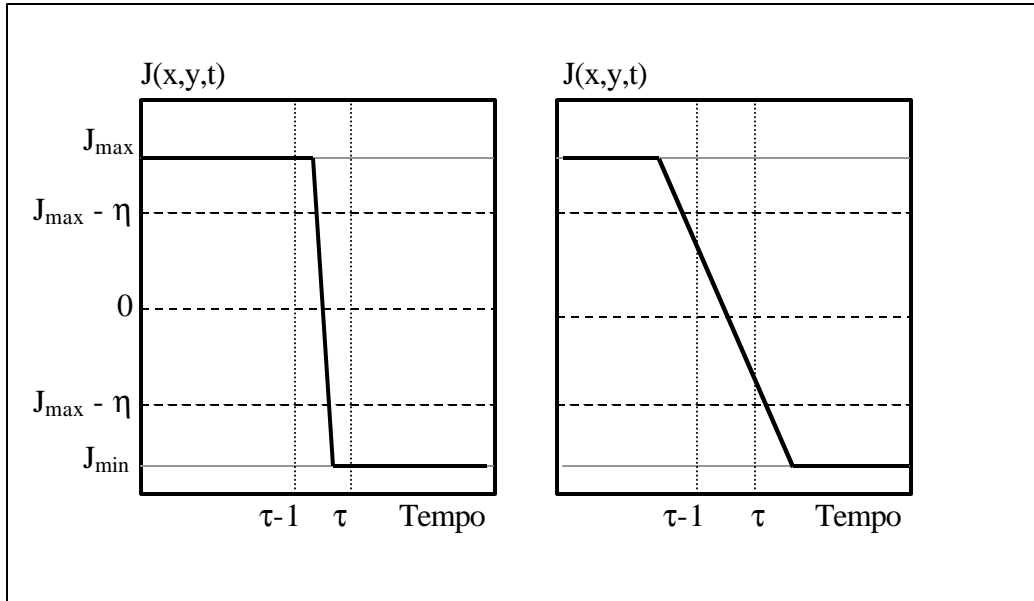
**Figura 6.10: Tipo di bordo dell'ombra proiettata.** Nella figura di sinistra viene proiettata un'ombra il cui bordo è molto ben definito. Ciò significa che la bacchetta è vicina al piano di terra. A destra invece il bordo dell'ombra presenta una zona di transizione molto ampia ( $\Delta$ ). Ciò significa che la bacchetta è piuttosto vicina alla sorgente luminosa

il bordo della striscia d'ombra sarà smussato. Tale fenomeno è direttamente osservabile nell'esempio di figura 6.10.

Il fenomeno in questione si ripercuote ovviamente sul profilo della funzione  $J(x,y,t)$  nel tempo. Un determinato pixel, coinvolto nel passaggio dell'ombra, vede una transizione brusca o meno del suo valore di luminosità a seconda se il bordo dell'ombra è brusco o meno. Si confronti la figura 6.11.



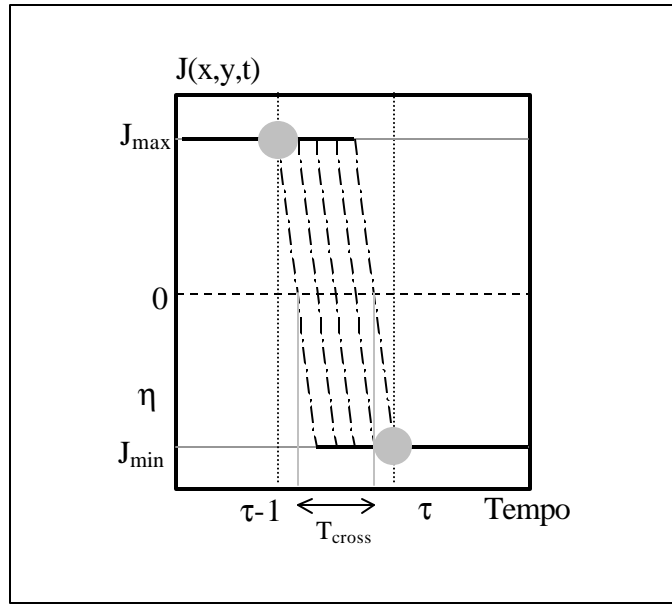
**Figura 6.11: Andamento nel tempo della densità normalizzata per un fissato pixel di lavoro.** Se il bordo dell'ombra è ben definito il profilo di  $J(x,y,t)$  presenta una elevata pendenza in corrispondenza dell'attraversamento (diagramma di sinistra). Se il bordo presenta una estesa zona di transizione la pendenza in corrispondenza dell'attraversamento è invece minore (diagramma di destra).



**Figura 6.12: Andamento nel tempo della densita' normalizzata per un fissato pixel di lavoro.** Se il passaggio tra  $J_{\max}$  e  $J_{\min}$  e' molto brusco, puo' facilmente capitare che  $J(x,y,\tau-1)$  e  $J(x,y,\tau)$  assumano valori praticamente coincidenti con  $J_{\max}$  e  $J_{\min}$  rispettivamente (cfr. figura di sinistra). Se invece il passaggio e' piu' graduale,  $J(x,y,\tau-1)$  e  $J(x,y,\tau)$  devono per forza assumere valori all'interno della transizione tra  $J_{\max}$  e  $J_{\min}$  come nel caso della figura di destra.

Tutto cio' a parita' di velocita' di spostamento dell'ombra. Se si assume invece costante la zona di transizione  $\Delta$ , la velocita' con cui si muove l'ombra influenza in modo perfettamente analogo l'andamento della  $J(x,y,t)$  in corrispondenza dell'attraversamento per lo zero. A fronte di spostamenti rapidi si ha un profilo come quello di figura 6.11 a sinistra, laddove per movimenti lenti si ha un profilo come quello di figura 6.11 a destra. E' ovvio che questi due effetti (velocita' con cui si muove l'ombra e ampiezza della regione di transizione  $\Delta$ ) possono combinarsi tra loro dando luogo a profili piu' o meno pendenti a seconda dei casi.

Si consideri un certo pixel  $\underline{x} = (x,y)$  e si supponga che l'andamento della relativa funzione  $J(x,y,t)$  nel tempo sia come quello illustrato in figura 6.12 a sinistra, a prescindere da quali possano esserne le ragioni, e che il valore di luminosit  normalizzata di  $\underline{x}$  agli istanti  $\tau$  e  $\tau-1$  siano  $J(x,y,\tau) \cong J_{\min}$  e  $J(x,y,\tau-1) \cong J_{\max}$  rispettivamente. In tali condizioni, trovare il tempo  $T_{\text{cross}}$  tramite interpolazione lineare (cfr. §6.2, equazione (6.2), e §2.5) tra  $J(x,y,\tau)$  e  $J(x,y,\tau-1)$  perde di significato. Come si vede nella figura 6.13 i possibili profili lineari che congiungono  $J(x,y,\tau) \cong J_{\min}$  con  $J(x,y,\tau-1) \cong J_{\max}$  sono praticamente infiniti.



**Figura 6.13: Possibili curve di transizione dati  $J(x,y,t-1)$  e  $J(x,y,t)$ .** Se  $J(x,y,t-1)$  e  $J(x,y,t)$  assumono valori coincidenti con  $J_{max}$  e  $J_{min}$  e' praticamente impossibile determinare il valore di  $T_{cross}$  all'interno dell'intervallo tra  $\tau-1$  e  $\tau$ . Esistono infatti infiniti profili lineari che congiungono tali punti.

Pertanto  $T_{cross}$  puo' essere un qualsiasi valore compreso tra  $\tau-1$  e  $\tau$ . In definitiva, ogni qual volta si sta muovendo la bacchetta troppo velocemente oppure si genera un'ombra il cui bordo e' tagliente, si va incontro a quest'inconveniente.

Se i valori assunti da  $J(x,y,\tau)$  con  $J(x,y,\tau-1)$  sono distanti di una certa soglia  $\eta$  dal valore massimo e dal valore minimo, si confronti per esempio la figura 6.12 di destra, il profilo lineare tra  $J(x,y,\tau)$  e  $J(x,y,\tau-1)$  diventa univocamente definito e pertanto tale sara' anche  $T_{cross}$ . In questo modo si ottiene un valore affidabile per il tempo di attraversamento. Tale tempo e' calcolato tramite interpolazione lineare secondo la formula (2.5):

$$T_{cross} = \mathbf{t} + \frac{J(x, y, \mathbf{t}-1)}{J(x, y, \mathbf{t}-1) - J(x, y, \mathbf{t})} \quad (2.5)$$

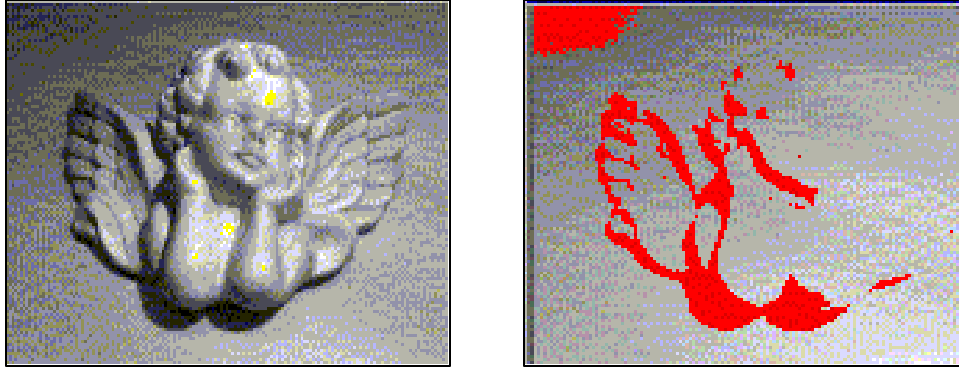
Non e' superfluo osservare che non necessariamente la (2.5) fornisce la migliore approssimazione per il reale tempo di attraversamento. E' infatti possibile utilizzare interpolazioni del secondo ordine o maggiori. Tuttavia si e' preferito adoperare quella del primo ordine per evitare di appesantire troppo l'elaborazione da effettuare in tempo

reale. In sostanza il vantaggio che se ne ricaverebbe non giustifica la perdita in termini di tempo di elaborazione che ne consegue.

Per non porre eccessive limitazioni sulla velocità con cui l'operatore deve muovere la bacchetta e sulla sua posizione relativa tra il piano di terra e la sorgente luminosa, si introducono ulteriori vincoli sulla selezione dei pixel da inserire nell'insieme  $\underline{X}_\tau$ . In altri termini, tra tutti i pixel appartenenti all'immagine quelli che soddisfano le seguenti condizioni faranno parte di  $\underline{X}_\tau$ :

$$\begin{aligned}
 & - \begin{cases} J(x, y, \mathbf{t}-1) \geq 0 \\ J(x, y, \mathbf{t}) < 0 \end{cases} \\
 & - \begin{cases} J(X_{\text{top}}, y_{\text{top}}, \mathbf{t}-1) > J(X_{\text{top}}, y_{\text{top}}, \mathbf{t}) \\ J(X_{\text{bot}}, y_{\text{bot}}, \mathbf{t}-1) > J(X_{\text{bot}}, y_{\text{bot}}, \mathbf{t}) \end{cases} \quad (\text{la direzione dell'ombra va da sinistra a destra}) \\
 & - J(x-1, y, \mathbf{t}) > J(x+1, y, \mathbf{t}) \\
 & - \begin{cases} J(x, y, \mathbf{t}) \in [J_{\min} + \mathbf{h}, J_{\max} - \mathbf{h}] \\ J(x, y, \mathbf{t}-1) \in [J_{\min} + \mathbf{h}, J_{\max} - \mathbf{h}] \end{cases} \\
 \\
 & \textit{oppure} \\
 \\
 & - \begin{cases} J(x, y, \mathbf{t}-1) < 0 \\ J(x, y, \mathbf{t}) \geq 0 \end{cases} \\
 & - \begin{cases} J(X_{\text{top}}, y_{\text{top}}, \mathbf{t}-1) < J(X_{\text{top}}, y_{\text{top}}, \mathbf{t}) \\ J(X_{\text{bot}}, y_{\text{bot}}, \mathbf{t}-1) < J(X_{\text{bot}}, y_{\text{bot}}, \mathbf{t}) \end{cases} \quad (\text{la direzione dell'ombra va da destra a sinistra}) \\
 & - J(x-1, y, \mathbf{t}) > J(x+1, y, \mathbf{t}) \\
 & - \begin{cases} J(x, y, \mathbf{t}) \in [J_{\min} + \mathbf{h}, J_{\max} - \mathbf{h}] \\ J(x, y, \mathbf{t}-1) \in [J_{\min} + \mathbf{h}, J_{\max} - \mathbf{h}] \end{cases} \\
 \\
 & \textit{per ogni } x, y \textit{ appartenenti all'immagine.} \tag{6.14}
 \end{aligned}$$

In questo modo scandire a elevate velocità e con la bacchetta vicino il piano di terra non porta più problemi nell'individuazione del tempo  $T_{\text{CURT}}$ , tuttavia il numero di



**Figura 6.14: Maschera definita sull'immagine.** Le parti con valori di contrasto molto bassi non vengono ulteriormente processate. Tali parti costituiscono la cosiddetta maschera. Nella figura a destra compare la maschera associata all'immagine di sinistra

pixel scartati, perché non soddisfacenti le condizioni (6.14), diventano molti e quindi la scansione integrale dello scenario richiede più tempo.

## 6.6 Condizioni sul contrasto

I pixel che presentano un basso valore di contrasto  $I_{\text{contrast}}(x,y) = I_{\text{max}}(x,y) - I_{\text{min}}(x,y)$  (cioè con  $I_{\text{max}}(x,y) \cong I_{\text{min}}(x,y)$ , cfr. §2.3) sono molto sensibili alle transizioni spurie. Infatti per questi pixel la luminosità normalizzata  $J(x,y,\tau)$  è molto vicina allo zero. La presenza del rumore provoca delle fluttuazioni su  $J(x,y,\tau)$  con il risultato che si possono verificare le condizioni:

$$\begin{cases} J(x,y,\mathbf{t}-1) \geq 0 \\ J(x,y,\mathbf{t}) < 0 \end{cases} \quad \text{e} \quad \begin{cases} J(x,y,\mathbf{t}-1) < 0 \\ J(x,y,\mathbf{t}) \geq 0 \end{cases}$$

anche per pixel non appartenenti al bordo attivo. Occorre quindi necessariamente eliminare a priori tali pixel. Per fare ciò si adopera la funzione  $I_{\text{mask}}(x,y)$  introdotta nel paragrafo 2.3. Se  $I_{\text{mask}}(x,y)$  assume valore nullo in presenza del pixel di lavoro, tale pixel non viene ulteriormente processato. L'individuazione di  $I_{\text{mask}}(x,y)$  avviene dopo aver valutato le funzioni  $I_{\text{max}}(x,y)$  e  $I_{\text{min}}(x,y)$  per ogni pixel (cfr. finestra 4). Tale funzione definisce la cosiddetta maschera, come illustrato nella figura 6.14.

Si osservi infine che l'aver introdotto la condizione:

$$\begin{cases} J(x, y, \mathbf{t}) \in [J_{\min} + \mathbf{h}, J_{\max} - \mathbf{h}] \\ J(x, y, \mathbf{t} - 1) \in [J_{\min} + \mathbf{h}, J_{\max} - \mathbf{h}] \end{cases}$$

elimina automaticamente quei pixel il cui valore di luminosità ha un range inferiore a  $2\eta$ . Tale condizione svolge pertanto un ruolo simile a quello della maschera.

## 6.7 L'algoritmo in tempo reale

Si è a questo punto in condizione di poter tracciare i principali passi dell'algoritmo in tempo reale.

Per ottenere il valori di  $I_{\max}(x,y)$ ,  $I_{\min}(x,y)$ ,  $I_{\text{contrast}}(x,y)$ ,  $I_{\text{mask}}(x,y)$  e  $I_{\text{thresh}}(x,y)$ , necessari per la fase di scansione vera e propria, si fa scorrere l'ombra sull'intero scenario. Un semplice algoritmo in real time aggiorna, per ogni immagine fornita dal frame grabber, le funzioni  $I_{\max}(x,y)$ ,  $I_{\min}(x,y)$ . Finito il processo, vengono calcolate  $I_{\text{mask}}(x,y)$ ,  $I_{\text{contrast}}(x,y)$  e  $I_{\text{thresh}}(x,y)$  secondo le equazioni presenti nel paragrafo 2.3 e 2.4. Tale procedura viene detta di *min\_max* ed è quella svolta nella finestra 4.

Impostati tutti i parametri secondo le necessità, si inizia la procedura di scansione. Ad ogni istante  $\tau$  il frame grabber fornisce un immagine (cio' equivale ad dire che istante per istante viene fornita la funzione  $I(x,y,\tau)$ ). La singola immagine, all'istante  $\tau$ , viene processata attraverso i seguenti passi:

1. Costruzione della funzione  $J(x,y,\tau)$  a partire da  $I(x,y,\tau)$  e da  $I_{\text{thresh}}(x,y)$  ( $J(x,y,\tau) = I(x,y,\tau) - I_{\text{thresh}}(x,y)$ ).
2. Valutazione delle ascisse di attraversamento  $x_{\text{top}}(\tau)$  e  $x_{\text{bot}}(\tau)$  nell'immagine corrente tramite la (4.2). Se nessun pixel appartenente tanto alla linea  $y_{\text{op}}$  quanto alla linea  $y_{\text{bot}}$  verifica la (4.2) la procedura si ferma e si passa all'immagine successiva (frame  $\tau+1$ ).

3. Vengono richiamate le ascisse di attraversamento  $x_{top}(\tau-1)$  e  $x_{bot}(\tau-1)$  relative all'immagine precedente (ovvero calcolate nel ciclo precedente). Se all'istante  $\tau-1$  nessun pixel appartenente tanto alla linea  $y_{top}$  quanto alla linea  $y_{bot}$  ha verificato la (4.2) la procedura si ferma e si passa all'immagine successiva (frame  $\tau+1$ ).
4. Determinazione della direzione dell'ombra tramite la tabella 6.3. Se non viene determinata alcuna direzione (ultime due righe della tabella) la procedura si ferma e si passa all'immagine successiva (frame  $\tau+1$ ).
5. Eliminazione dei pixel appartenenti alla maschera.
6. Determinazione dei pixel soddisfacenti le condizioni 6.14. L'insieme di tali pixel costituisce l'insieme  $\underline{X}_\tau = \{\underline{x}_{\tau 1}, \underline{x}_{\tau 2} \dots, \underline{x}_{\tau n}\}$ . Se non viene trovato alcun pixel soddisfacente la 6.14, la procedura si ferma e si passa all'immagine successiva (frame  $\tau+1$ ).
7. Valutazione del tempo  $T_{cross}$  per ciascun pixel appartenente a  $\underline{X}_\tau$  tramite la (6.2).
8. Valutazione dell'ascisse di attraversamento  $x_{top}(T_{cross})$  e  $x_{bot}(T_{cross})$  per ciascun pixel appartenente a  $\underline{X}_\tau$  facendo uso della (6.3).
9. Valutazione della retta di terra  $\bar{I}_h(T_{cross})$  per ciascun pixel appartenente a  $\underline{X}_\tau$  facendo uso della (6.4).
10. Valutazione del piano d'ombra  $\bar{w}_s(T_{cross})$  per ciascun pixel appartenente a  $\underline{X}_\tau$  facendo uso della (3.1).
11. Per ciascun pixel appartenente a  $\underline{X}_\tau$ , individuazione delle coordinate dei corrispondenti punti sul piano immagine tramite la (4.6).
12. Per ciascun pixel appartenente a  $\underline{X}_\tau$ , individuazione delle coordinate dei corrispondenti punti sullo scenario tramite la (3.6). La procedura e' completata e si passa all'immagine successiva (frame  $\tau+1$ ).

La sequenza di passi di cui sopra costituisce pertanto l'intero processo in tempo reale che deve essere svolto sulla singola immagine.

## 6.8 Tempi di elaborazione

Uno dei parametri fondamentali nei processi in tempo reale e' la velocita' con cui viene svolta l'elaborazione sull'immagine. Tale parametro dipende dalla velocita' del processore, dalla quantita' di memoria disponibile, dal numero e dalla complessita' delle operazioni da svolgere nell'algoritmo. Tipicamente un frame grabber e' impostato per acquisire (e quindi fornire al programma) 30 frame al secondo e cioe' un frame ogni 33ms. Pertanto se l'elaborazione sulla singola immagine impiega meno di 33ms, ogni frame fornito dal frame grabber e' in grado di essere processato. Se invece l'elaborazione impiega piu' di 33ms, un certo numero di frame verra' perso. Cio' significa che, pur muovendo con continuita' l'ombra sullo scenario, viene osservata una sequenza discontinua di immagini, facendo aumentare cosi' il tempo necessario per la scansione completa dell'oggetto.

Processing Time:	7.438932
FG Counter:	199
Program Counter:	100
Frames lost:	1
<input checked="" type="checkbox"/> Show image1	<input checked="" type="checkbox"/> Show image2

**Figura 6.15: Indicatori legati al tempo di elaborazione nel processo di scansione.** Il *processing time* rappresenta il tempo in millisecondi richiesto dal processo. Il processo in questione e' quello che determina  $I_{max}$  e  $I_{min}$ . *FG counter* indica il numero di frame che il frame grabber ha catturato da quando e' iniziata la scansione. *Program Counter* indica il numero di frame fin'ora processati ed infine *Frame Lost* indica il numero di frame persi per ogni ciclo di processo. Tale indicatore mostra un frame perso a causa dell'uso di risorse di cui non si tiene conto nel Processing Time.

Nella 6.15 sono evidenziate alcune delle informazioni fornite dal pannello di controllo per l'operazione di scansione.

## 6.9 Fase di merging

Come e' stato visto nel paragrafo 6.7, il passo conclusivo dell'algoritmo in tempo reale consiste valutare per ciascun pixel appartenente a  $\underline{X}_\tau$  (detto *pixel attivo*), le coordinate spaziali del corrispondente punto nel riferimento della telecamera. Detto  $\underline{x}_{\tau i}$  uno dei pixel di  $\underline{X}_\tau$ , si indichino con  $\bar{x}_{\tau i}=[x_{\tau i} \ y_{\tau i} \ z_{\tau i}]^T$  e  $\bar{X}_{\tau i}=[X_{\tau i} \ Y_{\tau i} \ Z_{\tau i}]^T$  rispettivamente il corrispondente punto sul piano immagine e il corrispondente punto nello scenario valutati nel riferimento della telecamera. Ricordando la (3.6), si ha:

$$Z_{\tau i} = \frac{1}{\langle \bar{\mathbf{w}}_{\tau i}, \bar{x}_{\tau i} \rangle}$$

da cui:

$$\bar{X}_{\tau i} = Z_{\tau i} \bar{x}_{\tau i} = \frac{\bar{x}_{\tau i}}{\langle \bar{\mathbf{w}}_{\tau i}, \bar{x}_{\tau i} \rangle} \quad (6.15)$$

dove si e' indicato con  $\bar{\mathbf{w}}_{\tau i}$  il piano ombra valutato per il pixel  $\underline{x}_{\tau i}$ . La coordinata  $Z_{\tau i}$  viene detta *profondita'* associata al pixel  $\underline{x}_{\tau i}$  ed e' quella quantita' che, come si puo' vedere dalla (6.15), consente di valutare interamente il vettore  $\bar{X}_{\tau i}$ . Pertanto l'elaborazione dell'immagine all'istante  $\tau$  e' completa quando, per tutti i pixel dell'insieme  $\underline{X}_\tau$ , viene fornita la relativa  $Z_{\tau i}$ .

Per ogni immagine fornita dal frame grabber, viene ripetuto il ciclo di passi della procedura in tempo reale, selezionando ogni volta un differente insieme  $Z_{\tau i}$  di pixel. Capitera' allora sicuramente che, data l'arbitrarieta' del movimento dell'ombra, stessi pixel possono essere processati anche piu' volte. Si ottengono cosi', per ciascun pixel  $\underline{x}_i$ , una successione di valori di profondita'  $\{Z_{1i}, \dots, Z_{\tau i}, Z_{(\tau+k)i}, \dots\}$  rappresentanti ciascuno una stima del vero valore di profondita'  $Z_i$  (ovviamente incognito) in un determinato istante di tempo. Si indichi con  $N$  il numero di valori di profondita' determinati fino all'istante  $\tau$ . Per ciascuno di questi valori, inoltre, e' possibile calcolare la relativa incertezza o varianza  $\sigma_{\tau i}^2$  (cfr. equazione 5.5).

Ebbene gli N valori della sequenza  $\{Z_{1i}, \dots, Z_{\tau i}, Z_{(\tau+1)i}, \dots, Z_{Ni}\}$  possono essere mediati opportunamente tramite gli  $\sigma_{\tau i}^2$ . Si ottiene così la stima  $\hat{Z}_i^{(N)}$  di  $Z_i$  sugli N valori della sequenza, dove  $\hat{Z}_i^{(N)}$  può essere calcolato ricorsivamente nel seguente modo:

$$\left(\hat{Z}_i^{(N)}\right)^2 = \left( \frac{\left(\hat{Z}_i^{(N-1)}\right)^2}{\left(\mathbf{s}_i^{(N-1)}\right)^2} + \frac{Z_{Ni}^2}{\mathbf{s}_{Ni}^2} \right) \left(\mathbf{s}_i^{(N)}\right)^2 \quad (6.16)$$

con

$$\left(\mathbf{s}_i^{(N)}\right)^2 = \frac{\left(\mathbf{s}_i^{(N-1)}\right)^2 \mathbf{s}_{Ni}^2}{\left(\mathbf{s}_i^{(N-1)}\right)^2 + \mathbf{s}_{Ni}^2} \quad (6.17)$$

e con valori iniziali  $\left(\mathbf{s}_i^{(1)}\right)^2 = \mathbf{s}_{i1}^2$  e  $\left(\hat{Z}_i^{(1)}\right)^2 = Z_{i1}^2$ .

La quantità  $\hat{Z}_i^{(N)}$  fornisce una buona stima di  $Z_i$  soprattutto quando N diventa sufficientemente grande (maggiore di circa 10, secondo risultati sperimentali).

## 6.10 Velocità e precisione

Da quanto detto nei precedenti paragrafi risulta evidente che i due parametri fondamentali nell'algoritmo in tempo reale sono: precisione nella ricostruzione e velocità di scansione.

Per velocità di scansione si intende il tempo che ci vuole per scandire l'intero volume di lavoro oppure, in altre parole, il tempo necessario per ottenere l'informazione di profondità per ogni pixel dello scenario. Per precisione si intende quanto è accurato il modello tridimensionale ottenuto rispetto all'originale, ovvero in altre parole, lo scostamento medio del valore di profondità stimato da quello reale. Di seguito tale grandezza viene considerata dal punto di vista locale (cioè relativa al singolo pixel e non come valor medio).

La velocità di scansione è direttamente legata alle seguenti quantità:

- Tempo richiesto per processare un'immagine. Se tale tempo è ridotto viene perso un numero limitato di frame al secondo. In generale poi il numero di frame al secondo processati fornisce un'indicazione di quanto veloce può spostarsi l'ombra sullo scenario.
- Spessore del bordo dell'ombra. Più il bordo dell'ombra presenta un'ampia zona di transizione da ombra a luce, maggiore sarà la possibilità che vengano soddisfatte le condizioni (6.14). Ciò quindi aumenta per ogni istante di tempo  $\tau$  il numero di pixel attivi migliorando così la velocità di scansione (ovvero è necessario un numero minore di passaggi dell'ombra sull'oggetto).

La precisione nella ricostruzione è influenzata dalle seguenti grandezze:

- Varianza  $\sigma_i^2$  del rumore legato alla profondità stimata  $Z_i$ , relativamente al pixel  $\underline{x}_i$ . Tale varianza è peraltro funzione di varie quantità (cfr equazione 5.5). In particolare è inversamente proporzionale a  $I_x^2$  che rappresenta il quadrato del gradiente di luminosità del bordo dell'ombra in corrispondenza del pixel  $\underline{x}_i$ . Pertanto, come già osservato nel capitolo V, più il bordo è ben definito minore è lo scostamento della stima dal valore vero, cioè la stima è più affidabile.
- Numero  $N$  di volte in cui un singolo pixel è stato processato (cioè numero di volte in cui il pixel  $\underline{x}_i$  è diventato attivo). La varianza  $(\mathbf{s}_i^{(N)})^2$  espressa dalla (6.17) diminuisce all'aumentare di  $N$ .

Quindi velocità e precisione possono risultare spesso in conflitto tra loro. Per esempio avere un bordo dell'ombra più smussato incrementa il numero di pixel attivi immagine per immagine ma peggiora l'accuratezza delle singole stime. D'altro canto, a parità di tempo, avere per ogni pixel una notevole quantità di stime va a favore della precisione.