



Consiglio Nazionale delle Ricerche  
Istituto di Calcolo e Reti ad Alte Prestazioni

## Un algoritmo innovativo per il calcolo del flusso ottico

E. Francomano, C. Macaluso

***Rapporto Tecnico N.:16***  
**RT-ICAR-PA-04-16**

**dicembre 2004**



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)  
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: [www.icar.cnr.it](http://www.icar.cnr.it)  
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: [www.na.icar.cnr.it](http://www.na.icar.cnr.it)  
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: [www.pa.icar.cnr.it](http://www.pa.icar.cnr.it)



Consiglio Nazionale delle Ricerche  
Istituto di Calcolo e Reti ad Alte Prestazioni

## Un algoritmo innovativo per il calcolo del flusso ottico

E. Francomano<sup>1,2</sup>, C. Macaluso

**Rapporto Tecnico N.:16**  
**RT-ICAR-PA-04-16**

**Data:**  
**dicembre 2004**

---

<sup>1</sup> Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sezione di Palermo Viale delle Scienze edificio 11 90128 Palermo

<sup>2</sup> Università degli Studi di Palermo Dipartimento di Ingegneria Informatica Viale delle Scienze 90128 Palermo

*I rapporti tecnici dell'ICAR-CNR sono pubblicati dall'Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche. Tali rapporti, approntati sotto l'esclusiva responsabilità scientifica degli autori, descrivono attività di ricerca del personale e dei collaboratori dell'ICAR, in alcuni casi in un formato preliminare prima della pubblicazione definitiva in altra sede.*

**Sommario:** si propone un algoritmo per il calcolo del flusso ottico che si avvale di un processo di approssimazione mediante un operatore quasi interpolante. Tale operatore consente di valutare globalmente i movimenti presenti in una scena utilizzando un campionamento adattivo nelle zone di interesse.

## 1. Introduzione

L'elaborazione di una immagine, o di una "scena statica" trova ormai applicazioni in diversi campi della tecnologia quali: il telecontrollo, il movimento autonomo di robot, l'inseguimento di obiettivi mobili, la compressione d'immagini, la televisione digitale, ed altro.

L'analisi di scene statiche si articola nella seguente successione di passi: acquisizione, preprocesso, segmentazione, descrizione delle informazioni, ricognizione ed interpretazione. Ogni passo del processo è oggetto di studio nel campo della elaborazione digitale.

Nel caso di acquisizione di più immagini è possibile estrarre informazioni dal movimento degli oggetti nelle scene. In questo caso, nella fase di preprocesso è importante la valutazione del campo del flusso ottico.

Il moto della scena è utile per la comprensione di base dell'immagine, infatti tale fase di preprocesso favorisce lo studio dei passi computazionali successivi.

I dati di ingresso per i processi di comprensione del moto sono quelli derivati dal campo di velocità bidimensionale (flusso ottico) ottenuto dalle proiezioni delle velocità tridimensionali dei punti in moto nello spazio.

La stima del flusso ottico è uno dei problemi fondamentali della fase di preprocesso (*low level vision*) e la ricerca condotta negli ultimi anni sull'argomento ha fornito diverse metodologie risolutive proprio in relazione allo sviluppo tecnologico.

L'informazione presente nel moto riveste un ruolo di primaria importanza in molte problematiche di visione quali la segmentazione, l'inseguimento di obiettivi mobili (*tracking*), la predizione di un certo assetto dell'ambiente circostante, la ricostruzione 3D ed è un prerequisito rilevante per i tasks finalizzati al movimento di robot autonomi e la loro interazione con l'ambiente.

E' dunque essenziale avvalersi di stime accurate di flussi ottici calcolate con algoritmi robusti ed affidabili per poter ottenere informazioni significative per i successivi processi computazionali.

In questo lavoro si propone un algoritmo per il calcolo del flusso ottico utilizzando un operatore quasi interpolante e avvalendosi di un processo di minimo non lineare.

## 2. Operatori quasi interpolanti

Un operatore quasi-interpolante monodimensionale è definito dalla seguente formulazione:

$$(Qf)(x) = \sum_{i=1}^k (\lambda_i f)(x) \phi_i(x)$$

dove  $\phi_1(x), \phi_2(x), \dots, \phi_k(x)$  sono opportune funzioni e  $\lambda_1, \lambda_2, \dots, \lambda_k$  sono appropriati funzionali.

Sia  $N_d(x)$  la funzione B-spline cardinale di ordine  $d$  e sia  $(\lambda f)(l)$  una combinazione lineare di un insieme di valori  $\{f_j\}_{j \in \mathbb{Z}} : (\lambda f)(l) = \sum_{j \in \mathbb{Z}} \lambda_{l-j} f_j$ .

L'operatore quasi-interpolante:

$$(Qf)(x) = \sum_{l \in \mathbb{Z}} (\lambda f)(l) N_d(x + \frac{d}{2} - l)$$

proietta  $C_b(\mathfrak{R}) = \{f \in C(\mathfrak{R}) : \sup|f(x)| < \infty\}$  nello spazio delle funzioni B-spline cardinali centrali.

Indicando con  $P_{d-1}$ ,  $d \geq 1$ , lo spazio dei polinomi algebrici di grado al più  $d-1$ , l'operatore  $Q$  verifica la seguente notevole proprietà:

$$(Qp)(x) = p(x) \quad p(x) \in P_{d-1}$$

Tale operatore è locale, nel senso che  $(Qf)(x)$  dipende solo dai valori di  $f$  in un piccolo intorno di  $x$ . Pertanto, l'utilizzo dell'operatore quasi-interpolante  $Q$  risulta particolarmente valido per fornire buone approssimazioni polinomiali e il processo di calcolo relativo risulta efficiente.

Per esempio, si consideri il seguente funzionale lineare:

$$(\lambda f)(l) = \frac{1}{6}[-f(l+1) + 8f(l) - f(l-1)]$$

l'operatore quasi-interpolante monodimensionale con  $d=4$  risulta essere:

$$(Qf)(x) = \sum_{l \in Z} \frac{1}{6}[-f(l+1) + 8f(l) - f(l-1)]N_4(x+2-l)$$

Nella formulazione si nota che è possibile costruire un processo multirisolutivo: infatti dalla sequenza di dati iniziale  $\{f_j\}_{j \in Z}$ , è possibile ottenere un nuovo campionamento dei dati fissando l'intervallo di campionamento arbitrariamente piccolo:  $h = 2^{-\tau}$ .

In questo modo, l'operatore  $Q^h$  fornisce l'approssimazione migliore passando da un campionamento a livello di risoluzione 0, cioè  $x = q$ ,  $q \in Z$ , ad un campionamento a livello di risoluzione  $\tau$ , cioè  $x = q \cdot 2^{-\tau}$ , verificando il seguente teorema:

**Teorema:** Per ogni  $f \in C_b(\mathfrak{R}) \cap C^d(\Omega)$ , con  $\Omega$  insieme aperto, contenente un insieme compatto  $K$  in  $\mathfrak{R}$ , esiste una costante positiva  $\alpha$  dipendente solo da  $f$  e  $K$  tale che:

$$\max_{x \in K} |(Q^h f - f)(x)| \leq \alpha \cdot h^d$$

per ogni valore di  $h > 0$  sufficientemente piccolo.

A partire dalla formulazione monodimensionale, mediante prodotto tensoriale, è possibile costruire un operatore quasi-interpolante in due variabili.

Ad esempio, riferendosi al caso precedentemente riportato si ha:

$$\begin{aligned} (Qf)(x, y) = \sum_{l \in Z} \sum_{k \in Z} \left\{ \left[ \frac{1}{36}f(l+1, k+1) - \frac{2}{9}f(l+1, k) + \frac{1}{36}f(l+1, k-1) + \right. \right. \\ \left. \left. - \frac{2}{9}f(l, k+1) + \frac{16}{9}f(l, k) - \frac{2}{9}f(l, k-1) + \frac{1}{36}f(l-1, k-1) - \frac{2}{9}f(l-1, k) + \right. \right. \\ \left. \left. + \frac{1}{36}f(l-1, k-1) \right] N_4(x+2-l) N_4(y+2-k) \right\} \quad (1) \end{aligned}$$

Se si pone  $x = q$  e  $y = r$ , con  $q, r \in Z$ , le due sommatorie di (1) sono estese soltanto ai termini con indici  $l$  e  $k$  che verificano le due condizioni:

$$q-1 \leq l \leq q+1 \quad \text{e} \quad r-1 \leq k \leq r+1$$

L'operatore (1) può essere riformulato facendo intervenire solo punti con indice pari e in questo modo si possono ottenere mediante la computazione di (1) tutte le valutazioni nei punti con indice dispari; ciò equivale a dimezzare l'ampiezza  $h$  della tabulazione dei dati e quindi a passare da una griglia di campionamento ad una più fitta.

### 3. Il calcolo del flusso ottico

Tra le modalità di calcolo del flusso ottico figurano le tecniche *gradient based* che calcolano il vettore degli spostamenti  $(u, v)$  facendo riferimento alla funzione  $I(x, y)$  rappresentante l'intensità luminosa dell'immagine nel punto di coordinate  $(x, y)$ .

Sia data una sequenza di immagini  $I_t(x, y)$  che si suppone ottenuta dall'immagine  $I(x, y)$  di riferimento mediante movimento, orizzontale e verticale al tempo  $t$  con velocità istantanea  $u_t(x, y)$  e  $v_t(x, y)$ , cioè:

$$I_t(x + u_t, y + v_t) = I(x, y)$$

Per suddetta sequenza si vuole calcolare il campo degli spostamenti  $u_t$  e  $v_t$  in relazione all'immagine  $I(x, y)$  minimizzando l'errore quadratico:

$$\sum_t \iint [I_t(x + u_t, y + v_t) - I(x, y)]^2 dx dy \quad (2)$$

In pratica, dato un insieme discreto di campionamenti dell'immagine, si sostituisce la (2) con una sommatoria estesa a tutti i pixel  $\{(x_i, y_i)\}$  dell'immagine.

Se i campi degli spostamenti  $u_t$  e  $v_t$ , per istanti di tempo differenti, risultano indipendenti l'uno dall'altro e la funzione di riferimento  $I(x, y)$  risulta nota, il problema (2) si decompone in distinti problemi di minimo, ciascuno per ogni *frame* della sequenza dell'immagine.

Pertanto, assumendo questa ipotesi, il problema si riconduce allo studio di soli due *frame* della sequenza.

$$E(\{u_i, v_i\}) = \sum_i [I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i)]^2 \quad (3)$$

Questa equazione è chiamata *Sum of Squared Differences* (SSD) ed in letteratura sono note diverse tecniche per ottenere una stima del flusso.

Per esempio Anandan (1989) svolge l'algoritmo SSD avendo suddiviso in quadrati di dimensione prefissata i pixel dell'immagine. Più recentemente Okutomi e Kanade (1993) hanno usato varianti adattive per suddividere i punti dell'immagine.

Ancora, si sono sviluppati algoritmi basati su processi di regolarizzazione dei dati per ottenere buone valutazioni del flusso ottico (Horn e Schunck 1983, Poggio 1985) ed algoritmi basati su processi multiscala del tipo *coarse-to-fine* per affinare le valutazioni iniziali del campo di moto.

L'alternativa che propone Szeliski [22] agli approcci suddetti, è quella di rappresentare il flusso ottico mediante funzioni spline bidimensionali costruite su un limitato insieme di punti base, cioè stime prefissate del flusso  $\hat{u}_j$  e  $\hat{v}_j$ , e ricostruire da tali dati il valore del flusso ottico in ogni pixel dell'immagine.

Il valore  $u_i = u(x_i, y_i)$  è calcolato mediante l'espressione:

$$u_i = \sum_j \hat{u}_j B_j(x_i, y_i) = \sum_j \hat{u}_j w_{ij}$$

dove  $B_j(x, y)$  rappresentano le funzioni base che risultano non nulle solo su un supporto finito.

I valori  $w_{ij} = B_j(x_i, y_i)$  sono chiamati pesi per sottolineare che le stime di  $(u_j, v_j)$  sono calcolate mediante combinazione lineare delle quantità note  $(\hat{u}_j, \hat{v}_j)$ .

Nella realizzazione del processo Szeliski considera sottoinsiemi regolari della griglia dei pixel:  $\hat{x}_j = mx_i$  e  $\hat{y}_j = my_i$ , in tal modo ad ogni insieme di  $m \times m$  pixel risulta associato una singola funzione spline. L'utilizzo di tali funzioni per rappresentare il campo di flusso, presenta il vantaggio di non dover ricorrere a termini di regolarizzazione nella equazione (3); inoltre, si elimina la necessità di sovrapporre gli

insiemi di suddivisione dei pixel della immagine, poiché ciascuna stima del flusso si ottiene solo mediante il contributo dei pixel di supporto di ogni quadrato di  $m \times m$  pixel.

#### 4. Il nuovo processo di calcolo

Un passo significativo rispetto al modello di Szeliski si può realizzare se si evita il partizionamento in sottoinsiemi dell'immagine, pur mantenendo il vantaggio di una approssimazione locale. Si propone una valutazione del campo di moto mediante un operatore quasi-interpolante, realizzato mediante funzioni B-spline, come già descritto, che supera la suddivisione del dominio in sottodomini e che risulta computazionalmente efficiente. Per applicare tale operatore è opportuno fissare come nodi base dell'approssimazione, punti equidistanti a cui si attribuiscono valori arbitrari del flusso ottico. Mediante processo di minimizzazione è possibile correggere le stime iniziali e ripetere il processo per una più corretta valutazione dei vettori di moto.

Il processo si basa sulla ricerca dell'insieme dei vettori  $(u, v)$  a partire da due immagini di una sequenza rappresentate mediante la funzione intensità luminosa:  $I_0(x, y)$  e  $I_1(x, y)$ .

Si riscrive il problema (11) come segue:

$$E(\{u_{ij}, v_{ij}\}) = \sum_i \sum_j [I_1(x_i + u_{ij}, y_j + v_{ij}) - I_0(x_i, y_j)]^2 \quad (4)$$

La valutazione del campo delle velocità riguarda tutti i pixel dell'immagine:  $u_{ij} = u(x_i, y_j)$  e  $v_{ij} = v(x_i, y_j)$  con  $i = 1, \dots, r$ ,  $j = 1, \dots, c$

Il processo di calcolo inizia fissando arbitrariamente alcuni valori del campo di moto su punti equidistanti:

$$\{(\hat{u}_{ij}, \hat{v}_{ij})\}_{i=i_1, \dots, i_{\hat{r}}; j=j_1, \dots, j_{\hat{c}}} \quad \text{con } \hat{r} < r \text{ e } \hat{c} < c$$

Questi valori costituiscono i nodi del processo di approssimazione locale.

Naturalmente i valori iniziali rappresentano una stima molto approssimata del campo delle velocità. Per valutare meglio il flusso ottico si sviluppa un processo di minimo che corregge le stime iniziali. Per affinare i valori del flusso si è applicata la tecnica di minimo non lineare di Levenberg-Marquardt alla formula (4).

Il processo richiede il calcolo della matrice hessiana  $H$  formata da blocchi del tipo:

$$\begin{pmatrix} \frac{\partial^2 E}{\partial \hat{u}_{mn} \partial \hat{u}_{pq}} & \frac{\partial^2 E}{\partial \hat{u}_{mn} \partial \hat{v}_{pq}} \\ \frac{\partial^2 E}{\partial \hat{u}_{mn} \partial \hat{v}_{pq}} & \frac{\partial^2 E}{\partial \hat{v}_{mn} \partial \hat{v}_{pq}} \end{pmatrix} \quad m, p = i_1, \dots, i_{\hat{r}} \quad n, q = j_1, \dots, j_{\hat{c}}$$

ed il vettore  $\gamma$  formato da blocchi gradiente del tipo:

$$\begin{pmatrix} \frac{\partial E}{\partial \hat{u}_{mn}} \\ \frac{\partial E}{\partial \hat{v}_{mn}} \end{pmatrix} \quad m = i_1, \dots, i_{\hat{r}} \quad n = j_1, \dots, j_{\hat{c}}$$

Ad ogni passo del processo deve essere risolto un sistema lineare:

$$(H + \lambda D) \Delta \varepsilon = -\gamma \quad (5)$$

dove  $\lambda$  è il parametro di stabilizzazione dello schema di Levenberg-Marquardt,  $D = \text{diag}(H)$  e  $\Delta \varepsilon$  è il vettore a blocchi incognito, di componenti:

$$\begin{pmatrix} \varepsilon_{\hat{u}_{mn}} \\ \varepsilon_{\hat{v}_{mn}} \end{pmatrix} \quad m = i_1, \dots, i_{\hat{r}} \quad n = j_1, \dots, j_{\hat{c}}$$

La soluzione del sistema (5) fornisce le correzioni che devono essere apportate alle valutazioni correnti:

$$\begin{aligned}\hat{u}_{mn}^* &= \hat{u}_{mn} + \varepsilon_{\hat{u}_{mn}} \\ \hat{v}_{mn}^* &= \hat{v}_{mn} + \varepsilon_{\hat{v}_{mn}} \quad m = i_1, \dots, i_r \quad n = j_1, \dots, j_c\end{aligned}\quad (6)$$

Con queste valutazioni è possibile approssimare il flusso ottico in ogni pixel dell'immagine mediante l'utilizzo dell'operatore quasi-interpolante. Si ottiene così un nuovo campo dei vettori di moto  $\{u_{ij}^*, v_{ij}^*\}$ ,  $i = 1, \dots, r$  e  $j = 1, \dots, c$ .

Se si verifica la relazione:

$$E(\{u_{ij}^*, v_{ij}^*\}) < E(\{u_{ij}, v_{ij}\}) \quad i = 1, \dots, r \text{ e } j = 1, \dots, c$$

i valori correnti vengono sostituiti con i valori nuovi e il parametro  $\lambda$  viene diminuito; altrimenti il valore corrente viene conservato e il parametro  $\lambda$  viene aumentato.

Il processo di calcolo può essere così descritto:

*Dati:*

1.  $I_0(x, y)$  e  $I_1(x, y)$ , due *frame* dell'immagine
2. un insieme di valutazioni iniziali del flusso ottico
3. *maxit*, numero massimo di iterate, e *toll*, zero di calcolo

*Obiettivo:*

Calcolo del flusso ottico per tutti i pixel dell'immagine

*Passi:*

1. Applicazione dell'operatore quasi-interpolante sull'insieme dei dati iniziali per ottenere una approssimazione delle velocità su tutti i punti dell'immagine.
2. Calcolo della funzione (4)  $E_{old}$
3. Inizializzazione del parametro  $\lambda$
4. Ripetere
  - 4.1. Calcolo del vettore  $\gamma$
  - 4.2. Calcolo della matrice hessiana  $H$
  - 4.3. Costruzione della matrice  $(H + \lambda D)$
  - 4.4. Risoluzione del sistema lineare (5)
  - 4.5. Calcolo del flusso ottico mediante l'operatore quasi-interpolante sui nuovi valori (6)
  - 4.6. Calcolo della funzione (4)  $E_{new}$
  - 4.7. Controllo della relazione  $E_{new} < E_{old}$  e aggiornamento di  $\lambda$ .

Finché non si verifica che  $(E_{new} - E_{old}) < toll$  oppure si è raggiunto il numero massimo di iterate *maxit*.

## 5. L'approssimazione mediante l'operatore quasi-interpolante

Siano dati i valori iniziali del campo di moto:

$$\{(\hat{u}_{ij}, \hat{v}_{ij})\}_{i=i_1, \dots, i_r; j=j_1, \dots, j_c}$$

che corrisponderanno ai nodi nel processo di approssimazione. Una prima valutazione del flusso ottico può essere espressa come:

$$\begin{aligned}u(x_i, y_j) &\approx (Qu)(x_i, y_j) \\ v(x_i, y_j) &\approx (Qv)(x_i, y_j) \quad i = 1, \dots, r \quad j = 1, \dots, c\end{aligned}$$

Indicando con:

$$e_{ij} = I_1(x_i + u_{ij}, y_j + v_{ij}) - I_0(x_i, y_j) \quad i = 1, \dots, r \quad j = 1, \dots, c$$

e facendo riferimento all'algoritmo di Levenberg-Marquardt, si calcolano le componenti del vettore  $\mathcal{Y}$  come segue:

$$\frac{\partial E}{\partial \hat{u}_{mn}} = 2 \sum_{i=1}^r \sum_{j=1}^c e_{ij} I_{ij}^x R_{ij}^{(mn)}$$

$$\frac{\partial E}{\partial \hat{v}_{mn}} = 2 \sum_{i=1}^r \sum_{j=1}^c e_{ij} I_{ij}^y R_{ij}^{(mn)} \quad m = i_1, \dots, i_r \quad n = j_1, \dots, j_c$$

dove:

$$(I_{ij}^x, I_{ij}^y) = \nabla I_1(x_i + u_{ij}, y_j + v_{ij}) \quad i=1, \dots, r \quad j=1, \dots, c$$

ed inoltre:

$$R_{ij}^{(mn)} = \left( \frac{\partial(Qu)}{\partial \hat{u}_{mn}} \right)_{ij} = \left( \frac{\partial(Qv)}{\partial \hat{v}_{mn}} \right)_{ij} \quad i=1, \dots, r \quad j=1, \dots, c$$

$$m = i_1, \dots, i_r \quad n = j_1, \dots, j_c$$

mentre per gli elementi della matrice hessiana si ha:

$$\frac{\partial^2 E}{\partial \hat{u}_{mn} \partial \hat{u}_{pq}} = 2 \sum_{i=1}^r \sum_{j=1}^c (I_{ij}^x)^2 R_{ij}^{(mn)} R_{ij}^{(pq)}$$

$$\frac{\partial^2 E}{\partial \hat{v}_{mn} \partial \hat{v}_{pq}} = \frac{\partial^2 E}{\partial \hat{u}_{mn} \partial \hat{v}_{pq}} = 2 \sum_{i=1}^r \sum_{j=1}^c (I_{ij}^x)(I_{ij}^y) R_{ij}^{(mn)} R_{ij}^{(pq)}$$

$$m = i_1, \dots, i_r \quad n = j_1, \dots, j_c$$

Per calcolare facilmente le derivate  $R_{ij}^{(mn)}$ , l'operatore quasi-interpolante può essere riscritto come segue:

$$(Qf)(x, y) = \sum_{l \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \left\{ \sum_{i=-2}^1 \sum_{j=-2}^1 f(l+i, k+j) [c_{ij} N_4(2x-2l+2) N_4(2y-2k+2) + \right.$$

$$d_{ij} N_4(2x-2l+3) N_4(2y-2k+3) + g_{ij} N_4(2x-2l+2) N_4(2y-2k+3) +$$

$$\left. h_{ij} N_4(2x-2l+3) N_4(2y-2k+2)] \right\}$$

Poiché  $\text{supp}(N_4) = [0, 4]$ , i valori  $R_{ij}^{(mn)}$  soddisfano la seguente relazione:

se  $i \in \{i_1, \dots, i_r\}$  e  $j \in \{j_1, \dots, j_c\}$  allora

$$R_{ij}^{(mn)} = \begin{cases} 1 & \text{se } (i, j) = (m, n) \\ 0 & \text{se } (i, j) \neq (m, n) \end{cases}$$

altrimenti

$$R_{ij}^{(mn)} \neq 0 \quad m-2\rho \leq i \leq m+2\rho \quad n-2\rho \leq j \leq n+2\rho$$

dove  $\rho$  rappresenta la distanza tra due pixel in cui è fissato un valore iniziale di velocità.

Per calcolare gli elementi della matrice  $H$  basta calcolare solo alcuni prodotti  $R_{ij}^{(mn)} R_{ij}^{(pq)}$

Infatti:

$$R_{ij}^{(mn)} R_{ij}^{(pq)} \neq 0$$

per i valori  $p$  e  $q$  tali che:

$$m-4\rho \leq p \leq m+4\rho \quad n-4\rho \leq q \leq n+4\rho$$

La struttura della matrice  $H$  è strettamente collegata all'operatore quasi-interpolante utilizzato. L'ordine della matrice è pari a  $2\hat{c}$  ma l'operatore adottato rende la matrice  $H$  a banda. Esattamente si tratta di una matrice a nove bande con massima ampiezza di banda uguale a 18. Tale struttura della matrice è indipendente dal numero iniziale dei punti, pertanto la sparsità della matrice aumenta al crescere dei punti base del processo. Lo schema di calcolo sviluppato risulta pertanto competitivo confrontato con



l'approssimazione mediante spline adottata da Szeliski. Quest'ultimo modello genera una matrice i cui elementi nulli sono strettamente dipendenti dal numero di suddivisioni che si considerano nel partizionamento dell'immagine.

Per la risoluzione del sistema lineare che si presenta ad ogni passo dell'algoritmo, si è lavorato con metodi iterativi. Una buona velocità di convergenza è stata raggiunta negli esperimenti svolti quando si è considerato come preconditionatore una matrice diagonale a blocchi. Esattamente ogni blocco diagonale è ottenuto dall'inversione della seguente matrice:

$$\begin{pmatrix} (1+\lambda) \frac{\partial^2 E}{\partial \hat{u}_{mn}^2} & \frac{\partial^2 E}{\partial \hat{u}_{mn} \partial \hat{v}_{mn}} \\ \frac{\partial^2 E}{\partial \hat{u}_{mn} \partial \hat{v}_{mn}} & (1+\lambda) \frac{\partial^2 E}{\partial \hat{v}_{mn}^2} \end{pmatrix} \quad m = i_1, \dots, i_f \quad n = j_1, \dots, j_c$$

## 6. Risultati sperimentali

Per verificare la validità del processo sviluppato si sono effettuati diversi test su differenti sequenze di immagini. I risultati riportano l'andamento del campo dei vettori di moto e l'errore relativo corrispondente. I risultati sono stati ottenuti senza utilizzare tecniche di pre o post processo, cioè non si è provveduto a una regolarizzazione sui dati iniziali e non è stato applicato nessun tipo di soglia sulla valutazione finale. L'errore relativo tra la velocità corretta e quella approssimata è stata valutata in ogni pixel. Si è adottata una scala di livelli di grigio per una rappresentazione grafica dell'errore calcolato. Come già detto la scelta dei punti iniziali è arbitraria.

Negli esperimenti riportati si è adottata la seguente scelta:

$$\hat{u}_{mn} = \hat{v}_{mn} = k[I_0(x_m, y_n) - I_1(x_m, y_n)] \quad m = i_1, \dots, i_f \quad n = j_1, \dots, j_c$$

dove  $k$  è fissato in modo che  $u, v \leq 10^{-3}$ .

### Sequenza 1

In questa sequenza si sono considerati due quadrati che si avvicinano con una velocità pari a  $(1;-1)$  e  $(-1;1)$  pixels/frame rispettivamente. I quadrati e lo sfondo dell'immagine sono caratterizzati da una luminosità che varia uniformemente. La prima immagine della sequenza è riportata nella figura 1. La dimensione del frame è di  $114 \times 114$  pixel. I risultati ottenuti risultano soddisfacenti. Il campo di moto è ben ricostruito come mostra la figura 2. L'errore è più evidente nei contorni dei quadrati ed è circa del 30%, e raggiunge il suo massimo valore del 61% ai vertici.

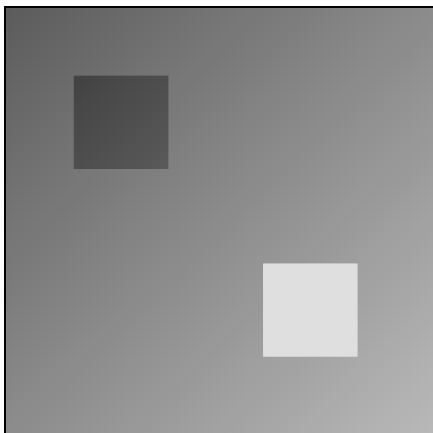


figura 1 Un frame della sequenza 1

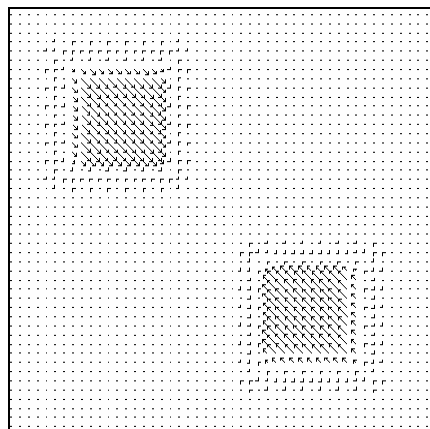


figura 2 Flusso Ottico

### Sequenza 2

Questa sequenza è stata ricavata da una scena reale (figura 3) producendo spostamenti su tre diversi oggetti, la dimensione del frame è di  $120 \times 120$  pixel. Esattamente il cappello si sposta verso l'alto con una velocità di  $(0;1)$  pixels/frame. Il pannello situato tra la libreria ed il muro si sposta a sinistra con una velocità di  $(-0.8;0)$  pixels/frame. Il video del monitor si sposta verso destra con una velocità pari a  $(1;0)$  pixels/frame. Il flusso ottico dei tre oggetti è mostrato in figura 4 dove i tre movimenti sono ben individuati. L'errore relativo in media è del 15%. L'errore più significativo, anche in questo caso, è localizzato ai bordi ed è circa del 50%. L'errore massimo raggiunge il valore di 88% e si trova nella parte bassa del cappello



figura 3 Un frame della sequenza 2

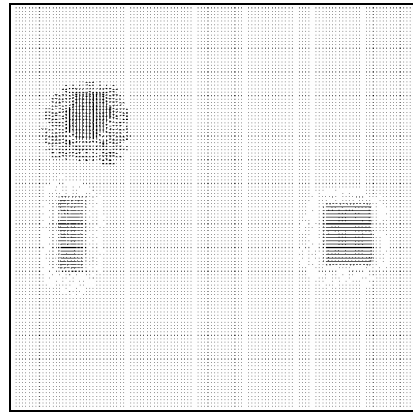


figura 4 Flusso ottico

### Sequenza 3

La sequenza 2 è stata modificata applicando un rumore random del 10% a tutti i pixel dell'immagine.



figura 5 Un frame della sequenza 3

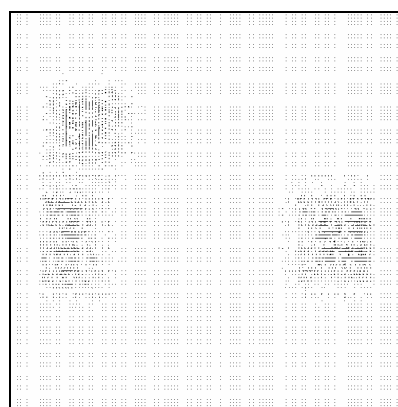


figura 6 Flusso ottico

La prima immagine della sequenza è riportata in figura 5. Come mostra la figura 6 il flusso ottico è apprezzabile e l'errore relativo è in media del 67%.

### Bibliografia

1. G.Adiv, *Inherent ambiguities in recovering 3-D motion and structure from noisy flow field*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (5), pp. 477-489 (1999).
2. P.Anandan, *A computational framework and an algorithm for the measurement of visual motion*, International Journal of Computer Vision, 2(3):283-310, (1989).

3. J.L.Barron, J.Fleet and S.S.Beauchemin, *Systems and Experiment-Performance of optical flow techniques*, International Journal of Computer Vision, 12, pp. 43-77, (1994).
4. J.L.Barron and A.Liptay, *Optical flow to measure minute increments in plant growth*, BioImaging, 2:1:57-61, (1994).
5. J.L.Barron, A.Liptay, T.Jewett and I.van Wesebeeck, *Oscillations in Corn Seedling Growth as Measured by Optical Flow*, Journal of American Society for Horticultural Science, 120 (3):379:385, (1995).
6. P.Boutemy and A.Mitiche, *Computation and analysis of image motion : a synopsis of current problems and methods*, International Journal of Computer Vision, 19, pp. 29-55, (1996).
7. V.Calderone, E.Francomano and A.Tortorici, *Optical flow computation in a multi-scale framework*, Applicable Mathematics its perspective and Challenges, Ed. J.C. Misra Narosa Publishing, pp.153-161, (2000).
8. C.K.Chui, *An introduction to wavelets*, Academic Press, (1992).
9. C.K.Chui, *Multivariate splines*, CBMS-NSF Regional Conference Series in Applied SIAM, 54, (1988).
10. C.K.Chui and H.Diamond, *A natural formulation of quasi-interpolation by multivariate splines*, Proc. Amer. Math. Soc., 99, pp.643-646, (1987).
11. C.deBoor, *On uniform approximation by splines*, Journal of Approximation Theory, pp.219-235, (1968).
12. K.Horn, *Robot vision*, MA MIT Press, Cambridge, (1986).
13. H.Lai and B.C.Vemuri, *Reliable and efficient computation of optical flow*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 5 (4), pp.598-610, (1996).
14. H.N. Mhaskar, F.J. Narcowich and J.D.Ward, *Quasi-interpolation in shift invariant spaces*, J. Math. Anal. Appl., 251, pp.356-363, (2000).
15. S.Negahdaripour and S.Lee, *Motion recovery from image sequences using only first order optical flow*, International Journal of Computer Vision, 9(3), pp. 163-184, (1992).
16. K. Prazdny, *On the information in optical flow*, Computer Vision Graphics, 23, pp.239-259, (1983).
17. W.H.Press, B.P.Flannery, S.A.Teukolsky and W.T.Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 23, Cambridge Univ. Press, 2nd ed. (1992).
18. J.L.Price and E.R.McVeigh, *Motion estimation from tagged mr image sequences*, IEEE Trans. On Medical Images, 11(2):238-249, (1992).
19. H.Quenòt, J.Pakleza and T.A.Kowalewski, *Particle image velocimetry with optical flow*, Experiments in fluids, 25, pp. 177-189, (1998).
20. L.Schumaker, *Spline functions: Basic theory*, J. Wiley & Sons, (1981).
21. G.Schunck, *Image flow: fundamentals and algorithms*, W. Martin and J.K.Aggarwal, Eds. Boston Kluwer, (1988).
22. R.Szeliski and H.Shum, *Motion estimation with quadtree splines*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18 n.12, pp. 1199-1210, (1996).
23. R.Szeliski, *Fast surface interpolation using hierarchical basis functions*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 (6), pp. 513-528, (1990).