DOTTORATO DI RICERCA
in
SCIENZE COMPUTAZIONALI E INFORMATICHE

Ciclo XVII

Consorzio tra Università di Catania, Università di Napoli Federico II,
Seconda Università di Napoli, Università di Palermo, Università di Salerno

SEDE AMMINISTRATIVA: UNIVERSITÀ DI NAPOLI FEDERICO II

Livia Marcellino

SU ALCUNI METODI NUMERICI PER IL RESTAURO AUTOMATICO DIGITALE DI IMMAGINI

TESI DI DOTTORATO DI RICERCA

Indice

Ringraziamenti.

Esprimo la mia profonda riconoscenza al Prof. Almerico Murli, che mi ha formato ed avviato alla ricerca e che, quale mio tutore in questo corso di dottorato, mi ha consigliato e guidato.

Desidero, inoltre, ringraziare la Prof.ssa Luisa D'Amore per la sua attiva e costante disponibilità, per i suoi consigli e per il notevole supporto che ho ricevuto dalla sua esperienza e conoscenza dell'argomento.

Un ringraziamento particolare va a tutte le persone con cui ho vissuto le mie giornate nel laboratorio del centro di calcolo e soprattutto ai dottori Francesco Gregoretti, Diego Romano e Nicla Palladino per il supporto tecnico, operativo e morale.

Infine, i miei ringraziamenti si rivolgono ai miei genitori, alla mia famiglia e agli amici di sempre che, con il loro continuo, costante e instancabile sostegno morale, mi hanno mostrato una fiducia cieca e priva di incertezze, spronandomi sempre ad andare per la mia strada.

Introduzione.

Le pellicole cinematografiche costituiscono un patrimonio artistico e culturale inestimabile. Purtroppo, però, sono molto fragili e non sempre ben conservate: il tempo e l'uso frequente favoriscono il formarsi di polvere e muffa, perdita del colore, graffi, tagli e ulteriori danni meccanici.
Nonostante gli interventi di prevenzione e i continui trasferimenti da una pellicola all'altra, spesso è necessario intervenire, manualmente o meccanicamente, per riparare filmati permanentemente danneggiati.
I procedimenti di ***restauro*** e/o ***ricostruzione*** mirano a migliorare la qualità delle copie dei vecchi film, mediante un insieme di operazioni che restituiscano una versione quanto piu simile all'originale.
Precisamente, il *restauro* consiste nella correzione delle informazioni degradate, mentre la *ricostruzione* riguarda il ripristino di parti parzialmente o totalmente mancanti.

Il restauro tradizionale consiste in una prima fase di "*rigenerazione*", in cui si tenta di rimuovere lo strato di grasso tamponando delicatamente le due facce del negativo mediante dei panni imbevuti di soluzioni chimiche. Si procede, poi ad un ulteriore lavaggio in macchina, immergendo la pellicola in uno speciale liquido detergente. Per il trattamento delle deformazioni si utilizzano, invece, particolari metodi per estendere, ammorbidire e riumidificare la pellicola.
Segue la fase di "*restauro*", in cui si tenta di eliminare le ulteriori anomalie analizzando e trattando ogni fotogramma della pellicola singolarmente.

Evidentemente, tali procedimenti risultano molto lenti e dispendiosi e permettono di riparare solo un certo tipo di degradazioni.
I tempi di procedura oscillano da un minimo di qualche mese ad oltre due anni di lavorazione e non sempre i risultati ottenuti sono soddisfacenti. E' impossibile, ad esempio, eliminare i graffi sulla pellicola, oppure ridar vita ai colori sbiaditi.

Il restauro digitale basato sull'approccio numerico-computazionale, permette, mediante le moderne tecnologie di calcolo, di rendere automatico un gran numero di fasi di riconoscimento e di eliminazione delle anomalie presenti in un filmato, accelerando i tempi e soprattutto riducendo i costi rispetto ai metodi tradizionali. Inoltre, consente di superare i limiti qualitativi imposti dalle procedure ottico-chimiche. Ad esempio, è possibile *ricostruire* parti dell'immagine che sono andate definitivamente perdute o inserire nuovi fotogrammi.

L'impiego delle tecnologie digitali nel campo cinematografico nasce, in primo luogo, per i cosidetti ***effetti speciali***, ovvero per creare ambienti, situazioni, e perfino personaggi, non realizzabili con normali riprese o che comporterebbero costi di produzione proibitivi.
Nel 1989 **[31]**, l'approccio digitale per il restauro cinematografico fu introdotto come un progetto arduo e a lunga scadenza. Tuttavia, quattro anni dopo, nel 1993, la *Walt Disney Studio* in collaborazione con la *Kodak*, presentò il primo film restaurato interamente in digitale, "*Biancaneve e i sette nani*" **[14]**, mediante i primi software non automatici per la rimozione dei difetti.
Quasi contemporaneamente, l'azienda cinematografica *FOX-Vision* iniziò il trasferimento in digitale di tutte le pellicole cinematografiche in bianco e nero in proprio possesso.

A partire da questi primi due episodi, negli ultimi vent'anni il restauro digitale di pellicole cinematografiche e video ha ricevuto grande attenzione da parte della comunità scientifica.
In quest'ambito, parallelamente al potenziamento dei supporti digitali,

nascono continuamente progetti che mirano a sviluppare processi specifici per la correzione dei difetti caratteristici dei vecchi film, mediante sistemi di restauro sempre più veloci e affidabili.
In particolare, le ricerche sui problemi posti dal restauro hanno dato origine a numerosi progetti in collaborazione fra università e aziende.
Tra i più noti vi sono: AURORA (1995), EUREKA LIMELIGHT (1997), FRAME (1998), LOWRY DIGITAL IMAGE (2000); i primi dispositivi di restauro digitale automatici e semi-automatici in grado di correggere i contrasti e le alterazioni di luminosità e di trattare le abrasioni della pellicola ricostruendo, mediante operazioni morfologiche o interpolazione, parti mancanti o troppo deteriorate.

La presente tesi si inserisce in questo contesto, con l'obbiettivo di affrontare le problematiche numeriche relative alla progettazione, implementazione e valutazione di algoritmi per il restauro digitale cinematografico.
Precisamente, lo scopo di questo lavoro consiste nel presentare un algoritmo numerico, implementato in ambiente di calcolo parallelo, per il restauro di sequenze di immagini digitali affette della presenza di ***blotch*** (sporcizia). L'approccio numerico consiste in una accurata stima del moto della sequenza e nell'impiego di tali informazioni sia per il riconoscimento che per l'eliminazione dei difetti.
L'intero sistema costituisce un software automatico suddiviso in tre moduli consecuitivi ed eventualmente indipendenti, sviluppati in ambienti computazionali di tipo MIMD a memoria distribuita (IBM SP RS6000).

I capitoli che seguono forniscono una descrizione delle metodologie impiegate e dei corrispondenti aspetti numerici.
Una guida all'utilizzo del software sviluppato e i più significativi risultati sperimentali ottenuti, ne illustrano le prestazioni in termini di accuratezza, efficienza e complessità. Segue un breve profilo per ogni capitolo.

Nel ***Capitolo 1***, si descrivono le funzioni principali del restauro digitale cinematografico, fornendo una classificazione delle più comuni degradazioni presenti in un filmato e illustrandone le cause principali.

Nel ***Capitolo 2***, si esamina il problema del riconoscimento e dell'eliminazione di un particolare difetto denominato *blotch*, analizzandone le proprietà spazio-temporali. Quindi, si definisce il problema matematico relativo al restauro come un problema inverso e si propone un algoritmo per la risoluzione, basato su un metodo di regolarizzazione *edge-preserving*.

Nel ***Capitolo 3***, si definiscono le condizioni iniziali del problema di regolarizzazione corrispondente al restauro, mediante l'uso delle informazioni relative al moto apparente (*flusso ottico*) della sequenza di immagini. In particolare, si descrive un algoritmo per la stima del flusso ottico, anch'esso basato su una tecnica di regolarizzazione *edge-preserving*.

Nel ***Capitolo 4***, si fornisce una descrizione dei metodi illustrati nei capitoli precedenti e si presenta l'intero sistema di restauro, introducendo l'analisi in multirisoluzione e l'utilizzo di operazioni morfologiche, al fine di raffinare e ottimizzare i risultati.

Nel ***Capitolo 5***, si descrive in dettaglio il software, implementato in ambiente di calcolo parallelo, analizzando le complessità computazionali, la robustezza e l'efficienza degli algoritmi progettati.

Infine, nel ***Capitolo 6***, sono riportati i risultati sperimentali ottenuti su diversi tipi di sequenze test, costituite da immagini sintetiche e reali, e le relative osservazioni e conclusioni.

Capitolo 1

Il restauro digitale dei film cinematografici.

Si stima che almeno il 90% dei film muti siano andati definitivamente danneggiati, così come più del 50% dei film girati prima degli anni '40.

Le cause principali dei deterioramenti sono da addurre ad una cattiva conservazione, che favorisce il formarsi di muffa, umidità, polvere e sporcizia; oppure ad un'impropria manipolazione: la pellicola spesso risulta piegata o presenta impronte digitali impresse su immagini successive, aggiunte fatte con lo scotch o con pezzi di altre pellicole.
Altri difetti sono dovuti all'erosione della macchina da presa, che rovina la pellicola producendo lunghi e profondi graffi permanenti.
Inoltre, il tipo di materiale utilizzato per la pellicola stessa è soggetto ad alti rischi di combustione e nel tempo ad acidità e ulteriori reazioni chimiche, che sono causa di restringimento o deformazione.

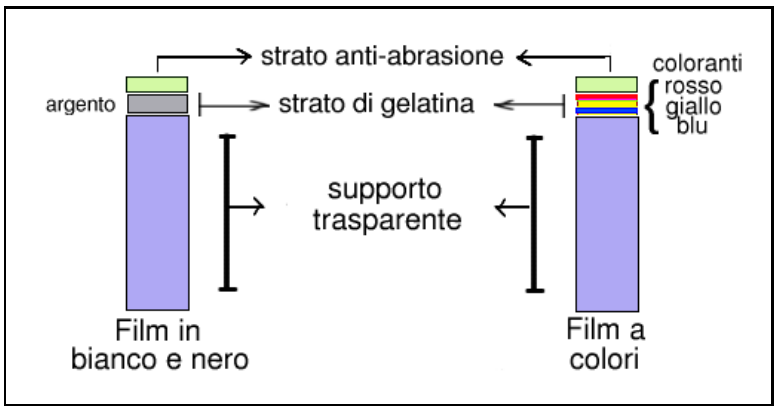

Figura 1: Pellicola cinematografica (bianco e nero/colori) - Sezione trasversale.

1.1 Caratteristiche della pellicola.

La pellicola cinematografica è un nastro continuo di materiale plastico costituito di tre strati (fig **1**):

- un ***supporto trasparente***, che fino alla fine degli anni '50 era per lo più in *nitrato di cellulosa* (o ***celluloide***), un materiale altamente infiammabile e facilmente soggetto a deformazioni.
  Oggi si utilizza il *triacetato di cellulosa* o il *poliestere,* che tra tutti si è rivelato il più stabile e duraturo;

- uno ***strato di gelatina*** composto da emulsioni fotosensibili a base d'argento, nel caso di una pellicola in bianco e nero, oppure a base di coloranti vegetali, nel caso di una pellicola a colori.
  Le emulsioni, impressionate dalla luce, generano le immagini del filmato;

- uno ***strato anti-abrasione***, che fornisce un rivestimento superficiale per proteggere le emulsioni dalla polvere e dagli agenti chimici.

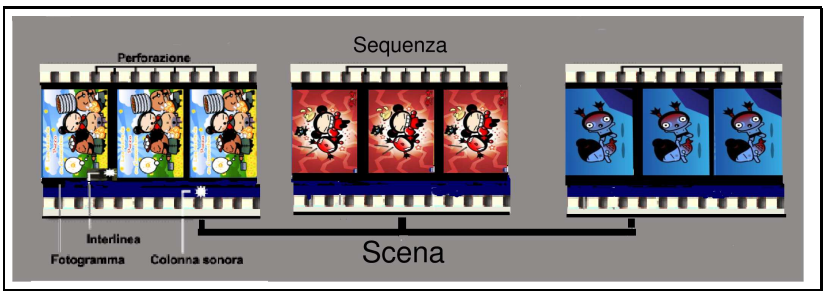

Figura 2: Pellicola cinematografica - Suddivisione per fotogrammi, sequenze, scene.

Sul nastro è impressa (fig. **2**) una lunga sequenza di ***fotogrammi*** (le immagini) posti uno di seguito all'altro all'incirca al centro e separati fra loro da un margine chiamato ***interlinea***. Sulla destra di ogni singolo fotogramma vi è una banda scura su cui è registrata la ***colonna sonora***, ovvero la pista sonora analogica.
I fotogrammi e la colonna sonora sono fiancheggiati su ciascun lato da un serie di fori detti ***perforazioni***, affinchè i rulli dentati del proiettore possano agganciare la pellicola e farla muovere in modo regolare.

Complessivamente, una successione di fotogrammi costituisce una ***sequenza*** cinematografica, ovvero una ripresa non interrotta. Un insieme di sequenze definisce una ***scena*** e, infine, più scene costituiscono l'intero *prodotto cinematografico*.

1.2 Il sistema di restauro digitale.

Gli attuali sistemi di elaborazione digitale per il restauro di sequenze di immagini, comprendono le seguenti operazioni:

- ***digitalizzazione*** *del filmato*,
- ***elaborazione*** *del filmato digitalizzato,*
- ***stoccaggio*** *e* ***restituzione*** *alla pellicola.*

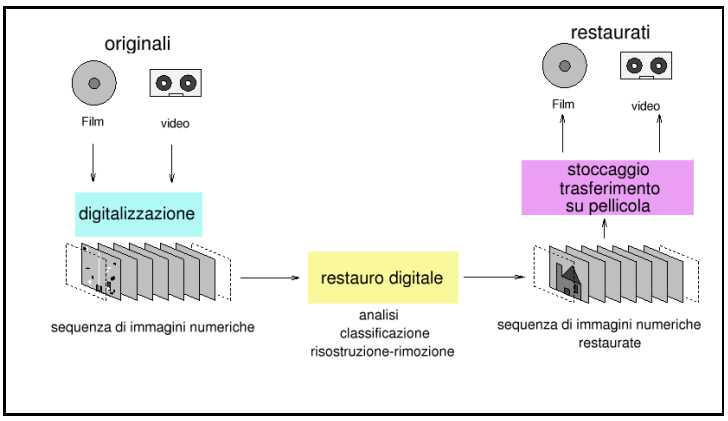

Figura 3: Schema generale del restauro digitale.

Ognuna di queste fasi (fig **3**) deve essere inclusa, necessariamente, all'interno del sistema di restauro, ma la ripartizione fisica delle macchine che svolgono tali funzioni può essere molteplice.
E' possibile costruire un unico corpo completo, ma anche suddividerlo in più stazioni di lavoro, conservando su un unico server le immagini da trattare oppure distribuendole ad ognuno di esse.

1.2.1 Digitalizzazione e restituzione alla pellicola.

Digitalizzazione, stoccaggio e restituzione finale alla pellicola costituiscono le fasi di "interfaccia" tra il filmato in forma analogica e le immagini in formato digitale e, di fatto, hanno un rilievo fondamentale nelle prestazioni dell'intero sistema di restauro.
Per una corretta elaborazione è necessario che la fase di digitalizzazione avvenga nella risoluzione massima possibile: l'aumentare delle informazione è di ausilio fondamentale per il riconoscimento dei difetti.
Allo stesso modo, i procedimenti di stoccaggio e stampa non devono alterare l'originalità del filmato, tanto meno i risultati ottenuti.

Il procedimento di ***digitalizzazione*** di un filmato avviene mediante un apparecchio di acquisizione (***scanner***) che realizza la conversione della serie di fotogrammi, che costituiscono l'intera pellicola, in un insieme di "immagini numeriche".
Le immagini, in formato analogico, vengono catturate dal dispositivo ad intervalli discreti di tempo $t \in [t_0, ..., t_n]$ (fig. **4**). In questo modo si ottiene una sequenza $\{\Omega_t\}_{t\in[t_0,...,t_n]}$, costituita da matrici bidimensionali di $N \times M$ picture element (***pel***), detti anche ***pixel***.

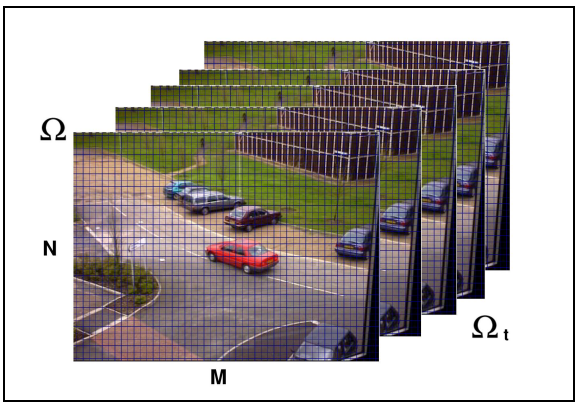

Figura 4: Sequenza digitale

Il *pixel* rappresenta il valore, codificato in digitale, dell'intensità luminosa della sequenza di immagini in quel punto.
Se il filmato è in bianco e nero, ogni immagine sarà costituita da pixel il cui valore misura l'intensità del grigio corrispondente al punto che rappresenta. Se invece, si tratta di una sequenza a colori, un solo valore non è sufficiente. La più nota rappresentazione delle immagini a colori in formato digitale associa tre valori ad ogni pixel, basandosi sul fatto che tutti i colori possono essere ottenuti con un opportuna miscela di tre colori di base: rosso, verde e blu (come nel caso analogico).
Tale rappresentazione viene detta ***RGB*** (Red, Green, Blue) e fornisce il peso delle componenti di colore rosso, verde e blu per lo specifico pixel.

Inoltre, quando un segnale analogico viene convertito in un segnale

digitale esso viene ***quantizzato***.
La quantizzazione è il processo mediante il quale un insieme di valori continuo viene associato in modo univoco ad un insieme finito di simboli. Per ogni pixel risulta, quindi, definita la quantità ***pixel depht*** (profondità del pixel), ovvero il numero di bit di informazione associati al pixel.
Per un'immagine in bianco e nero, si usa generalmente, una profondità di 8 bit; il che significa che i suoi pixel possono assumere $256 = 2^8$ valori diversi, o in altre parole che ci possono essere al massimo 256 diverse intensità di grigio per ogni pixel. Nelle rappresentazioni a colori *RGB,* si è soliti usare un byte per ogni componente, quindi 24 bit per ogni pixel.

Infine, per definire una sequenza, si utilizza il parametro ***frame rate***, che indica il numero di immagini che vengono visualizzate al secondo durante la proiezione del filmato. Comunemente il valore di questo parametro viene indicato in ***fps*** (fotogrammi al secondo). L'illusione del movimento si ottiene già con 12 immagini al secondo, ma in generale si usano 24 immagini al secondo per i formati analogici e 25/30 per i formati video-VHS.

Al termine del procedimento di *restauro,* i risultati ottenuti, ancora in forma digitale, sono sottoposti ad una fase preliminare di ***stoccaggio,*** in cui la sequenza di immagini numeriche viene rimontata e conservata opportunamente.

Infine, nella fase di ***ripristino***, il filmato in digitale è riconvertito in forma analogica e restituito alla pellicola mediante un apparecchio di stampa (***recorder***) che sfrutta un dispositivo basato su una tecnologia laser, con la qualità di una ripresa cinematografica.

1.2.2 Elaborazione numerica delle immagini.

L'effettivo interesse di questo lavoro di tesi riguarda il procedimento di ***elaborazione numerica delle immagini***.
In generale, attraverso lo sviluppo e l'applicazione di metodologie dell'a-

nalisi numerica si mira ad aumentare il grado di automazione del processo di restauro e, quindi, a minimizzare il tempo globale e i costi di esecuzione.
Precisamente, il sistema di elaborazione numerica deve risultare ***robusto***, ***efficiente***, ***automatico*** e ***veloce***. Ciò può essere realizzato combinando, opportunamente, architetture di calcolo avanzate (*hardware/software*) e metodologie efficienti (*algoritmi*).

E' difficile definire precisamente l'architettura di calcolo più adeguata alla risoluzione del problema. E' evidente , però, che il restauro numerico dei film concerne l'analisi di sequenze molto lunghe[1], per cui le operazioni richiedono, generalmente, tempi di elaborazione elevati.
Le soluzioni possono essere numerose e le tecnologie evolvono molto rapidamente. In generale, il sistema di calcolo utilizzato dovrebbe consentire un trattamento rapido, quindi un'elevata potenza di calcolo (*super calcolatori*, *claster*, *sitemi multiprocessore*) e, al tempo stesso, deve poter essere utilizzato anche da personale non necessariamente specializzato (*interfacce grafiche*, *moduli organizzati*).

Inoltre, per preservare la qualità del filmato stesso un'immagine non può assolutamente essere filtrata o elaborata globalmente, una tale operazione è nettamente percepibile durante la proiezione del film e spesso vi è il rischio di introdurre nuovi difetti, anche se se ne eliminano molti altri.
Dunque, i difetti devono essere identificati e manipolati localmente in maniera automatica, nel senso che, a parte la scelta iniziale di pochi parametri, la procedura di restauro deve essere capace di trattare un'intera sequenza senza intervento dell'operatore.
Risulta, quindi, necessaria una preliminare analisi e classificazione dei difetti e delle loro caratteristiche.

[1] 100 minuti di un filmato a colori in 2k byte corrispondono a 2000 linee di 1500 pixel. In genere, per 144000 immagini si ha un totale di 1300 miliardi di pixel.

1.3 Classificazione dei difetti.

La tabella **1**, presentata dalla "*Commission Supérieure Technique de l'image et du son*" **[14]**, costituisce una prima ripartizione dei classici deterioramenti presenti in un filmato in base alla necessità dell'intervento di un operatore nelle fasi di riconoscimento e rimozione.

	Rilevamento interattivo	**Rilevamento automatico**
Correzione interattiva	Riflessi Patine o strati di grasso	Abrasioni della pellicola, Macchie non stazionarie e ampie
Correzione automatica	Graffi fissi Macchie statiche Instabilità luminose locali	Capelli, Polvere Macchie e garffi non stazionari, Vibrazioni, Alereazioni della luminosità

Tabella 1: Classificazione dei difetti.

Nella colonna di sinistra della tabella sono riportate tutte le imperfezioni che non possono essere rilevate automaticamente; ovvero tutti quei difetti che possono essere confusi con elementi dell'immagine. In tal caso, l'identificazione non può farsi se non con l'ausilio di informazioni contestuali.

Nella parte superiore, invece, si collocano i difetti che non possono essere corretti automaticamente. Le ragioni principali sono dovute alla perdita totale delle informazioni, oppure all'impossibilità di definire un modello matematico rigoroso che descriva il problema.
Questo tipo di classificazione, ripresa in **[24]**, influenza anche la scelta e lo sviluppo del software di restauro, che può essere di tre tipologie:

- *sistema interamente manuali* (es: ***Paint***),
- *sistema automatico e manuale* (es: ***Frame***, ***Revival***)
- *sistema completamente automatico* (es: ***Archangel***)

Particolare interesse destano i difetti per cui i procedimenti di individuazione e rimozione possono essere completamente o in parte automatizzati. Questa categoria comprende tutte le anomalie per le quali il modello matematico è univoco e consente di stabilire una regola automatica di correzione. Il grado di automazione, per le fasi di riconoscimento e rimozione, dipende dal tipo di informazione che caratterizza il difetto in esame.
Infatti, i difetti possono essere ulteriormente classificati in base al tipo di informazione che forniscono **[15]**:

- *informazione spaziale:* estensione dell'area del difetto nella singola immagine della sequenza,
- *informazione temporale:* estensione del difetto nella sequenza di immagini,
- *quantità di informazione perduta.*

Sulla base delle informazioni di tipo spaziale, le degradazioni si suddividono in due classi fondamentali: difetti *globali* e difetti *locali*.
I ***difetti globali*** sono tutte quelle degradazioni o anomalie che si distribuiscono su tutta l'immagine, tra cui:

- l'***alterazione della luminosità*** o ***del colore***; può essere sia periodica che casuale e dipende da svariati fattori, ad esempio da reazioni chimiche, che favoriscono il degrado dell'emulsione, oppure dall'ossidazione dei materiali che compongono lo strato di gelatina sbiadendo i colori,
- le ***vibrazioni*** o ***instabilità posizionali,*** spesso dovute a problemi di sincronizzazione durante il passaggio del film da una pellicola all'altra oppure nel trasferimento di un film dalla pellicola a un supporto video-VHS;

- i ***frame mancanti***, che possono essere generati da abrasioni estese o dalla rottura della pellicola.

I ***difetti locali*** sono posizionati in zone limitate all'interno dell'immagine, ad esempio:

- il ***rumore,*** che può essere dovuto al degrado fisico della pellicola oppure a problemi di ricezione o trasmissione del segnale.

La differenza sostanziale fra questi due tipi di difetti è che, sia per l'individuazione che per la rimozione dei difetti locali, è possibile utilizzare le informazioni contenute nell'immagine stessa in zone adiancenti a quella corrotta.

Sulla base delle informazioni temporali, invece, i difetti possono essere distinti in: difetti *stabili* e difetti *aleatori*.
I ***difetti stabili*** o ***fissi***, appaiono esattamente nella stessa posizione in più fotogrammi consecuitivi, ad esempio:

- i ***graffi verticali bianchi*** o ***neri***; i più diffusi sono i cosidetti "graffi lineari", abrasioni del sottile strato di emulsione del nastro di supporto, che percorrono interi fotogrammi dal basso verso l'alto lungo la direzione di scorrimento della striscia di pellicola.

I ***difetti aleatori*** o ***mobili*** (*one-frame*), difficilmente si trovano nella stessa posizione nel passaggio da un fotogramma all'altro e, dunque, la loro posizione non è fissa nel tempo, tra questi:

- i ***blotch***, con questo termine si indicano anomalie di diversa natura, tra cui: polvere, macchie, lacerazioni, capelli, abrasioni, graffi non stabili o obliqui. I blotch sono dovuti a svariate cause, ma soprattutto ad un'impropria manipolazione o archiviazione della pellicola.

Per l'individuazione e la rimozione dei difetti mobili è possibile utilizzare le informazioni contenute nei fotogrammi adiacenti a quello corrotto. Nel caso dei difetti fissi, invece, l'informazione contenuta nei fotogrammi adiacenti è corrotta allo stesso modo di quella dell'immagine da restaurare, per cui è in genere inutilizzabile, almeno per la rimozione.

Infine, i difetti possono essere classificati anche in base alla quantità di informazione persa, che può essere:

- ***totale*** (*missing data*), l'informazione originaria contenuta in una regione di uno o più fotogrammi è andata totalmente perduta,
- ***parziale***, l'informazione originaria contenuta in una regione di uno o più fotogrammi è corrotta, ma non completamente assente.

Nel caso di perdita totale delle informazioni, il difetto deve essere ricostruito a partire da eventuali informazioni spazio-temporali. Al contrario, nel caso di perdita parziale è possibile ricostruire l'informazione originaria anche a partire da un modello idoneo della degradazione avvenuta.

Alcuni tipi di difetti forniscono, contemporaneamente, informazioni di diversa natura (spaziale, temporale, quantitativa). In tal caso, è possibile scegliere l'informazione che conduce al più opportuno metodo risolutivo o meglio, combinare le informazioni a disposizione rappresentandole con un unico modello.

Capitolo 2

Riconoscimento ed eliminazione dei blotch.

Quando una particella "estranea" si deposita sulla superficie della pellicola oppure l'invecchiamento della pellicola stessa causa la perdita di parte della gelatina protettiva che la ricopre, può capitare che su un fotogramma si formino macchie chiare o scure, comunemente note con il termine: ***blotch***.
Questo tipo di difetto è, in genere, dovuto alle (precarie) condizioni di conservazione o ad una impropria manipolazione della pellicola cinematografica.

I blotch fanno parte della categoria dei *difetti locali mobili* e sono una delle degradazioni più comuni nel metraggio archiviato.
Essi comprendono grandi parti di sporcizia, capelli, impronte digitali, muffa, acqua, graffi instabili, fori nell'emulsione della pellicola e polvere

distribuita e sono presenti spesso su un solo fotogramma (1/25 di secondo), per cui l'effetto visivo, durante la proiezione della pellicola, è quello di piccoli flash che appaiono e scompaiono molto velocemente (fig. **1**).

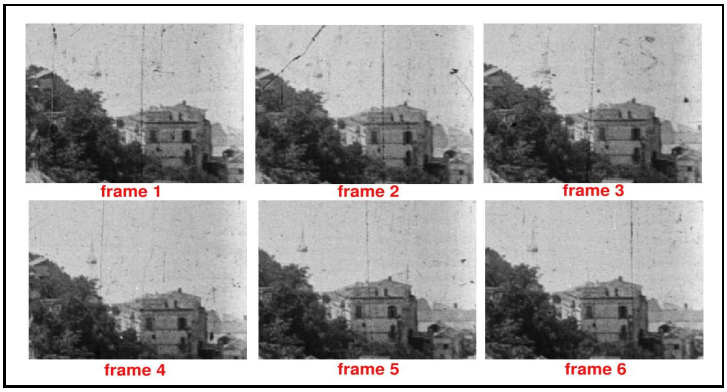

Figura 1: Esempio di una squenza di immagini digitali corrotta da *Blotch*.

Riassumendo, in una sequenza di immagini, questo tipo di difetto può essere caratterizzato dalle seguenti proprietà **[3]**:

1. *l'intensità luminosa di un blotch è significativamente differente dalle intensità delle zone non corrotte*;
2. *i blotch sono temporalmente indipendenti, ovvero difficilmente si presentano, soprattutto nella stessa posizione, in fotogrammi adiacenti*;
3. *i blotch formano regioni connesse*.

2.1 Il modello di degradazione.

In una sequenza di immagini digitali un fotogramma affetto da blotch, per la ***proprietà 1***, è caratterizzato da punti con intensità luminosa nettamente differente da quella delle zone non corrotte. A partire da tale ipotesi, è possibile definire il modello di degradazione che rappresenta

numericamente il problema.
Indicato con $J \subset \Re$ un intervallo di tempo e con $\Omega \subset \Re^2$ il piano immagine, una **sequenza di immagini** è definita dalla funzione:

$$t \in J \subset \Re \longrightarrow (P(t), t) \in \Omega \times J$$

quindi, l'**intensità luminosa** di una *sequenza di immagini* "ideale", ovvero non affetta da alcun difetto, è una funzione dello spazio e del tempo:

$$I: \ t \in J \longrightarrow I(P(t), t) \in \Re$$

Al contrario, l'**intensità luminosa** di una *sequenza di immagini*, "eventualmente danneggiata", è descritta dalla funzione:

$$\widetilde{I}: \ t \in J \longrightarrow \widetilde{I}(P(t), t) \in \Re$$

che, $\forall t \in J$, si può scrivere come**[3]** :

$$\widetilde{I}(P(t), t) = (1 - b(P(t), t))\ I(P(t), t) + b(P(t), t)\ I^B(P(t), t)$$

dove:

- la funzione b è la ***funzione caratteristica*** *incognita* del sottoinsieme $(B \times J) \subset (\Omega \times J)$, che rappresenta il dominio del blotch e definisce, dunque, la posizione dei punti danneggiati della sequenza di immagini, come segue:

$$b: t \in J \rightarrow b(P(t), t) = \begin{cases} 1 & se\ (P(t), t) \in (B \times J) \subset (\Omega \times J) \\ 0 & se\ (P(t), t) \in (\Omega \times J) \setminus (B \times J) \end{cases}$$

- la funzione I^B definisce l'***intensità luminosa*** *incognita*, corrispondente ai punti della sequenza corrotti da blotch:

$$I^B: t \in J \longrightarrow I^B(P(t), t) \in \Re$$

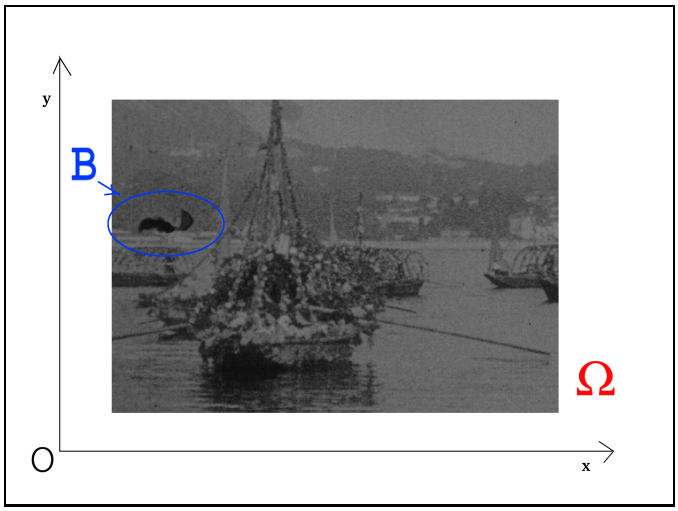

Figura 2: Fissato un istante $t \in J$, il dominio del blocth risulta definito dall'insieme $B \subset \Omega$.

In altre parole, nella zona non corrotta $\forall\,(P(t),t) \in (\Omega \times J) \setminus (B \times J)$, si ha:

$$b\,(P\,(t)\,,\,t) = 0 \Longrightarrow \widetilde{I}\,(P\,(t)\,,\,t) = I\,(P\,(t)\,,\,t)\,,$$

mentre, nel dominio del blotch $\forall\,(P\,(t)\,,\,t) \in (B \times J) \subset (\Omega \times J)$, risulta:

$$b\,(P\,(t)\,,\,t) = 1 \Longrightarrow \widetilde{I}\,(P\,(t)\,,\,t) = I^{B}\,(P\,(t)\,,\,t)$$

Quindi, nota la funzione b, il recupero delle informazioni corrispondenti alle zone danneggiate (fig. **2**), si può ricondurre al sottoinsieme $(B \times J)$ di $(\Omega \times J)$ e alla funzione che in esso assume valore:

$$I^{B} : t \in J \longrightarrow I^{B}\,(P\,(t)\,,t) \in \Re$$

In definitiva, come la maggior parte dei problemi di restauro, il problema descritto rientra tra i problemi *inversi mal posti* **[7]**.
Le difficoltà legate al restauro di immagini corrotte da blotch è dovuto, principalmente, a una *mal posizione* intrinseca del problema rappresentato dall'equazione:

$$\widetilde{I} = \varphi(I) = [1 - b]\, I + bI^B, \quad \forall (P(t), t)$$

che rende necessario *determinare*, innanzitutto, la funzione caratteristica b, nota $\widetilde{I}$ (*individuazione del blotch*) e quindi nel *correggere* l'intensità corripondente alla zona danneggiata I^B, nota b (*rimozione del blotch*).

I metodi proposti per il restauro di immagini affette dalla presenza di blotch, si basano su approcci di tipo statistico **[3, 21, 33]**, o su combinazione di opportune proprietà morfologiche e confronti temporali **[15]**.
La maggior parte di essi tratta la ricostruzione dei dati appartenenti a una regione connessa B del dominio dell'immagine Ω, mediante interpolazione dei valori adiacenti a B (***inpainting digitale***).
Ad esempio in **[3]**, il modello di restauro impiega un'***interpolazione spaziale*** dei valori dell'immagine in esame che si trovano in prossimità dei blotch.
Altre tecniche **[6, 10]**, propongono un modello più complesso, ma sicuramente più attendibile basato sia sull'uso delle informazioni relative all'immagine in esame, che sull'impiego delle informazioni relative a immagini ad essa adiacenti nella sequenza effettuando, dunque, un'***interpolazione spazio-temporale***.

La scelta del metodo, in generale, è legata alla grandezza e al tipo di difetto: quanto più i difetti sono grandi tanto più un'interpolazione puramente spaziale può dare risultati imprecisi; al contrario, per difetti stabili, ovvero presenti su più frame, un'interpolazione spazio-temporale non dà alcun contributo alla ricostruzione dell'informazione perduta.

2.2 La traiettoria del moto.

Il modello proposto, per il riconoscimento e la rimozione dei blotch si basa sull'utilizzo delle informazioni fornite dal moto di ogni punto della sequenza.

Definizione 1: Per ogni punto $(P(t), t)$ appartenente al dominio $\Omega \times J$, la ***traiettoria del moto*** L è la linea, o arco di linea, luogo delle successive posizioni occupate da $P(t)$ al variare di $t \in J$ (fig. **3**).

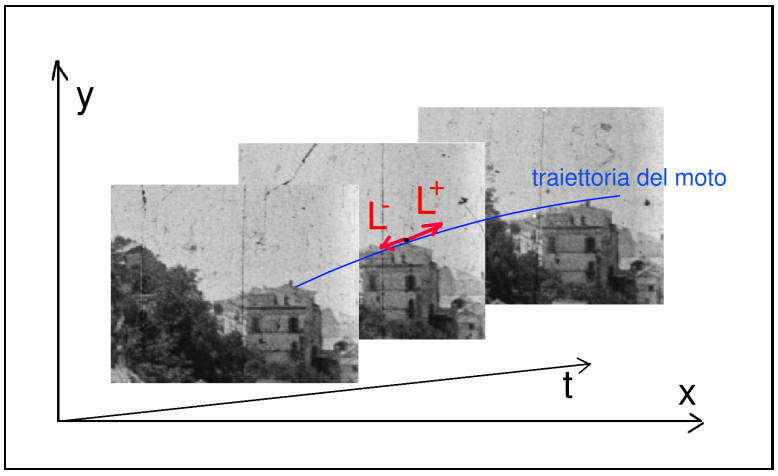

Figura 3: L'arco di linea L, che definisce la traiettoria del moto di un punto.

Considerati tre istanti successivi della sequenza: $t - \Delta t$, t, $t + \Delta t$, è possibile dare le seguenti definizioni.

Definizione 2: Per ogni punto $(P(t - \Delta t), t - \Delta t) \in \Omega \times J$, il vettore:

$$v^+(t - \Delta t) = \left(v_x^+(t - \Delta t),\ v_y^+(t - \Delta t)\right)$$

definisce il ***campo di moto*** nell'intervallo di tempo Δt^+, ovvero dall'istante $t - \Delta t$ all'istante t. Mentre, per ogni punto $(P(t + \Delta t), t + \Delta t)$, il vettore:

$$v^-(t + \Delta t) = \left(v_x^-(t + \Delta t), v_y^-(t + \Delta t)\right)$$

definisce il ***campo di moto*** nell'intevallo di tempo Δt^-, ovvero dall'istante $t + \Delta t$ all'istante t.

Definizione 3: Considerati gli spostamenti, di ogni punto dell'immagine, nell'intervallo di tempo Δt^+ (*verso positivo del moto*):

$$\begin{cases} \Delta x^+ = \Delta t^+ \cdot v_x^+(t - \Delta t) \\ \Delta y^+ = \Delta t^+ \cdot v_y^+(t - \Delta t) \end{cases}$$

la ***direzione positiva della traiettoria del moto*** può essere rappresentata mediante il vettore $L^+ = (\Delta x^+, \Delta y^+)$. Allo stesso modo, se gli spostamenti, di ogni punto dell'immagine, nell'intervallo di tempo Δt^- (*verso negativo del moto*), sono:

$$\begin{cases} \Delta x^- = \Delta t^- \cdot v_x^-(t + \Delta t) \\ \Delta y^- = \Delta t^- \cdot v_y^-(t + \Delta t) \end{cases}$$

la ***direzione negativa della traiettoria del moto*** è definita dal vettore $L^- = (\Delta x^-, \Delta y^-)$

Definizione 4: Se

$$L: \quad \begin{cases} x = x(t) \\ y = y(t) \end{cases}$$

è l'equazione parametrica della traiettoria del moto L, allora la ***derivata direzionale***[1] dell'intensità luminosa I, *lungo la direzione positiva* della traiettoria L, è :

$$\frac{\partial I}{\partial L^+} = \lim_{\Delta t^+ \to 0} \frac{I\left(x\left(t-\Delta t\right)+\Delta x^+, y\left(t-\Delta t\right)+\Delta y^+, t-\Delta t\right) - I\left(x\left(t\right), y\left(t\right), t\right)}{\Delta t^+}$$

mentre, la ***derivata direzionale*** dell'intensità I, lungo la *direzione negativa* della traiettoria L, è:

$$\frac{\partial I}{\partial L^-} = \lim_{\Delta t^- \to 0} \frac{I\left(x\left(t+\Delta t\right)-\Delta x^-, y\left(t+\Delta t\right)-\Delta y^-, t+\Delta t\right) - I\left(x\left(t\right), y\left(t\right), t\right)}{\Delta t^-}$$

Le definizioni riportate consentono di formalizzare l'assunzione secondo cui: "***l'intensità luminosa di una sequenza di immagini resta costante lungo la traiettoria del moto***" **[5, 20]**.
Infatti, tale ipotesi implica che le derivate direzionali dell'intensità luminosa I, valutate rispettivamente lungo la traiettoria (*positiva* L^+ e *negativa* L^-) del moto, sono uguali a zero. Ovvero:

$$\frac{\partial}{\partial L^+} I\left(P\left(t\right), t\right) = 0 \tag{1}$$

$$\frac{\partial}{\partial L^-} I\left(P\left(t\right), t\right) = 0 \tag{2}$$

[1] Considerata una funzione di due valiabili $f(x,y)$ e un vettore $\nu = (\nu_1, \nu_2) \neq (0,0)$, se esiste finito il limite del rapporto incrementale:

$$\lim_{h \to 0} \frac{f\left(x_0 + h\nu_1,\, y_0 + h\nu_2\right) - f\left(x_0,\, y_0\right)}{h}$$

allora tale valore si denota con $\frac{\partial}{\partial \nu} f\left(x_0,\, y_0\right)$ ed è detto ***derivata direzionale*** di f rispetto alla direzione ν, calcolata nel punto $(x_0,\, y_0)$

Il legame fra il campo di moto e l'intensità luminosa, espresso dalle relazioni (**1**) e (**2**), costituisce il punto di partenza ai metodi impiegati sia per l'*individuazione* e che per la *rimozione* dei blotch.

2.3 Individuazione dei blotch.

Il campo di moto, in una sequenza di immagini danneggiata, presenta delle discontinuità temporali in corrispondenza delle zone corrispondenti ai blotch. Dunque, al variare di $t \in J$, per alcuni punti $(P(t), t)$ del dominio $\Omega \times J$, si ha:

$$\frac{\partial}{\partial L^+}\widetilde{I}(P(t), t) \neq 0, \quad \frac{\partial}{\partial L^-}\widetilde{I}(P(t), t) \neq 0$$

Ciò implica che il dominio dei blotch per l'intera sequenza può essere caratterizzato dall'insieme:

$$B \times J = \left\{ (P(t), t) \in \Omega \times J : \quad \min\left\{ \frac{\partial \widetilde{I}}{\partial L^+}, \frac{\partial \widetilde{I}}{\partial L^-} \right\} \neq 0 \right\}$$

e quindi determinato valutando, innanzitutto, il campo vettoriale $\overrightarrow{v}(t)$, che definisce la traiettoria del moto in ogni punto del dominio Ω e ad ogni istante $t \in J$, e di seguito i vettori unitari L^- ed L^+, per il calcolo delle derivate direzionali:

$$\left| \frac{\partial \widetilde{I}}{\partial L^+} \right| = \left| \widetilde{I}\left(x(t-\Delta t) + \Delta x^+, y(t-\Delta t) + \Delta y^+, t-\Delta t\right) - \widetilde{I}(x(t), y(t), t) \right| + O\left(\triangle t^+\right)$$

$$\left| \frac{\partial \widetilde{I}}{\partial L^-} \right| = \left| \widetilde{I}\left(x(t+\Delta t) - \Delta x^-, y(t+\Delta t) - \Delta y^-, t+\Delta t\right) - \widetilde{I}(x(t), y(t), t) \right| + O\left(\triangle t^-\right)$$

In definitiva, la ***funzione caratteristica***, che rappresenta il dominio del blotch $B \times J$, risulta:

$$b\left(x\left(t\right),\, y\left(t\right),\, t\right) = \begin{cases} 1 & se\ \min\left\{\left|\frac{\partial \widetilde{I}}{\partial L^{+}}\right|,\, \left|\frac{\partial \widetilde{I}}{\partial L^{-}}\right|\right\} \neq 0 \\ \\ 0 & altrimenti \end{cases}$$

2.4 Rimozione dei blotch.

Nota la funzione *caratteristica* b e, quindi, l'informazione della posizione delle zone danneggiate nell'immagine, allora il recupero delle informazioni perdute si focalizza al dominio del blotch e riguarda la stima della funzione I^B ad esso corrispondente.
L'idea di base consiste nel calcolare I^B risolvendo un problema di *migliore approssimazione nel senso dei minimi quadrati*, del tipo:

$$\widetilde{I}^B = arg \min_{I^B} \left\| I^C - I^B \right\|_{L^2} \tag{3}$$

dove I^C rappresenta un'approssimazione iniziale della soluzione, la cui definizione sarà trattata nel capitolo successivo.

Il problema può essere risolto mediante un metodo di regolarizzazione *edge-preserving*, che impiega il funzionale TV (*totale variazione*) **[4]**:

$$E\left(I^B\right) = \int_B \left| I^C - I^B \right| d\underline{x} + \lambda \int_{B-\partial B} \phi\left(\left\| \nabla I^B \right\|\right) d\underline{x}$$

In pratica, a partire dal dato iniziale I^C e scegliendo opportunamente il valore del parametro λ, si determina la migliore approssimazione della funzione I^B, imponendo che siano preservate le discontinuità nei punti in

cui il gradiente ∇I^B è diverso da zero (ad esempio lungo i contorni).
Quindi, il problema:

$$I^B = \min_{I^B} E\left(I^B\right) \tag{4}$$

si riconduce ad un classico problema di regolarizzazione *edge-preserving* (appendice **A**), in cui la funzione ϕ deve essere tale che la soluzione risulti una funzione costante a tratti.
In questo modo, l'immagine corrisponde alla soluzione risulta costituita da regioni omogenee delimitate da contorni ben definiti, che può essere valutata utilizzando la decomposizione descritta in appendice **B**.
Precisamente, si introduce la *funzione duale* b per cui:

$$\phi(s) = \min_b \left[bs^2 + \psi(b)\right]$$

purchè $\phi\left(\sqrt{s}\right)$ sia concava, e il funzionale E risulta modificato come segue:

$$E^d\left(I^B\right) = \int_B \left|I^B - I_C\right|^2 d\underline{x} + \lambda \int_B \min_{L \le b \le M} \left(b\left|\nabla I^B\right|^2 \psi(b)\right) d\underline{x}$$

Da cui:

$$\min_{I^B} E^d\left(I^B\right) = \min_{I^B} \left\{\int_B \left|I^B - I^C\right|^2 d\underline{x} + \lambda \int_B \min_b \left(b\left|\nabla I^B\right|^2 + \psi(b)\right) d\underline{x}\right\}$$

$$= \min_{(I^B, b)} \left\{\int_B \left|I^B - I^C\right|^2 d\underline{x} + \lambda \int_B \left(b\left|\nabla I^B\right|^2 + \psi(b)\right) d\underline{x}\right\}$$

In definitiva, per determinare I^B bisogna valutare il minimo del funzionale:

$$E^d\left(I^B,\, b\right) = \int_B \left|I^B - I^C\right|^2 d\underline{x} + \lambda \int_B \left(b\left|\nabla I^B\right|^2 + \psi(b)\right) d\underline{x} \tag{5}$$

tenendo conto di entrambe le variabili che in esso intervengono $\left(I^B,\, b\right)$, mediante una *minimizzazione alternata* rispetto ad ogni variabile (appendice **C**).

Assegnate le condizioni iniziali $I_0^B = I^C$, si procede mediante lo ***schema del punto fisso alternato***:

$$\begin{cases} b^{n+1} = arg\min_{b\in L^2(B)} E^d\left(I_n^B,\, b\right) \\ \\ I_{n+1}^B = arg\min_{I^B\in W^{1,2}(B)} E^d\left(I^B,\, b^{n+1}\right) \end{cases}$$

Precisamente, si determina iterativamente la migliore approssimazione della funzione duale b, calcolando:

$$b^{n+1} = arg \min_{b\in L^2(B)} E^d\left(I_n^B,\, b\right)$$

cioè, risolvendo il problema (**5**) rispetto a b:

$$\min_b \int_B \left[b\left|\nabla I_n^B\right|^2 + \psi\left(b\right)\right] d\underline{x}$$

che ammette un'unica soluzione quando:

$$b^{n+1} = \frac{\phi'\left(\nabla I_n^B\right)}{2\left|\nabla I_n^B\right|}$$

Allo stesso modo si ricava la migliore approssimazione della funzione I^B calcolando:

$$I_{n+1}^B = arg \min_{I^B\in W^{1,2}(B)} E^d\left(I^B,\, b^{n+1}\right)$$

cioè risolvendo il problema (**5**) rispetto a I^B:

$$\min_{I^B} \int_B \left[\left(I^B - I^C\right)^2 + \lambda\, b^{n+1}\left|\nabla I_n^B\right|\right] d\underline{x}$$

imponendo che la derivata del funzionale sia nulla. In tal caso, si ha l'equazione di *Eulero-Lagrange*:

$$I_{n+1}^B - I_n^B + \lambda\, div\left(b^{n+1}\nabla I_n^B\right) = 0.$$

Schema del punto fisso alternato
$I_0^B = I^C$ ***repeat*** $b^{n+1} = \frac{\phi'\left(\nabla I_n^B\right)}{2\lvert\nabla I_n^B\rvert}$ $I_{n+1}^B = I_n^B - \lambda\, div\left(b^{n+1}\nabla I_n^B\right)$ ***until*** (convergenza**a**)

In termini matriciali, l'iterazione $n+1$ si può descrivere come un sistema lineare nelle due incognite $\left(I_{n+1}^B,\, b^{n+1}\right)$, la cui matrice dei coefficienti risulta essere una matrice triangolare superiore:

$$\begin{bmatrix} 1 & L\left(I_n^B\right) I_n^B \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_{n+1}^B \\ b^{n+1} \end{bmatrix} = \begin{bmatrix} G \star I_n^B \\ \frac{\phi'\left(\nabla I_n^B\right)}{2\lvert\nabla I_n^B\rvert} \end{bmatrix}$$

in cui, l'operatore divergenza è indicato con:

$$L\left(I_n^B\right) I_n^B = -div\left(b^{n+1}\nabla I_n^B\right) - \nabla \cdot \left[b^{n+1}\nabla I_n^B\right]$$

Capitolo 3

Interpolazione temporale.

Utilizzando opportunamente le informazioni relative al moto della sequenza, è possibile definire il dato iniziale I^C nella risoluzione del problema di regolarizzazione introdotto in **2.4**.
Precisamente, utilizzando la ***proprietà 2,*** secondo cui i blotch sono temporalmente indipendenti e difficilmente si presentano, nella stessa posizione, in fotogrammi adiacenti, l'intensità corrotta può essere recuperata "interpolando" lungo la traiettoria del moto.
Tale procedimento è anche detto ***compensazione del moto*** e consiste nel considerare un'immagine di riferimento nella sequenza e modificare le immagini, ad essa adiacenti, in modo da annullare il movimento.
Ad esempio, considerati tre istanti successivi della sequenza $t - \triangle t$, t, $t + \triangle t$, se è nota la traiettoria del moto e quindi gli spostamenti di ogni punto $(P(t), t)$, al variare di $t \in J$ è possibile far *coincidere* parti di immagini consecutive in corrispondenza temporale. Infatti, se sono noti gli

spostamenti $\Delta x^{\pm}$ e $\Delta y^{\pm}$, è possibile modificare l'intensità luminosa corrispondente agli istanti $t-\Delta t$ e $t+\Delta t$, in modo da ottenere due immagini la cui intensità luminosa coincide, punto per punto con quella dell'intensità luminosa corrispondente all'immagine centrale all'istante t (fig. **1**).

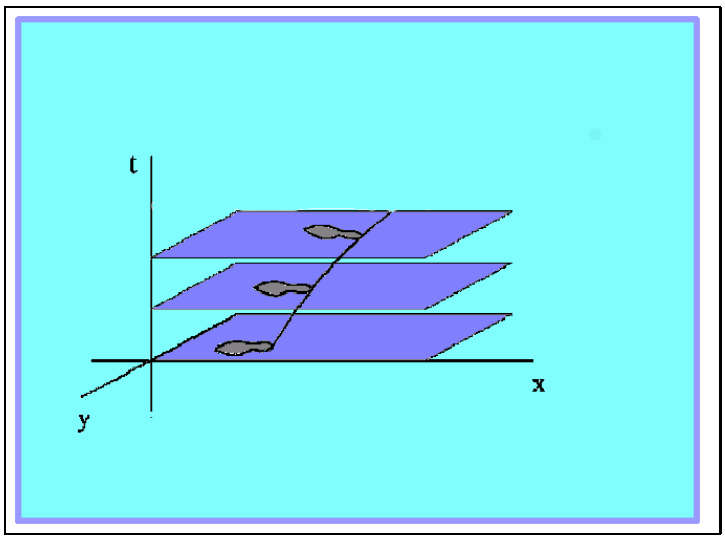

Figura 1: Compensazione del moto.

Precisamente, seguendo il punto $(P(t-\Delta t), t-\Delta t)$ lungo la tariettoria *positiva* del moto è possibile determinare la sua posizione al tempo t, usando gli spostamenti Δx^{+}, Δy^{+}:

$$\begin{cases} x(t) = x(t-\Delta t) + \Delta x^{+} \\ y(t) = y(t-\Delta t) + \Delta y^{+} \end{cases}$$

per cui si ottiene:

$$I^{C}_{t-\Delta t}(x(t), y(t), t) = I\left(x(t-\Delta t) + \Delta x^{+}, y(t-\Delta t) + \Delta y^{+}, t-\Delta t\right)$$

l'**immagine compensata in avanti, da** $t-\Delta t$ **a** t.
Al contrario, seguendo il punto $(P(t+\Delta t), t+\Delta t)$ lungo la tariettoria *negativa* del moto è possibile determinare la sua posizione al tempo t,

usando gli spostamenti Δx^{-}, Δy^{-}:

$$\begin{cases} x(t) = x(t + \Delta t) - \Delta x^{-} \\ y(t) = y(t + \Delta t) - \Delta y^{-} \end{cases}$$

da cui:

$$I^{C}_{t+\Delta t}(x(t),\, y(t),\, t) = I\left(x(t + \Delta t) - \Delta x^{-},\, y(t + \Delta t) - \Delta y^{-},\, t + \Delta t\right)$$

che rappresenta l'**immagine compensata all'indietro, da** $t + \Delta t$ **a** t. Tali immagini possono essere impiegate per definire la funzione I^{C}, come segue:

$$I^{C}(x(t), y(t), t) = \frac{I^{C}_{t-\Delta t}(x(t), y(t), t) + I^{C}_{t+\Delta t}(x(t), y(t), t)}{2},\ \forall t \in J$$

A tal fine è necessario determinare gli spostamenti $\Delta x^{\pm}$, $\Delta y^{\pm}$ che definiscono il *campo di moto* per ogni punto della sequenza.

3.1 Il campo di moto.

Il *campo di moto*, in una sequenza di immagini, è definito, in ogni suo punto, dal vettore che ne rappresenta lo spostamento, al variare del tempo $t \in J$ nello spazio euclideo $\Re^3$.
Sfortunatamente, nel passaggio dalla rappresentazione *3D* del mondo reale alla sua proiezione sul piano *immagine 2D*, il campo moto subisce una notevole perdita di informazioni. In generale, l'unica informazione disponibile risulta *l'intensità luminosa* di ogni pixel, che costituisce l'immagine in esame.
Per questo motivo, nella stima del campo di moto quello che in effetti viene rilevato è il "moto 2-D apparente" che prende il nome di ***Flusso Ottico*** (Fig. **2**).

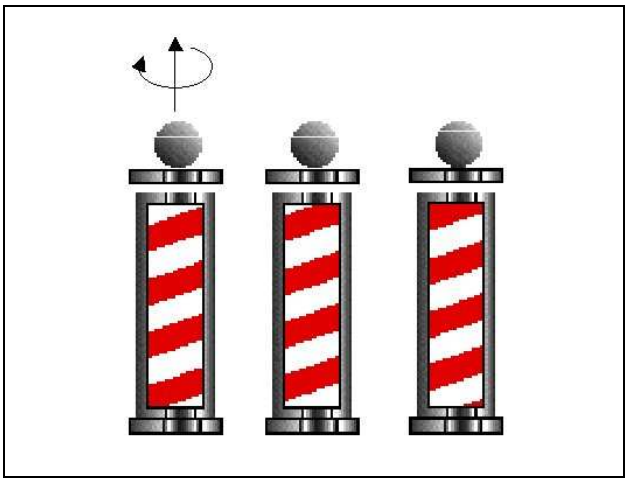

Figura 2: **Un esempio della differenza fra il campo di moto ed il flusso ottico.** Il reale campo di moto 3-D è rotatorio, ma dall'analisi della sequenza, rappresentata da immagini 2-D, si osserva che il flusso ottico risulta essere traslatorio verticale.

Il problema del calcolo del *flusso ottico* consiste nel valutare il campo vettoriale $\overrightarrow{v}(t)$, $\forall t \in J$, ovvero la funzione:

$$\overrightarrow{v} : t \in J \longrightarrow \overrightarrow{v}(t) = \frac{dP(t)}{dt} \in \Re^2$$

che descrive la variazione dell'*intensità luminosa* della sequenza nel tempo.

La risoluzione di tale problema si basa fondamentalmente sull'ipotesi secondo cui: "***l'intensità luminosa, di ogni punto della sequenza resta costante nel tempo***". Ciò implica che, fissato un punto $P_0 = P(t_0)$ nell'immagine iniziale della sequenza all'istante $t = t_0$, lungo la traiettoria del moto deve valere la relazione:

$$I(P(t), t) = I(P_0, t_0) \quad \forall t \in J$$

Tale relazione definisce l'**equazione costante del flusso ottico**:

$$\frac{d}{dt} I\left(P\left(t\right), t\right) = 0$$

che si può scrivere esplicitamente come segue:

$$\frac{d}{dt} I\left(x\left(t\right), y\left(t\right), t\right) = \frac{\partial I}{\partial x}\frac{d}{dt}x\left(t\right) + \frac{\partial I}{\partial y}\frac{d}{dt}y\left(t\right) + \frac{\partial I}{\partial t} = 0$$

Da quest'ultima, posto:

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right), \quad I_t = \frac{\partial I}{\partial t}, \quad \overrightarrow{v} = (v_x, v_y) = \left(\frac{d}{dt}x\left(t\right), \frac{d}{dt}y\left(t\right)\right)$$

e indicato con il simbolo $\times$ il *prodotto scalare* tra vettori, si ottiene l'***equazione costante del flusso ottico in forma vettoriale***:

$$\overrightarrow{v} \times \nabla I + I_t = 0 \tag{1}$$

Dall'equazione (**1**) si può ricavare solo la *componente normale* della velocità $\overrightarrow{v}$, considerando $s = \overrightarrow{v} \cdot \widehat{n}$, ovvero la proiezione del vettore $\overrightarrow{v}$ sulla normale $\widehat{n}$ alla traiettoria del moto:

$$\overrightarrow{v}_{\perp} = s\widehat{n} = \left(\overrightarrow{v} \cdot \widehat{n}\right)\widehat{n} = \left(\frac{-I_t}{\nabla I} \cdot \frac{\nabla I}{\|\nabla I\|_{L^2}}\right)\frac{\nabla I}{\|\nabla I\|_{L^2}} = \frac{-I_t \nabla I}{\|\nabla I\|_{L^2}^2}$$

dove $\cdot$ indica il prodotto *vettoriale* tra vettori.
Quindi, nelle regioni dell'immagine in cui il gradiente della funzione I, ha un'unica direzione[1] è possibile valutare da (**1**), solo il ***flusso ottico normale,*** ovvero la componente della velocità perpendicolare alla direzione del gradiente stesso. Invece, nelle zone omogenee, in cui l'intensità luminosa è uniforme, non si ha alcuna informazione sul gradiente e non è possibile ottenere alcuna stima dalla velocità.

[1] almeno una delle due componenti è nulla.

Tale indeterminatezza è alla base del problema noto come **problema dell'apertura**. Il nome deriva dal tipico esempio "visivo" (fig **3**) che in genere si utilizza per spiegarne il significato e mostrare che l'equazione costante del moto non è sufficiente al calcolo di entrambe le componenti del vettore $\overrightarrow{v}$ in maniera univoca.

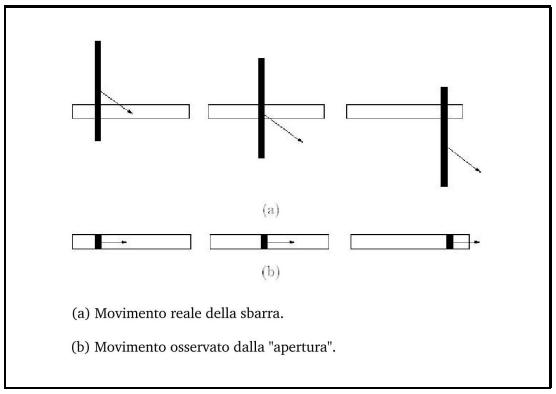

Figura 3: **Problema dell'apertura**: se si osserva la sbarra, che si muove diagonalmente verso destra e verso il basso (**a**), attraverso una "***apertura***" di dimensioni minori della sbarra (**b**), l'unico moto che può essere rilevato è quello verso destra, proprio nella direzione del gradiente dell'intensità luminosa.

In altre parole, il problema del calcolo del flusso ottico risulta *mal posto nel senso di Hadamard*.
La maggior parte dei metodi per il calcolo del flusso ottico si basa sull'idea di imporre ulteriori restrizioni alla soluzione che determinino nuovi vincoli **[5]** e si distinguono in due classi fondamentali: *Tecniche di regolarizzazione* e *Tecniche di matching*.

Tecniche di regolarizzazione.

Le tecniche di regolarizzazione si basano, innanzitutto, sulla formulazione del problema originale (ovvero l'*equazione costante del moto*) in termini di un problema di migliore approssimazione nel senso dei minimi quadrati:

$$\min_{\overrightarrow{v}} \|\nabla I \times \overrightarrow{v} + I_t\|_{L^2}$$

a cui segue l'aggiunta di un termine (di *regolarizzazione*) che tiene conto di vincoli sulla regolarità della soluzione.
In generale, la condizione aggiuntiva si basa sull'ipotesi che: "***punti vicini nell'immagine si spostano alla medesima velocità, cioè hanno lo stesso vettore spostamento***".
Ciò implica che le derivate spaziali delle componenti (v_x, v_y) del vettore $\overrightarrow{v}$, in tali punti, risultano tali che:

$$\nabla v_x = \nabla v_y = 0$$

Rientrano in questo tipo di approccio le tecniche proposte in **[23, 20, 28, 18]**, in cui il termine di regolarizzazione si ottiene minimizzando i gradienti delle componenti della velocità. In particolare, si può utilizzare il modello proposto da Tikhonov **[36]** scegliendo il funzionale:

$$\int_\Omega \left(\overrightarrow{v} \cdot \nabla I + I_t\right)^2 d\underline{x} + \alpha^r \int_\Omega \left(\|\nabla v_x\|^2 + \|\nabla v_y\|^2\right) d\underline{x}$$

dove α^r è una costante positiva fissata e valutando il flusso ottico come soluzione del problema:

$$\min_{\overrightarrow{v}} \left\{\|\overrightarrow{v} \times \nabla I + I_t\|_{L^2}^2 + \|\nabla v_x\|_{L^2}^2 + \|\nabla v_y\|_{L^2}^2\right\}$$

Tuttavia, tale scelta introduce nelle equazioni di *Eulero-Lagrange,* associate al problema, gli operatori di Laplace $2\Delta v_x$, $2\Delta v_y$ che determinano un'eccessiva regolarità della soluzione e quindi un flusso ottico troppo omogeneo soprattutto lungo i contorni.
Il vincolo aggiuntivo può essere modificato come proposto in **[18, 28]**, ovvero considerando l'ipotesi secondo cui le componenti del vettore velocità debbano essere quanto più piccole possibile nelle zone omogenee, ma non necessariamente nelle zone corrispondenti ai bordi.
Il particolare in **[28]**, il funzionale di regolarizzazione impiegato è il se-

guente:

$$\int_{\Omega} |\overrightarrow{v} \times \nabla I + I_t| \, d\underline{x} + \alpha^r \int_{\Omega} \left[\phi\left(\|\nabla v_x\|\right) + \phi\left(\|\nabla v_y\|\right)\right] d\underline{x} +$$

$$+\alpha^h \int_{\Omega} c\left(\|\nabla I\| \cdot \underline{x}\right) \cdot \|\overrightarrow{v}\| \, d\underline{x}$$

con α^r e α^h costanti positive fissate. In cui il primo addendo fonisce la misura dell'affidabilità dei dati, il secondo addendo misura il termine di regolarizzazione e generalizza il modello proposto da Tikhonov, mentre nel terzo addendo si introduce l'uso di una funzione $c \in L^{\infty}(\Omega)$, tale che:

$$\exists m_c \approx 0 : \; c(\underline{x}) \in [m_c, \, 1] \quad q.o. \, in \, \Omega$$

e cosi definita: $\forall \underline{x} = (x, \, y) \in \Omega$

$$c(\underline{x}) = \begin{cases} m_c & nelle \, zone \, omogenee \\ 1 & lungo \, i \, contorni \end{cases}$$

In questo modo, nelle zone omogenee dell'immagine, caratterizzate da bassi valori sia del modulo del gradiente $\|\nabla I\|$ che delle derivate temporali $|I_t|$, la funzione assume valore $c(\underline{x}) = m_c$ e il flusso ottico risulta fortemente limitato. Al contrario, nelle zone ricche d'informazioni (contorni, tessiture ...), la funzione assume valore $c(\underline{x}) = 1$ e il flusso ottico, in quei punti, risulta evidenziato.

Tecniche di Matching.

Considerate due immagini della sequenza relative a due istanti di tempo successivi, t, $t + \Delta t$, la conservazione dell'intensità luminosa rispetto al tempo implica che:

$$I\left(x\left(t\right), y\left(t\right), t\right) - I\left(x\left(t + \Delta t\right) + \Delta x, y\left(t + \Delta t\right) + \Delta y, t + \Delta t\right) = 0 \quad (2)$$

in cui $\Delta = (\Delta x, \, \Delta y)$ indica lo spostamento del punto $(x(t), \, y(t), \, t)$

nell'intervallo di tempo Δt.

L'equazione (**2**) è il punto di partenza delle *tecniche di matching* per il calcolo del flusso ottico proposte in **[2, 35, 29]**. Ad esempo in **[2]**, fissati gli istanti t, $t + \Delta t \in J$ e le rispettive intensità luminose:

$$I\left(x\left(t\right),\, y\left(t\right),\, t\right) = I_1\left(x,\, y\right)$$

$$I\left(x\left(t + \Delta t\right),\, y\left(t + \Delta t\right),\, t + \Delta t\right) = I_2\left(x,\, y\right)$$

si determina lo spostamento $\Delta = (\Delta x,\, \Delta y)$ di ogni punto $(x,\, y) \in \Omega$, definendo, innanzitutto, la funzione:

$$W_n\left(x,\, y\right) = \{(\widetilde{x},\, \widetilde{y}) \in \Omega :\, |\widetilde{x} - x| \leq n,\, |\widetilde{y} - y| \leq n\}$$

che rappresenta, $\forall \underline{x} = (x,\, y) \in \Omega$ un intorno di raggio n.
In tale intorno (fig. **4**), si minimizza la "distanza" fra le funzioni I_1 e I_2, ovvero:

$$SSD_{1,2}\left(x,\, y,\, \Delta x,\, \Delta y\right) = W_n\left(x,\, y\right) \star \left[I_1\left(x,\, y\right) - I_2\left(x + \Delta x,\, y + \Delta y\right)\right]^2 =$$

$$= \int_{\Omega} W_n\left(\underline{x}\right) \left[I_1\left(\underline{x}\right) - I_2\left(\underline{x} + \Delta\right)\right]^2 d\underline{x}$$

in cui il simbolo $\star$ rappresenta il prodotto di convoluzione bidimensionale.

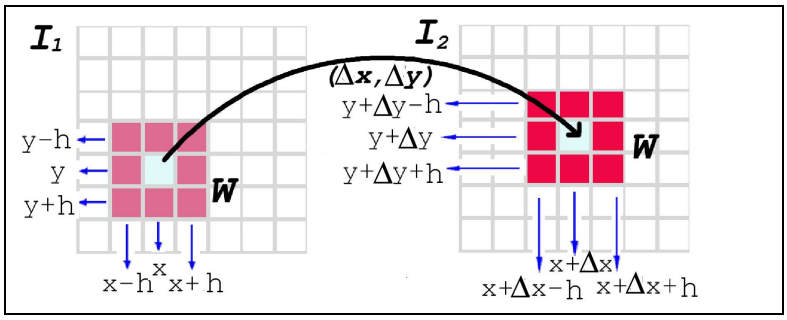

Figura 4: Tutti i punti appartenenti all'intorno discreto W di centro $\underline{x}$ e raggio h hanno lo stesso vettore spostamento.

Quindi, gli spostamenti $\Delta = (\Delta x,\, \Delta y)$ si ottengono risolvendo il problema:

$$\min_{(\Delta x, \Delta y)} SSD_{1,2}\,(x,\, y,\, \Delta x,\, \Delta y) =$$

$$= \min_{(\Delta x, \Delta y)} \left\{ \sum_{r=-n}^{n} W_n\,(r, s)\,[I_1\,(x + r, y + s) - I_2\,(x + r + \Delta x, y + s + \Delta y)]^2 \right\}$$

ovvero minimizzando la norma euclidea della differenza tra tutti i punti dell'intorno $W_n\,(r,\, s)$ rispettivamente in I_1 e in I_2, secondo l'ipotesi che punti vicini si spostano alla stessa velocità e dunque hanno lo stesso vettore spostamento.

In **[35, 29]**, invece, sono proposte alcune varianti per la stima del valore SSD in base ad ulteriori carartteristiche dell'immagine.

3.2 Stima del flusso ottico.

L'algoritmo per il calcolo del flusso ottico, implementato in questo lavoro di tesi, si basa sulla tecnica di regolarizzazzione proposta in **[18]**, secondo cui la soluzione al problema è data da:

$$\overrightarrow{v} = arg \min_{\overrightarrow{v} \in W^{1,2}(\Omega)} E\,(\overrightarrow{v})\,, \quad I \in W^{1,2}\,(\Omega) \tag{3}$$

Come nel caso della rimozione del blotch, si tratta di un problema di regolarizzazione *edge-preserving* (appendice **A**), in cui il funzionale energia è definito come segue:

$$E\,(\overrightarrow{v}) = \int_\Omega \phi_1\,(|\,\overrightarrow{v} \times \nabla I + I_t|)\,d\underline{x} + \alpha^r \int_\Omega [\phi_2\,(\|\nabla v_x\|) + \phi_2\,(\|\nabla v_y\|)]\,d\underline{x} +$$

$$+\alpha^h \int_\Omega c\,(\|\nabla I\|\,\underline{x})\,\|\overrightarrow{v}\|\,d\underline{x}$$

Dunque, il problema di minimo:

$$\min_{\overrightarrow{v}} E\left(\overrightarrow{v}\right)$$

ammette un'unica soluzione, la cui variazione risulta trascurabile nelle zone omogenee e rilevante lungo i contorni.
Per valutare:

$$\overrightarrow{v} = arg \min_{\overrightarrow{v}} E\left(\overrightarrow{v}\right)$$

si utilizza la decomposizione descritta in appendice **B**, per le funzioni ϕ_1 e ϕ_2, purchè $\phi_1\left(\sqrt{s}\right)$ e $\phi_2\left(\sqrt{s}\right)$ siano concave. Precisamente:

- se a è la variabile duale associata a $\left(\overrightarrow{v} \times \nabla I + I_t\right)$ si ottiene:

$$\phi_1\left(\overrightarrow{v} \times \nabla I + I_t\right) = \min_a \left[a\left(\overrightarrow{v} \times \nabla I + I_t\right)^2 + \psi_1\left(a\right)\right] =$$

$$= a\left(\overrightarrow{v} \times \nabla I + I_t\right)^2 + \frac{1}{a}$$

- Mentre per b_x e b_y, rispettivamente le variabili duali associate a $\|\nabla v_x\|$ e $\|\nabla v_y\|$, risulta:

$$\begin{cases} \phi_2\left(\|\nabla v_x\|\right) = \min_{b_x}\left[b_x \|\nabla v_x\|^2 + \psi_2\left(b_x\right)\right] \\ \phi_2\left(\|\nabla v_y\|\right) = \min_{b_y}\left[b_y \|\nabla v_y\|^2 + \psi_2\left(b_y\right)\right] \end{cases}$$

Con tali notazioni, il funzionale duale associato a E diventa:

$$= E^d\left(\overrightarrow{v},\, a,\, b\right) = \int_\Omega \min_{L_1 \le a \le M_1} \left(a\left(\overrightarrow{v} \times \nabla I + I_t\right)^2 + \frac{1}{a}\right) d\underline{x} +$$

$$+ \int_\Omega \min_{L_2 \le b_x \le M_2} \left(b_x \|\nabla v_x\|^2 + \psi_2\left(b_x\right)\right) dx +$$

$$+ \int_\Omega \min_{L_2 \le b_y \le M_2} \left(b_y \|\nabla v_y\|^2 + \psi_2\left(b_y\right)\right) d\underline{x} + \int_\Omega c\left(\underline{x}\right) \|\overrightarrow{v}\|\, d\underline{x}$$

Minimizzando rispetto a $\overrightarrow{v}$, si ottiene:

$$\min_{\overrightarrow{v}} E\left(\overrightarrow{v}\right) = \min_{\overrightarrow{v}} \left\{ \int_{\Omega} \min_{L_1 \leq a \leq M_1} \left(a\left(\overrightarrow{v} \times \nabla I + I_t\right)^2 + \frac{1}{a} \right) d\underline{x} + \right.$$

$$+ \int_{\Omega} \min_{L_2 \leq b_x \leq M_2} \left(b_x \left\| \nabla v_x \right\|^2 + \psi_2\left(b_x\right) \right) d\underline{x} +$$

$$+ \int_{\Omega} \min_{L_2 \leq b_y \leq M_2} \left(b_y \left\| \nabla v_y \right\|^2 + \psi_2\left(b_y\right) \right) d\underline{x} +$$

$$\left. + \int_{\Omega} c\left(\underline{x}\right) \left\| \overrightarrow{v} \right\| d\underline{x} \right\}$$

da cui:

$$\min_{\overrightarrow{v}} E^d\left(\overrightarrow{v}\right) =$$

$$\min_{\left(\overrightarrow{v}\, b_x, b_y, a\right)} \left\{ \int_{\Omega} \left(a\left(\overrightarrow{v} \times \nabla I + I_t\right)^2 + \frac{1}{a} \right) d\underline{x} + \int_{\Omega} \left(b_x \left\| \nabla v_x \right\|^2 + \psi_2\left(b_x\right) \right) d\underline{x} + \right.$$

$$\left. + \int_{\Omega} \left(b_y \left\| \nabla v_y \right\|^2 + \psi_2\left(b_y\right) \right) d\underline{x} + \int_{\Omega} c\left(\underline{x}\right) \left\| \overrightarrow{v} \right\| d\underline{x} \right\}$$

Quindi, per minimizzando il funzionale $E^d\left(\overrightarrow{v}, a, b_x, b_y\right)$ bisogna tener conto di tutte le variabili che intervengo: $\left(\overrightarrow{v},\, a,\, \underline{b}\right)$.
Fissate le condizioni iniziali $\overrightarrow{v}^0 = \left(v_x^0,\, v_y^0\right) = (0,\, 0)$ si utilizza una variante dello ***schema del punto fisso alternato***, descritto in appendice **C**:

$$\begin{cases} \left(b_x^{n+1},\, b_y^{n+1}\right) = arg\min_{\underline{b} \in L^2(\Omega)} E^d\left(v_x^n,\, v_y^n,\, a^n, b_x,\, b_y\right) \\ a^{n+1} = arg\min_{a \in L^2(\Omega)} E^d\left(v_x^n,\, v_y^n,\, a,\, b_x^{n+1}\, b_y^{n+1}\right) \\ \left(v_x^{n+1},\, v_y^{n+1}\right) = arg\min_{\overrightarrow{v} \in W^{1,2}(\Omega)} E^d\left(v_x,\, v_y,\, a^{n+1},\, b_x^{n+1},\, b_y^{n+1}\right) \end{cases} \tag{4}$$

Quindi, iterativamente, si determina la migliore approssimazione della variabile duale $\underline{b} = (b_x,\, b_y)$, calcolando:

$$\left(b_x^{n+1},\, b_y^{n+1}\right) = arg \min_{\underline{b} \in L^2(\Omega)} E^d \left(v_x^n,\, v_y^n,\, a^n,\, b_x,\, b_y\right)$$

cioè, risolvendo il problema rispetto a b_x e b_y:

$$\min_{\underline{b}} \left\{ \int_\Omega \left[b_x \left\| \nabla v_x^n \right\|^2 + \psi\left(b_x\right) + b_y \left\| \nabla v_x^n \right\|^2 + \psi\left(b_y\right) \right] d\underline{x} \right\}$$

che ammette un'unica soluzione per:

$$b_x^{n+1} = \frac{\phi_2' \left(\left\| \nabla v_x^n \right\| \right)}{2 \left\| \nabla v_x^n \right\|}, \quad b_y^{n+1} = \frac{\phi_2' \left(\left\| \nabla v_y^n \right\| \right)}{2 \left\| \nabla v_y^n \right\|}$$

Analogamente, per calcolare:

$$a^{n+1} = arg \min_{a \in L^2(\Omega)} E^d \left(v_x^n,\, v_y^n,\, a,\, b_x^{n+1}\, b_y^{n+1}\right)$$

si risolve il problema di minimo, rispetto alla variabile duale a:

$$\min_a \int_\Omega \left[a \left| \overrightarrow{v}^n \times \nabla I + I_t \right|^2 + \frac{1}{a} \right] d\underline{x}$$

e si ottiene:

$$a^{n+1} = \frac{\phi_{1,\varepsilon}' \left(\left| \overrightarrow{v}^n \times \nabla I + I_t \right| \right)}{2 \left| \overrightarrow{v}^n \times \nabla I + I_t \right|}$$

Infine, la migliore approssimazione del flusso ottico, ad ogni iterazione, si ricava valutando:

$$\left(v_x^{n+1},\, v_y^{n+1}\right) = arg \min_{\overrightarrow{v} \in W^{1,2}(\Omega)} E^d \left(v_x, v_y, a^{n+1},\, b_x^{n+1},\, b_y^{n+1}\right)$$

cioè risolvendo, rispetto a $\overrightarrow{v} = (v_x,\, v_y)$, il problema:

$$\min_{\overrightarrow{v}} \left\{ \int_\Omega \left[a^{n+1} \left| \overrightarrow{v} \times \nabla I + I_t \right|^2 + \alpha^r \, b_x^{n+1} \left\| \nabla v_x \right\|^2 + \right.\right.$$

$$\left.\left. + + \alpha^r b_y^{n+1} \left\| \nabla v_y \right\|^2 + \alpha^h c\,(\underline{x})\, \overrightarrow{v} \right] d\underline{x} \right\}$$

Imponendo che la derivata prima del funzionale, rispetto alle componenti x e y sia nulla, si ottengono le seguenti equazioni di *Eulero-Lagrange*, nelle incognite v_x^{n+1} e v_y^{n+1}:

$$\begin{cases} a^{n+1} \left(\overrightarrow{v}^{n+1} \times \nabla I + I_t \right) I_x + \alpha^r div(b_x^n \nabla v_x^{n+1}) + \alpha^h c\,(\underline{x})\, v_x^{n+1} = 0 \\ a^{n+1} \left(\overrightarrow{v}^{n+1} \times \nabla I + I_t \right) I_y + \alpha^r div(b_y^n \nabla v_y^{n+1}) + \alpha^h c\,(\underline{x})\, v_y^{n+1} = 0 \end{cases}$$

L'algoritmo per il calcolo del flusso ottico può essere schematizzato come segue.

Schema del punto fisso alternato

$v_x^0 = 0 \qquad v_y^0 = 0$

repeat

$b_x^{n+1} = \frac{\phi_2'(\|\nabla v_x^n\|)}{2\|\nabla v_x^n\|}, \quad b_y^{n+1} = \frac{\phi_2'(\|\nabla v_y^n\|)}{2\|\nabla v_y^n\|}$

$a^{n+1} = \frac{\phi_{1,\varepsilon}'(|\overrightarrow{v}^n \times \nabla I + I_t|)}{2|\overrightarrow{v}^n \times \nabla I + I_t|}$

$\alpha^h c\,(\underline{x})\, v_x^{n+1} = a^{n+1}\,(\overrightarrow{v}^n \times \nabla I + I_t)\, I_x + \alpha^r div(b_x^{n+1} \nabla v_x^n)$

$\alpha^h c\,(\underline{x})\, v_y^{n+1} = a^{n+1}\,(\overrightarrow{v}^n \times \nabla I + I_t)\, I_y + \alpha^r div(b_y^{n+1} \nabla v_y^n)$

until (convergenza)

In termini matriciali, l'iterazione $n+1$ si può descrivere mediante due sistemi lineari: il primo relativo alla componente v_x di $\overrightarrow{v}$, nelle incogni-

te $(v_x^{n+1},\, a^{n+1},\, b_x^{n+1})$; il secondo relativo alla componente v_y di $\overrightarrow{v}$, nelle incognite $\left(v_y^{n+1},\, a^{n+1},\, b_y^{n+1}\right)$.

$$\begin{bmatrix} \alpha^h c\left(\underline{x}\right) & \left(\overrightarrow{v}^{\,n} \times \nabla I + I_t\right) I_x & L\left(v_x^n\right) v_x^n \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x^{n+1} \\ a^{n+1} \\ b_x^{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\phi'_{1,\varepsilon}\left(\left|\overrightarrow{v}^{\,n} \times \nabla I + I_t\right|\right)}{2\left|\overrightarrow{v}^{\,n} \times \nabla I + I_t\right|} \\ \frac{\phi'_2\left(\left\|\nabla v_x^n\right\|\right)}{2\left\|\nabla v_x^n\right\|} \end{bmatrix}$$

$$\begin{bmatrix} \alpha^h c\left(\underline{x}\right) & \left(\overrightarrow{v}^{\,n} \times \nabla I + I_t\right) I_y & L\left(v_y^n\right) v_y^n \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_y^{n+1} \\ a^{n+1} \\ b_y^{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\phi'_{1,\varepsilon}\left(\left|\overrightarrow{v}^{\,n} \times \nabla I + I_t\right|\right)}{2\left|\overrightarrow{v}^{\,n} \times \nabla I + I_t\right|} \\ \frac{\phi'_2\left(\left\|\nabla v_y^n\right\|\right)}{2\left\|\nabla v_y^n\right\|} \end{bmatrix}$$

In entrambi i casi la matrice dei coefficienti risulta essere una matrice triangolare superiore; mentre L è l'operatore che definisce la divergenza:

$$L\left(v_x^n\right) v_x^n = -div\left(b_x^{n+1} \nabla v_x^{n+1}\right) = -\nabla \cdot \left[b_x^{n+1} \nabla \nu_x^n\right]$$

$$L\left(v_y^n\right) v_y^n = -div\left(b_y^{n+1} \nabla v_y^{n+1}\right) = -\nabla \cdot \left[b_y^{n+1} \nabla \nu_y^n\right]$$

Capitolo 4

Approssimazione numerica.

Fissato un vettore $J \subset \aleph$, la cui componente $i - sima$ rappresenta l'$i - simo$ istante temporale: $J = (t_0,\, t_1,\, ...,\, t_n)$, una sequenza di immagini digitali è rappresentata da una sequenza di $n + 1$ matrici, ognuna delle quali corrisponde al generico istante t_i e può essere identificata con una porzione finita del piano:

$$\Omega = [0,\, N] \times [0,\, M] \subset \aleph^2$$

Al variare di $t_i \in J$, ogni elemento (*pixel*) della matrice è individuato da una coppia di coordinate $(x,\, y)$, in esso centrate (come mostrato in figura **1**), tali che:

$$x \in [0,\, N] = \Omega_x, \;\; y \in [0,\, M] = \Omega_y$$

Con tali notazioni, la funzione che definisce la sequenza di immagini digitali risulta:

$$t \in J \longrightarrow (x\,(t)\;, y\,(t)\;, t) \in \Omega_x \times \Omega_y \times J$$

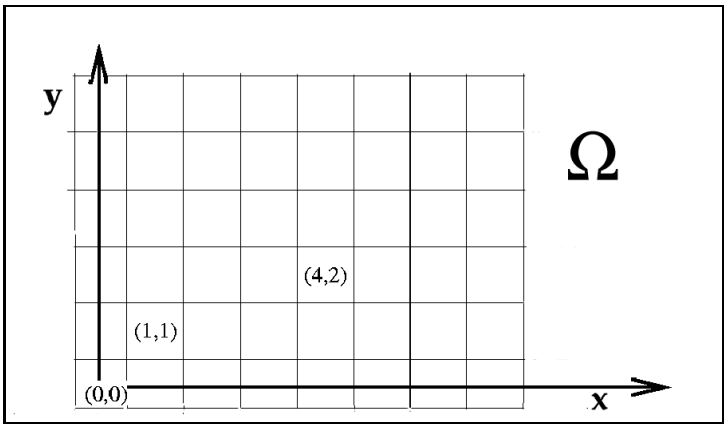

Figura 1: Rappresentazione di un'immagine digitale: gli elementi della matrice (pixel) sono individuati da una coppia di coordinate.

Se la sequenza di immagine digitali è in bianco e nero, ad ogni pixel risulta associato un valore appartenente all'intervallo:

$$[0,\, 255] \subset \aleph$$

che rappresenta l'intensità di grigio in quel punto. Quindi, la funzione che definisce l'intensità luminosa della sequenza è:

$$t \in J \longrightarrow I\left(x\left(t\right)\,, y\left(t\right),\, t\right) \in [0,\, 255]$$

Se la sequenza di immagini digitali è a colori, ad ogni pixel corrispondono tre valori distinti, uno per ogni banda della rappresentazione RGB. In tal caso, la funzione che descrive l'intensità luminosa della sequenza è:

$$t \in [0,\, T] \longrightarrow I\left(x\left(t\right)\,, y\left(t\right),\, t\right) = (R,\, G,\, B) \in [0,\, 255]^3$$

Supposto che $\forall i = 0, ..n$, l'intervallo di tempo tra un istante e l'altro della sequenza sia sempre lo stesso $\Delta t = t_{i+1} - t_i$, allora, al variare di $t \in J = (t_0,\, t_1, ..., t_n)$, lo schema numerico corrispondente al modello proposto per il riconoscimento e l'eliminazione dei blotch in sequenze di

immagini, è composto dalle seguenti fasi:

- ***Approssimazione del moto***:
  in cui si determinano gli spostamenti di ogni punto dall'istante $t-\Delta t$ all'istante t; e gli spostamenti, per ogni punto dall'istante t all'istante $t+\Delta t$.

- ***Individuazione delle zone corrotte***:
  in cui si utilizzano le informazioni relative agli spostamenti da $t-\Delta t$ a t e da t a $t+\Delta t$, per definire il dominio B, che corrisponde alla zona deteriorata dell'immagine all'istante t.

- ***Ricostruzione delle intensità corrotte***:
  in cui si valuta un'approssimazione dell'intensità luminosa corretta all'istante t, ricostruendo le zone daneggiate mediante le informazioni degli spostamenti dall'istante precedente $t-\Delta t$ e successivo $t+\Delta t$.

Per ognuna di queste fasi è possibile definire un metodo numerico di risoluzione, che si basa sul rispettivo modello teorico, facendo riferimento a quanto stabilito nei capitoli precedenti.

4.1 Approssimazione del moto.

L'algoritmo per il calcolo del flusso ottico $\overrightarrow{v}(t)$ che definisce il campo di moto, dunque la variazione di intensità luminosa tra l'immagine $I(x(t), y(t), t)$ e l'immagine $I(x(t+\Delta t), y(t+\Delta t), t+\Delta t)$, si basa sul calcolo della migliore approssimazione del vettore $\overrightarrow{v}$, descritta nel capitolo **3**.
Assegnate le condizioni iniziali $\left(v_x^0,\, v_y^0\right) = (0,\, 0)$, lo schema iterativo, impiegato per il calcolo della soluzione, può essere riassunto nella tavola seguente.

Algoritmo numerico per la stima del flusso ottico
repeat
$b_x^{n+1} = \frac{\phi_2'(\|\nabla v_x^n\|)}{2\|\nabla v_x^n\|}, \quad b_y^{n+1} = \frac{\phi_2'(\|\nabla v_y^n\|)}{2\|\nabla v_y^n\|}$
$a^{n+1} = \frac{\phi_{1,\varepsilon}'(\lvert \overrightarrow{v}^n \times \nabla I + I_t \rvert)}{2\lvert \overrightarrow{v}^n \times \nabla I + I_t \rvert}$
$\alpha^h c(\underline{x})\, v_x^{n+1} = a^{n+1}(\overrightarrow{v}^n \times \nabla I + I_t) I_x + \alpha^r div(b_x^{n+1} \nabla v_x^n)$
$\alpha^h c(\underline{x})\, v_y^{n+1} = a^{n+1}(\overrightarrow{v}^n \times \nabla I + I_t) I_y + \alpha^r div(b_y^{n+1} \nabla v_y^n)$
until (convergenza)

In particolare, i valori delle variabili *duali* a, b dipendono dalla scelta della funzioni ϕ_1 e ϕ_2 (appendici **A** e **B**). Posto: $\phi_1(s) \equiv id(s) = s$ e $\phi_2(s) = 2\sqrt{1+s^2} - 2$, rispettivamente, si ha:

- $\phi_1'(s) = 1 \Rightarrow\ a = \frac{\phi_1'(s)}{2s} = \frac{1}{2s}$
- $\phi_2'(s) = \frac{2s}{\sqrt{1+s^2}} \Rightarrow\ b = \frac{\phi_2'(s)}{2s} = \frac{1}{\sqrt{1+s^2}}$

Quindi, la variabile duale a, all'iterazione $n+1$, risulta:

$$a^{n+1} = \frac{\phi_1'(\lvert \overrightarrow{v}^n \times \nabla I + I_t \rvert)}{2\lvert \overrightarrow{v}^n \times \nabla I + I_t \rvert} = \frac{1}{2\lvert \overrightarrow{v}^n \times \nabla I + I_t \rvert}$$

in cui:

$$\nabla I = \nabla I(P(t), t) = \left(\frac{\partial I(P(t), t)}{\partial x}, \frac{\partial I(P(t), t)}{\partial y} \right), \quad I_t = \frac{\partial I(P(t), t)}{\partial t}$$

Tali quantità, sono da intendersi come approssimazione delle derivate parziali mediante differenze finite (metodo ***sub-pixel***), per cui:

$$\frac{\partial}{\partial x} I(x(t), y(t), t) \approx \Delta^x I(P(t), t) = I\left(x(t) + \tfrac{1}{2}, y(t), t\right) - I(x(t), y(t), t)$$

$$\frac{\partial}{\partial y} I(x(t), y(t), t) \approx \Delta^y I(P(t), t) = I\left(x(t), y(t) + \tfrac{1}{2}, t\right) - I(x(t), y(t), t)$$

$$\frac{\partial}{\partial t} I(x(t), y(t), t) \approx I(x(t + \Delta t), y(t + \Delta t), t + \Delta t) - I(x(t), y(t), t)$$

Inoltre, le componenti della variabile duale $\underline{b} = (b_x, b_y)$ risultano:

$$b_x^{n+1} = \frac{\phi_2^{'}\left(\|\nabla v_x^n\|\right)}{2\|\nabla v_x^n\|} = \frac{1}{\sqrt{1+\|\nabla v_x^n\|^2}},\ b_y^{n+1} = \frac{\phi_2^{'}\left(\left\|\nabla v_y^n\right\|\right)}{2\left\|\nabla v_y^n\right\|} = \frac{1}{\sqrt{1+\left\|\nabla v_y^n\right\|^2}}$$

in cui:

$$\nabla v_x^n = \left(\frac{\partial v_x^n}{\partial x}, \frac{\partial v_x^n}{\partial y}\right),\ \nabla v_y^n = \left(\frac{\partial v_y^n}{\partial x}, \frac{\partial v_y^n}{\partial y}\right)$$

possono essere approssimate come segue:

$$\begin{array}{l}
\frac{\partial}{\partial x} v_x^n\,(x,\,y) \approx \Delta^x v_x^n = v_x^n\left(x+\frac{1}{2},\,y\right) - v_x^n\,(x,\,y) \\
\frac{\partial}{\partial y} v_x^n\,(x,\,y) \approx \Delta^y v_x^n = v_x^n\left(x,\,y+\frac{1}{2}\right) - v_x^n\,(x,\,y) \\
\frac{\partial}{\partial x} v_y^n\,(x,\,y) \approx \Delta^x v_y^n = v_y^n\left(x+\frac{1}{2},\,y\right) - v_y^n\,(x,\,y) \\
\frac{\partial}{\partial y} v_y^n\,(x,\,y) \approx \Delta^y v_y^n = v_y^n\left(x,\,y+\frac{1}{2}\right) - v_y^n\,(x,\,y)
\end{array}$$

Lo stesso tipo di approssimazione può essere impiegata per risolvere le equazioni di Eulero-Lagrange, relative alle componenti v_x, v_y del flusso ottico:

$$\alpha^h c\,(\underline{x})\, v_x^{n+1} = a^{n+1}\left(\overrightarrow{v}^n \times \nabla I + I_t\right) I_x + \alpha^r div(b_x^{n+1}\nabla v_x^n)$$

$$\alpha^h c\,(\underline{x})\, v_y^{n+1} = a^{n+1}\left(\overrightarrow{v}^n \times \nabla I + I_t\right) I_y + \alpha^r div(b_y^{n+1}\nabla v_y^n)$$

In particolare, per discretizzare gli operatori:

$$div\left(b_x^{n+1}\nabla v_x^n\right) = \nabla\cdot\left[b_x^{n+1}\nabla v_x^n\right]\ e\ div\left(b_x^{n+1}\nabla v_x^n\right) = \nabla\cdot\left[b_x^{n+1}\nabla v_x^n\right]$$

si utilizza lo schema descritto in **4.4**.

Come si osserva in figura **2**, per ogni pixel appartenente all'immagine, l'area di applicazione per il calcolo delle derivate parziali è rappresentata da una regione **S** di dimensione 3×3.
Tale regione è definita dagli operatori utilizzati per il calcolo delle diffe-

renze finite (***stencil a 3 punti***) in cui l'elemento centrale corrisponde all'elemento in esame.

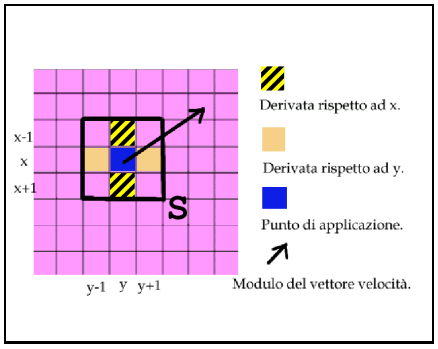

Figura 2: Calcolo delle derivate parziali mediante il metodo alle differenze finite.

Per migliorare l'affidabilità dell'approssimazione numerica realizzata dalle differenze finite è stata utilizzata la ***multirisoluzione*** (appendice **D**). Questo tipo di approssimazione risulta utile per il calcolo del flusso ottico, perchè permette di calcolare gli spostamenti macroscopici percepili su larga scala (***coarse***), e via via spostamenti microscopici percepibili su piccola scala (***fine***) **[25]**.
Lo schema adottato si può rappresentare come un ***V-ciclo multigrid*** (fig. **3**) e consiste di due fasi: ***decomposizione*** e ***correzione.***

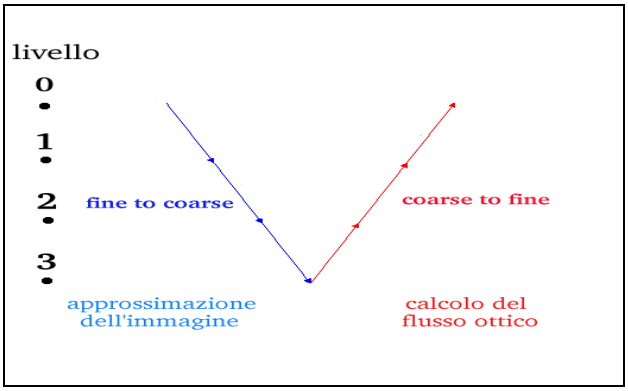

Figura 3: V-ciclo multigrid.

Decomposizione (*fine-to-coarse*):

A partire dall'immagine originale I, il cui dominio Ω è costituito da $N \times M$ punti (dove $N = 2^{2n}$ e $M = 2^{2m}$) e a cui è associato un passo di discretizzazione h, si costruisce la struttura in multirisoluzione (***piramide***), aumentando il passo di discretizzazione della griglia di risoluzione.

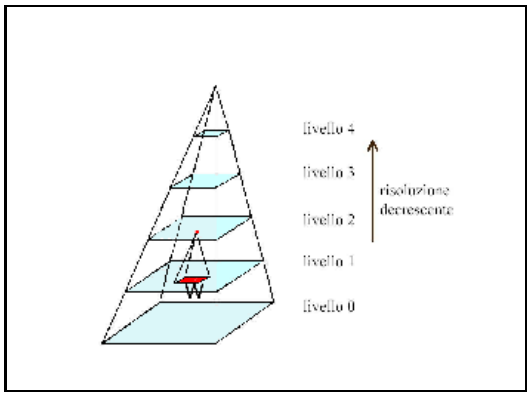

Figura 4: Struttura Piramidale.

Tale struttura risulta una *sequenza* di rappresentazioni della stessa immagine (fig. **4**) $\{\Omega_i : \Omega_0 \equiv \Omega\}_{1 \leq i \leq level}$, in cui ad ogni *livello* i:

- il dominio Ω_i si ottiene da quello del livello precedente Ω_{i-1}, raddoppiando il passo di discretizzazione:

  $$h_i = 2h_{i-1}$$

  In generale, se $\Omega \equiv \Omega_0$ ha dimensione $N \times M$ ed è costituito da $2^{2n} \times 2^{2m}$ pixel, allora il generico dominio Ω_i, corrispondente al livello di risoluzione i, ha dimensione:

  $$\frac{N}{2^i} \times \frac{M}{2^i}$$

  ed è composto da $2^{\frac{2n}{i+1}} \times 2^{\frac{2m}{i+1}}$pixel;

- l'intensità luminosa I_i, di ogni punto $(x,\ y) \in \Omega_i$, è calcolata come:

$$I_i\,(x,\ y) = I_{i-1}\,(x,\ y) \star \omega_i\,(x,\ y)$$

dove: il simbolo $\star$ indica il prodotto di convoluzione e ω_i è la *base ortonormale* relativa alla funzione generatrice gaussiana ω.

Infine, il numero dei livelli *level*, nella piramide in multirisoluzione, dipende dalle dimensioni originali dell'immagine. Fissati due valori di soglia minima per la decomposizione, ad ogni passo i si procede con la costruzione di un nuovo livello finchè:

$$\frac{N}{2^i} \geq minN \quad e \quad \frac{M}{2^i} \geq minM$$

Correzione (*coarse-to-fine*):

In questa seconda fase si effettua una prima stima del flusso ottico a partire dall'apice della struttura, che corrisponde alla risoluzione più bassa ed è rappresentato da una griglia con passo di discretizzazione "*grande*" (*large displacement*). Si procede, quindi, per raffinamenti successivi ottenuti all'aumentare della risoluzione (fig **5**).

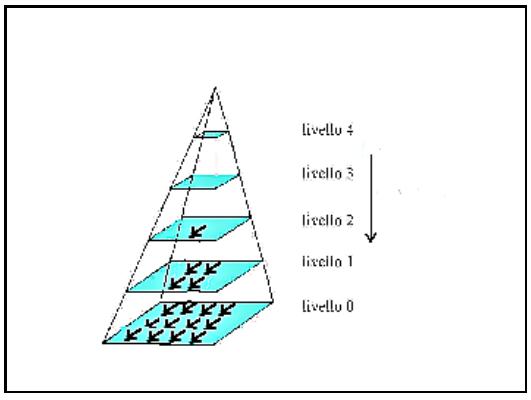

Figura 5: Prolungamento per duplicazione

Precisamente, il flusso ottico è calcolato ad ogni livello di risoluzione i, risolvendo il problema di regolarizzazione descritto nel capitolo **3**. La soluzione, una volta ***prolungata*** sul livello successivo $i-1$, rappresenta l'incremento alla nuova soluzione valutata al livello corrente.
Il prolungamento al livello i dal livello precedente $i-1$, corrispondente a una risoluzione spaziale doppia, viene effettuato per ***duplicazione*** ed è illustrato nello schema seguente.

Approccio in multirisoluzione per la stima del flusso ottico

for *(* $i = level, 0, step - 1$ *)* ***do***
    %% prolungamento per duplicazione
    if *(* $i = level$ *)* ***then*** $\delta\vec{v}_i(x, y) := 0$
                else $\delta\vec{v}_i(x, y) := \vec{v}_{i-1}\left(\frac{x}{2}, \frac{y}{2}\right)$
    endif
    %% algoritmo per la stima del flusso ottico (pag. 47)
    $\vec{v}_i$
    %% aggiornamento dei dati
    $\vec{v}_i = \vec{v}_i + \delta\vec{v}_i$
endfor

La strategia impiegata si basa sull'idea che "*punti vicini nell'immagine si muovono alla stessa velocità*", per cui sulla griglia *coarse* (immagine a bassa risoluzione) si rilevano gli spostamenti relativi a sottogruppi di punti vicini, che vengono via via differenziati quando si passa ad una griglia più *fine* (immagine ad alta risoluzione).

4.2 Individuazione delle zone corrotte.

Il metodo per l'individuazione della posizione dei blotch impiega le informazioni relative agli spostamenti di ogni punto della sequenza per definire il dominio corrispondente alle zone danneggiate.

Precisamente, se indichiamo con:

- $\Delta^+ = (\Delta x^+, \Delta y^+)$, il vettore spostamento che descrive il flusso ottico valutato dall'istante $t - \Delta t$ all'istante t;
- $\Delta^- = (\Delta x^-, \Delta y^-)$, il vettore spostamento che definisce il flusso ottico stimato dall'istante $t + \Delta t$ all'istante t,

lo schema numerico può essere riassunto come segue.

Algoritmo numerico per il riconoscimento dei blotch

%% Calcolo delle immagini compensate

$$I^C_{t-\Delta t}(x(t), y(t), t) = I(x(t-\Delta t) + \Delta x^+, y(t-\Delta t) + \Delta y^+, t - \Delta t)$$

$$I^C_{t+\Delta t}(x(t), y(t), t) = I(x(t+\Delta t) - \Delta x^-, y(t+\Delta t) - \Delta y^-, t + \Delta t)$$

%% Differenze con l'immagine di riferimento

$$W_{t-\Delta t}(x(t), y(t), t) = \left| I^C_{t-\Delta t}(x(t), y(t), t) - I(x(t), y(t), t) \right|$$

$$W_{t+\Delta t}(x(t), y(t), t) = \left| I^C_{t+\Delta t}(x(t), y(t), t) - I(x(t), y(t), t) \right|$$

%% Definizione della maschera

$$B(x(t), y(t), t) = \begin{cases} 1 & se \min\{W_{t-\Delta t}, W_{t+\Delta t}\} \geq K \\ 0 & altrimenti \end{cases}$$

In pratica, mediante le prime due relazioni, si determinano le immagini compensate. Queste ultime possono essere utilizzate per valutare un'approssimazione numerica delle derivate direzionali lungo le traiettorie positive e negative del moto.
Con tali valori, fissata una tolleranza $\boldsymbol{K}$ che rappresenta la soglia di grigio per il riconoscimento delle zone corrotte, si può defire la ***maschera del blotch*** $B(x(t), y(t))$.
Tale maschera è rappresentata da un'immagine binaria di riferimento, che

ha le stesse dimensioni dell'immagine in esame ed in cui sono evidenziati i pixel corrispondenti ai punti che fanno parte del dominio del blotch.

E' possibile che l'immagine ottenuta, al termine dell'operazione di rimozione dei difetti, presenti delle discontinuità dei valori di grigio in corrispondenza dei contorni delle zone corrotte e rielaborate. Ciò può dipendere dall'accuratezza della maschera del blocth.
Utilizzando la ***proprietà 3,*** secondo cui i blotch formano regioni connesse[1] all'interno dell'immagine, è possibile correggere la maschera B, affinchè verifichi tale proprietà. Allora, ogni zona evidenziata, in B come locazione del dominio del blotch viene sottoposta all'*operazione morfologica di apertura* (appendice **E**).
Si costruisce una matrice più piccola S, generalmente di dimensione 3×3, i cui elementi sono tutti uguali a uno:

$$S = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

e si effettua prima l'operazione di *erosione* e poi la *dilatazione*.
Per ogni elemento (x, y) della matrice B, si determina:

$$val_E = \sum_{i=-1}^{1} \sum_{j=-1}^{1} S(i,j) \cdot B(x-i, y-j)$$

e si pone:

$$B_E(x,y) = \begin{cases} 1 & se\ \ val_E = 9 \\ 0 & altrimenti \end{cases}$$

Quindi, per ogni elemento (x, y) della nuova matrice B_E si valuta:

$$val_D = \sum_{i=-1}^{1} \sum_{j=-1}^{1} S(i,j) \cdot B_E(x-i, y-j)$$

[1]Una ***regione connessa*** è una porzione di piano S, in cui per ogni coppia di punti $(x, y) \in S$, esiste una curva che li collega γ, interamente contenuta in S.

e si pone:

$$B_{E_D}(x,y) = \begin{cases} 1 & se\, val_D \geq 1 \\ 0 & altrimenti \end{cases}$$

Il risultato di tale operazione B_{E_D}, viene infine ricopiato in B è rappresenta la *nuova maschera del blotch*, che verra impiegata nella fase di ricostruzione.

4.3 Ricostruzione delle intensità corrotte.

Nella fase di ricostruzione dell'intensità luminosa corrispondente alle zone danneggiate si risolve il problema di approssimazione, descritto nel capitolo **2**, mediante le informazioni relatve alla maschera del blotch B e alle immagini compensate $I^C_{t+\Delta t}$ e $I^C_{t-\Delta t}$.
Per ogni punto del dominio Ω, per cui $B(x,\, y) = 1$, si definisce una prima approssimazione dei valori corrispondenti alle zone corrotte, mediante la relazione:

$$I^B_0(x,\, y) = \frac{I^C_{t-\Delta t}(x,\, y) + I^C_{t+\Delta t}(x,\, y)}{2}$$

In queste condizioni inziali, si procede al calcolo della soluzione mediante il seguente medoto iterativo.

Algoritmo numerico per la rimozione dei blocth
repeat $b^{n+1} = \frac{\phi'(\nabla I^B_n)}{2\lvert\nabla I^B_n\rvert}$ $I^B_{n+1} = G \star I^B_n - \lambda div\ \left(b^{n+1}\nabla I^B_n\right)$ ***until*** (convergenza)

Per risolvere l'equazione relativa alla variabile *duale* b, si definisce la funzione:

$$\phi(s) = 2\sqrt{1+s^2} - 2 \;\Rightarrow\; \phi'(s) = \frac{2s}{\sqrt{1+s^2}} \;\Rightarrow\; b = \frac{\phi'(s)}{2s} = \frac{1}{\sqrt{1+s^2}}$$

quindi, si ottiene:

$$b^{n+1} = \frac{\phi'\left(\left\|\nabla I_n^B\right\|\right)}{2\left\|\nabla I_n^B\right\|} = \frac{1}{\sqrt{1+\left\|\nabla I_n^B\right\|^2}}$$

Le componenti del gradiente della funzione I_n^B sono le derivate parziali, valutate mediante differenze finite:

$$\frac{\partial}{\partial x} I_n^B(x, y) \approx \triangle^x I_n^B(x, y) = I_n^B\left(x + \frac{1}{2}, y\right) - I_n^B(x, y)$$

$$\frac{\partial}{\partial y} I_n^B(x, y) \approx \triangle^y I_n^B(x, y) = I_n^B\left(x, y + \frac{1}{2}\right) - I_n^B(x, y)$$

Invece, per risolvere l'equazione:

$$I_{n+1}^B = G \star I_n^B - \lambda\, div\left(b^{n+1} \nabla I_n^B\right)$$

si effettua una media locale (***smoothing***), pesata in funzione della distanza di raggio $\sigma^2 = 1$ dal centro, di tutti i punti appartenenti al dominio B. In pratica, si applica la seguente operazione di convoluzione **[32]**:

$$G(x, y) \star I_n^B(x, y) = \sum_{h=-1}^{1} \sum_{k=-1}^{1} \omega(h, k)\; I_n^B(x + h, y + k)$$

dove ω è una matrice, di dimensione 3×3, i cui coefficienti sono:

$$\omega = \frac{1}{4} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

che rappresenta la funzione gaussiana bidimensionale:

$$G\left(x,\, y\right) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\}$$

Infine, per discretizzare il termine relativo alla divergenza:

$$div\left(b^{n+1}\nabla I_n^B\right) = \nabla \cdot \left[b^{n+1}\nabla I_n^B\right]$$

si utilizza il metodo descritto nel paragrafo successivo.

4.4 Discretizzazione dell'operatore divergenza.

Per ottenere un'approssimazione numerica dell'operatore divergenza, indotto dal termine di regolarizzazione (appendici **A**, **B**), si applica lo schema proposto in **[30]**.
Considerate due marici d ed a, si osserva innanzitutto che:

$$div\left(d^{n+1}\cdot\nabla a^n\right) = \frac{\partial}{\partial x}\left(d^{n+1}\cdot\frac{\partial}{\partial x}a^n\right) + \frac{\partial}{\partial y}\left(d^{n+1}\cdot\frac{\partial}{\partial y}a^n\right) \approx$$

$$\approx \triangle^x\left(d^{n+1}\cdot\triangle^x a^n\right) + \triangle^y\left(d^{n+1}\cdot\triangle^y a^n\right)$$

Per approssimare le derivate parziali, si definiscono, per ogni elemento di d^{n+1}, le matrici P e D combinando differenze finite in avanti e all'indietro:

$$P = \begin{bmatrix} 0 & d^{n+1}\left(x-\frac{\triangle x}{2},\, y\right) & 0 \\ d^{n+1}\left(x,\, y-\frac{\triangle y}{2}\right) & -S^P & d^{n+1}\left(x,\, y+\frac{\triangle y}{2}\right) \\ 0 & d^{n+1}\left(x+\frac{\triangle x}{2},\, y\right) & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} d^{n+1}\left(x - \frac{\triangle x}{2},\, y - \frac{1}{2}\right) & 0 & d^{n+1}\left(x - \frac{\triangle x}{2}, y - \frac{\triangle y}{2}\right) \\ 0 & -S^D & 0 \\ d^{n+1}\left(x + \frac{\triangle x}{2},\, y + \frac{\triangle y}{2}\right) & 0 & d^{n+1}\left(x + \frac{\triangle x}{2}, y + \frac{\triangle y}{2}\right) \end{bmatrix}$$

dove:

$$S^P = d^{n+1}\left(x - \tfrac{\triangle x}{2},\, y\right) + d^{n+1}\left(x + \tfrac{\triangle x}{2},\, y\right) +$$

$$+ d^{n+1}\left(x,\, y - \tfrac{\triangle y}{2}\right) + d^{n+1}\left(x,\, y + \tfrac{\triangle y}{2}\right)$$

$$S^D = d^{n+1}\left(x - \tfrac{\triangle x}{2},\, y - \tfrac{\triangle y}{2}\right) + d^{n+1}\left(x - \tfrac{\triangle x}{2},\, y + \tfrac{\triangle y}{2}\right) +$$

$$+ d^{n+1}\left(x + \tfrac{\triangle x}{2},\, y - \tfrac{\triangle y}{2}\right) + d^{n+1}\left(x + \tfrac{\triangle x}{2},\, y + \tfrac{\triangle y}{2}\right)$$

In definitiva, mediante tali operatori è possibile valutare, per ogni elemento della matrice a^n:

$$L\left(a^n\right) a^n = div\left(d^{n+1} \nabla a^n\right)(x,\, y) = \alpha_P\, P \star a^n\,(x,\, y) + \alpha_D\, D \star a^n\,(x,\, y)$$

in cui $\star$ indica l'operazione di convoluzione ed i parametri α_P e α_D verificano la condizione:

$$\alpha_P + 2\alpha_D = 1$$

e possono essere scelti in modo tale che siano privilegiare le direzioni principali relative agli assi cartesiani, ad esempio ponendo: $(\alpha_P,\, \alpha_D) = \left(\frac{1}{2},\, \frac{1}{4}\right)$; oppure considerando le direzioni relative all'orientamento del gradiente.

Capitolo 5

Un software parallelo per l'eliminazione del blotch.

Il software sviluppato costituisce un sistema di restauro digitale, per le sequenze di immagini affette da blotch, che funziona secondo lo schema mostrato in figura **1**.
Precisamente, nel primo modulo si determinano le informazioni relative al flusso ottico, successivamente utilizzate per il calcolo delle *immagini compensate*, necessarie sia al modulo di individuazione che a quello di rimozione dei difetti.

Nei pragrafi che seguono sono descritti in dettaglio gli algoritmi, implementati in linguaggio C, che corrispondono ai singoli moduli del sistema, analizzando le relative complessità computazionali, l'accuratezza e l'efficienza.

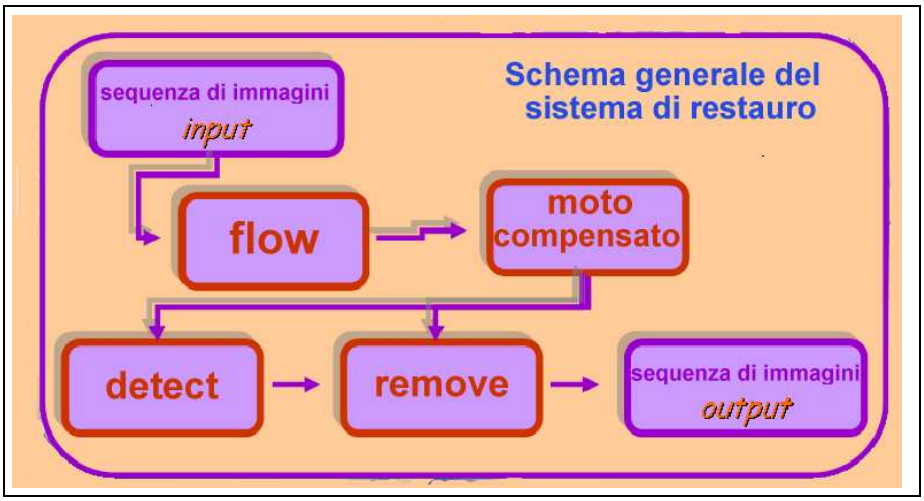

Figura 1: Schema generale del sistema di restauro.

5.1 Modulo 1: algoritmo *flow*.

Specifiche:

flow *<path> <frame> <option>*

dove:

<path> è il nome della sequenza in esame.

<frame> è il numero corrispondente al fotogramma da restaurare.

<option> le opzioni per l'uso

- ***-l*** *level;* permette di scegliere il livello di risoluzione per il calcolo del flusso ottico. Per default ha valore zero, ovvero la risoluzione originale.
- ***-s****;* nel caso in cui si desidera salvare le immagini ai livelli di risoluzione intermedi.

- ***-D*** *MaxIt;* è il massimo numero di iterazioni nel processo di minimizzazione. Per default ha valore 10.
- ***-G*** *Sg;* è il valore di soglia per il modulo del gradiente spaziale dell'immagine. Può essere scelto nell'intervallo [0, 1] e per default ha valore 0.5.
- ***-T*** *St;* è il valore di soglia della derivata temporale dell'immagine. Può essere scelto nell'intervallo [0, 1] e per default ha valore 0.5.

Algorithm 1 **Schema generale**

%% Lettura dell'immagine corrispondente al numero dato in input %%
I_{nr}
%% Lettura del'immagine precedente e successiva %%
I_{nr-1}, I_{nr+1}
%% Le tre immagini vengono organizzate a coppie, quindi la prima volta %%
$I_{LEFT} = I_{nr-1}$ $I_{RIGHT} = I_{nr}$
%% La seconda volta, invece: %%
$I_{LEFT} = I_{nr}$ $I_{RIGHT} = I_{nr+1}$
%% Il seguito dell'algoritmo è descritto per la generica coppia I_{LEFT} - I_{RIGHT} %%

%% Calcolo dei livelli di risoluzione %%
Level ($MinDim$, M, N, $level$)

%% Costruzione della piramide in multirisoluzione %%
Pyramid (Pyr, $level$)

%% *Ciclo piu esterno sui livelli di risoluzione* %%

for ($k = level$, 0, $step - 1$) do

  %% Prolungamento del risultati dal livello precedente %%
  Proy (Pyr, $level$)

  %% Calcolo del flusso ottico al livello corrente %%
  Optical_Flow (Pyr, $level$)

  %% Incremento al flusso ottico %%
  Add (Pyr, $level$)

endfor

Il modulo per il calcolo del flusso ottico fornisce in output le componenti degli spostamenti:

$$V^x_{nr-1},\ V^y_{nr-1}$$

relativi alla coppia di immagini I_{nr-1} e I_{nr} e le componenti degli spostamenti:

$$V^x_{nr+1},\ V^y_{nr+1}$$

relativi alla coppia I_{nr} e I_{nr+1}.

La complessità computazionale, relativa all'implementazione dell'algoritmo su una sola coppia di immagini, è legata al numero di livelli: $level$, che definisce l'altezza della piramide in multirisoluzione, dal livello 0, che corrisponde all'immagine originale fino alla sua rappresentazione al livello $level - 1$ (appendice **D**).

Il numero dei livelli dipende dalle dimensioni dell'immagine iniziale: se indichiamo con $N \times M$ le dimensioni della matrice che rappresenta l'immagine, le nuove dimensioni si determinano iterativamente mediante la procedura **Level** (alg. **2**): ad ogni iterazione, i valori ottenuti sono confrontati con un valore predefinito $MinDim$ e, quando almeno uno di essi risulta minore di tale quantità, il procedimento si interrompe.

Algorithm 2 **Level** - Calcolo del Numero dei livelli $MinDim = 25$

%% Al livello zero le dimensioni originali %%
$Pyr[0].row = N;\quad Pyr[0].col = M$;

%% Un ciclo iterativo controlla le dimensioni dei livelli successivi %%
$k = 0$
while $((Pyr[k].row > MinDim)$ && $(Pyr[k].col > MinDim))$ do
  $k = k + 1;$
  $Pyr[k].row = \frac{N}{2^k};\ Pyr[k].col = \frac{M}{2^k}$
endwhile
$level = k + 1;$

La definizione di una soglia minima per le dimensioni dell'immagine evita la costruzione di immagini "troppo piccole" e dunque prive di dettagli significativi. In tal caso, infatti, i risultati ottenuti dall'analisi del moto non sono rilevanti.

Stabilito il numero di livelli, la struttura dati dell'algoritmo risulta definita da un array di strutture $Pyr\,[k]$ i cui campi, $\forall k = 0, level - 1$ sono:

- I_{LEFT}, I_{RIGHT} le immagini in esame,
- δv_x, δv_y, v_x, v_y le componenti del vettore velocità e gli incrementi relativi al livello precedente.
- $\frac{\partial v_x}{\partial x}$, $\frac{\partial v_y}{\partial x}$, $\frac{\partial v_x}{\partial y}$, $\frac{\partial v_y}{\partial y}$ le derivate parziali delle componenti del vettore velocità,
- b_x, b_y, a le variabili ausiliarie,
- c la funzione che regola la densità del flusso ottico,
- I_x, I_y, I_t, ∇I le derivate spaziali e temporali dell'immagine I_{LEFT} e i valori del modulo del gradiente.
- $divX$, $divY$ i valori corrispondenti ai termini dell'operatore divergenza,
- Eq i valori relativi all'equazione costante del flusso ottico
- row, col le dimensioni al livello corrente.

Per ogni livello di risoluzione k, sono necessari 21 array di lunghezza $\left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$. Quindi, sommando sui livelli:

$$\sum_{k=0}^{level-1} 21 \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right) = 21 \cdot (N \times M) \sum_{k=0}^{level-1} \left[\frac{1}{4^k}\right]$$

Poichè, per $level \to \infty$, si ha:

$$21 \cdot (N \times M) \sum_{k=0}^{\infty} \left[\frac{1}{4^k} \right] = 21 \cdot (N \times M) \cdot \frac{4}{3}$$

la ***complessità di spazio*** dell'algoritmo, al crescere di $level$, risulta quindi:

$$T_S\,(N \times M) \simeq O\,(N \times M)$$

Per quanto rigurada la complessità computazionale, analizziamo in dettaglio le singole procedure riportate nello schema generale (alg. **1**). Nella costruzione della piramide in multirisoluzione, ad ogni livello k, per $1 \leq k \leq level - 1$, si determina un'approssimazione dell'immagine I_k, mediante l'operazione di convoluzione discreta:

$$I_k\,(x,\, y) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} \omega\,(m,\, n)\; I_{k-1}\,(2x + m,\, 2y + n) \qquad (1)$$

in cui I_{k-1} è la matrice che rappresenta l'immagine al livello di risoluzione precedente, mentre:

$$\omega = \frac{1}{4} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

è la matrice di dimensione 3×3, che rappresenta la base wavelet gaussiana al discreto (alg. **3**).

Algorithm 3 **Pyramid** - Costruzione della piramide

%% Inizializzazione %%
for $(x = 1,\, Pyr\,[0]\,.row\,;\, y = 1,\, Pyr\,[0]\,.col)$ do
$Pyr\,[0]\,.I_{LEFT}\,(x,\, y) = I_{LEFT}\,(x,\, y)\,;\quad Pyr\,[0]\,.I_{RIGHT}\,(x,\, y) = I_{RIGHT}\,(x,\, y)\,;$
endfor

%% Ciclo sui livelli %%
for $(k = 1; level - 1, step + 1)$ do

%% Convoluzione con kernel gaussiano 3x3 %%
for $(x = 1,\, Pyr\,[k]\,.row\,;\, y = 1,\, Pyr\,[k]\,.col)$ do
$Pyr\,[k]\,.I_{LEFT}\,(x,\, y) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} \omega\,(m,\, n) \cdot Pyr\,[k+1]\,.I_{LEFT}\,(x-m,\, y-n)\,;$
$Pyr\,[k]\,.I_{RIGHT}\,(x,\, y) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} \omega\,(m,\, n) \cdot Pyr\,[k+1]\,.I_{RIGHT}\,(x-m,\, y-n)\,;$
endfor
endfor

Se indichiamo, rispettivamente con:

- m l'operazione di moltiplicazione/divisione,
- a l'operazione di addizione/sottrazione,
- c l'operazione di confronto,

è possibile valutare la complessità di tempo necessaria al calcolo dei livelli e alla costruzione delle due strutture piramidali in multirisoluzione per due fotogrammi consecutivi della sequenza:

$$2 \cdot T_{pyr}\,(N \times M) = 2 \cdot \sum_{k=1}^{level-1} T_{pyr}^{k} \left(\frac{N}{2^k} \times \frac{M}{2^k}\right) =$$

$$= 2 \cdot \sum_{k=1}^{level-1} [2a + 3m + 3a] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

Inoltre, ad ogni livello di risoluzione, escluso il più alto $k = level - 1$, i valori ottenuti corrispondenti alle componenti v_x e v_y, vengono prolungati dal livello precedente al livello in esame (alg **4**.).

Algorithm 4 **Proy** - Prolungamento dei risultati al livello corrente.

```
%% Ciclo sui livelli di risoluzione %%
for (k = level − 1; 0, step − 1) do
  if (k = level − 1) then
   for (x = 1, Pyr[k].row ; y = 1, Pyr[k].col) do
     Pyr[k].δv_x (x, y) = 0;
     Pyr[k].δv_y (x, y) = 0;
   endfor
  else
   %% I valori vengono duplicati in un intorno 2x2 %%
   for (x = 1, Pyr[k].row ; y = 1, Pyr[k].col) do
     Pyr[k].δv_x (x, y) = Pyr[k + 1].v_x (x/2, y/2) ;
     Pyr[k].δv_y (x, y) = Pyr[k + 1].v_y (x/2, y/2) ;
   endfor
  endif
endfor
```

La complessità di tempo per quest'operazione è:

$$2 \cdot T_{proy}(N \times M) = 2 \cdot \sum_{k=0}^{level-2} T^k_{proy}\left(\frac{N}{2^k} \times \frac{M}{2^k}\right) = 2 \cdot \sum_{k=0}^{level-2} 2m \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

A questo punto, possiamo analizzare la procedura per il calcolo del flusso ottico (alg. **5**)

Algorithm 5 **Optical_Flow** - Calcolo del flusso ottico

%% Ciclo sui livelli di risoluzione %%
for ($k = level - 1$; 0, $step - 1$) do

%% *Fase di inizializzazione* %%

%% Calcolo delle derivate del vettore velocità%%
Deriv_Vels (Pyr, $level$)

%% Calcolo delle componenti della variabile duale b %%
B_calc (Pyr, $level$)

%% Calcolo delle derivate, del gradiente e dell'equazione
%% costante del flusso ottico %%
Deriv_Imag (Pyr, $level$)

%% Calcolo della variabile duale a %%
A_calc (Pyr, $level$)

%% Calcolo della funzione c %%
C_calc (Pyr, $level$)

%% Schema del punto fisso %%
Fix_point_scheme (Pyr, $level$, $MaxIt$)

endfor

Precisamente, per valutare $\underline{b} = (b_x, b_y)$ ad ogni livello, escluso il più alto ($level - 1$), in cui i valori per le componenti di $\underline{b}$ sono tutti uguali a uno, bisogna determinare innanzitutto le derivate parziali delle componenti del vettore velocità (v_x, v_y) per poi valutare $\underline{b}$ come descritto in **4.2**.

Algorithm 6 **Deriv_Vels** - Derivate delle velocità.

%% Ciclo sui livelli %%
for $(k = level - 2;\ 0,\ step - 1)$ do

%% Differenze finite %%
for $(x = 1,\ Pyr\,[k]\,.row\ ;\ y = 1,\ Pyr\,[k]\,.col)$ do

$$Pyr\,[k]\,.\frac{\partial v_x}{\partial x}\,(x,\ y) = Pyr\,[k]\,.v_x\left(x + \tfrac{1}{2},\ y\right) - Pyr\,[k]\,.v_x\,(x,\ y)\,;$$

$$Pyr\,[k]\,.\frac{\partial v_x}{\partial y}\,(x,\ y) = Pyr\,[k]\,.v_x\left(x,\ y + \tfrac{1}{2}\right) - Pyr\,[k]\,.v_x\,(x,\ y)\,;$$

$$Pyr\,[k]\,.\frac{\partial v_y}{\partial x}\,(x,\ y) = Pyr\,[k]\,.v_y\left(x + \tfrac{1}{2},\ y\right) - Pyr\,[k]\,.v_y\,(x,\ y)\,;$$

$$Pyr\,[k]\,.\frac{\partial v_y}{\partial y}\,(x,\ y) = Pyr\,[k]\,.v_y\left(x,\ y + \tfrac{1}{2}\right) - Pyr\,[k]\,.v_y\,(x,\ y)\,;$$

endfor
endfor

Algorithm 7 **B_calc** - Componenti della variabile b.

%% Ciclo sui livelli %%
for $(k = level - 2;\ 0,\ step - 1)$ do

%% Secondo la formula (**7**) %%
for $(x = 1,\ Pyr\,[k]\,.row\ ;\ y = 1,\ Pyr\,[k]\,.col)$ do

$$norm^2 = \left(Pyr\,[k]\,.\frac{\partial v_x}{\partial x}\,(x,\ y)\right)^2 + \left(Pyr\,[k]\,.\frac{\partial v_x}{\partial y}\,(x,\ y)\right)^2$$

$$Pyr\,[k]\,.b_x\,(x,\ y) = \frac{1}{1+nrom^2}$$

$$norm^2 = \left(Pyr\,[k]\,.\frac{\partial v_y}{\partial x}\,(x,\ y)\right)^2 + \left(Pyr\,[k]\,.\frac{\partial v_y}{\partial y}\,(x,\ y)\right)^2$$

$$Pyr\,[k]\,.b_y\,(x,\ y) = \frac{1}{1+nrom^2}$$

endfor
endfor

La complessità di tempo per le quattro derivate parziali è:

$$4{\cdot}T_{deriv}\,(N \times M) = 4{\cdot}\sum_{k=0}^{level-2} T^k_{deriv}\left(\frac{N}{2^k} \times \frac{M}{2^k}\right) = 4{\cdot}\sum_{k=0}^{level-2} [1m + 2a]{\cdot}\left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

Quindi, per le due componenti di $\underline{b}$, risulta:

$$T_b\left(N \times M\right) = 4 \cdot T_{deriv}\left(N \times M\right) + 2 \cdot \sum_{k=0}^{level-2} \left[3m + 2a\right] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right) =$$

$$= \sum_{k=0}^{level-2} \left[10m + 12a\right] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

Analogamente, per valutare a ad ogni livello di risoluzione, è necessario determinare le derivate spaziali e temporali ed il gradiente della funzione che rappresenta l'intensità luminosa, da cui si ricava l'equazione costante del moto e infine il valore corrispondente alla variabile duale a (alg. **8-9**).

Algorithm 8 Deriv_Imag - Derivate spaziali e temporali, calcolo del gradiente.

%% Ciclo sui livelli %%

for $(k = level - 2;\ 0,\ step - 1)$ do

  %% Differenze finite %%

  for $(x = 1,\ Pyr\left[k\right].row\ ;\ y = 1,\ Pyr\left[k\right].col)$ do

    $Pyr\left[k\right].I_x\left(x,\ y\right) = Pyr\left[k\right].I_{LEFT}\left(x + \frac{1}{2},\ y\right) - Pyr\left[k\right].I_{LEFT}\left(x,\ y\right);$

    $Pyr\left[k\right].I_y\left(x,\ y\right) = Pyr\left[k\right].I_{LEFT}\left(x,\ y + \frac{1}{2}\right) - Pyr\left[k\right].I_{LEFT}\left(x,\ y\right);$

    $Pyr\left[k\right].I_t\left(x,\ y\right) = Pyr\left[k\right].I_{LEFT}\left(x,\ y\right) - Pyr\left[k\right].I_{RIGHT}\left(x,\ y\right);$

    $Pyr\left[k\right].\nabla I\left(x,\ y\right) = \sqrt{\left(Pyr\left[k\right].I_x\left(x,\ y\right)\right)^2 + \left(Pyr\left[k\right].I_y\left(x,\ y\right)\right)^2};$

  endfor

endfor

Algorithm 9 **A_calc** - Calcolo di a.

%% Ciclo sui livelli %%

for $(k = level - 1;\ 0,\ step - 1)$ do

  %% Secondo la formula (**6**) %%

  for $(x = 1,\ Pyr[k].row\ ;\ y = 1,\ Pyr[k].col)$ do

$$Pyr[k].Eq(x,\ y) = Pyr[k].v_x(x,\ y) \cdot Pyr[k].I_x(x,\ y) + \\ + Pyr[k].v_y(x,\ y) \cdot Pyr[k].I_y(x,\ y) + Pyr[k].I_t(x,\ y)$$

$$Pyr[k].a(x,\ y) = \frac{1}{1+(Pyr[k].Eq(x,\ y))^2}$$

  endfor

endfor

La complessità di tempo per questa fase è:

$$2 \cdot T_{deriv}(N \times M) + T_{grad}(N \times M) + T_{Eq}(N \times M) =$$

$$= \sum_{k=0}^{level-1} [5m + 6a] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

Unendo i termini, la complessità di tempo per valutare a è:

$$T_a(N \times M) = \sum_{k=0}^{level-1} [9m + 9a] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

Infine, bisogna valutare la funzione c. Stabiliti i valori dei parametri ***Sg*** e ***St***, la funzione $c(x)$, che regola la densità del flusso ottico, è definita come:

$$c(x) = \begin{cases} 0 & se\ \frac{\|\nabla I\|}{\max(\|\nabla I\|)} \geq Sg\ \ e\ \ \frac{|I_t|}{\max(|I_t|)} \geq St \\ \\ 1 & altrimenti \end{cases}$$

Si procede come mostrato in alg. **10**.

Algorithm 10 **C_calc** - Calcolo di c.

%% Ciclo sui livelli %%
for ($k = level - 1$; 0, $step - 1$) do
  %% Ricerca del max in ∇I %%
  $max\left(\|Pyr\left[k\right].\nabla I\left(x,\, y\right)\|\right)$
  %% Ricerca del max in I_t %%
  $max\left(|Pyr\left[k\right].I_t\left(x,\, y\right)|\right)$
  for ($x = 1$, $Pyr\left[k\right].row$; $y = 1$, $Pyr\left[k\right].col$) do
    if $\left(\frac{\sqrt{Pyr[k].\nabla I(x,\, y)}}{max(\|Pyr[k].\nabla I\|)} \geq Sg \;\&\&\; \frac{|Pyr[k].I_t(x,\, y)|}{max(|Pyr[k].I_t(x,\, y)|)} \geq St\right)$then
      $Pyr\left[k\right].c\left(x,\, y\right) = 1$;
    else
      $Pyr\left[k\right].c\left(x,\, y\right) = 0$;
    endif
  endfor
endfor

La complessità di tempo, in questo caso, è:

$$T_c\left(N \times M\right) = \sum_{k=0}^{level-1} \left[3m + 4c\right] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

Terminata la fase di inizializzazione, l'algoritmo procede con il calcolo della soluzione mediante il metodo iterativo corrispondente allo schema del punto fisso alternato. Indicato con $MaxIt$ il massimo numero di iterazioni, ad ogni livello di risoluzione, si valutano le componenti v_x, v_y della velocità, per ogni $n = 1, MaxIt$ e ogni volta si aggiornano le variabili $\underline{b}$ ed a con i nuovi risultati ottenuti (alg. **11**).

Algorithm 11
Fix_point_scheme - Schema del punto fisso per la minimizzazione alternata.

%% Ciclo sui livelli %%
for $(k = level - 1;\ 0,\ step - 1)$ do

%% Ciclo sul numero di iterazioni %%
for $(n = 1;\ MaxIt;\ step + 1)$ do

for $(x = 1,\ Pyr[k].row\ ;\ y = 1,\ Pyr[k].col)$ do

%% Calcolo dei coefficienti di D[3][3] e %%
%% P[3][3] come indicato in **4.5** %%

$$Pyr[k].divX(x, y) = \sum_{m=-1}^{1}\sum_{n=1}^{1} P(m, n)\cdot Pyr[k].v_x^n(x-m, y-n) + \\ + \sum_{m=-1}^{1}\sum_{n=1}^{1} D(m, n)\cdot Pyr[k].v_x^n(x-m, y-n)$$

$$Pyr[k].divY(x, y) = \sum_{m=-1}^{1}\sum_{n=1}^{1} P(m, n)\cdot Pyr[k].v_x^n(x-m, y-n) \\ + \sum_{m=-1}^{1}\sum_{n=1}^{1} D(m, n)\cdot Pyr[k].v_x^n(x-m, y-n)$$

endfor

%% Risoluzione delle equazioni di Eulero-Lagrange %%
for $(x = 1,\ Pyr[k].row\ ;\ y = 1,\ Pyr[k].col)$ do

if ($Pyr[k].c(x, y) = 1$) then

$$Pyr[k].v_x^n(x, y) = \frac{\alpha^r}{\alpha^h}\cdot Pyr[k].divX(x, y) - \\ -\frac{1}{\alpha^h}\cdot Pyr[k].a(x, y)\cdot Pyr[k].Eq(x, y)$$

$$Pyr[k].v_y^n(x, y) = \frac{\alpha^r}{\alpha^h}\cdot Pyr[k].divY(x, y) - \\ -\frac{1}{\alpha^h}\cdot Pyr[k].a(x, y)\cdot Pyr[k].Eq(x, y)$$

endif

endfor

%% Aggiornamento delle variabili ausiliarie %%
Deriv_Vels $(Pyr,\ level)$

B_calc $(Pyr,\ level)$

Deriv_Imag $(Pyr,\ level)$

A_calc $(Pyr,\ level)$

endfor
endfor

In pratica, per valutare v_x e v_y è necessario determinare i due termini relativi alla divergenza, con complessità:

$$2 \cdot MaxIt \cdot T_{div}\left(N \times M\right) = 2 \cdot MaxIt \cdot \sum_{k=0}^{level-1} \left[18m + 28a\right] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

da cui si ricavano le componenti del vettore velocità v_x, v_y, per cui:

$$MaxIt \cdot T_{vels}\left(N \times M\right) = MaxIt \cdot \sum_{k=0}^{level-1} \left[5m + 2a\right] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

Inoltre la complessità computazionale necessaria ad aggiornare le variabili $\underline{b}$ ed a, ad ogni iterazione esclusa l'ultima, è:

$$\left(MaxIt - 1\right) \cdot T_b\left(N \times M\right) = \left(MaxIt - 1\right) \cdot \sum_{k=0}^{level-1} \left[10m + 12a\right] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

$$\left(MaxIt - 1\right) \cdot T_a\left(N \times M\right) = \left(MaxIt - 1\right) \cdot \sum_{k=0}^{level-1} \left[9m + 9a\right] \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

Infine, terminato il procedimento iterativo, ai risultati ottentuti v_x e v_y vengono aggiunti gli incrementi ottenuti dal livello precedente (alg. **12**).

Algorithm 12 Add - Incremento al flusso ottico dal livello di risoluzione precedente.

```
%% Ciclo sui livelli %%
for (k = level − 2; 0, step − 1) do
    for (x = 1, Pyr[k].row ; y = 1, Pyr[k].col) do
      Pyr[k].v_x(x, y) = Pyr[k].v_x(x, y) + Pyr[k].δv_x(x, y)
      Pyr[k].v_y(x, y) = Pyr[k].v_y(x, y) + Pyr[k].δv_y(x, y)
    endfor
  endfor
endfor
```

Per tale operazione la complessità di tempo è:

$$T_{Inc}\left(N \times M\right) = \sum_{k=0}^{level-2} 2a \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)$$

Raggruppando i termini corrispondenti alle singole procedure, si ottiene la ***complessità di tempo*** per l'intero algoritmo:

$$T_{flow}\left(N \times M\right) = T_{Ind}\left(N \times M\right) + MaxIt \cdot T_{Fix}\left(N \times M\right) =$$

$$\simeq \sum_{k=0}^{level-1} \left[13m + 4c + 12a\right] \left(\frac{N}{2^k} \times \frac{M}{2^k}\right) + MaxIt \cdot \left[\sum_{k=0}^{level-1} \left[60m + 79a\right] \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)\right]$$

$$\simeq MaxIt \cdot \left[\sum_{k=0}^{level-1} \left[4c + 73m + 91a\right] \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)\right]$$

dove si è indicato con $T_{Ind}\left(N \times M\right)$ la complessità di tempo di tutta la fase preliminare al calcolo, che comprende la costruzione della piramide, l'inizializzazione delle variabili e la proiezione dei risultati al livello di risoluzione successivo; e con $MaxIT \cdot T_{Fix}\left(N \times M\right)$ la complessità di tempo necessaria allo schema iterativo del punto fisso alternato per il calcolo della soluzione.

Assunto che il tempo di esecuzione di una moltiplicazione sia comparabile a quello di esecuzione di una addizione e che l'operazione di confronto è trascurabile rispetto a un'operazione ***floating-point*** (moltiplicazione/addizione), si ha:

$$T_{flow}\left(N \times M\right) \simeq MaxIt \cdot \left[\sum_{k=0}^{level-1} 164\, flop \cdot \left(\frac{N}{2^k} \times \frac{M}{2^k}\right)\right] \simeq O\left(N \times M\right)$$

dove $flop$ indica l'operazione *floating-point*.

5.2 Modulo 2: algoritmo *detect*.

Specifiche:

detect <*path*> <*frame*> <*file1*> <*file2*> <*option*>

dove:

<*path*> è il nome della sequenza in esame.

<*frame*> è nr, il numero corrispondente al fotogramma da restaurare.

<*file1*> il file che contiene il flusso ottico valutato da $nr-1$ a nr.

<*file2*> il file che contiene il flusso ottico valutato da nr a $nr+1$.

<*option*> le opzioni per l'uso

- ***-s;*** nel caso in cui si desidera salvare le immagini durante l'elaborazione.
- ***-K*** *Kval;* è la soglia per il calcolo della maschera. Per default ha valore 50.

Specificato il path che contiene il nome della sequenza da esaminare e i file relativi alle informazioni del flusso ottico valutato dal modulo precedente, l'algoritmo funziona come riportato in alg **13**.

Algorithm 13 **Schema generale**

%% Lettura dei dati %%
Immagini della sequenza:
I_{nr}, $Inr-1$, $Inr+1$ rispettivamente il frame in esame, il precedente e il successivo
Flusso ottico:
V^x_{nr-1}, V^y_{nr-1} campo di moto valutato tra $Inr-1$ e Inr
V^x_{nr+1}, V^y_{nr+1} campo di moto valutato tra Inr e $Inr+1$
%% Compensazione del moto —> immagini compensate: I^C_{next}, I^C_{last} %%
Warped $\left(I_{nr-1}, I_{nr}, I_{nr+1}, V^x_{nr-1}, V^y_{nr-1}, V^x_{nr+1}, V^y_{nr+1}\right)$
%% Individuazione del blocth —> maschera del blotch B %%
Mask $\left(I^C_{next}, I^C_{last}, Kval, I_{nr}, I_{nr-1}, I_{nr+1}\right)$
%% Perfezionamento della maschera del blocth %%
Open (B)

Il modulo per l'individuazione dei difetti fornisce in output la maschera del blotch B e le immagini compensate I^C_{next} e I^C_{last}.

Per l'analisi della complessità si spazio relativa all'algoritmo, osserviamo che se l'immagine da restaurare I_{nr} ha dimensione $N \times M$, la struttura dati nessaria risulta costituita da:

- I_{nr}, I_{nr-1}, I_{nr+1} rispettivamente l'immagine della sequenza in esame, la precedente e la successiva,
- I^C_{last}, I^C_{next} rispettivamente l'immagine compensata in avanti e l'immagine compensata all'indietro,
- W_{last}, W_{next}, W_{diff}, B per le differenze e la maschera del blotch.

Sono, dunque, necessari 9 array di memoria di lunghezza $(N \times M)$ e la ***complessità di spazio*** dell'algoritmo risulta:

$$T_S\left(N \times M\right) = 9 \cdot \left(N \times M\right) \simeq O\left(N \times M\right)$$

Analizziamo ora in dettaglio i singoli moduli dello schema generale (alg **13**).
Il primo passo consiste nel calcolo delle immagini compensate (alg. **14**), la cui complessità di tempo è:

$$T_{Warp}\left(N \times M\right) = 2 \cdot \left[2a\right] \cdot \left(N \times M\right)$$

Algorithm 14 **Warped** - Immagini compensate

%% Compensazione del moto con l'immagine precedente e la successiva %%

for $(x = 1,\ row,\ step + 1;\ y = 1,\ col,\ step + 1)$ do

$I^C_{last}(x,\ y) = I_{n-1}\left(x + V^x_{nr-1}(x,\ y),\ y + V^y_{nr-1}(x,\ y)\right);$
$I^C_{next}(x,\ y) = I_{n+1}\left(x - V^x_{nr+1}(x,\ y),\ y - V^y_{nr+1}(x,\ y)\right);$

endfor

Quindi, si procede con la fase di individuazione del dominio del blotch come descritto in **4.3** (alg. **15**).

Algorithm 15 **Mask** - Individuazione dei difetti

%% Stima delle differenze %%

for $(x = 1,\ row,\ step + 1;\ y = 1,\ col,\ step + 1)$ do

$W_{last}(x,\ y) = \left|I^C_{last}(x,\ y) - I_{nr}(x,\ y)\right|;$
$W_{next}(x,\ y) = \left|I^C_{next}(x,\ y) - I_{nr}(x,\ y)\right|;$
$W_{diff}(x,\ y) = |W_{last}(x,\ y) - W_{next}(x,\ y)|;$

endfor

%% Maschera dei difetti %%

for $(x = 1,\ row,\ step + 1;\ y = 1,\ col,\ step + 1)$ do

if $((W_{last}(x,\ y) \geq Kval\ ||\ W_{next}(x,\ y) \geq Kval)\ \&\&\ W_{diff}(x,\ y) \leq Kval)$;
$B(x,\ y) = 1;$
else
$B(x,\ y) = 0;$
endif

La complessità di tempo per l'individuazione dei difetti è:

$$T_{mask}(N \times M) = [3a + 3c] \cdot (N \times M)$$

Il risultato è un'immagine binaria B che viene successimante, sottoposta all'operazione di *apertura* (alg. **16**).

Algorithm 16 **Open** - Operazione morfologica di apertura

%% Erosione %%
for ($x = 1$, row, $step + 1$; $y = 1$, col, $step + 1$) do
  $val = \sum_{m=-1}^{1} \sum_{n=-1}^{1} S(m, n) \cdot B(x - m, y - n)$;
  if ($val == 9$) then
    $B(x, y) = 1$;
  else
    $B(x, y) = 0$;
endfor
%% Dilatazione %%
for ($x = 1$, row, $step + 1$; $y = 1$, col, $step + 1$) do
  $val = \sum_{m=-1}^{1} \sum_{n=-1}^{1} S(m, n) \cdot B(x - m, y - n)$;
  if ($val \geq 1$) then
    $B(x, y) = 1$;
  else
    $B(x, y) = 0$;
  endif
endfor

La complessità di tempo per quest'ultima operazione è:

$$T_{open}(N \times M) = [9m + 10a + 1c] \cdot (N \times M)$$

Trascurando i termini che coinvolgono l'operazione di confronto in T_{mask} e T_{open}, si ha che la ***complessità di tempo*** dell'algoritmo *detect* è:

$$T_{detect}(N \times M) = T_{Warp}(N \times M) + T_{mask}(N \times M) + T_{open}(N \times M) =$$
$$= 4a \cdot (N \times M) + 3a \cdot (N \times M) + [9m + 10a] \cdot (N \times M) \cong$$
$$\cong 26flop(N \times M)$$

5.3 Modulo 3: algoritmo *remove*.

Specifiche:

remove <*path*> <*frame*> <*file c1*> <*file c2*> <*mask*> <*option*>

dove:

<*path*> è il nome della sequenza in esame.

<*frame*> è nr, il numero corrispondente al fotogramma da restaurare.

<*file c1*> l'immagine compensata I^C_{nr-1}.

<*file c2*> l'immagine compensata I^C_{nr+1}.

<*mask*> la maschera del blotch.

<*option*> le opzioni per l'uso

- ***-s;*** nel caso in cui si desidera salvare le immagini intermedie durante l'elaborazione.
- ***-D*** *MaxIt;* è il massimo numero di iterazioni nel processo di minimizzazione. Per default ha valore 10.

Specificato il path che contiene il nome della sequenza da esaminare, la maschera del blotch e le immagini compensate valutate, il modulo funziona come riportato in alg. **17**.

Algorithm 17 **Schema generale**

%% *Lettura dei dati* %%
Immagine da restaurare: I_{nr}
Machera del blotch e le immagini compensate: I^C_{nr-1}, I^C_{nr+1}, B
%% *Inizializzazione* %%
%% media delle immagini compensate I^C–> I_0^{alg} %%
Temp $\left(B, I^C_{last}, I^C_{next}\right)$
%% calcolo di b %%
Deriv_Imm_B $\left(I_0^{alg}\right) - - > b^1$
%% *Schema del punto fisso* %%
Fix-point-scheme $\left(I_0^{alg}, MaxIT, b^1\right)$

Il modulo per la rimozione dei difetti fornisce in output la nuova immagine $I^{\star}_{nr}$, così definita:

$$\forall\,(x,\,y):\,I^{\star}_{nr}\,(x,\,y)=\begin{cases} I^{alg}_{nr}\,(x,\,y) & se\,B\,(x,\,y)=1 \\ I_{nr}\,(x,\,y) & se\,B\,(x,\,y)=0 \end{cases}$$

La complessità di spazio dell'algoritmo dipende dalle dimensioni dell'immagine I_{nr} da restaurare e dalla cardinalità del dominio del blotch. La struttura dati necessaria è costituita da:

- I_{nr} l'immagine della sequenza in esame,
- I^{C}_{last}, I^{C}_{next} rispettivamente l'immagine compensata in avanti e l'immagine compensata all'indietro,
- I^{C} per l'interpolazione temporale e per l'interpolazione spaziale,
- div, I_x, I_y per la divergenza e per le derivate spaziali,
- b per la variabile ausiliaria di *Geman-Reynold,*
- $I^{\star}_{nr}$ per l'immagine finale restaurata.

Sono necessari , dunque,3 array di memoria di lunghezza $N \times M$ e 5 array di lunghezza $N_B \times M_B$.

Quest'ultimo valore indica il numero di pixel corrotti, ovvero la cardinalità dell'insieme discreto B che definisce la maschera del blotch.
Poichè $N_B \times M_B \leq N \times M$, la ***complessità di spazio*** dell'algoritmo risulta:

$$T_S\,(N \times M) = 3 \cdot (N \times M) + 5 \cdot (N_B \times M_B) \simeq O\,(N \times M)$$

Analizziamo ora in dettaglio i singoli moduli dello schema generale (alg **17**).

Le operazioni che seguono vengono effettuate solo per i pixel dell'immagine per cui $B(x,\, y)$ è contrassegnato dal valore uno, ovvero solo per i punti appartenenti alla zona del blocth.

Alla fase di lettura, segue quella di inizializzazione in cui vengono calcolate le condizioni inziali per il successivo schema iterativo (alg. **18-19**):

Algorithm 18 **Temp** - Inizializzazione**.**

%% Interpolazione temporale %%
for $(x = 1,\, row,\, step + 1;\, y = 1,\, col,\, step + 1)$ do
  if $(B(x,\, y) == 1)$ then
    $I^C(x,\, y) = \frac{I^C_{last}(x,\, y) + I^C_{next}(x,\, y)}{2}$
  endif
endfor

La complessità computazionale necessaria a valutare la media (alg. **18**) delle immagini compensate è:

$$T_{temp}(N \times M) = 1c \cdot (N \times M) + [1a + 1m] \cdot (N_B \times M_B)$$

Algorithm 19 **Deriv_Imm -** calcolo delle derivate

%% Calcolo delle derivate di I_i^{alg}, $i = 0, ..., MaxIt$ e quindi di b variabile duale %%
  for $(x = 1,\, row,\, step + 1;\, y = 1,\, col,\, step + 1)$ do
    if $(B(x,\, y) = 1)$ then
      $I_x(x,\, y) = I_{n-1}\left(x + \frac{1}{2},\, y\right) - I_{n-1}(x,\, y)$
      $I_y(x,\, y) = I_{n-1}\left(x,\, y + \frac{1}{2}\right) - I_{n-1}(x,\, y)$
      $norm^2 = (I_x(x,\, y))^2 + (I_y(x,\, y))^2$
      $b(x,\, y) = \frac{1}{1 + norm^2}$;
    endif
  endfor

La complessità computazionale per l'algoritmo **19**, risulta:

$$T_b\,(N \times M) = 1c \cdot (N \times M) + [4m + 3a] \cdot (N_B \times M_B)$$

Infine, il calcolo della soluzione si basa sullo schema di minimizzazione del punto fisso alternato descritto in appendice **C**.

Algorithm 20 **Fix-Point-Scheme** - Rimozione dei difetti

%% Schema di minimizzazione %%
%% Ciclo sul numero di iterazioni %%
for $(n = 1,\, MaxIt,\, step + 1)$ do

%% Calcolo dei coefficienti di $P\,[3]\,[3]$ e $D\,[3]\,[3]$ definite in **4** %%
for $(x = 1,\, row,\, step + 1;\, y = 1,\, col,\, step + 1)$ do

if $(B\,(x,\, y) == 1)$ then

$$div\,(x,\, y) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} P\,(m,\, n) \cdot I_{n-1}\,(x - m,\, y - n) + \\ + \sum_{m=-1}^{1} \sum_{n=-1}^{1} D\,(m,\, n) \cdot I_{n-1}\,(x - m,\, y - n)\,;$$

endif
endfor
%% calcolo di I^B all'iterazione n %
for $(x = 1,\, row,\, step + 1;\, y = 1,\, col,\, step + 1)$ do

if $(B\,(x,\, y) == 1)$ then

$$I_n\,(x,\, y) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} G\,(m,\, n) \cdot I_{n-1}\,(x - m,\, y - n)\,;$$
$$I_n\,(x,\, y) = I_n\,(x,\, y) - \lambda \cdot div\,(x,\, y)\,;$$

endif
endfor
%% aggiornamento di b %%
Deriv_Imm_B $(I_n) - - > b^{n+1}$
endfor
%% fine del ciclo sulle iterazioni %%

La complessità per il calcolo per lo schema del punto fisso alternato (alg. **20**) è data dalla somma delle complessità relative alle seguenti operazioni:

- Il calcolo della divergenza:

$$T_{div}\left(N \times M\right) = MaxIt \cdot \left[1c \cdot \left(N \times M\right) + \left[18m + 28a\right] \cdot \left(N_B \times M_B\right)\right]$$

- L'operazione di convoluzione con kernel gaussiano:

$$T_{conv}\left(N \times M\right) = MaxIt \cdot \left[1c \cdot \left(N \times M\right) + \left[2m + 4a\right] \cdot \left(N_B \times M_B\right)\right]$$

- Il calcolo della soluzione:

$$T_{sol}\left(N \times M\right) = \left(MaxIt\right) \cdot \left[1c \cdot \left(N \times M\right) + \left[1m + 1a\right] \cdot \left(N_B \times M_B\right)\right]$$

- Aggiornamento di b:

$$T_b\left(N \times M\right) = \left(MaxIt - 1\right) \cdot \left[1c \cdot \left(N \times M\right) + \left[4m + 3a\right] \cdot \left(N_B \times M_B\right)\right]$$

Quindi, la ***complessità di tempo*** per il modulo di rimozione è:

$$T_{remove}\left(N \times M\right) = 1c\left(N \times M\right) + \left[1m + 1a\right]\left(N_B \times M_B\right) +$$

$$+MaxIt\left[2c\left(N \times M\right) + \left[4m + 3a\right]\left(N_B \times M_B\right) + \left[18m + 28a\right]\left(N_B \times M_B\right)\right] +$$

$$+MaxIt\left[2c\left(N \times M\right) + \left[2m + 4a\right]\left(N_B \times M_B\right) + \left[1m + 1a\right]\left(N_B \times M_B\right)\right] \cong$$

$$\cong MaxIt\left[4\left(N \times M\right)c + \left[24m + 37a\right]\left(N_B \times M_B\right)\right]$$

Anche in questo caso, l'operazione di confronto è trascurabile ed è possibile eliminare il primo termine in T_{remove}, per cui:

$$T_{remove}\left(N \times M\right) \cong \left(N_B \times M_B\right) 61 flop$$

5.4 Stima dell'errore.

Per ognuno dei moduli proposti è possibile dare una stima dell'errore osservando i risultati ottenuti e confrontandoli, nel caso di sequenze di immagini sintetiche, con i valori della soluzione già noti.

Precisamente, la misura dell'errore nel calcolo del flusso ottico, può essere valutata solo nel caso in cui il campo del moto originale è noto a priori. Ciò è sempre possibile per sequenze di immagini di tipo sintetico per cui il moto è generato artificialmente.

Se, per ogni punto di coordinate $(x,\, y)$, si indica con $\overrightarrow{v}_c\,(x,\, y)$ la velocità corretta e con $\overrightarrow{v}_e\,(x,\, y)$ la velocità stimata, si può definire l'**errore angolare** tra i due vettori velocità (fig. **2**):

$$\psi_E\,(x,\, y) = \arccos\left(\overrightarrow{v}_c\,(x,\, y) \times \overrightarrow{v}_e\,(x,\, y)\right)$$

e dare una una misura dell'accuratezza della soluzione calcolata dall'algoritmo, osservando che tanto più piccolo è il valore $\psi_E\,(x,\, y)$, tanto più sono affidabili i risultati ottenuti.

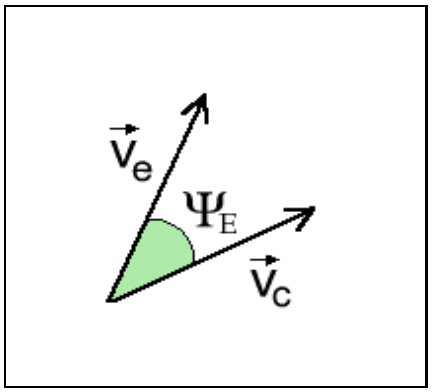

Figura 2: Errore angolare.

Inoltre, indicato con nr il numero di pixel per cui è stata valutata la velocità, è possibile stimare l'*errore angolare medio*, mediante la relazione:

$$AveError = \frac{\sum_{x,y} (\psi_E (x, y))}{nr},$$

e la *densità* del flusso ottico ottenuto confrontando nr con il numero totale di pixel nell'immagine: $n = N \times M$.

Anolagamente, per ottenere una stima dell'errore relativo agli algoritmi di individuazione e rimozione è possibile corrompere artificialmente una sequenza di immagini digitali introducendo, su alcuni fotogrammi, macchie di cui si conosce la posizione e l'intensità luminosa corretta.
Quindi, l'accuratezza relativa all'algoritmo di individuazione dei difetti può essere valutata a partire dalla maschera del blotch ottenuta.
In particolare, il numero dei pixel corrotti individuati e la loro posizione possono essere confrontati con i dati corrispondeti alla maschera creata, al fine di ottenere una stima della percentuale d'errore nei risultati.
Mentre, l'accuratezza dell'algoritmo di rimozione si ricava confrontando i valori ottenuti per le zone ricostruite e i corrispondenti valori che appartengono all'immagine non corrotta. Tale confronto può essere rappresentato graficamente mediante *profilo trasversale*.

5.5 Analisi della convergenza.

Nell'analisi computazionale degli algoritmi relativi ai moduli ***flow*** e ***detect*** si è fissato un massimo numero di iterazioni ***MaxIt***, per il calcolo della soluzione mediante lo schema iterativo del punto fisso alteranto (appendice **C**).

Tale valore dipende dalla velocità di convergenza del metodo e dai parametri di regolarizzazione impiegati e può essere determinato osservando l'andamento della funzione corrispondente all'***errore quadratico medio*** (***MSE***) valutato ad ogni iterazione.

Se indichiamo con $\overrightarrow{v}_k(x,\, y)$ il flusso ottico valutato, per il pixel $(x,\, y)$ dell'immagine Ω di dimensione $N \times M$, all'iterazione k, il valore dell'*MSE* relativo al modulo per la stima del campo di moto, è dato dalla relazione:

$$mse(k) = \frac{1}{N \times M} \sum_{x=1}^{M} \sum_{y=1}^{N} [\overrightarrow{v}_k(x,\, y) - \overrightarrow{v}_{k-1}(x,\, y)]^2$$

Allo stesso modo si osserva la convergenza dello schema iterativo relativo al modulo di rimozione dei difetti.
Posto $I_k^B(x,\, y)$ l'intensità luminosa valutata, per il pixel $(x,\, y) \in B$ di cardinalità $N_B \times M_B$, all'iterazione k, il valore dell'*MSE* è dato dalla relazione

$$mse(k) = \frac{1}{N_B \times M_B} \sum_{x=1}^{N_B} \sum_{y=1}^{M_B} \left[I_k^B(x,\, y) - I_{k-1}^B(x,\, y)\right]^2$$

Gli algoritmi implementati prevedono un criterio d'arresto automatico basato sul confronto, ad ogni iterazione k, tra il valore dell'errore quadratico medio $mse(k)$ e una tolleranza prefissata Tol, interrompendo il procedimento quando $mse(k) \leq Tol = 10^{-3}$.

L'accuratezza richiesta nel calcolo della soluzione è, dunque, fino a tre cifre decimali. Con questa scelta è possibile determinare una buona approssimazione sia dei vettori spostamento che dovranno essere successivamente utilizzati per il calcolo delle immagini compensate, sia per i

valori di grigio corrispondenti alle zone restaurate minimizzando l'onere computazionale.
Un'accuratezza maggiore risulta superflua, in quanto nella rappresentazione grafica dell'immagine ricostruita tutti i valori di grigio vengono convertiti in numeri interi compresi fra 0 e 255.

5.6 L'introduzione del parallelismo.

Il nucleo computazionale degli algoritmi implementati consiste nella risoluzione numerica di un sistema di *PDE* definite in tutto il dominio $\Omega \times J$ della sequenza in esame. Ciò implica che decomponendo il dominio è possibile suddividere il problema differenziale in sottoproblemi. La strategia di parallelismo impiegata si basa, dunque, su una discretizzazione del dominio locale in cui le equazioni stesse sono definite (fig. **3**).

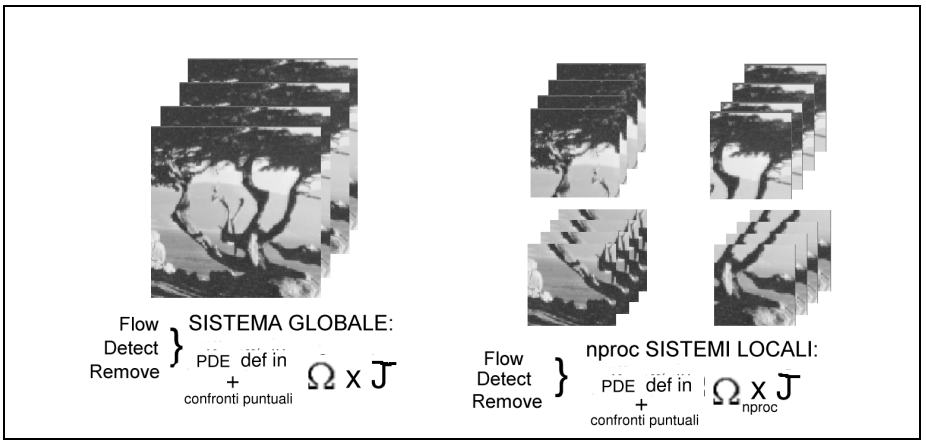

Figura 3: Suddivisione del dominio della sequenza in più sottodomini.

Se $nproc$ è il numero di processi tra cui si può suddividere il problema iniziale, ogni immagine della sequenza può essere ripartita in più immagini di dimensioni minori secondo una topologia virtuale denominata ***griglia cartesiana virtuale***.
Ad ogni processo, individuato dalla coppia di coordinate cartesiane $(x,\ y)$, tali che:

$$\begin{cases} x \in [0,\ p-1] \subset \aleph \\ \\ y \in [0,\ q-1] \subset \aleph \end{cases} \qquad dove:\ nproc = p \times q$$

è assegnata una porzione dell'immagine (fig. **4**) di dimensione:

$$\frac{N}{p} \times \frac{M}{q}$$

Quindi, ognuno di essi calcola la soluzione locale sul proprio dominio.

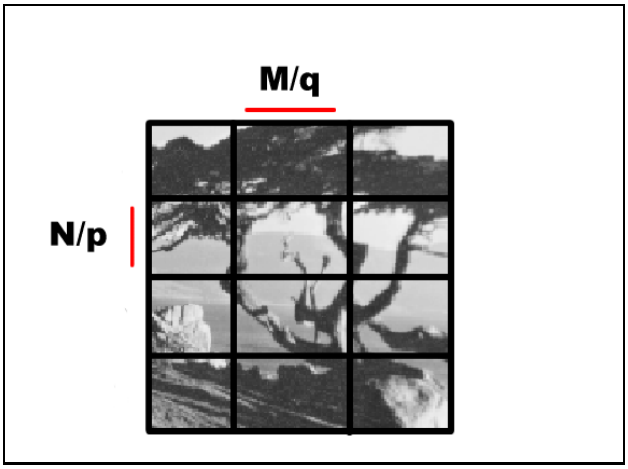

Figura 4: Suddivisione dei dati.

Inoltre, è necessario che ogni processo abbia a disposizione anche i dati della frontiera, coinvolti nel calcolo della soluzione locale.

Pertanto, poichè ad ogni pixel dell'immagine è associato un intorno discreto di raggio ω (ovvero una finestra di dimensione $\omega \times \omega$ centrata nel pixel in esame[1]), allora ad ogni processo è necessario un blocco dell'immagine di dimensione:

$$\left(\frac{N}{p} + \omega\right) \times \left(\frac{M}{q} + \omega\right)$$

che contiene la porzione di dati fissa di dimensione $\frac{N}{p} \times \frac{M}{q}$ e una porzione di "contorno" (fig. **5**) di ampiezza ω, relativa alle informazioni necessarie ai pixel che si trovano sulla frontiera dei blocchi.

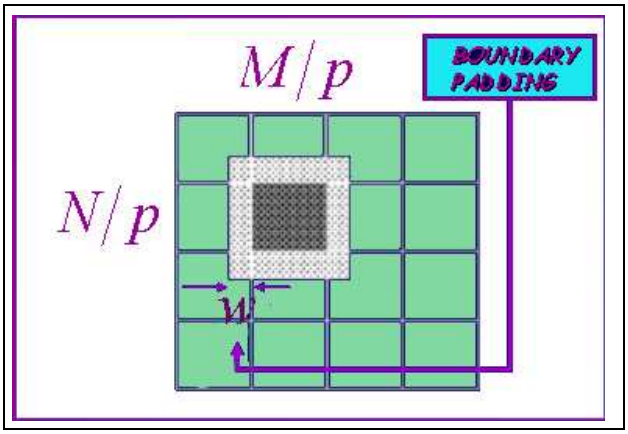

Figura 5: Suddivisione dei dati per ogni processo.

Durante il procedimento di calcolo, tali valori sono impiegati ed, eventualmente, modificati. Quindi, ogni processo comunica i dati aggiornati appartenenti alla zona di *contorno* ai processi ad esso associati (secondo la topologia considerata) come mostrato in figura **6**.

[1]In questo lavoro si utilizza $\omega = 3$,che corrisponde all'ampiezza degli operatori utilizzati per il calcolo delle derivate parziali (differenze finite - *stecil a 3 punti*) e delle *maschere* utilizzate sia per la costruzione della struttura piramidale in multirisoluzione che per le operazioni morfologiche.

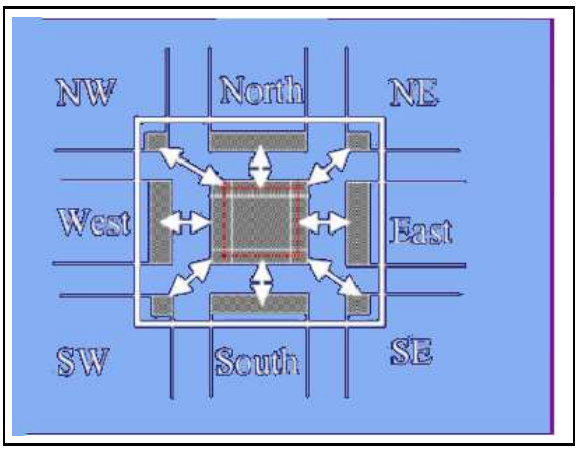

Figura 6: Stategia di comunicazione.

Le prestazioni del software parallelo implementato possono essere valutate osservando i grafici di ***speedup*** ed ***efficienza*** relativi ad ogni singolo modulo.
Indicato con T_1 il tempo di esecuzione dell'algoritmo su un unico processore e in generale, con T_{nproc} il tempo di esecuzione dell'algoritmo su *nproc* processori, lo *speedup* è dato dalla relazione:

$$S_{nproc} = \frac{T_1}{T_{nproc}}$$

e misura la riduzione del tempo d'esecuzione, ovvero l'aumento della velocità di esecuzione all'aumentare del numero di processori coinvolti nel calcolo. Quindi, l'*efficienza* risulta:

$$E_{nproc} = \frac{S_{nproc}}{nproc}$$

Capitolo 6

Risultati sperimentali.

In questo capitolo sono riportati i risultati ottenuti per sequenze di immagini sintetiche e reali, analizzando le prestazioni del software proposto, in termini di accuratezza ed efficienza.

In particolare, per le sequenze di tipo sintetico è possibile fornire una stima dell'errore confrontando i valori ottenuti con quelli già noti, come descritto in **5.4**.
Mentre per ognuno degli esempi considerati è possibile osservare l'andamento della funzione che rappresenta l'errore quadratico medio, per verificare i tempi necessari alla convergenza dei metodi iterativi, relativi ai moduli per la stima del flusso ottico e per l'eliminazione dei difetti individuati.

6.1 Sequenze di immagini sintetiche.

L'esempio riportato in figura **1** mostra un'immagine reale, a partire dalla quale è possibile costruire una sequenza di immagini corrotta dalla presenza di blotch: simulando il moto (*zoom* della telecamera) e introducendo macchie artificiali in zone prestabilite.

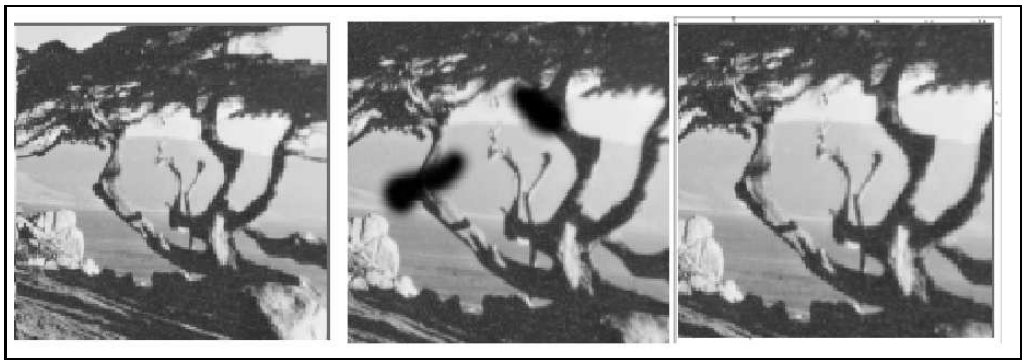

Figura 1: **TREED** - Sequenza di immagini sintetica. Il moto è simulato con uno zoom sull'immagine e nel fotogramma centrale sono state introdotte due macchie. Dimensioni originali 156x156 pixel.

In figura **2** si osserva il flusso ottico stimato (modulo ***flow***), che può essere confrontato con il campo di moto originale.
Scegliendo opportunamente i valori dei parametri ***Sg*** e ***St***, che regolano la densità del flusso ottico, è possibile ricavare solo le informazioni necessarie al modulo successivo, ovvero quelle relative ai contorni degli oggetti presenti nella scena, che risultano sufficienti al calcolo delle immagini compensate.

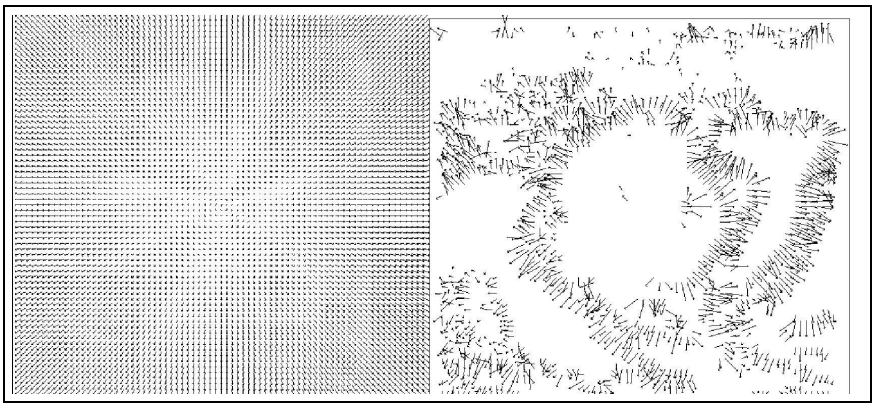

Figura 2: **TREED -** A sinistra il campo di moto originale, a destra il flusso ottico valutato dal modulo ***flow*.** Tempo di cpu (versione sequenziale)- **IBM SP RS6000: 6.67 secs.**

In figura **3** è riportata la maschera del blotch (modulo ***detect***), in cui sono evidenziati i pixel corrispondenti alla zona corrotta e l'immagine restaurata (modulo ***remove***).

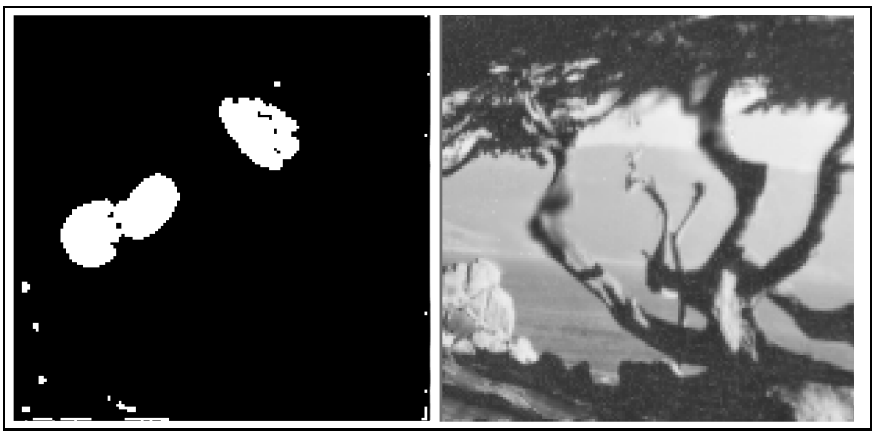

Figura 3: **TREED -** A sinistra la maschera del blotch valutata dal modulo **detect**. Tempo di cpu (versione sequenziale)- **IBM SP RS6000: 1.721 secs**. A destra l'immagine restaurata mediante il modulo **remove**. Tempo di cpu (versione sequenziale)- **IBM SP RS6000: 2.924 secs.**

L'accuratezza dei risultati ottenuti per la stima del flusso ottico è mostrata in tabella **1**, in cui sono riportati i valori corrispondenti all'errore medio, valutato rispetto alla densità del campo di moto.
Tali valori possono essere confrontati con i risultati ottenuti dai più noti algoritmi per la stima del moto **[5]**.

Metodo	Errore Medio	Densità
Horn-Schunck	2.50°	32.9%
Lucas-Kanade	1.65°	24.3%
Uras	3.83°	60.2%
Nagel	3.21°	53.5%
Anandan	7.64°	100%
Singh	7.09°	3.9%
Feature Matching	0.34°	65.9%
Modulo 1	0.22°	56.49%

Tabella 1: **TREED -** Accuratezza del modulo ***flow***. Confronto tra l'errore medio e la densità del flusso ottico.

L'accuratezza dei risultati relativi al modulo di individuazione dei difetti dipende, invece, dalla scelta del parametro $\boldsymbol{K}$ nell'intervallo [0, 255]. In tabella **2** è riportato il numero di pixel della maschera originale del blotch e, per ogni scelta del valore di soglia, il numero dei pixel ottenuti e la percentuale in posizione corretta.

K	Numero di pixel individuati	Percentuale dei pixel individuati in posizione corretta
maschera originale	3009	100%
10	4267	65%
50	3273	89%
100	2701	71%
150	2035	79%

Tabella 2: **TREED -** Accuratezza del modulo ***detect***. Confronto tra la maschera originale del blotch e i risultati ottenuti al variare della soglia $\boldsymbol{K}$.

Si osserva che per valori di K relativamente bassi si individua il maggior numero dei pixel corrotti, ma non sempre nella posizione esatta. Invece, incrementando K il numero di pixel individuato diminuisce, ma aumenta la precisione. In generale poi, per valori troppo grandi della soglia K l'individuazione del numero dei pixel appartenenti alla zona corrotta risulta fortemente penalizzato.

Infine, per la stima dell'accuratezza relativa al modulo di rimozione dei difetti, si possono osservare i grafici che descrivono l'andamento dei valori corrispondenti ai profili trasversali riportati in figura **4** e in figura **5**.

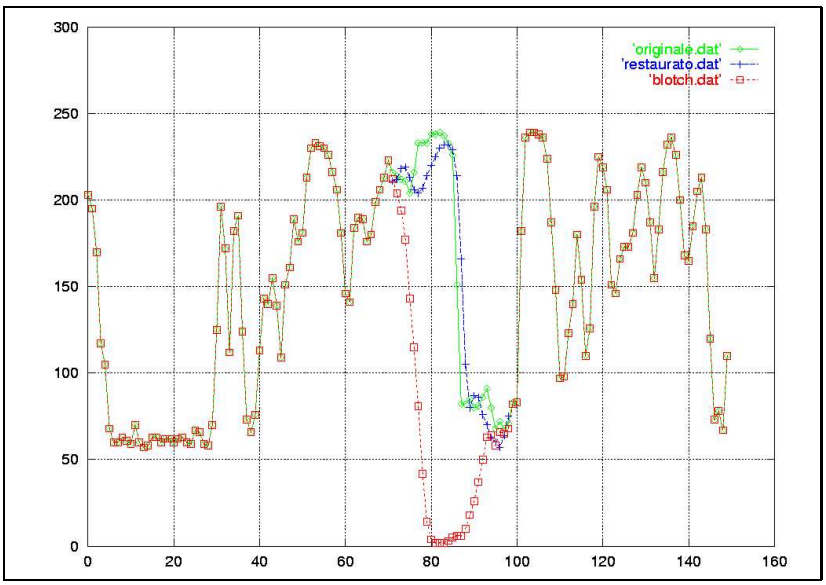

Figura 4: **TREED** - Accuratezza del modulo ***remove***. Profilo trasversale in corrispondenza della zona corrotta in alto a destra.

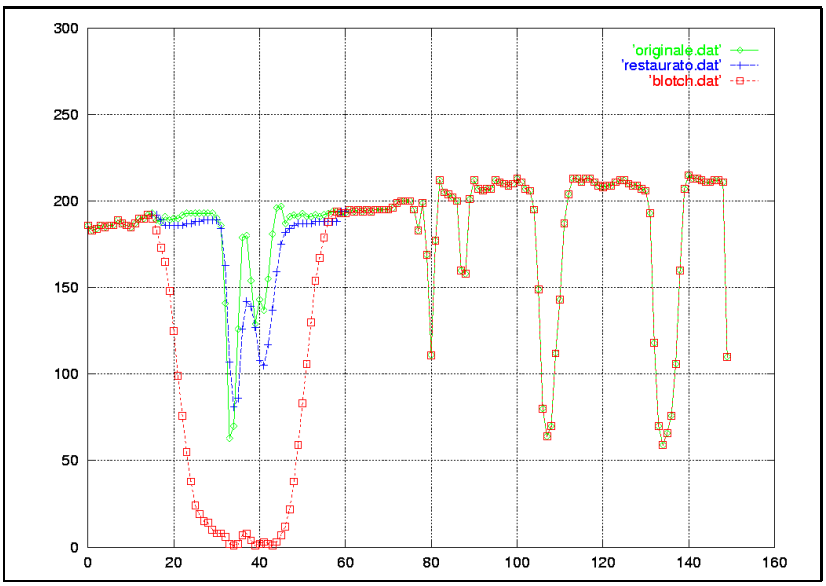

Figura 5: **TREED** - Accuratezza del modulo ***remove***. Profilo trasversale in corrispondenza della zona corrotta in alto a sinistra.

In figura **6** e in figura **7** si osserva la convergenza dei metodi iterativi (schema del punto fisso alternato). L'accuratezza scelta è 10^{-3} e i valori corrispondenti all'*mse* sono riportati in scala logaritmica, fino al raggiungimento di tale soglia.

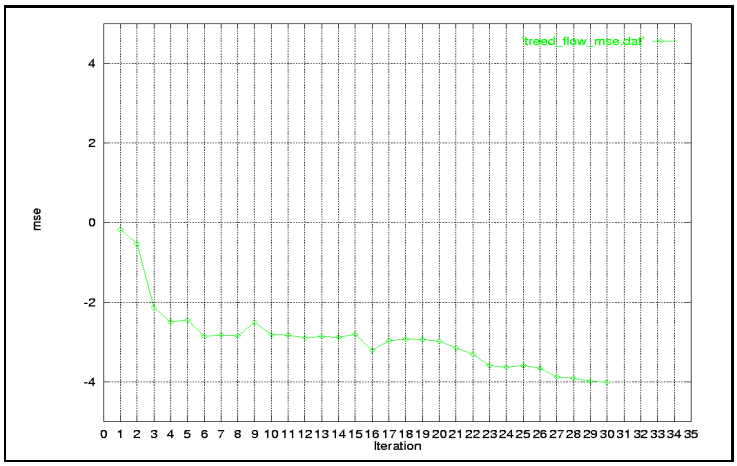

Figura 6: **TREED** - Andamento dell'mse in scala logaritmica - Modulo ***flow***.

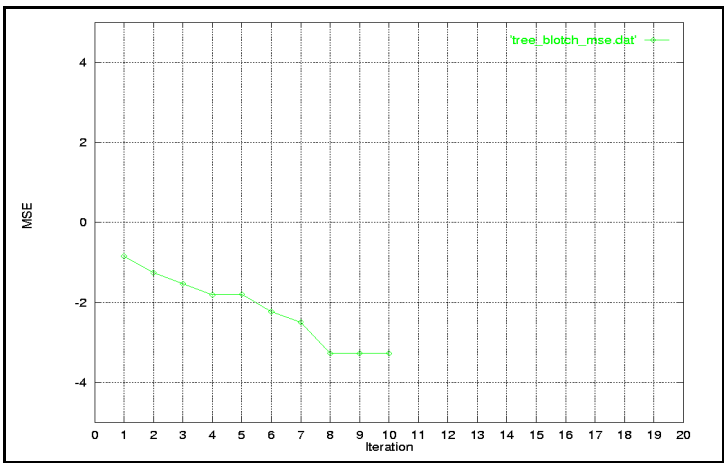

Figura 7: **TREED** - Andamento dell'mse in scala logaritmica - Modulo ***remove***.

Infine, si riportano i grafici corrispondenti ai valori di speedup ed efficienza relativi ad ognuno dei moduli, per l'analisi delle prestazioni del software in parallelo.

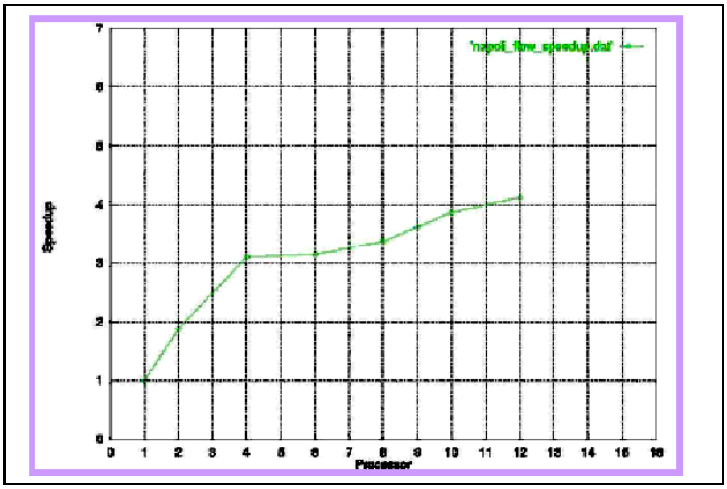

Figura 8: **TREED** - Speedup relativo al modulo ***flow*** in parallelo.

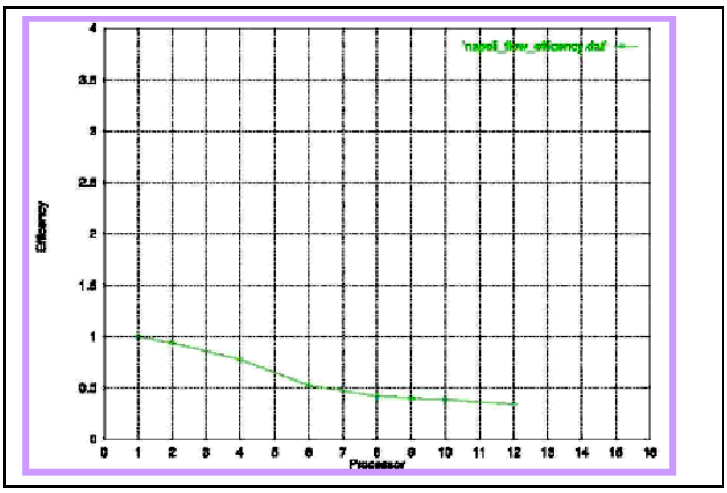

Figura 9: **TREED** - Efficienza relativa al modulo ***flow*** in parallelo.

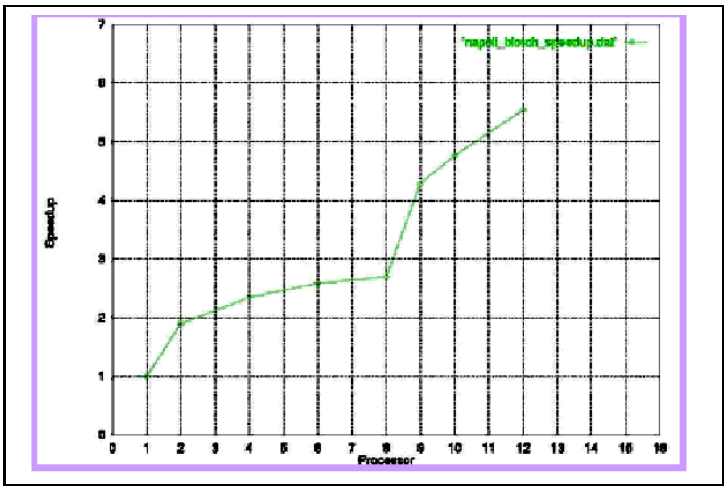

Figura 10: **TREED** - Speedup relativo al modulo ***detect*** in parallelo.

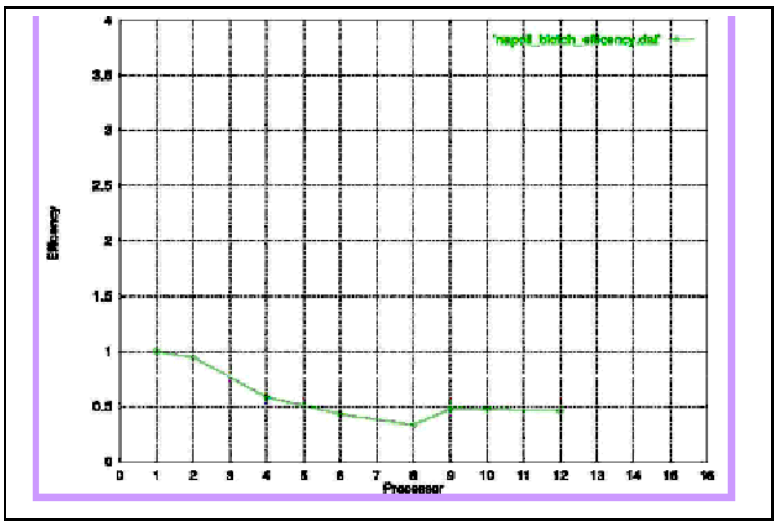

Figura 11: **TREED** - Efficienza relativa al modulo ***detect*** in parallelo.

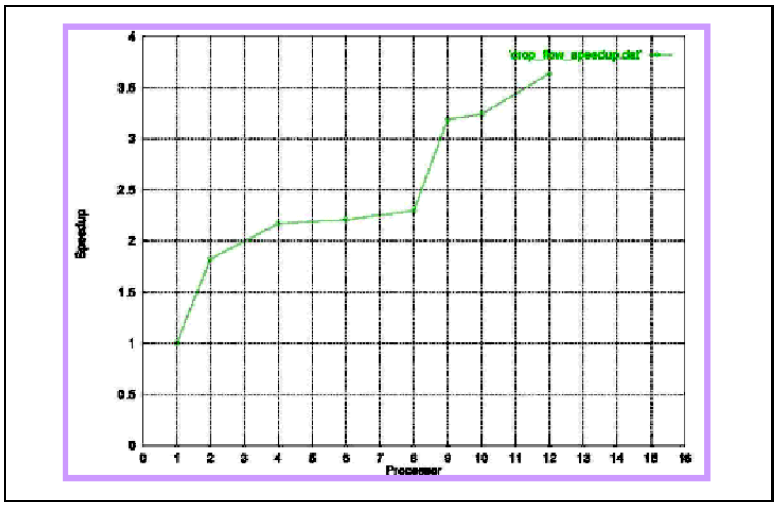

Figura 12: **TREED** - Speedup relativo al modulo ***remove*** in parallelo.

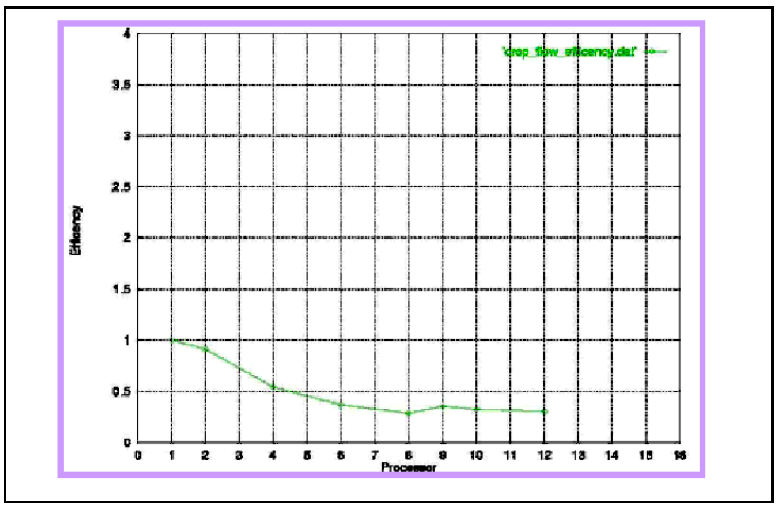

Figura 13: **TREED** - Efficienza relativa al modulo ***remove*** in parallelo.

6.2 Sequenze di immagini reali.

Il software è stato testato anche su sequenze di immagini reali, in particolare su brevi documentari piuttosto rovinati. Come si osserva dai risultati ottenuti, quasi tutti i difetti, che verificano le proprietà spazio-temporali dei blotch (capitolo **2**), risultano individuati ed eliminati.

La sequenza, riportata in figura **14,** permette di effettuare interessanti osservazioni dai risultati ottenuti.
Nonostante le immagini siano notevolmente corrotte, il flusso ottico è puramente traslatorio e inoltre la maggior parte dei difetti difficilmente si presenta, nella stessa posizione, in fotogrammi consecutivi.

Figura 14: ***Documentario Napoli*** - Al centro l'immagine da restaurare. A sinistra e a destra il fotogramma precedente ed il successivo nella sequenza. Dimendioni originali: 512x720.

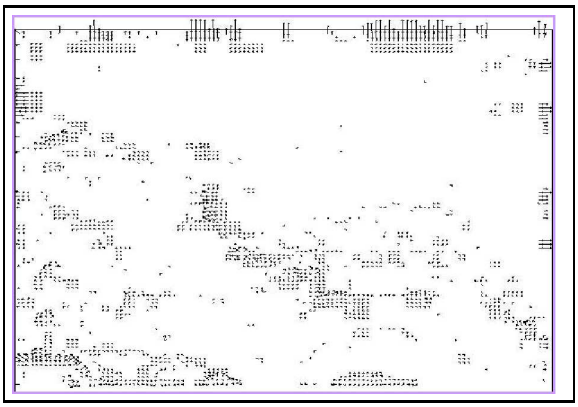

Figura 15: ***Documentario Napoli*** - Il flusso ottico valutato mediante il modulo ***flow***. Tempo di cpu (versione sequenziale)- **IBM SP RS6000: 87.237 secs.**

In particolare, si osserva che anche il graffio verticale, che generalmente appartiene alla categoria dei *difetti locali fissi*, cambia posizione rispetto alla scena per cui è possibile ricostruire quasi interamente la parte mancante.

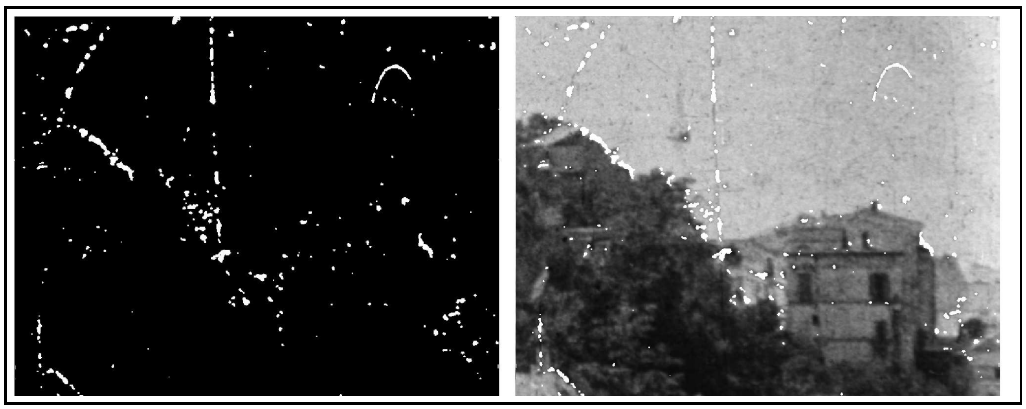

Figura 16: ***Documentario Napoli*** - A sinistra la maschera dei difetti valutata dal modulo ***detect***. A destra la stessa maschera sovrapposta all'immagine centrale in esame. Tempo di cpu (versione sequenziale)- **IBM SP RS6000: 16.69 secs.**

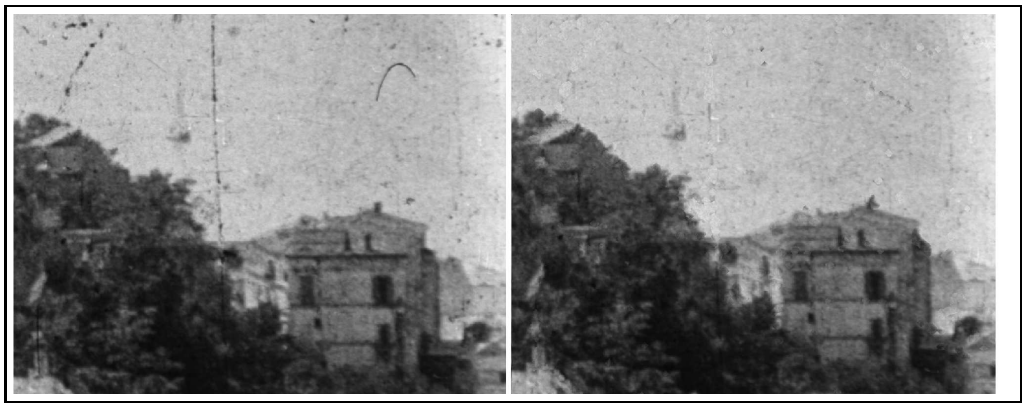

Figura 17: ***Documentario Napoli*** - A sinistra l'immagine originale. A destra l'immagine restaurata mediante il modulo ***remove***. Tempo di cpu (versione sequenziale)- **IBM SP RS6000: 26.328 secs.**

Le prestazioni del sistema di restauro, in termini di accuratezza, robustezza ed efficienza si osservano dai grafici che seguono.

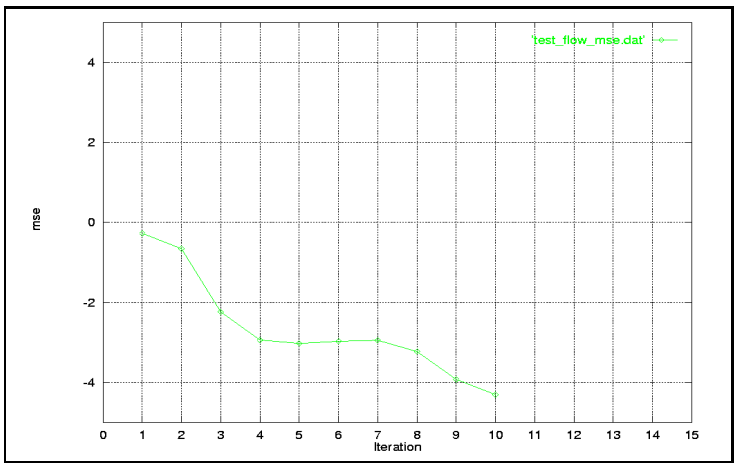

Figura 18: ***Documentario Napoli*** - Andamento dell'mse in scala logaritmica - Modulo ***flow***.

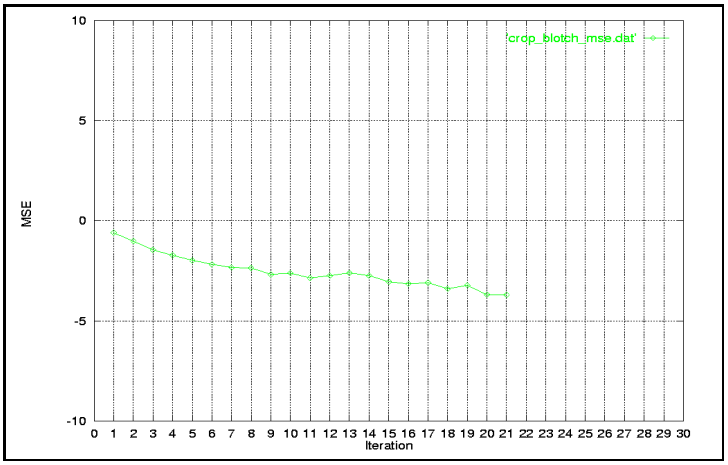

Figura 19: ***Documentario Napoli*** - Andamento dell'mse in scala logaritmica - Modulo ***remove***.

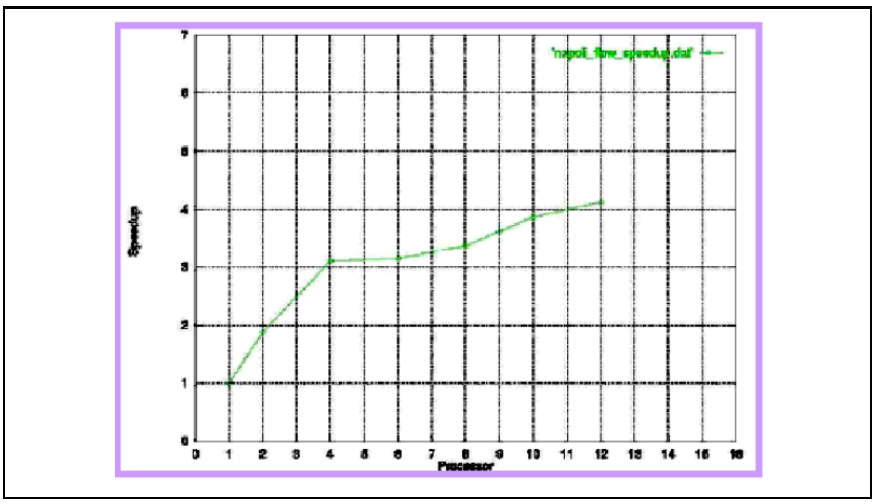

Figura 20: ***Documentario Napoli*** - Speedup relativo al modulo ***flow*** in parallelo.

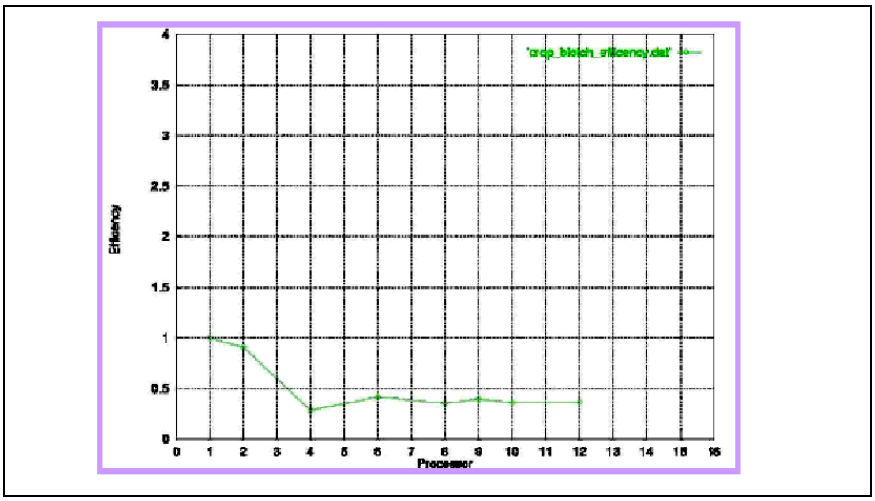

Figura 21: ***Documentario Napoli*** - Efficienza relativa al modulo ***flow*** in parallelo.

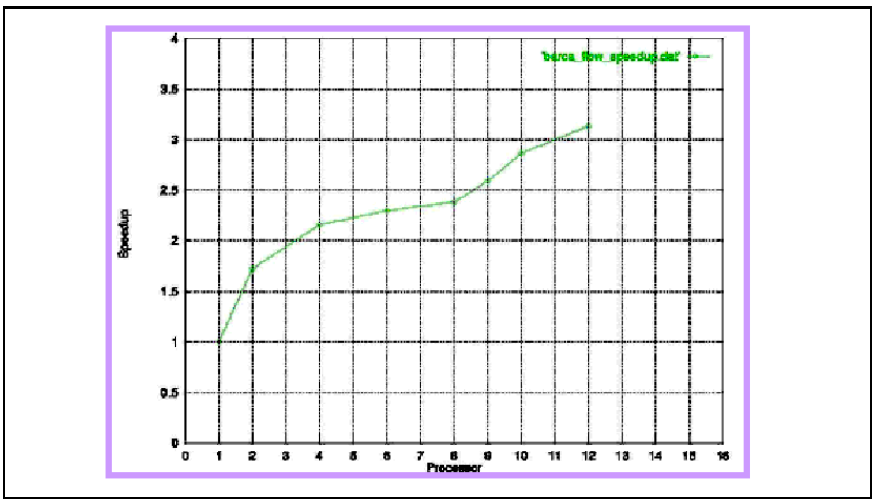

Figura 22: ***Documentario Napoli*** - Speedup relativo al modulo ***detect*** in paralleo.

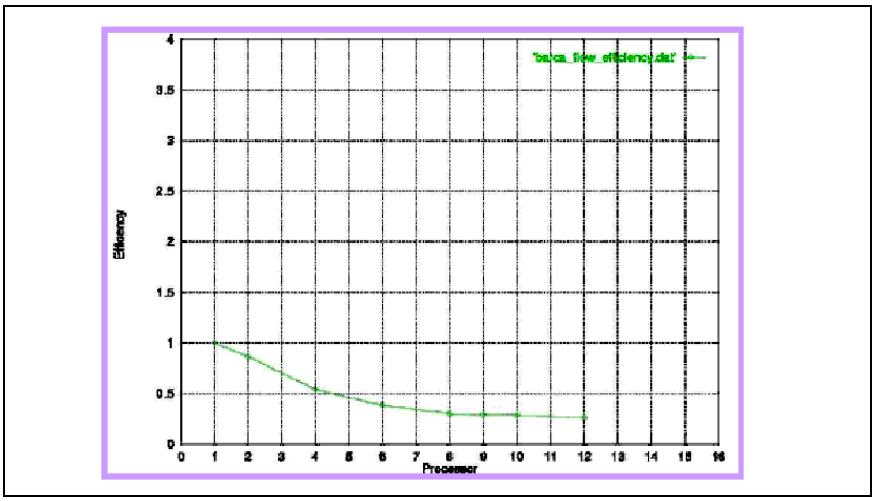

Figura 23: ***Documentario Napoli*** - Efficienza relativa al ***detect*** in parallelo.

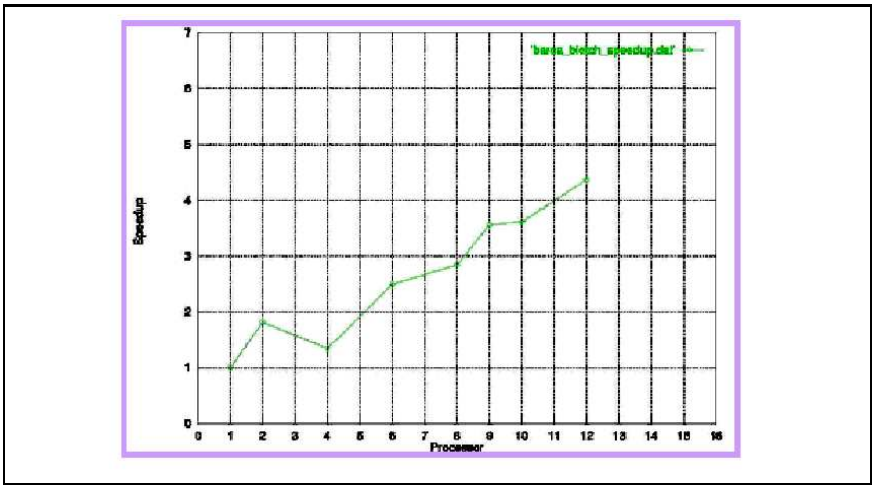

Figura 24: ***Documentario Napoli*** - Speedup relativo al modulo ***remove*** in paralleo.

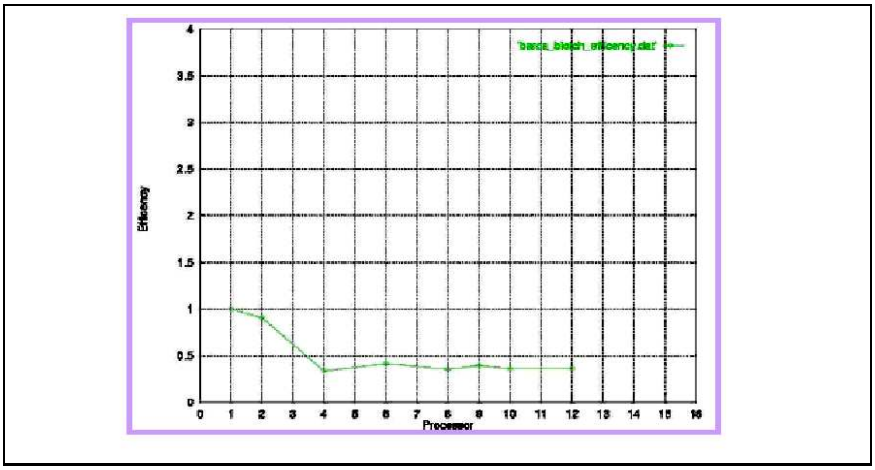

Figura 25: ***Documentario Napoli*** - Efficienza relativa al modulo ***remove*** in parallelo.

6.3 Osservazioni e sviluppi futuri.

I grafici corrispondenti alle prestazioni del software parallelo implementato, consentono di effettuare interessanti osservazioni riguardo la strategia impiegata per la suddivisione iniziale dei dati e per lo scambio delle informazioni necessarie durante la fase di calcolo.
Gli esempi riportati mostrano i risultati ottenuti su una singola immagine della sequenza mediante la strategia descritta in **5.6**, che costituisce un primo approccio al parallelismo e, dunque, non tiene conto di un efficiente bilanciamento fra il tempo di elaborazione e il tempo di comunicazione (*granularità dei processi*).
Infatti, quando le immagini della sequenza hanno dimensione relativamente bassa, l'impiego di un numero elevato di processori risulta superfluo, oltre a condizionare le prestazioni del software in termini di efficienza e scalabilità. Per cui, soddisfacenti risultati si osservano fino all'utilizzo di al più quattro/sei processori.

Tali osservazioni costituiscono il punto di partenza al miglioramento delle prestazione del software parallelo. Al fine di mantenere bilanciato il carico di elaborazione dei dati, l'intenzione è quella di analizzare e implementare differenti strategie per l'allocazione delle attività ai vari processi. Ad esempio, piuttosto che suddividere i dati appartenenti ad ogni singolo fotogramma, il modello teorico descritto per la risoluzione del problema di restauro, suggerisce di suddividere la sequenza di immagini in gruppi costituiti da almeno tre fotogrammi ad ogni processo ed eventualmente scambiare i dati al termine dell'elaborazione.

Conclusioni.

In questo lavoro di tesi è stato presentato un algoritmo numerico e il relativo software per il riconoscimento e la rimozione di *difetti locali aleatori* (blotch) in sequenze di immagini digitali
La relazione fra il campo di moto e l'intensità luminosa di una sequenza di immagini suggerisce l'approccio al problema di restauro utilizzato. Precisamente, utilizzando l'ipotesi secondo cui l'intensità luminosa della sequenza resta costante lungo la traiettoria del moto, è possibile impiegare le informazioni relative al flusso ottico della sequenza, sia per il riconoscimento che per l'eliminazione dei blotch (interpolazione spazio-temporale).
L'algoritmo proposto si articola in tre fasi: una fase preliminare (*pre-processing*), in cui si effettua la stima del flusso ottico, una fase di individuazione delle anomalie presenti nella sequenza di immagini ed una fase finale di restauro.
Precisamente, l' algoritmo per il calcolo del flusso ottico si serve dell'approssimazione in multirisoluzione spaziale per valutare, mediante un metodo iterativo, il campo di moto ad ogni livello di risoluzione. Le informazioni relative al flusso ottico forniscono le premesse ai successivi algoritmi di individuazione e rimozione dei difetti, mediante approssimazione basata su informazioni spazio-temporali.

Lo schema numerico corrispondente si basa sulla risoluzione di equazioni differenziali alle derivate parziali e sulla tecnica della compensazione del moto, mediante schemi alle differenze finite.

Il software, implementato in ambiente di calcolo parallelo, costituisce un sistema di restauro digitale automatico composto di tre moduli (***flow-detect-remove***) e consente l'utilizzo indipendente di ognuno di essi.

La strategia di parallelizzazione impiegata determina, in base alle dimensioni dei fotogrammi della sequenza di immagini in esame, il numero più efficiente di processi necessari allo svolgimento di ognuno dei compiti e, quindi, suddivide ad ognuno di essi il lavoro sfruttando le proprietà di una griglia topologica virtuale per lo scambio dei dati durante la fase di calcolo.

Appendice A

Regolarizzazione edge-preserving nell'elaborazione di immagini.

La maggior parte dei problemi che nascono nell'ambito dell'analisi di immagini digitali, è un problema *mal posto* nel senso di *Hadamard*[1].
In generale, esso consiste nel determinare i valori di una funzione f, a partire da un modello matematico del tipo:

$$R(f) = g \tag{1}$$

in cui la funzione g rappresenta il dato iniziale, mentre R è un operatore lineare che descrive il processo di aquisizione dell'immagine[2].

Numericamente, il problema (**1**) equivale al calcolo di un'approssimazione della funzione f, indicata con $\widetilde{f}$, tale che:

$$\left\| R\left(\widetilde{f}\right) - g \right\|^2 < \varepsilon, \ \forall \varepsilon > 0$$

Tra i metodi di risoluzione impiegati, il più noto si basa sulla risoluzione del problema:

$$\widetilde{f} = arg \min_f J(f)$$

[1] Un problema ben posto nel senso di Hadamard se la soluzione esiste, è unica e dipende con continuità dai dati iniziali.

[2] L'equanzione (**1**), nel caso della stima del flusso ottico, corrisponde all'equazione costante del moto:

$$\vec{v} \times \nabla I = -I_t;$$

mentre nel caso dell'eliminazione dei difetti, coincide con il modello proposto:

$$I^B = \frac{I^C_{t-1} + I^C_{t+1}}{2}$$

in cui il funzionale $J(f)$ è costituito dalla somma di due termini:

$$J(f) = J_1(f) + \lambda J_2(f) = \|R(f) - g\|^2 + \lambda \|\phi(\nabla f)\|$$

Il primo termine controlla l'affidabilità della soluzione, mentre il secondo è detto ***termine*** di ***regolarizzazione*** (*edge-preserving*) e coinvolge i valori del gradiente della funzione f, opportunamente pesati mediante la *funzione* ϕ ed il ***parametro*** di ***regolarizzazione*** λ (*energy-based models PDE*) **[13, 26, 27]**.

In particolare, $\phi(\|\nabla f\|)$ induce nelle equazioni di *Eulero-Lagrange*, associate al problema di minimizzazione, l'operatore divergenza:

$$div\left(\frac{\phi'(\|\nabla f\|)}{\|\nabla f\|} \cdot \nabla f\right) = \frac{\phi'(\|\nabla f\|)}{\|\nabla f\|} f_{\xi\xi} + \phi''(\|\nabla f\|)_{\eta\eta}$$

dove:

ϕ' e ϕ'' sono rispettivamente le derivate prime e seconde di ϕ rispetto ad s, mentre $f_{\eta\eta}$ è la derivata direzionale di f, rispetto alla direzione del gradiente:

$$\eta = \frac{\nabla f}{\|\nabla f\|}$$

e, infine, $f_{\xi\xi}$ la derivata direzionale di f rispetto alla direzione ξ ortogonale al gradiente.

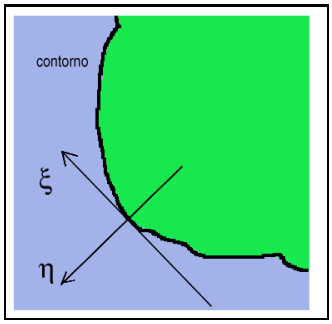

Figura 1: Direzione della tangente e della normale rispetto a una curva di contorno.

Ciò implica che la funzione ϕ deve essere regolare nei punti in cui $\|\nabla f\|$ assume valore relativamente piccolo (nella direzione di η, fig. **1**), ovvero:

$$\phi'(0) = 0, \ \lim_{s\to 0^+} \frac{\phi'(s)}{s} = \lim_{s\to 0^+} \phi''(s) = \phi''(0) > 0$$

da cui, posto: $\triangle f = f_{\xi\xi} + f_{\eta\eta}$, si ha:

$$\frac{\phi'(\|\nabla f\|)}{\|\nabla f\|} \cdot f_{\xi\xi} + \phi''(\|\nabla f\|) \cdot f_{\eta\eta} = \phi''(0) \cdot [f_{\xi\xi} + f_{\eta\eta}] = \phi''(0) \cdot \triangle f$$

Invece, nei punti in cui $\|\nabla f\|$ assume valore grande, ad esempio in prossimità dei contorni (nella direzione di ξ, fig. **1**) il coefficiente del termine deve essere non nullo, cioè:

$$\lim_{s\to+\infty} \phi''(s) = 0, \ \lim_{s\to+\infty} \frac{\phi'(s)}{s} = \beta > 0$$

Poichè queste ultime due condizioni sono incompatibili fra loro, si giunge a un compromesso osservando che entrambe le quantità $\phi''(s)$ e $\phi'(s)/s$ convergono a zero al crescere di s, ma $\phi''(s)$ con ordine superiore, per cui:

$$\lim_{s\to+\infty} \phi''(s) = \lim_{s\to+\infty} \frac{\phi'(s)}{s} = 0, \qquad \lim_{s\to+\infty} \frac{\phi''(s)}{\phi'(s)} s = 0$$

Riassumendo, deve essere:

- $\lim_{s\to 0} \phi''(s) = \lim_{s\to 0} \frac{\phi'(s)}{s} = \gamma_1 > 0, \ \phi'(0) = 0$
- $\lim_{s\to\infty} \phi''(s) = \lim_{s\to\infty} \frac{\phi'(s)}{s} = 0$
- $\lim_{s\to\infty} \frac{\phi''(s)}{\phi'(s)} \cdot s = 0$

ovvero la funzione ϕ deve essere definita come segue:

1. $\phi : R \longrightarrow R^+$ è pari, convessa e crescente in R^+,
2. $\exists c > 0, \ b \geq 0 : \ cs - b \leq \phi(s) \leq cs + b, \ \forall s \in R^+$,
3. $\phi \in C^\infty(R) : \ \phi^\infty(1) = 1$

In tali ipotesi è possibile dimostrare l'esistenza e l'unicità della soluzione del problema **[4]**.

Teorema 1: *Sia* $f \in W^{1,\infty}(\Omega)$, *considerata la funzione* ϕ *tale che, verifichi le condizioni 1, 2, 3, allora esiste un'unica soluzione al problema di regolarizzazione:*

$$\min_{f \in BV(\Omega)} J(f)$$

nello spazio $BV(\Omega) = \{f \in L^{\infty}(\Omega) : |Df|(\Omega) < +\infty\}$ *delle funzioni a variazione limitata, dove:*

$$|Df|(\Omega) = \sup \left\{ \sum_{i=1}^{n} \|\nabla f\| : \bigcup_{i=1}^{n} A_i \subseteq \Omega, \ aperti\, i \neq j \ A_i \cap A_j = \emptyset \right\}$$

Per trattare il problema è, dunque, necessario prolungare il funzionale $J(f)$ definito in $W^{1,2}(\Omega)$ su tutto lo spazio $BV(\Omega)$ al fine di ottenere una soluzione generalizzata del problema di partenza.
Tale operazione, nota come ***tecnica del rilassamento*** **[9]**, può essere realizzata associando alla funzione ϕ un prolungamento ϕ_ε e utilizzando la nozione di $\Gamma - convergenza$.

Proposizione 1: *Considerata la funzione* ϕ_ε, *tale che:*

$$\phi_\varepsilon(s) = \begin{cases} \frac{\phi'(\varepsilon)}{2\varepsilon} s^2 + \phi(\varepsilon) - \frac{\varepsilon \phi'(\varepsilon)}{2} & se\ 0 \leq s \leq \varepsilon \\ \phi(s) & se\ \varepsilon \leq s \leq \frac{1}{\varepsilon} \\ \frac{\varepsilon \phi'\left(\frac{1}{\varepsilon}\right)}{2} s^2 + \phi\left(\frac{1}{\varepsilon}\right) - \frac{\phi'\left(\frac{1}{\varepsilon}\right)}{2\varepsilon} & se\ s \geq \varepsilon \end{cases}$$

allora (fig. **2**)*:*

$$\forall \varepsilon > 0, \ \phi_\varepsilon \geq \phi \ \ \forall s, \ \lim_{\varepsilon \to 0} \phi_\varepsilon(s) = \phi(s)$$

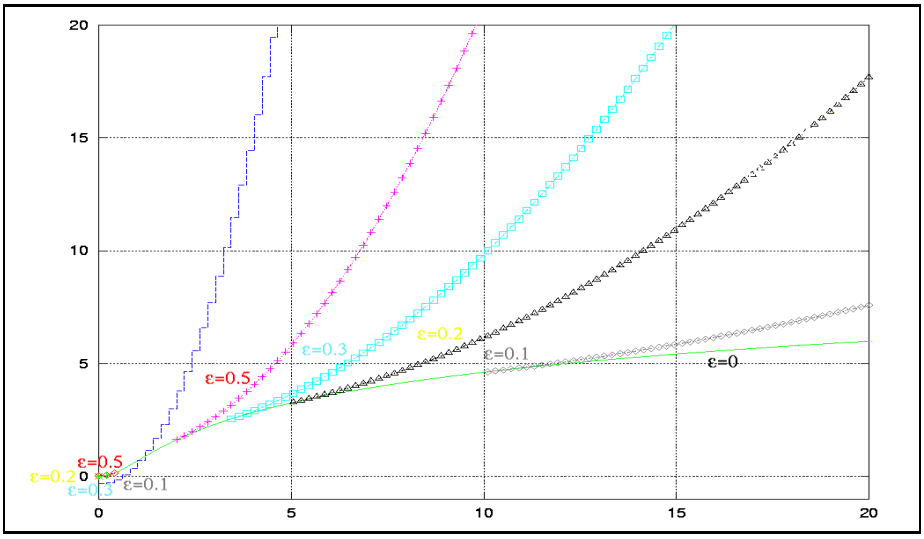

Figura 2: Se $\phi(x) = log(1+x^2)$ si può osservare l'andamento della successione $\{\phi_\varepsilon(x)\}_\varepsilon$, per $\varepsilon = 0$, $\varepsilon = 0.1$, $\varepsilon = 0.3$, $\varepsilon = 0.5$, $\varepsilon = 1$

Definizione 1: *Una successione di funzioni* $\{f_\varepsilon\}$ $\Gamma-$*converge ad una funzione* f *se, e solo se, sono verificate le due seguenti ipotesi:*

- $\forall\{x_\varepsilon\}:\ \lim_{\varepsilon\to 0} x_\varepsilon = x \quad \liminf_{\varepsilon\to 0} f_\varepsilon(x_\varepsilon) \geq f(x)$
- $\exists\{x_\varepsilon\}:\ \lim_{\varepsilon\to 0} x_\varepsilon = x \quad \limsup_{\varepsilon\to 0} f_\varepsilon(x_\varepsilon) \leq f(x)$

Con tali premesse si ottiene un nuovo funzionale:

$$J_\varepsilon(f) = \begin{cases} \|R(f) - g\|^2 + \lambda \|\phi_\varepsilon(\nabla f)\| & se\ f \in W^{1,2}(\Omega) \\ +\infty & altrimenti \end{cases}$$

che $\Gamma-$converge, per $\varepsilon \to 0$ alla soluzione.

Proposizione 2 (*Demengel-Teman* **[17]**)*: Sia* $f \in W^{1,\infty}(\Omega)$ *e* ϕ_ε *una funzione che verifica le condizioni* ***1-2-3****, allora la successione* $\{J_\varepsilon\}_\varepsilon$ $\Gamma-$*converge al funzionale* J*, per* $\varepsilon \to 0$.

Teorema di convergenza* [1, 16]**: *Sia* $f \in W^{1,2}(\Omega)$ *e* ϕ_ε *una funzione che verifica le condizioni* ***1-2-3*, il problema di minimo:*

$$\min_{f_\varepsilon \in BV(\Omega)} J_\varepsilon (f_\varepsilon)$$

ammette un'unica soluzione f_ε.
Inoltre la successione $\{f_\varepsilon\}_\varepsilon$ $\Gamma - converge$ *per* $\varepsilon \to 0$, *all'unica soluzione* f *del problema di minimo:*

$$\min_{f \in W^{1,2}(\Omega)} J(f)$$

Appendice B

Regolarizzazione semi-quadratica.

La soluzione del problema di regolarizzazione, definito in appendice **A**:

$$\min_{f_\varepsilon \in W^{1,2}(\Omega)} J_\varepsilon\left(f_\varepsilon\right) = \min_{f_\varepsilon} \left\{ \left\| R\left(f_\varepsilon\right) - g \right\|^2 + \lambda \left\| \phi_\varepsilon\left(\left| \nabla f_\varepsilon \right| \right) \right\| \right\} \qquad (1)$$

deve verificare l'equazione di Eulero-Lagrange ad esso associata; ovvero:

$$\frac{\partial J_\varepsilon\left(f_\varepsilon\right)}{\partial f_\varepsilon} = 0.$$

Tale equazione, nella sua forma più generale, può essere scritta nel seguente modo:

$$2R^\star\left(R\left(f_\varepsilon\right) - g\right) + div\left(\frac{\phi_\varepsilon'\left(\left|\nabla f_\varepsilon\right|\right)}{\left|\nabla f_\varepsilon\right|} \nabla f_\varepsilon \right) = 0$$

con condizione a contorno di *Neumman*:

$$\frac{\phi_\varepsilon'\left(\left|\nabla f_\varepsilon\right|\right)}{\left|\nabla f_\varepsilon\right|} \frac{\partial f_\varepsilon}{\partial N} = 0$$

Il problema, dunque, è rappresentato da un'equazione alle derivate parziali (PDE) di tipo ellittico, non lineare.

L'approccio proposto in **[19]**, permette di trasformare il funzionale, non quadratico, in **(1)**, in un funzionale semi-quadratico (ovvero quadratico rispetto a ciascuna variabile separatamente), introducendo una *variabile* ***ausiliare*** b e una *funzione* ***duale*** ψ.

Teorema: *Se* $\phi : [0, +\infty[\to [0, +\infty[$ *è tale che* $\phi(\sqrt{x})$ *è strettamente concava in* $]0, +\infty[$, *posto:*

$$L = \lim_{x \to +\infty} \frac{\phi'(x)}{2x} \; e \; M = \lim_{x \to 0^+} \frac{\phi'(x)}{2x}$$

valgono le seguenti proprietà:

- *Esiste una funzione:* $\psi : [L,\, M] \longrightarrow [\alpha,\, \beta]$, *dove:*

$$\alpha = \lim_{x \to +\infty} \left(\phi(x) - x^2 \frac{\phi'(x)}{2x} \right) \; e \; \beta = \lim_{x \to 0^+} \phi(x)$$

  per cui, si ha: $\phi(x) = \min_{L \leq b \leq M} (bx^2 + \psi(b))$

- $\forall x \geq 0$ *il minimo di* $\phi(x)$ *esiste, è unico ed è dato da:*

$$b_x = \frac{\phi'(x)}{2x}$$

Scelta, dunque, ϕ in modo tale che $\phi(\sqrt{s})$ sia strettamente concava, la funzione duale ψ risulta convessa e può essere espressa mediante ϕ (alcuni esempi in tab. **1** e in fig. **1-2**).
Allora, la decomposizione del teorema:

$$\phi(s) = \min_b bs^2 + \psi(b)$$

si può sostituire in (**1**) per ottenere il ***funzionale duale*** associato a J_ε:

$$J_\varepsilon^d(f_\varepsilon,\, b) = \|R(f_\varepsilon) - g\|^2 + \lambda \left\| \min_{L_\varepsilon \leq b \leq M_\varepsilon} \left\{ b\, |\nabla f_\varepsilon|^2 + \psi_\varepsilon(b) \right\} \right\|$$

ed il criterio di minimizzazione semi-quadratico:

$$\min_{f_\varepsilon} J_\varepsilon^d(f_\varepsilon,\, b) = \min_{f_\varepsilon} \left\{ \|R(f_\varepsilon) - g\|^2 + \lambda \left\| \min_{L_\varepsilon \leq b \leq M_\varepsilon} \left\{ b\, |\nabla f_\varepsilon|^2 + \psi_\varepsilon(b) \right\} \right\| \right\} =$$

$$= \min_{f_\varepsilon,\, L_\varepsilon \leq b \leq M_\varepsilon} \left\{ \|R(f_\varepsilon) - g\|^2 + \lambda \left\| b\, |\nabla f_\varepsilon|^2 + \psi_\varepsilon(b) \right\| \right\}$$

Autori	$\phi(s)$	$\phi(\sqrt{s})$	$\frac{\phi'(s)}{2s}$	$\psi(b)$
Geman-Reynolds	$\frac{s^2}{1+s^2}$ non convessa	$\frac{s}{1+s}$ concava	$\frac{1}{(1+s^2)^2}$	$b-2\sqrt{b}+1$
Rudin-Osher	s non convessa	$\sqrt{s}$ concava	$\frac{1}{2s}$	$\frac{1}{4b}$
Perona-Malik	$\lg(1+s^2)$ non convessa	$\lg(1+s)$ concava	$\frac{1}{(1+s^2)}$	$b-\lg(b)-1$
Aubert	$2\sqrt{1+s^2}-2$ convessa	$2\sqrt{1+s}-2$ concava	$\frac{1}{\sqrt{1+s^2}}$	$b+\frac{1}{b}$

Tabella 1: Sostituendo i valori delle funzioni $\phi(s)$ e $b=\frac{\phi'(s)}{2s}$ per cui è verificato il minimo, si ricavano i valori della funzione convessa $\psi(b)$.

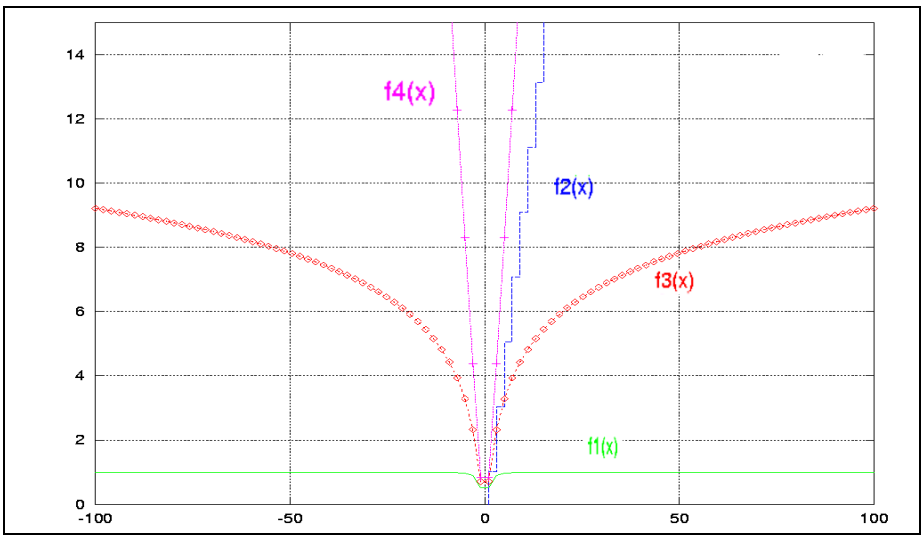

Figura 1: I grafici delle funzioni $\phi(s)$ riportate in tabella 1.
$f_1(x)=\frac{x^2}{1+x^2}$, $f_2(x)=x$, $f_3(x)=\log\left(1+x^2\right)$, $f_4(x)=2\sqrt{1+x^2}-2$

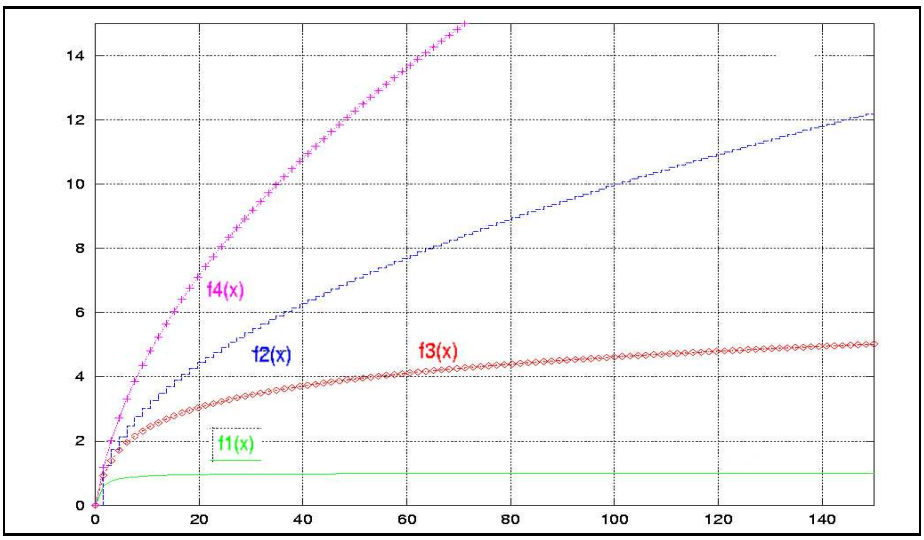

Figura 2: I grafici delle rispettive funzioni $\phi(\sqrt{s})$:
$f_1(\sqrt{x}) = \frac{x}{1+x}$, $f_2(\sqrt{x}) = \sqrt{x}$, $f_3(\sqrt{x}) = \log(1+x)$, $f_4(\sqrt{x}) = 2\sqrt{1+x} - 2$

Appendice C

Metodo del punto fisso alternato.

Come si è visto nelle precedenti sezioni il problema di regolarizzazione originale (appendice **A**) è equivalente al problema di regolarizzazione "duale" (appendice **B**), ovvero:

$$\min_{f_\varepsilon} J_\varepsilon\left(f_\varepsilon\right) = \min_{f_\varepsilon,\, L_\varepsilon \le b \le M_\varepsilon} J_\varepsilon^d\left(f_\varepsilon,\, b\right) \tag{1}$$

Poichè si ha:

$$\min_{f_\varepsilon} J_\varepsilon\left(f_\varepsilon\right) = \min_{f_\varepsilon} \min_{b} J_\varepsilon^d\left(f_\varepsilon,\, b\right) = \min_{b} \min_{f_\varepsilon} J_\varepsilon^d\left(f_\varepsilon,\, b\right)$$

posto:

$$T\left(b\right) = \min_{f_\varepsilon} J_\varepsilon^d\left(f_\varepsilon,\, b\right)$$

allora, si può scrivere:

$$\min_{f_\varepsilon} J_\varepsilon\left(f_\varepsilon\right) = \min_{b} T\left(b\right)$$

e dimostrare che **[12]**, se b è fissato il funzionale $J_\varepsilon^d\left(f_\varepsilon,\, b\right)$ risulta quadratico rispetto alla variabile f_ε ed ammette un'unica soluzione, che può essere indicata con $f_{\varepsilon,\, b}$. In particolare:

$$\widehat{b} = arg\min_{b} T\left(b\right) \implies f_{\varepsilon,\widehat{b}} = arg\min_{f_\varepsilon} J_\varepsilon\left(f_\varepsilon\right)$$

Tali risultati possono essere impiegati per valutare un'approssimazione dei valori $\widehat{b}$ e $f_{\varepsilon,\widehat{b}}$ utilizzando una strategia di minimizzazione alternata rispetto ad ogni variabile (***schema del punto fisso alternato***).

In questo modo, il problema di regolarizzazione iniziale, rappresentato da un'equazione di *Eulero-Lagrange* non lineare, può essere risolto mediante una successione di problemi lineari **[11]**.

Teorema: *Indicato con* $(J_\varepsilon)_n = J_\varepsilon^d\left(f_\varepsilon^n,\, b^{n+1}\right)$ *il funzionale duale corrispondente all'iterazione* n**,** *se la funzione* ϕ_ε *verifica le ipotesi* ***1-2-3*** *del teorema di esistenza e unicità* (app. **1**) *allora, la successione* $(J_\varepsilon)_n$ *converge totalmente all'unica soluzione* $J_\varepsilon^d\left(f_{\widehat{b},\varepsilon}^n,\, \widehat{b}^{n+1}\right)$.
Inoltre, localmente si ha:

$$\left(b^{n+1} - b^n\right) \to_{n\to\infty} 0 \;\; e \;\; \left(f_\varepsilon^{n+1} - f_\varepsilon^n\right) \to_{n\to\infty} 0$$

purchè la variabile ausiliaria b^{n+1} *sia tale che:*

$$0 \le c \le b^{n+1} \le 1,\; con\; c = cost$$

Lo schema generale di risoluzione può essere riassunto come segue:

$f_0^\varepsilon = 0$
repeat

$b^{n+1} = arg\min_b \left\{J_\varepsilon^d\left(f_\varepsilon^n,\, b\right)\right\}$ (**2**)
$f_\varepsilon^{n+1} = arg\min_{f_\varepsilon} \left\{J_\varepsilon^d\left(f_\varepsilon,\, b^{n+1}\right)\right\}$ (**3**)

until (**convergenza**)

In definitiva, la successione $(J_\varepsilon)_n$ converge alla soluzione del problema duale $J_\varepsilon^d\left(f_{\widehat{b},\varepsilon}^n,\, \widehat{b}^{n+1}\right)$**,** quando sono verificate le seguenti ipotesi:

1. $\phi_\varepsilon : R \longrightarrow R^+$ è pari, convessa e crescente in R^+,
2. $\exists c > 0,\; b \ge 0 :\; cs - b \le \phi_\varepsilon(s) \le cs + b,\; \forall s \in R^+$,
3. $\phi_\varepsilon \in C^\infty(R) :\; \phi_\varepsilon^\infty(1) = 1$

In particolare, se la funzione ϕ_ε risulta stettamente convessa, lo schema del punto fisso alternato conduce al calcolo dell'unica soluzione $\left(f_{\widehat{b},\varepsilon},\, \widehat{b}\right)$ al problema di minimizzazione (**1**).

In tali ipotesi, è possibile impiegare i risultati relativi allo schema del punto fisso e al teorema delle contrazioni che ne garantisce la convergenza[1], per mostrare che la convergenza locale dell'equazione di Eulero-Lagrange, associata al problema (**2**):

$$b^{n+1} = \frac{\phi_\varepsilon' \left(\nabla f_\varepsilon^n\right)}{2\left|\nabla f_\varepsilon^n\right|}$$

risulta ***lineare***, purchè la variabile ausiliaria b^{n+1} sia tale che: $0 \leq c \leq b^{n+1} \leq 1$, *con* $c = cost$.
Ciò implica la convergenza ***lineare*** dell'equazione di Eulero-Lagrange associata al problema (**3**):

$$2R^\star \left(R\left(f_\varepsilon^{n+1}\right) - g\right) + div\left(b^{n+1}\nabla f_\varepsilon^n\right) = 0$$

purchè l'operatore R sia simmetrico e definito positivo.

[1] ***Teorema delle contrazioni:*** Se il funzionale $F:\ C \subseteq \Re \longrightarrow C \subseteq \Re$ è una contrazione, ovvero:

$$\exists \lambda = cost:\ 0 < \lambda < 1:\ \forall x,\, y \in C \subseteq \Re\ \ \left|F\left(x\right) - F\left(y\right)\right| \leq \lambda \left|x - y\right|$$

allora ha un unico punto fisso $s \in C \subseteq \Re:\ F\left(s\right) = s$.

Appendice D

Multirisoluzione.

La ***multirisoluzione*** è una tecnica di approssimazione basata su un modello di tipo "*gerarchico*", ovvero su una struttura costituita da una catena di sottospazi $\{M_k\}_{k\in Z}$, $M_k \subset L^2(\Re)$, che verificano le seguenti proprietà:

1. $\forall k \in Z \quad M_k \supset M_{k+1}, \quad dim(M_k) \geq dim(M_{k+1})$
2. $\forall k \in Z \quad f(t) \in M_k \Longleftrightarrow f\left(t - 2^k n\right) \in M_k, \;\; n \in Z$
3. $\forall k \in Z \quad f(t) \in M_k \Longleftrightarrow f\left(\frac{t}{2}\right) \in M_{k+1}$
4. $\cap_{k\in Z} M_k = \{\emptyset\}, \quad \overline{\cap_{k\in Z}} M_k = L^2(\Re)$

Per ogni struttura $\{M_k\}_{k\in Z}$, esiste una ***funzione generatrice*** $\phi(t) \in L^2(\Re)$, mediante la quale è possibile definire una famiglia di funzioni:

$$\left\{\phi_{n,k}(t) = \frac{1}{\sqrt{2^k}}\phi\left(\frac{t-n}{2^k}\right)\right\}_{n,k\in Z}$$

tale che $\forall k \in Z, \;\; \phi_{n,k}(t)$ è una ***base ortonormale*** del singolo sottospazio M_k.

Da un punto di vista numerico, nell'approssimazione in multirisoluzione a ciascun sottospazio corrisponde un'opportuna discretizzazione dello spazio continuo. In particolare, nell'analisi di immagini, ogni sottospazio corrisponde ad una risoluzione spaziale del dominio dell'immagine Ω e fornisce le informazioni desiderate su una griglia più o meno fitta (***multiscala***, fig. **1**).

Figura 1: Immagine in multiscala

In questo lavoro si utilizza una struttura in multirisoluzione ***gaussiana***, ottenuta mediante l'impiego di ***basi wavelets ortonormali*** **[18]**, la cui *funzione generatrice* è una funzione *gaussiana standard*:

$$\omega\left(x,\, y\right) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\}$$

Quindi:

$$\forall k \in Z,\; I_k\left(x,\, y\right) = I_{k-1}\left(x,\, y\right) \star \omega\left(x,\, y\right) \tag{1}$$

dove: il simbolo $\star$ indica il prodotto di convoluzione.
Le caratteristiche della base ω rendono l'operazione di convoluzione, definita in (**1**), poco dispendiosa computazionalmente. Infatti poichè ω risulta *separabile* nelle sue componenti x, y e a *simmetria centrale,* la convoluzione bidimensionale può essere calcolata velocemente convolvendo la funzione $I_k\left(x,\, y\right)$ prima con il vettore $\widehat{\omega}\left(x\right)$ e successivamente con $\widehat{\omega}\left(y\right)$, che rappresentano rispettivamente la base gaussiana monodimensionale nella direzione x e nella direzione y.
Ciò significa che se, in genere, per ottenere la convoluzione di un pixel dell'immagine per una generica matrice di dimensione $n \times n$ sono necessarie $n \times n$ operazioni, per una matrice separabile sono sufficenti $2 \times n$ operazioni.
Inoltre, la proprietà di *normalizzazione* assicura una riproduzione delle

informazioni relative all'immagine originale in modo da preservare i dettagli fondamentali.

Infine, si può osservare che la complessità di spazio per l'intera struttura piramidale è:

$$N^2 \cdot \left(1 + \frac{1}{4} + \frac{1}{16} + ...\right) = N^2 \cdot \sum_{k=0}^{\infty} \left(\frac{1}{4}\right)^k = N^2 \cdot \frac{1}{1 - \frac{1}{4}} = \frac{4}{3} \cdot N^2$$

Quindi solo $\frac{1}{3}$ in più di quanto occorra per l'immagine originale.

Appendice E
Morfologia matematica.

La ***morfologia matematica***, nell'ambito dell'elaborazione di immagini, si basa sull'esame della struttura geometrica di un'immagine al fine di rendere evidenti le sue connessioni topologiche con un elemento di confronto detto *elemento strutturante*. Tali connessioni dipendono, oltre che dalla geometria della struttura da evidenziare, anche dalla sua posizione all'interno dell'immagine da esaminare **[34]**.

Le operazioni morfologiche elementari sono la *dilatazione* e l'*erosione*, anche dette *somma e sottrazione di Minkowski*, e l'operatore *Hit or Miss*. Vengono, inoltre frequentemente indicati come operatori elementari anche le trasformazioni di *apertura* e di *chiusura* ottenute dall'opportuna combinazione delle trasformazioni di erosione e dilatazione.

Erosione:
Si definisce erosione di B da parte di S, l'insieme:

$$B \ominus S = \{P \in B : \forall Q \in S,\, P - Q \in B\} = \bigcap_{Q \in S} B_Q$$

Nell'operazione di erosione l'elemento strutturante S scorre su ogni punto dell'immagine eliminando gli elementi che non lo contengono interamente.

Dilatazione:
Si definisce dilatazione di B da parte di S, l'insieme:

$$B \oplus S = \{P \in B : \exists Q \in S,\, P - Q \in B\} \bigcup_{Q \in S} B_Q$$

Nell'operazione di dilatazione l'elemento strutturante S scorre sull'immagine di riferimento B estendendo il dominio di quest'ultima alle porzioni di immagine che hanno almeno un punto in comune con B.

Apertura e Chiusura:
Gli operatori di apertura e chiusura si ottengono combinando erosione e dilatazione al seguente modo:

$$Apertura: \ B \circ S = (B \ominus S) \oplus S$$

$$Chiusura: \ B \bullet S = (B \oplus S) \ominus S$$

Le parentesi evidenziano la successione delle operazioni, poichè erosione e dilatazione non godono della proprietà associativa.