

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Matematica

**Metodo di Edge-Detection
basato sul calcolo di aree parziali**

Tesi di Laurea in Analisi

Relatore:
Chiar.ma Prof.ssa
Giovanna Citti

Presentata da:
Andrea De Rosa

III Sessione
Anno Accademico 2017/2018

Indice

1	Rappresentazione Multi-Scala	7
1.1	Preliminari	8
1.2	Filtro di diffusione lineare	11
1.2.1	Filtro Gaussiano	11
1.3	Rappresentazione Multi-Scala	13
1.3.1	Altre proprietà importanti	15
2	Metodi di Denoising	17
2.1	Steepest Descent	18
2.2	Total Variation	19
2.2.1	Formula Variazionale	19
2.3	Metodo di Perona e Malik	20
2.3.1	Formula Variazionale	21
2.4	Mean Motion Curvature	21
2.4.1	Varietà differenziabile	21
2.4.2	Varietà Riemanniana	23
2.4.3	Metrica indotta su un grafico	23
2.4.4	Descrizione del metodo e formula variazionale	25
2.5	Metodo di Alvarez, Lions e Morel	26
3	Metodo di Perona-Malik	27
3.1	Descrizione dell'equazione	28
3.2	Proprietà della diffusione anisotropica	29
3.2.1	Il principio di propagazione dei massimi	30
3.2.2	Effetto sull'edge	31
3.2.3	Alcune scelte del coefficiente di diffusione	33
3.2.4	Scelta del parametro	33
3.2.5	Mal-posizione del problema	34
3.3	Implementazione	35
3.3.1	Miglioramento della griglia	38

4	Metodi di Deblurring	41
4.1	Esempi di Blur	42
4.2	Tecniche di Deblurring	42
4.2.1	Metodo di Richardson-Lucy	42
4.2.2	Metodo di Van Cittert	43
4.2.3	Metodo di Landweber	43
4.2.4	Metodo di Poisson	43
4.3	Metodo proposto	43
5	Subpixel Edge-Detection	45
5.1	Gruppi principali dei metodi subpixel	46
5.1.1	Metodi basati sull'adattamento di curve	46
5.1.2	Metodi basati sul calcolo dei momenti	47
5.1.3	Metodi basati sulla ricostruzione	49
5.1.4	Metodo proposto	50
6	Subpixel Edge-Detector basato sul calcolo di aree parziali	51
6.1	Acquisizione dell'immagine	51
6.2	Analisi dell'errore nel calcolo del gradiente	52
6.3	Subpixel-Detection di un edge di rettilineo	53
6.4	Subpixel-Detection di un edge di curvilineo	55
6.4.1	Generalizzazione a tutti gli ottanti	57
6.5	Edge-Detection in condizioni di rumore	58
7	Il modello di Edge-Detection	61
7.1	Il modello di subpixel edge-detection	62
7.1.1	Scelta del coefficiente di diffusione del Perona-Malik	62
7.1.2	Subpixel analysis	64
7.1.3	Sistema di acquisizione	64
7.2	Test su edge rettilinei	64
7.2.1	Test su immagini reali	70
7.2.2	In caso di angoli	72
7.3	Test su circonferenze	74

Introduzione

Il lavoro di questa tesi volge allo studio di un metodo per il *Subpixel Edge-Detection*, ovvero la localizzazione dei contorni (edge) di un oggetto, acquisito come immagine digitale, con una precisione molto alta. Il pixel è infatti l'unità fondamentale in cui viene suddivisa un'immagine, ma le rappresentazioni digitali delle immagini forniscono di fatto una codifica a precisione maggiore rispetto al pixel, rappresentando in toni di grigio la posizione dell'edge all'interno di esso. L'obiettivo centrale del nostro lavoro sarà quindi la localizzazione dell'edge all'interno del pixel con una precisione subpixel.

Il problema è stato posto da Marposs, azienda che fornisce sistemi di misura di alta precisione, in ambienti come quello dell'officina e dell'industria, ad esempio per le macchine utensili. E in grado di fornire sia prodotti per la singola misura di un componente, sia macchine e tutto ciò che è necessario per il miglioramento della produzione ed il controllo della qualità in ambiente industriale. Gli apparecchi ed i sistemi di misura forniti devono essere in grado di rilevare misure di pezzi meccanici ed utensili, con precisioni molto alte, in molti casi fino al millesimo di millimetro, e devono spesso eseguire rilevazioni sulla superficie di un utensile di precisione. Quindi è necessario poter studiare dettagliatamente i contorni dell'oggetto, acquisiti in forma di immagine digitale, e si impone la necessità di un'analisi subpixel dell'immagine.

I capitoli sono organizzati nel modo seguente. Nel **Capitolo I** viene introdotta l'analisi multi-scala, alla base del filtraggio delle immagini, che permette di confrontarne le caratteristiche a seconda del parametro della scala. Nel **Capitolo II**, vengono richiamati alcuni metodi di denoising: Total Variation, Perona-Malik, Mean Motion Curvature, metodo di Alvarez-Lions-Morel. Vengono descritti brevemente, spiegandone i vantaggi e gli svantaggi. Il **Capitolo III** viene interamente dedicato al metodo di Perona e Malik, centrale in questo lavoro, per la sua capacità di rimuovere il rumore dalle immagini, mantenendone inalterate le caratteristiche, poichè esegue una diffusione tangenziale alla direzione dell'edge. Un altro aspetto importante nell'analisi delle immagini è il deblurring, di cui viene data una panoramica di alcuni importanti metodi nel **Capitolo IV**. In seguito viene spiegata la ragione per cui non sono stati utilizzati, motivando le scelte della tecnica utilizzata con la necessità di non alterare i valori dell'immagine. Nel **Capitolo V** viene inquadrato storicamente il problema dell'Edge-Detection e vengono introdotti i principali metodi per l'analisi Subpixel. Il **Capitolo VI** è stato dedicato alla

descrizione del metodo di Subpixel dedge-detection che è stato ritenuto piú consono al problema in questione. Esso si basa sul calcolo di aree parziali in un intorno dei pixel dell'immagine per cui passa l'edge. Infine, nel **Capitolo VII** vengono mostrati i risultati ottenuti dall'analisi Subpixel con le tecniche sopra citate. Inoltre, vengono fatte delle osservazione sul comportamento del metodo in relazione ad immagini affette da blur e noise e se ne verifica la ripetibilità.

Capitolo 1

Rappresentazione Multi-Scala

Le Equazioni alle Derivate Parziali (PDE) sono uno dei principali strumenti nel campo della regolarizzazione e/o ricostruzione delle immagini. In questo contesto, l'immagine acquistata viene considerata come lo stato iniziale di un processo diffusivo parabolico, da cui se ne estraggono delle versioni "filtrate" nel corso della sua evoluzione temporale. Questa evoluzione viene detta "Multi-Scala", è un'immersione dell'immagine iniziale in una famiglia di sue rappresentazioni gradualmente piú semplici. Cosí facendo viene introdotta una gerarchia nelle strutture delle immagini, che permette di usare questa rappresentazione Multi-Scala per estrarre importanti informazioni dall'immagine.

1.1 Preliminari

Durante il processo di acquisizione di un'immagine, essa viene deteriorata. Solitamente, questa degradazione è il risultato di due fenomeni. Il primo è deterministico, lo sfuocamento (blur), legato al metodo di acquisizione dell'immagine, a causa di eventuali difetti del sistema (sbagliata regolazione delle lenti, movimento, ecc), o fenomeni "atmosferici" all'interno di esso. Il secondo è di tipo casuale, il rumore (noise), causato da una qualsiasi trasmissione di un segnale.

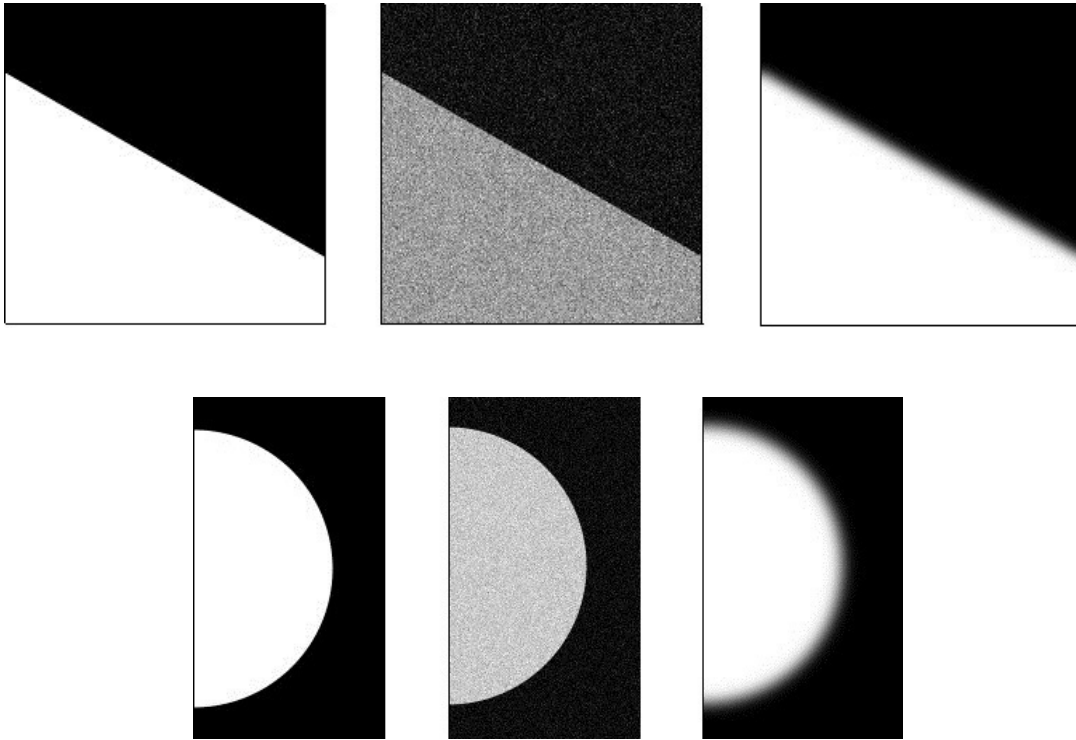


Figura 1.1: Da sinistra a destra: Immagine ideale, affetta da rumore, da sfuocamento

Si cerca di scegliere un modello di degradazione il piú possibile fedele alla realtà. Ogni modello è caratterizzato da una distribuzione probabilistica e, in molti casi, si assume quella Gaussiana. Tuttavia, varie applicazioni ne richiedono una piú specifica.

In questo contesto, si può definire un'*immagine digitale* come una funzione

$$I_0 : \Omega \ni (x, y) \mapsto I_0(x, y) \in \mathbb{R}, \quad (1.1)$$

dove Ω è un aperto limitato di \mathbb{R}^2 ed, indicando con I_0 l'immagine ideale ed I quella acquisita, è possibile definire il *modello di degradazione* nel modo seguente:

$$D : I_0 \rightarrow I := f(I_0) + \eta, \quad (1.2)$$

dove η indica il rumore introdotto al momento dell'acquisizione (solitamente di tipo Gaussiano) ed f è una funzione che dipende dallo strumento di acquisizione considerato, che causa lo sfuocamento.

Si dice che un'immagine digitale I è affetta da rumore se

$$I = I_0 + \eta, \quad (1.3)$$

dove η è una distribuzione di probabilità (Gaussiana, di Poisson ecc), mentre, si può pensare allo sfuocamento, come ad un processo di convoluzione di un'immagine, ad esempio con una Gaussiana.

Ora, ha senso definire la ricostruzione di un'immagine come un processo finalizzato alla determinazione di un'immagine \tilde{I} che minimizza un funzionale, che nel caso più semplice è una norma, come

$$\|\tilde{I} - I\| < \epsilon,$$

con la norma $\|\cdot\|$ ed $\epsilon > 0$ scelti opportunamente.

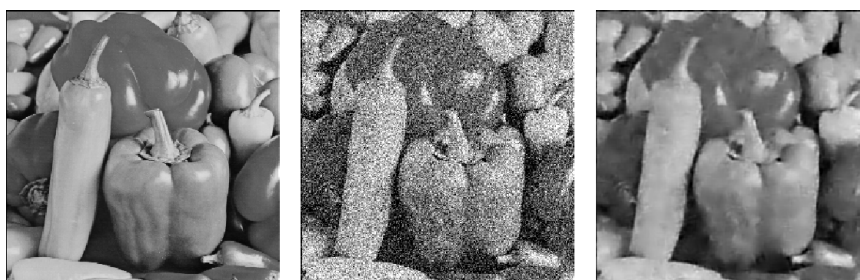


Figura 1.2: esempio della ricostruzione di un'immagine

In generale possono essere di vari tipi, come si osserva dalla figura seguente:

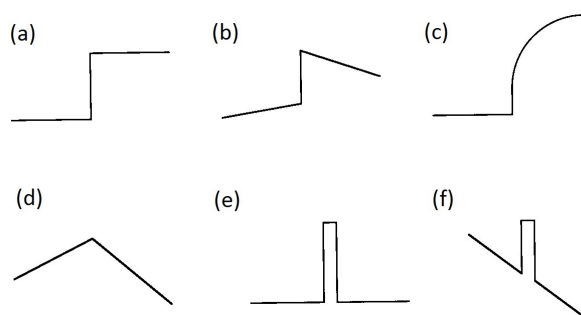


Figura 1.3: tipi di edge

Il caso (a) della figura è quello preso in considerazione in questo lavoro, di cui si può dare la definizione seguente.

Definizione 1.1. Siano J un intervallo di \mathbb{R} , con $\bar{x} \in \text{int}(J)$, ed $u : J \rightarrow \mathbb{R}$ una funzione. Si dice che \bar{x} è un edge gradino di u se

$$u(x) = \begin{cases} a, & \text{con } x < \bar{x} \\ b, & \text{con } x \geq \bar{x} \end{cases}$$

con $a \neq b$ e costanti in \mathbb{R} .

È la formalizzazione di una percezione visiva di una forte variazione nei valori dell'intensità dell'immagine. Parlando di variazione, la funzione che comunemente si usa per quantificarla è $\|\nabla u\|$, oppure una $g(\|\nabla u\|)$, con g una funzione scelta opportunamente. Trattandosi di analisi delle immagini, le quali presentano discontinuità e non derivabilità, quando si calcola il gradiente, o si fanno altre operazioni del calcolo differenziale, si considera tutto in "senso discreto".

Qualitativamente, si possono definire come *regioni omogenee*, quelle regioni in cui $\|\nabla u\|$ è costante, a meno di una soglia fissata a priori. Sono delimitate da curve, gli edge, che stabiliscono la "separazione" di una dall'altra, in cui il valore di $\|\nabla u\|$ aumenta drasticamente.

Nella seguente figura vengono mostrati l'edge ed i tipi di degradazioni:

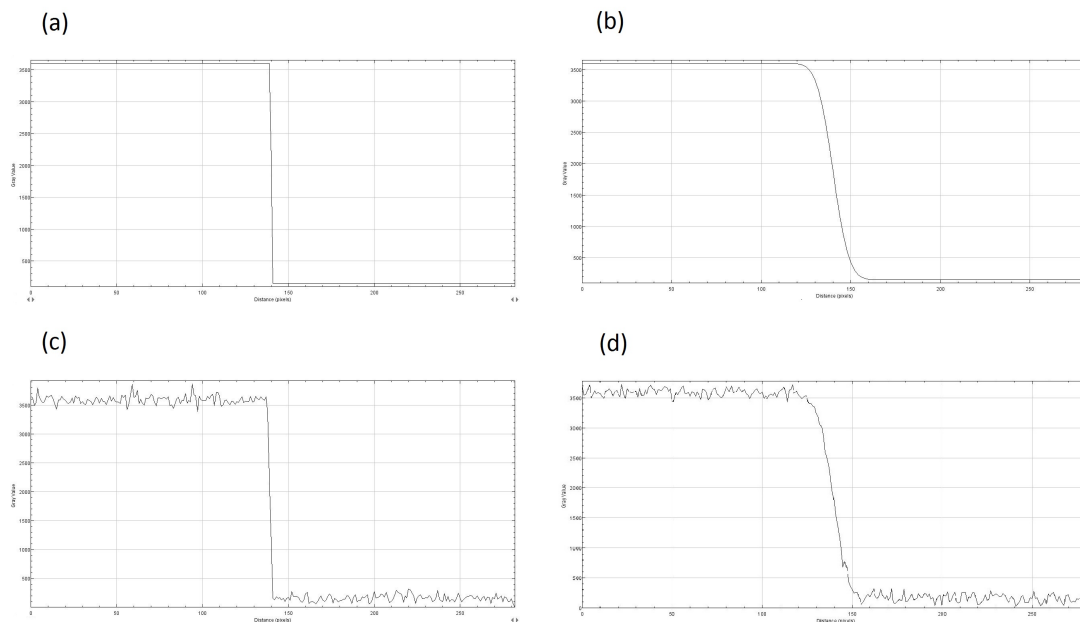


Figura 1.4: (a) edge ideale, (b) blurred, (c) noised, (d) blurred e noised

1.2 Filtro di diffusione lineare

Il piú tradizionale metodo di regolarizzazione delle immagini, usando le PDE, è un processo di diffusione lineare. L'attenzione cade sulla relazione tra il filtro lineare di diffusione e la convoluzione con una Gaussiana, l'analisi delle proprietà dell'immagine, come le sue derivate, e le caratteristiche principali della rappresentazione Multi-Scala con filtro Gaussiano.

1.2.1 Filtro Gaussiano

Si supponga di poter rappresentare un'immagine in scala di grigio I come una funzione a valori reali $I \in L^1(\mathbb{R}^2)$. La tecnica piú classica per la regolarizzazione è la convoluzione:

$$(G_\sigma * I)(x) := \int_{\mathbb{R}^2} G_\sigma(x - y)I(y) dy, \quad (1.4)$$

dove G_σ denota la funzione Gaussiana

$$G_\sigma(x) := \frac{1}{2\pi\sigma^2} \exp^{-\frac{\|x\|^2}{2\sigma^2}}. \quad (1.5)$$

con deviazione standard $\sigma > 0$.

Vi sono diverse ragioni per cui la convoluzione è un ottimo metodo di regolarizzazione. In primo luogo, poichè $G_\sigma \in C^\infty(\mathbb{R}^2)$, è possibile ottenere $G_\sigma * I \in C^\infty(\mathbb{R}^2)$ anche se I è solo assolutamente integrabile.

Inoltre, analizzando le frequenze e definendo la trasformata di Fourier come

$$(\mathcal{F}I)(\omega) := \int_{\mathbb{R}^2} I(x) \exp^{-i\langle \omega, x \rangle} dx, \quad (1.6)$$

per il teorema della convoluzione, si ha

$$(\mathcal{F}(G_\sigma * I))(\omega) = (\mathcal{F}G_\sigma)(\omega) \cdot (\mathcal{F}f)(\omega). \quad (1.7)$$

Poichè la trasformata di Fourier di una Gaussiana è ancora una Gaussiana,

$$(\mathcal{F}G_\sigma)(\omega) = \exp^{-\frac{\|\omega\|^2}{2/\sigma^2}}, \quad (1.8)$$

si osserva che (1.4) è un filtro a passo basso che attenua le alte frequenze in modo monotono. È interessante capire questo discorso nel contesto delle PDE.

Un risultato classico è dato dal seguente teorema:

Teorema 1. *Per ogni funzione $u_0 \in C(\mathbb{R}^2)$ limitata, il Problema del calore*

$$\begin{cases} \partial_t u = \Delta u \\ u(x, 0) = u_0(x) \end{cases} \quad (1.9)$$

ammette soluzione del tipo

$$u(x, t) = \begin{cases} u_0(x), & \text{se } t = 0 \\ G_{\sqrt{2t}} * u_0, & \text{se } t > 0 \end{cases} \quad (1.10)$$

Si ha anche l'unicità della soluzione a patto che u sia limitata, ad esempio restringendosi alle funzioni

$$u(x, t) \leq M \cdot e^{a\|x^2\|} \quad (M, a > 0).$$

Da (1.9) si osserva che il parametro di scala t è in relazione a $\sigma = \sqrt{2t}$ della Gaussiana. Si noti che u dipende in modo continuo dal dato iniziale u_0 , rispetto alla norma $\|\cdot\|_{L^\infty(\mathbb{R}^3)}$ e che verifica il **principio del massimo**:

Teorema 2. Siano Ω un aperto limitato di \mathbb{R}^2 , $Q_T = \Omega \times]0, T[$ e $T > 0$. Si definisce la frontiera parabolica come l'insieme:

$$\partial_p Q_T := (\bar{\Omega} \times \{t = 0\}) \cup S_T, \quad (1.11)$$

dove $S_T = \partial\Omega \times]0, T[$ ed $\bar{\Omega} = \Omega \cup \partial\Omega$. Data una funzione $u \in C^0(\bar{Q}_T) \cap C^2(Q_T)$ di variabile reale, valgono:

1. se $\frac{\partial u}{\partial t} - \Delta u \leq 0$ su Q_T , $\sup_{Q_T} u = \sup_{\partial_p Q_T} u$;
2. se $\frac{\partial u}{\partial t} - \Delta u \geq 0$ su Q_T , $\inf_{Q_T} u = \inf_{\partial_p Q_T} u$;
3. se $\frac{\partial u}{\partial t} - \Delta u = 0$ su Q_T , $\sup_{Q_T} u = \sup_{\partial_p Q_T} u$ e $\inf_{Q_T} u = \inf_{\partial_p Q_T} u$.

Per comprendere appieno la struttura dell'immagine è necessario analizzare le variazioni dei toni di grigio in un intorno di ogni punto (pixel) dell'immagine, cioè bisogna studiarne le derivate. Tuttavia, l'operazione di differenziazione quando è possibile calcolarla, è mal-posta, poichè a piccole perturbazioni dell'immagine iniziale possono corrispondere di grandi nella sua derivata. Da qui nasce la necessità della regolarizzazione, ad esempio con la convoluzione dell'immagine con una Gaussiana ed in seguito derivando, ovvero

$$\partial_{x_1}^n \partial_{x_2}^m (G_\sigma * u_0) = G_\sigma * (\partial_{x_1}^n \partial_{x_2}^m u_0) = (\partial_{x_1}^n \partial_{x_2}^m G_\sigma) * u_0. \quad (1.12)$$

Supponendo che u_0 sia sufficientemente regolare, si osservi che le derivate subiscono lo stesso processo di regolarizzazione Gaussiana dell'immagine, che è equivalente a convolvere l'immagine con delle derivate della Gaussiana. Questo porta ad un maggior effetto regolarizzante.

Esistono diversi metodi che operano questa sostituzione per localizzare gli edge in un'immagine, uno dei più famosi è il Canny edge-detector [15], che si basa sul calcolo di $\nabla(G_\sigma * u_0)$. È considerato il miglior metodo di edge-detection di tipo lineare. Ad

ogni iterazione, il tempo σ cresce ed avviene una nuova convoluzione con la derivata della Gaussiana. Infine l'edge viene localizzato dove il gradiente ha il massimo valore in norma. Analizzando l'evoluzione temporale degli zeri dell'immagine filtrata dalla diffusione lineare, si osserva un fenomeno molto interessante: *umentanto il parametro di scala σ , non si creano nuovi zeri (che non sono presenti ad una scala piú fine)*. Questa proprietà viene detta *causalità* ed è legata proprio al principio di del massimo di certi operatori parabolici.

Questa proprietà è la chiave che ha ispirato Witkin nella cosiddetta *teoria della rappresentazione Multi-Scala*.

1.3 Rappresentazione Multi-Scala

Le immagini spesso contengono una grande varietà di informazioni, a seconda della scala con cui le si “guarda”. In questi casi, in cui non si conosce a priori quale sia la scala giusta con cui analizzarle, è utile averne una rappresentazione, che permetta di comparare le strutture e facilitarne l'interpretazione. A tale scopo, Witkin introduce un tipo di rappresentazione detta “Multi-Scala”.

Si consideri un'immagine, a “livelli di grigio”, come una funzione $I_0 : \Omega \rightarrow \mathbb{R}$, dove Ω rappresenta un piano rettangolare o quadrato. L'immagine viene immersa in una famiglia

$$\{I_t := T_t(I_0) \mid t \geq 0\}, \quad (1.13)$$

dove t è detto parametro di scala e $T_t : L^1(\mathbb{R}^2) \rightarrow L^1(\mathbb{R}^2)$.

Questa famiglia verifica certi criteri, ad esempio:

- *Ricorsività:*

$$T_0(I_0) = I_0, \quad (1.14)$$

$$T_{t+s}(I_0) = T_t(T_s(I_0)), \quad \forall s, t \geq 0, \quad (1.15)$$

detta proprietà di semigruppato.

- *Linearità:*

$$T_t(aI_0 + bF_0) = aT_t(I_0) + bT_t(F_0), \quad \forall t \geq 0, \quad \forall a, b \in \mathbb{R}. \quad (1.16)$$

- *Fedeltà:*

$$T_t(I_0) \rightarrow I_0, \quad \text{se } t \rightarrow 0 \quad (1.17)$$

- *Causalità*: nella rappresentazione dell'immagine ad una scala piú grossolana, non devono nascere caratteristiche non presenti ad una scala piú fine (garantita dal principio del massimo);
- *Omogeneità ed Isotropia*: lo sfuocamento è una proprietà invariante nello spazio.

Le proprietà di regolarizzazione e la diminuzione delle informazioni nascono dalla speranza che il passaggio da una scala piú fine ad una piú grossolana non generi "artefatti" nell'immagine.

Alla luce di tutto questo, si può pensare che un'immagine sia una classe di equivalenza contenente delle immagini tutte utili per descrivere lo stesso oggetto. Due immagini della stessa classe di equivalenza differiscono, ad esempio, per i diversi livelli di grigio, possibili traslazioni e rotazioni, oppure trasformazioni piú complicate, come le trasformazioni affini. Questo rende plausibile la richiesta che la rappresentazione multi-scala sia invariante per il maggior numero possibile di queste trasformazioni.

Il lavoro pionieristico di Alvarez, Guichard, Lions e Morel [21], mostra che ogni analisi multi-scala verificante alcuni assiomi è governata da una PDE, dove l'immagine originale viene presa come condizione iniziale. Questo rende le PDE un ottimo strumento per questo tipo di analisi delle immagini.

La rappresentazione di tipo Gaussiano, introdotta nel paragrafo (1), è un caso particolare della rappresentazione multi-scala e si ottiene tramite la convoluzione con una funzione Gaussiana di varianza σ crescente ad ogni passo del metodo. In questo caso l'operatore T_σ rappresenta un filtro a passo basso definito da

$$T_\sigma(I_0) := G_\sigma * I_0 \quad (1.18)$$

Da quanto detto riguardo ai processi di diffusione lineare, si osserva che la convoluzione di un'immagine con un filtro Gaussiano, oppure applicare il problema del calore

$$\begin{cases} \partial_t I = \Delta I = \partial_{xx} I + \partial_{yy} I \\ I(x, y; 0) = I_0(x, y) \end{cases} \quad (1.19)$$

sono due operazioni del tutto analoghe. Dunque, operare su un'immagine un processo di diffusione, all'aumentare del tempo, significa darne una rappresentazione Multi-Scala. Si nota anche il legame di un processo regolarizzante con una eliminazione del rumore.

Purtroppo, questo modo di procedere è soggetto a forti limitazioni:

- La convoluzione con una Gaussiana non porta solo ad una riduzione del rumore, ma anche allo sfuocamento di importanti dettagli, come gli edge, rendendoli piú difficili da identificare. Inoltre, questo modo di procedere non prende in considerazione a priori alcuna informazione sull'immagine;

- I filtri di diffusione dislocano gli edge, passando da una scala piú fine ad una piú grossolana. Strutture che vengono identificate ad una scala piú grossolana in realtà non sono in posizione esatta, ma devono essere ricavate dall'immagine iniziale; questo causa problemi di instabilità al metodo.

1.3.1 Altre proprietà importanti

Dato un operatore $T_t : L^1(\mathbb{R}^2) \rightarrow L^1(\mathbb{R}^2)$ lineare definito come in (1.13) ed un'unica soluzione (1.10) del problema (1.9), si può dimostrare che la rappresentazione Multi-Scala gode di altre importanti proprietà:

- *Invarianza del livello di grigio:*

$$T_t(0) = 0 \quad \text{e} \quad T_t(I_0 + c) = T_t(I_0) + c; \quad (1.20)$$

per ogni costante c ;

- *Invarianza per traslazione:*

$$T_t(\tau_h(I_0)) = \tau_h(T_t(I_0)); \quad (1.21)$$

dove τ_h è una traslazione tale che $\tau_h(f)(x) = f(x + h)$;

- *Invarianza di scala:*

$$T_t(H_\lambda(I_0)) = H_\lambda(T_{t'}(I_0)); \quad (1.22)$$

con $t' = t\lambda^2$ dove $(H_\lambda(f))(x) = f(\lambda x)$;

- *Invarianza per isometria:*

$$T_t(\mathcal{R}(I_0)) = \mathcal{R}(T_t(I_0)); \quad (1.23)$$

per ogni trasformazione ortogonale \mathcal{R} di \mathbb{R}^2 , dove $(\mathcal{R}(f))(x) = f(\mathcal{R}(x))$;

- *Conservazione del valore medio:*

$$T_t(M(I_0)) = M(T_t(I_0)), \quad \text{con } M(f) := \frac{1}{|\Omega|} \int_{\Omega} f(x) dx; \quad (1.24)$$

- *Proprietà di semigruppato:* vista in precedenza;

- *Principio del confronto:*

$$\text{se } I_0 \leq \bar{I}_0, \text{ allora } T_t(I_0) \leq T_t(\bar{I}_0); \quad (1.25)$$

Queste proprietà sono di grande importanza. Per esempio, l'invarianza del livello di grigio significa che l'analisi non dipende dal range di luminosità dell'immagine iniziale. In generale, le altre proprietà geometriche traducono l'invarianza dell'analisi dell'immagine rispetto alle posizioni dell'osservatore e dell'osservato.

Purtroppo queste richieste non sono sufficienti. Ad esempio, l'equazione del calore è troppo regolarizzante ed anche se l'immagine iniziale $I_0(x, t)$ fosse di classe C^∞ in $x, t > 0$, rimarrebbe il rischio di perdere delle informazioni sull'edge. Questo tipo di diffusione è isotropica: non dipende dalle caratteristiche dell'immagine, ma è la stessa in ogni direzione, dunque non preserva gli edge.

Nel prossimo capitolo verranno mostrati ed analizzati alcuni metodi variazionali, basati sull'analisi multi-scala e finalizzati al denoising di un'immagine.

Capitolo 2

Metodi di Denoising

Nell'analisi di un'immagine digitale, è necessario trovare un metodo di denoising efficiente, che permetta di ridurre il rumore, senza perdere i dettagli dell'immagine. Ciò è particolarmente importante nel caso della localizzazione di un edge.

Ci sono un grande numero di filtri per il denoising, come quelli di Wiener, Bilaterale e Wavelet. Tuttavia, in questo lavoro sono stati privilegiati dei metodi di diffusione non lineare, basati su un funzionale che dipende fortemente dal ∇ , poichè esso individua le direzioni ortogonali ai bordi, permettendo una regolarizzazione dell'immagine tangente agli edge, senza alterarne la posizione. Di questi metodi ne vengono descritti alcuni.

Quello di Perona e Malik [30] verrà trattato più dettagliatamente in seguito, essendo uno degli argomenti di maggior interesse.

2.1 Steepest Descent

I metodi di denoising che verranno trattati si basano su un problema di minimo di un funzionale. Per calcolare computazionalmente una soluzione di, un modo possibile è applicare la cosiddetta *Steepest Descent* (o flusso di Steepest Descent):

$$\begin{cases} X'(t) = F(X(t)), \\ X(0) = u_0 \end{cases} \quad (2.1)$$

dove si sceglie $F = -\nabla S$, con S un funzionale su uno spazio di Banach \mathcal{E} , ed $u \in \mathcal{E}$, dunque F che sarà l'equazione di Eulero-Lagrange dei metodi che verranno descritti. Se $X(t)$ è della forma

$$X(t) = (x, y, u(t, x, y))^T,$$

si ricava

$$\frac{dX}{dt}(t) = \left(0, 0, \frac{\partial u}{\partial t}(t, x, y) \right)^T. \quad (2.2)$$

Dunque, il flusso è lungo la direzione verticale z .

Osservazione. Questo metodo sfrutta il fatto che il ∇S è ortogonale agli insiemi di livello, quindi il flusso nella direzione $-\nabla S$ abbassa i valori del funzionale e la soluzione tende ai minimi (locali) lungo il flusso. Se il funzionale è convesso, allora il flusso tende al minimo globale, altrimenti la scelta del dato iniziale è fondamentale per la determinazione del minimo.

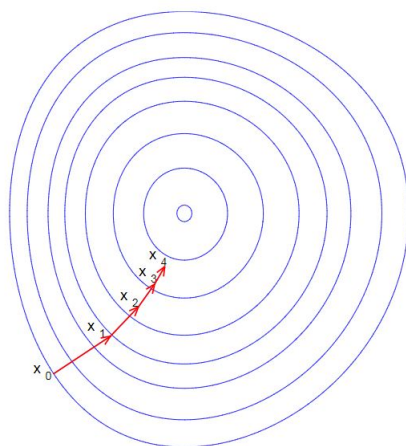


Figura 2.1: metodo di Steepest Descent

2.2 Total Variation

Questo metodo [3] è importante nell'ambito del denoising, poichè, come viene mostrato in [23], una buona norma per il gradiente è quella L^1 , piuttosto della norma L^2 . Soprattutto alla luce del fatto che verranno analizzati edge di tipo gradino.

Prima di arrivare al metodo, è necessario dare alcune definizioni preliminari:

Definizione 2.1. Siano Ω un aperto di \mathbb{R}^n ed $u \in L^1(\Omega, \mathbb{R}^n)$, si dice *Total Variation* di una funzione u

$$\|Du\| := \sup \left\{ \int_{\Omega} u \nabla g \, dx \mid g \in C_0^1(\Omega, \mathbb{R}^n), \|g\|_{L^\infty} \leq 1 \, \forall x \in \mathbb{R}^n \right\}, \quad (2.3)$$

denotata con $TV(u)$.

Ne viene la definizione dello spazio di Banach seguente:

Definizione 2.2. Lo spazio delle funzioni con TV limitata sono:

$$BV := \{u \in L^1 \mid \|u\|_{BV} < \infty\}, \quad (2.4)$$

dove

$$\|u\|_{BV} := \|u\|_{L^1} + \|Du\|. \quad (2.5)$$

2.2.1 Formula Variazionale

Considerata u_0 un'immagine reale affetta da rumore gaussiano, si vuole trovare un'immagine u incognita, che la individua a meno del rumore, cioè

$$u_0 = u + \eta. \quad (2.6)$$

ed modello che definisce questo problema variazionale è il seguente:

$$\min_{u \in BV(\Omega)} \int_{\Omega} \|\nabla u\| \, dx, \quad (2.7)$$

il cui minimo è assunto in \bar{u} .

Il rumore è di tipo Gaussiano, dunque

$$\int_{\Omega} u \, dx = \int_{\Omega} u_0 \, dx, \quad (2.8)$$

ovvero η ha media nulla, e

$$\frac{1}{2} \int_{\Omega} (u - u_0)^2 \, dx = \sigma^2, \quad \sigma > 0, \quad (2.9)$$

dove σ è la varianza.

Per risolvere il problema di minimo (2.7), con il vincolo (2.9), si calcola l'equazione di Eulero-Lagrange da cui si ottiene il problema seguente:

$$\begin{cases} \partial_t u = \operatorname{div} \left(\frac{\nabla u}{\|\nabla u\|} \right) - \lambda(u - u_0), & t > 0, x \in \Omega \\ \partial_{\bar{n}} u(x) = 0, & \text{su } \partial\Omega \\ u(x; 0) = u_0(x) \end{cases}$$

la cui soluzione è \bar{u} , minimo del problema (2.7). Inoltre, si osservi che, in generale, un problema di minimo verifica le condizioni al bordo di Neumann.

Osservazione. Come è stato introdotto precedentemente, il Total Variation risulta utile nell'ambito del denoising di immagini. Inoltre, in questo lavoro, la riduzione del rumore è la fase iniziale, con lo scopo di migliorare l'edge-detection. Siccome nelle immagini gli edge da localizzare sono per la maggior parte rettilinei, allora questo è un buon metodo, poichè gli insiemi di livello della norma L^1 sono quadrati, dunque i minimi ammettono spigoli, a differenza di quella L^2 , i cui minimi non li ammettono, poichè gli insiemi di livello sono dei cerchi.

2.3 Metodo di Perona e Malik

L'idea principale di Perona e Malik [30] è stata quella di permettere un'interazione tra le fasi della rappresentazione multi-scala, grazie ad un coefficiente di diffusione non costante. Il problema alle PDE proposto è il seguente:

$$\begin{cases} \partial_t u = \operatorname{div} (f(\|\nabla u\|)\nabla u) & t > 0, x \in \Omega \\ \partial_{\bar{n}} u(x; t) = 0 & \text{su } \partial\Omega \\ u(x; 0) = u_0 \end{cases}$$

Il coefficiente f , che determina la diffusione anisotropica, rappresenta una funzione non decrescente, tale che $f(0) = 1$ ed $f(s) \rightarrow 0$ per $s \rightarrow \infty$. L'aspetto fondamentale è quello di regolarizzare l'immagine dove il $\|\nabla u\|$ ha un valore "piccolo" e risaltarne le caratteristiche dove $\|\nabla u\|$ ha un valore "grande". Questo permette di preservare e mantenere le caratteristiche degli edge dell'immagine, e di regolarizzare le regioni interne. I risultati di Perona e Malik sono notevoli, poichè gli edge rimangono stabili anche dopo un lungo tempo di elaborazione.

2.3.1 Formula Variazionale

Data un'immagine u affetta da rumore come in (2.6) visto nella sezione precedente, il problema di minimo da risolvere è il seguente:

$$\min_{u \in L^1} \int_{\Omega} F(\|\nabla u\|^2) dx, \quad (2.10)$$

con $F(t) = \int_0^t f(s) ds$ ed Ω un aperto di \mathbb{R}^n .

Grazie all'equazione di Eulero-Lagrange, si riconduce il problema di minimo (2.10) alla risoluzione di

$$\begin{cases} \partial_t u = \operatorname{div}(f(\|\nabla u\|^2) \nabla u) + \lambda(u - u_0), & t > 0, x \in \Omega \\ \partial_{\bar{n}} u = 0, & \text{su } \partial\Omega \\ u(x; 0) = u_0 \end{cases}$$

Si osservi che l'aggiunta del termine $u - u_0$, dovuta a Nordstrom [26], ha lo scopo di risolvere il problema del tempo di arresto, facendo in modo che l'immagine ricostruita dal metodo rimanga fedele a quella iniziale.

Le due scelte più famose per il coefficiente di diffusione f sono:

$$f(s) = \frac{1}{1 + (\|\nabla u\|/K)^2} \quad \text{ed} \quad f(s) = e^{-(\|\nabla u\|/K)^2}, \quad k > 0. \quad (2.11)$$

Il metodo di Perona e Malik verrà trattato più dettagliatamente in seguito, essendo un argomento centrale di questo lavoro.

2.4 Mean Motion Curvature

Questo metodo nasce con lo scopo di diffondere solo nella direzione degli edge. Un'immagine viene considerata una superficie di dimensione 2 in uno spazio tridimensionale. Ad esempio, data un'immagine a toni di grigio, il valore di ogni pixel rappresenta un'altezza, dunque può essere rappresentata ed analizzata in \mathbb{R}^3 . Questo tipo di approccio fornisce dei nuovi strumenti per la ricostruzioni delle immagini. Prima di discutere il metodo, si richiamano alcuni concetti di geometria differenziale.

2.4.1 Varietà differenziabile

Si ricordano i concetti di varietà e spazio tangente.

Definizione 2.3. Sia M un sottoinsieme di \mathbb{R}^N , $M \neq \emptyset$. Si dice che M è una varietà differenziabile di \mathbb{R}^N di dimensione d , con $1 \leq d < N$, se per ogni $p \in M$ esistono un intorno U di p in M ed una funzione $f : U \rightarrow \mathbb{R}^{N-d}$, $f \in C^\infty$, tali che:

1. $M \cap U = \{u \in U \mid f(u) = 0\}$;
2. $J_f(p)$ ha rango massimo (cioè $N - d$).

In questo caso, si dice che f è un'equazione locale di M in p .

Osservazione. Il *teorema di Dini* sulle funzioni implicite, garantisce che ogni varietà differenziabile di dimensione d sia localmente il grafico di una funzione di d variabili reali. Di conseguenza, per ogni $p \in M$, esistono un intorno U di p in M ed un diffeomorfismo di classe C^∞

$$x : U \rightarrow \Omega, \quad (2.12)$$

dove Ω è un sottoinsieme aperto di \mathbb{R}^d .

Fissati $p \in M$, U ed x come sopra, si ha che localmente, ogni punto $q \in U$ univocamente determinato dalla sua immagine $x(q) \in \Omega$, con la quale si identifica per comodità di notazione. Le componenti q vengono dette *coordinate locali* di $x(q) \in \mathbb{R}^d$ e si scrive

$$x = (x^1, \dots, x^d) \quad (2.13)$$

per indicare le coordinate euclidee in \mathbb{R}^d , adottate come coordinate locali sulla varietà M .

Ora, è possibile dare la nozione seguente.

Definizione 2.4. Siano M una varietà differenziabile di \mathbb{R}^N , di dimensione d , e $p \in M$. Si dice che $h \in \mathbb{R}^N$ è un *vettore tangente ad M in p* se esiste una curva differenziabile $\psi :]-\epsilon, \epsilon[\rightarrow M$ tale che

$$\psi(0) = p, \quad \psi'(0) = h. \quad (2.14)$$

L'insieme dei vettori di \mathbb{R}^N tangenti in p ad M si chiama *spazio tangente in p ad M* e viene denotato con $T_p M$.

Un teorema importante sullo spazio tangente è il seguente:

Teorema 3. Sia M una varietà differenziabile di \mathbb{R}^N , di dimensione d , e sia f una sua equazione locale in $p \in M$. Allora

$$T_p M = \ker df(p). \quad (2.15)$$

Corollario 4. $T_p M$ è uno spazio vettoriale di dimensione d .

2.4.2 Varietà Riemanniana

Definizione 2.5. Una *metrica Riemanniana* su una varietà differenziabile M è definita da un prodotto scalare su ogni spazio tangente T_pM , che dipende in modo C^∞ dal punto base $p \in M$.

Ora, è possibile definire una *varietà Riemanniana*, cioè una varietà differenziabile su cui è definita una metrica Riemanniana.

Date le coordinate locali $x = (x^1, \dots, x^d)$ su M , una metrica si rappresenta in ogni punto mediante una matrice simmetrica e definita positiva del tipo

$$(g_{i,j}(x))_{i,j=1,\dots,d}, \quad \text{con } g_{i,j} \in C^\infty \forall i, j. \quad (2.16)$$

Il prodotto scalare di due vettori tangenti $v, w \in T_pM$ rappresentati rispettivamente dalle coordinate locali, $(v^1, \dots, v^d), (w^1, \dots, w^d)$, è

$$\langle v, w \rangle := g_{i,j}(x(p))v^i w^j. \quad (2.17)$$

Si osservi che definire $(g_{i,j}(x))_{i,j=1,\dots,d}$, in un particolare sistema di coordinate locali $x = (x^1, \dots, x^d)$ non comporta restrizioni, poichè la dipendenza di classe C^∞ da p , richiesta nella definizione, non dipende dalla scelta delle coordinate.

2.4.3 Metrica indotta su un grafico

Si consideri un'immagine

$$I : \Omega \rightarrow \mathbb{R}, \quad I(x, y) \in [0, 256], \quad (2.18)$$

con $\Omega \subset \mathbb{R}^2$ e si definisca

$$M := \{(x, y, I(x, y)) \mid (x, y) \in \Omega\}. \quad (2.19)$$

M è una varietà bidimensionale di \mathbb{R}^3 , dunque è possibile costruire su di essa una metrica Riemanniana. Si prenda $p = (x, y) \in M$, lo spazio tangente T_pM è il sottospazio vettoriale di \mathbb{R}^3 generato dai vettori

$$e_1 = (1, 0, I_x(x, y)), \quad e_2 = (0, 1, I_y(x, y)). \quad (2.20)$$

Su $T_pM : \text{Span}\{e_1, e_2\}$ si definisce il prodotto scalare $\langle \cdot, \cdot \rangle_g$ in questo modo:

$$\langle (\alpha_1, \alpha_2), (\beta_1, \beta_2) \rangle_g := \langle \alpha_1 e_1 + \alpha_2 e_2, \beta_1 e_1 + \beta_2 e_2 \rangle, \quad (2.21)$$

per ogni $(\alpha_1, \alpha_2), (\beta_1, \beta_2) \in T_pM$, dove $\langle \cdot, \cdot \rangle$ denota il prodotto scalare euclideo in \mathbb{R}^3 . La metrica indotta sull'immagine viene scritta esplicitamente così

$$\langle (\alpha_1, \alpha_2), (\beta_1, \beta_2) \rangle_g = \begin{pmatrix} \alpha_1 & \alpha_2 \end{pmatrix} \begin{pmatrix} \|e_1\|^2 & \langle e_1, e_2 \rangle \\ \langle e_2, e_1 \rangle & \|e_2\|^2 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}. \quad (2.22)$$

Di conseguenza, la metrica indotta sull'immagine è Riemanniana e definita come:

$$g_{i,j}(x, y) := \langle e_i(x, y), e_j(x, y) \rangle, \quad \forall i, j \in \{1, 2\}, \quad \forall (x, y) \in M. \quad (2.23)$$

Dopo queste premesse, si può considerare un'immagine a colori come una superficie immersa \mathcal{M} in \mathbb{R}^3 , definita da:

$$\Sigma \rightarrow \mathcal{M} \quad (2.24)$$

$$(\sigma_1, \sigma_2) \rightarrow X(\sigma_1, \sigma_2) = (X_1(\sigma_1, \sigma_2), X_2(\sigma_1, \sigma_2), X_3(\sigma_1, \sigma_2)) \quad (2.25)$$

dove Σ è detta *varietà immagine* ed \mathcal{M} è lo *spazio delle caratteristiche di σ* .

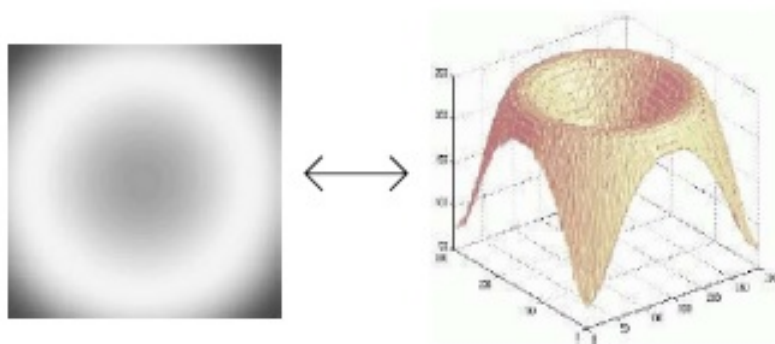


Figura 2.2: Interpretazione di un'immagine come una superficie

L'aspetto fondamentale è che Σ ed \mathcal{M} vengono viste come varietà Riemanniane, dunque equipaggiate di un'opportuna metrica (Riemanniana) indotta dall'immagine.

Per facilitare la comprensione, si consideri l'esempio seguente:

$\Sigma = \Omega$, il dominio dell'immagine;

$(\sigma_1, \sigma_2) = (x, y)$, le coordinate cartesiane;

$X(x, y) = ((x, y), u(x, y))$, dove $u(x)$ rappresenta l'intensità del livello di grigio.

La metrica su Σ è quella euclidea, $dx^2 + dy^2$, mentre su \mathcal{M} si ha quella indotta ds^2 e, con un breve calcolo, si osserva che

$$ds^2 = (dx \ dy) \cdot G \cdot \begin{pmatrix} dx \\ dy \end{pmatrix}.$$

dove, posti $u_x := \partial_x u$ ed $u_y = \partial_y u$,

$$G = \begin{pmatrix} 1 + u_x^2 & u_x u_y \\ u_x u_y & 1 + u_y^2 \end{pmatrix} \quad (2.26)$$

è la matrice associata simmetrica definita positiva.

2.4.4 Descrizione del metodo e formula variazionale

In generale, durante il processo di acquisizione, un'immagine viene affetta da rumore e deterioramento. Per ricostruire l'immagine ideale a partire da quella degradata, si cerca \mathcal{M} come la superficie minima, in questo modo le singolarità vengono regolarizzate. Posto $g := \det(G)$, si vuole calcolare il minimo, rispetto ad u , dell'integrale seguente:

$$\mathcal{S}(\mathcal{M}(u)) = \iint_{\Omega} \sqrt{g} \, dx dy = \iint_{\Omega} \sqrt{1 + u_x^2 + u_y^2} \, dx dy. \quad (2.27)$$

Se esiste una soluzione $\bar{u}(x, y)$ al problema di minimo, il suo grafico è una superficie minima

$$\min_u (\mathcal{S}(\mathcal{M}(u))) \quad (2.28)$$

allora \bar{u} verifica l'Equazione di Eulero-Lagrange, che è l'equazione dei punti critici del funzionale S ,

$$\frac{\partial}{\partial x} \left(\frac{u_x}{\sqrt{1 + u_x^2 + u_y^2}} \right) + \frac{\partial}{\partial y} \left(\frac{u_y}{\sqrt{1 + u_x^2 + u_y^2}} \right) = 0 \quad (2.29)$$

ovvero

$$\frac{u_{xy}(1 + u_y^2) - 2u_x u_y u_{xy} + u_{xy}(1 + u_x^2)}{(1 + u_x^2 + u_y^2)^{3/2}} = 0 \quad (2.30)$$

L'equazione (2.30) è equivalente a dire che la curvatura media H di \mathcal{M} è nulla.

Facendo riferimento a (2.1), se si sceglie $F = (0, 0, \alpha H)$, si ottiene

$$\frac{\partial u}{\partial t} = \alpha H = \alpha \frac{u_{xx}(1 + u_y^2) - 2u_x u_y u_{xy} + u_{yy}(1 + u_x^2)}{(1 + u_x^2 + u_y^2)^{3/2}}. \quad (2.31)$$

dove il coefficiente $\alpha \in \mathbb{R}$ può essere interpretato all'interno dell'equazione come un parametro che "pesa" la regolarizzazione. Posto il coefficiente

$$r = \frac{1}{\sqrt{1 + u_x^2 + u_y^2}}, \quad (2.32)$$

si può generalizzare l'equazione (2.31) scrivendo

$$\frac{\partial u}{\partial t} = r^{\gamma+3} (u_{xy}(1 + u_y^2) - 2u_x u_y u_{xy} + u_{xy}(1 + u_x^2)). \quad (2.33)$$

dove, al variare di γ , si ricavano i due metodi seguenti:

- per $\gamma = -1$ si ha il mean motion curvature

$$\frac{\partial u}{\partial t} = \frac{u_{xx}(1 + u_y^2) - 2u_x u_y u_{xy} + u_{yy}(1 + u_x^2)}{1 + u_x^2 + u_y^2} \quad (2.34)$$

- per $\gamma = +1$ si ha l'operatore di Laplace-Beltrami

$$\frac{\partial u}{\partial t} = \frac{u_{xx}(1 + u_y^2) - 2u_x u_y u_{xy} + u_{yy}(1 + u_x^2)}{(1 + u_x^2 + u_y^2)^2}. \quad (2.35)$$

2.5 Metodo di Alvarez, Lions e Morel

Alvarez, Lions e Morel in [21] propongono di analizzare un'immagine tramite il problema parabolico non lineare seguente:

$$\begin{cases} \partial_t u = f(\|\nabla G_\sigma * u\|) \|\nabla u\| \operatorname{div} \left(\frac{\nabla u}{\|\nabla u\|} \right) & t > 0, x \in \Omega \\ \partial_{\vec{n}} u = 0 & \text{su } \partial\Omega \\ u(x; 0) = u_0(x) \end{cases}$$

con Ω un aperto di \mathbb{R}^n .

I termini presenti nell'equazione si interpretano nel modo seguente:

- il termine $\|\nabla u\| \operatorname{div} \left(\frac{\nabla u}{\|\nabla u\|} \right)$ rappresenta la diffusione nella direzione ortogonale del gradiente. Ha lo scopo di regolarizzare l'immagine ai lati dell'edge e di preservarne le caratteristiche, diffondendo in maniera parallela ad esso.
- il termine $f(\|\nabla G_\sigma * u\|)$ viene usato per “migliorare” l'edge, ovvero controlla la velocità di diffusione. Se il valore di $\|\nabla u\|$ è “piccolo” in un intorno di un punto x dell'immagine, allora si considera x appartenente ad una regione interna e regolare. Invece se $\|\nabla u\|$ è “grande” si considera appartenente ad un edge.

Con questo metodo, viene generalizzata l'equazione, poichè diventa stabile, di cui si trova una soluzione, ma non è un minimo di un funzionale, dunque non è nemmeno un problema variazionale.

Capitolo 3

Metodo di Perona-Malik

I metodi di regolarizzazione adattiva si basano sull'idea di applicare un processo che dipende dalle proprietà locali dell'immagine. Sebbene questo sia ben noto nel campo dell'elaborazione delle immagini, una formulazione corrispondente, basata sulle PDE, è stata data in precedenza da Perona e Malik in [29], nel 1987.

È interessante discutere il loro modello di regolarizzazione, essendo uno dei più efficienti in letteratura, nonostante sia malposto dal punto di vista delle PDE ad esso accociato.

3.1 Descrizione dell'equazione

Pietro Perona e Jitendra Malik propongono una nuova definizione di “Multi-Scala”, che opera attraverso un processo di *diffusione non lineare*. Il coefficiente di diffusione viene scelto a favore della regolarizzazione delle zone intraregione dell'immagine, ma riducendo la diffusività nelle zone dell'immagine in cui c'è una maggiore probabilità della presenza di un edge, permettendone una buona localizzazione. La misura di questa probabilità è data dalla norma del gradiente $\|\nabla u\|^2$.

Come è stato detto nel capitolo precedente, l'idea principale della rappresentazione Multi-Scala è considerare una famiglia di immagini $I(x, y; t)$ ottenute tramite la soluzione di un'equazione.

Per migliorare il metodo, Perona e Malik hanno messo a punto tre criteri principali che devono essere soddisfatti dal metodo Multi-Scala:

1. *Causalità*: definita come prima;
2. *Immediata localizzazione dei bordi ad ogni scala*: ad ogni risoluzione, i bordi devono essere preservati, e non devono essere dislocati dalla loro posizione originaria;
3. *Regolarizzazione a tratti*: viene privilegiata la regolarizzazione nelle regioni interne, piuttosto che attraverso i loro bordi.

Il modo più semplice per trovare un metodo che verifichi questi tre criteri è quello di lavorare sul coefficiente di diffusione $c = c(x, y; t)$, fino ad ora considerato costante. Questa assunzione non è obbligatoria, ma si può scegliere c in modo opportuno.

Dunque, si ottiene l'*equazione di Perona-Malik di diffusione non lineare*

$$\frac{\partial I}{\partial t} = \operatorname{div} (c(x, y; t) \nabla I) = \nabla c \cdot \nabla I + c(x, y; t) \Delta I, \quad (3.1)$$

riducibile all'equazione di diffusione isotropa del calore nel caso di c costante.

Nonostante Perona e Malik definiscano il loro filtro come anisotropico, bisogna osservare che, nella loro terminologia, può essere visto come un modello isotropico, poiché utilizza una diffusività a valori scalari e non un tensore di diffusione.

I risultati sono impressionanti, anche dopo una lunga elaborazione dell'immagine, gli edge mantengono le loro caratteristiche. È stato dimostrato che i metodi di Edge-Detection basati sull'equazione (3.1) hanno prestazioni migliori rispetto al metodo lineare di Canny. Questo vantaggio è dovuto alle due operazioni di diffusione ed Edge-Detection che agiscono insieme, invece di due processi separati uno di seguito all'altro. Inoltre, c'è un'altra ragione per questi ottimi risultati sugli edge, che verrà trattata nella prossima sezione.

3.2 Proprietà della diffusione anisotropica

Si supponga che al tempo t (parametro di scala) si sappia dove sono localizzati i confini di una certa regione dell'immagine. Si vuole favorire la regolarizzazione all'interno della regione considerata, piuttosto che attraverso i suoi bordi. Di conseguenza, ponendo il coefficiente di conduzione “ c ” uguale ad 1 all'interno della regione e 0 sul bordo, gli sfocamenti di regioni distinte non interferiscono tra loro ed i confini rimangono netti. Ovviamente, non si conosce la posizione esatta dell'edge, ma con un'opportuna scelta di c se ne può dare un'ottima stima.

Sia $E = E(x, y; t)$ una funzione a valori vettoriali che rappresenta questa stima, definita sull'immagine e con le seguenti proprietà:

1. $E(x, y; t) = 0$ all'interno di ogni regione;
2. $E(x, y; t) = K \cdot e(x, y; t)$ in ogni punto dell'edge, dove e è un vettore unitario ed ortogonale all'edge in ogni suo punto e K è il contrasto locale, cioè la differenza di intensità in un intorno dell'edge.

Se si conoscesse una tale funzione $E(x, y; t)$, si potrebbe scegliere $c = g(\|E\|)$ come coefficiente di conduzione. In accordo con le proprietà di cui sopra, $g(\cdot)$ deve essere non negativa, monotonicamente decrescente e tale che $g(0) = 1$.

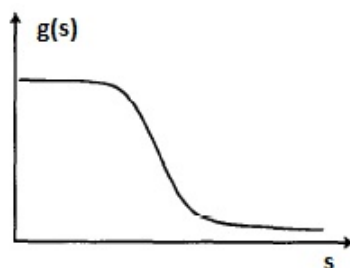


Figura 3.1: Forma qualitativa di $g(\cdot)$

In questo modo il processo di diffusione, dato dall'equazione

$$\partial_t u = \operatorname{div} (g(\|E\|) \cdot \nabla u), \quad (3.2)$$

agisce principalmente nelle regioni interne, lontane dai bordi, in cui il valore di E è basso, o nullo; al contrario il suo valore aumenta in prossimità di un edge, ma diminuisce l'azione dell'operatore.

Il successo di questo processo di diffusione nel soddisfare i tre criteri enunciati in precedenza dipende da quanto è accurata la scelta di E .

Sperimentalmente, si è visto che porre $E(x, y; t) = \nabla I(x, y; t)$ dà ottimi risultati. Così, dato Ω un aperto limitato di \mathbb{R}^2 , si arriva al *modello anisotropico di Perona-Malik*:

$$\begin{cases} \partial_t u = \operatorname{div} (g(|\nabla u|) \nabla u), & x \in \Omega, t > 0 \\ \frac{\partial u}{\partial \vec{n}}|_{\partial\Omega} u = 0, & t > 0 \\ u|_{t=0} = u_0, & x \in \Omega \end{cases} \quad (3.3)$$

dove $g : [0, +\infty) \rightarrow (0, +\infty)$ ed \vec{n} è la normale uscente.

Al problema sono state date le condizioni al bordo adiabatiche di Neumann che, numericamente, corrispondono ad un prolungamento dell'immagine, per riflessione, lungo i bordi.

L'equazione in (3.3) è l'*equazione di Eulero-Lagrange* associata al funzionale

$$F(u) = \frac{1}{2} \int_{\Omega} \bar{g} (|\nabla u|^2) dx, \quad (3.4)$$

dove $\bar{g}(s^2) = g(s)$.

Nordstrom in [26] introduce nel funzionale il termine $u - u_0$, dunque

$$F(u) = \frac{1}{2} \int_{\Omega} \bar{g} (|\nabla u|^2) dx + \frac{1}{2} \int_{\Omega} (u - u_0)^2 dx \quad (3.5)$$

la cui equazione di Eulero-Lagrange diventa

$$\frac{\partial u}{\partial t} = \operatorname{div} (\bar{g} (|\nabla u|^2) \nabla u) + (u - u_0). \quad (3.6)$$

In questo modo, u rimane "vicino" ad u_0 , risolvendo in parte il problema del tempo di arresto dell'algoritmo.

Osservazione. Ci sono varie scelte di $g(\cdot)$, la più ovvia delle quali è una funzione a valori binari in base al valore del gradiente, ovvero: si fissa una determinata soglia per il valore del gradiente di u , sotto la quale si suppone la "certezza" di non essere su un bordo, e si pone la funzione uguale ad u , mentre sopra al livello di soglia ci sono i valori del gradiente che corrispondono ai bordi da preservare, dunque $g = 0$.

3.2.1 Il principio di propogazione dei massimi

Il principio del massimo è un importante risultato delle PDE, grazie al quale si stabilisce che la diffusione anisotropica soddisfa il criterio di causalità. Inoltre, si vedrà come, con un'opportuna scelta di $g(\cdot)$, sia possibile risaltare l'edge, oltre a preservarlo.

Il criterio di causalità richiede che non si creino degli “artefatti” nell’immagine, nel passaggio da una scala più fine ad una più grossolana della rappresentazione Multi-Scala. Se si identificano questi “artefatti” con dei “picchi” della funzione intensità $I(x, y; t)$, al variare del parametro di scala t , allora la nascita di un nuovo “picco” implicherebbe la formazione di un nuovo massimo o minimo della funzione $I(x, y; t)$. Di conseguenza il criterio di causalità può essere stabilito mostrando che tutti i massimi ed i minimi della rappresentazione Multi-Scala appartengono all’immagine originale I_0 ed ai bordi del dominio di interesse, a patto che il coefficiente di conduzione sia positivo.

Osservazione. In seguito, verrà scelto un coefficiente di diffusione $g(\cdot)$ tale per cui, calcolato lungo l’edge, abbia segno negativo. Verrà mostrato che questo fatto ha un riscontro positivo dal punto di vista numerico, ma è importante osservare che l’equazione di Perona-Malik (3.1) non verifica il principio del massimo (ne forte, ne debole) poiché risulta un Laplaciano negativo.

3.2.2 Effetto sull’edge

Con i tradizionali filtri a basso passo e la diffusione lineare, per rimuovere il rumore grazie al metodo “Multi-scala”, si “paga” il prezzo del blurring dell’edge, che diventa più difficile da localizzare. Al contrario, per migliorare l’edge conviene usare un filtro a passo alto, od un’equazione di diffusione all’indietro nel tempo (parametro di scala), che è un problema mal-posto.

Verrà mostrato che, se il coefficiente di conduzione $c(\cdot)$ viene scelto come un’opportuna funzione del gradiente dell’immagine, si ha una diffusione anisotropica all’avanti che migliora l’edge, mantenendo la stabilità (garantita dal principio del massimo).

Senza perdere di generalità, si può modellare un edge come una “funzione gradino” ed assumere che sia allineato all’asse y . L’espressione della divergenza si semplifica in

$$\operatorname{div}(c(x, y; t)\nabla I) = \frac{\partial}{\partial x}(c(x, y; t)\partial_x I) \quad (3.7)$$

e si scelga c come una funzione del gradiente

$$c(x, y; t) = g(\partial_x I(x, y; t)). \quad (3.8)$$

Sia $\phi(\partial_x I) := g(\partial_x I)\partial_x I$ il flusso $c \cdot \partial_x I$. La versione 1D dell’equazione di diffusione precedente diventa:

$$\partial_t I = \frac{\partial}{\partial x}(\phi(\partial_x I)) = \phi'(\partial_x I) \cdot \partial_{xx} I. \quad (3.9)$$

Ciò che interessa è la variazione nel tempo della pendenza dell'edge: $\partial_t(\partial_x I)$. Se $c(\cdot) > 0$, la funzione $I(\cdot)$ è regolare, dunque è possibile scambiare l'ordine di derivazione:

$$\frac{\partial}{\partial t} \left(\frac{\partial}{\partial x} I \right) = \frac{\partial}{\partial x} \left(\frac{\partial}{\partial t} I \right) = \frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} (\phi(\partial_x I)) \right) = \quad (3.10)$$

$$= \phi'' \cdot (\partial_{xx} I)^2 + \phi' \cdot \partial_{xxx} I. \quad (3.11)$$



Figura 3.2: Da sinistra a destra: immagine I_0 , derivate I, II e III

Posti $I_x = \partial_x I$ ed $I_t = \partial_t I$, suppone che l'edge sia orientato in modo tale che $I_x > 0$, allora nel punto di flesso $I_{xx} = 0$ ed $I_{xxx} \ll 0$, poichè esso corrisponde al punto di massima pendenza (fig. 3.2). Di conseguenza, in un intorno del punto di flesso, $\partial_t(I_x)$ ha segno opposto a quello di $\phi'(I_x)$. Se ne deduce che il processo di sfuocamento/miglioramento dell'edge 'e governato dal segno di $\phi'(I_x)$:

- se $\phi'(I_x) > 0$, ovvero (3.9) è parabolica all'avanti, l'edge subisce uno sfuocamento (la pendenza diminuisce nel tempo);
- se $\phi'(I_x) < 0$, ovvero (3.9) è parabolica all'indietro, l'edge viene migliorato (la pendenza aumenta nel tempo).

Equivalentemente, se ϕ è una funzione crescente, il bordo viene smussato, mentre se ϕ è decrescente, viene preservato. Si osservi che questo aumento della pendenza non è

causato da una scalatura dell'edge, poichè violerebbe il principio del massimo.

3.2.3 Alcune scelte del coefficiente di diffusione

Perona e Malik proposero due importanti esempi di g , tali che il flusso ϕ rispetti il criterio di causalità:

$$g(\|\nabla u\|) = e^{-(\|\nabla u\|/K)^2}, \quad (3.12)$$

$$g(\|\nabla u\|) = \frac{1}{1 + \left(\frac{\|\nabla u\|}{K}\right)^2}. \quad (3.13)$$

Queste due funzioni privilegiano zone diverse dell'immagine: la prima agisce soprattutto sugli edge ad alto contrasto, la seconda sulle zone interegione. La costante K viene fissata a mano, oppure usando uno stimatore del rumore, descritto da Canny in [15]: a questo proposito è utile calcolare un istogramma dei valori assoluti del gradiente dell'immagine. Ad esempio Canny propone di fissare K al 90% del suo valore per ogni iterazione.

In generale, è possibile scegliere $g(I_x) = 1/(1 + (I_x/K)^{1+\alpha})$, con $\alpha > 0$, ma è necessario tenere conto di una soglia dei parametri K ed α , sotto la quale ϕ è monotonicamente crescente ed oltre la quale decrescente, dove quest'ultima permette di regolarizzare le regioni interne e preservarne i bordi.

Si osservi che in un intorno del punto di massima pendenza di un bordo, ovvero il punto di flesso, l'equazione del modello è $u_t = \phi'(u_x)u_{xx}$, di tipo parabolico ed all'indietro, poichè $\phi'(I_x) < 0$. Questo fatto potrebbe destare qualche preoccupazione, poichè equazioni di diffusione all'indietro a coefficienti costanti, sono instabili ed amplificano il rumore, generando artefatti. Nel nostro caso tale preoccupazione è ingiustificata: il principio di massimo garantisce almeno formalmente, che questi non vengano prodotti, ovvero la validità del principio di causalità. Sperimentalmente si osserva che nelle aree dell'immagine in cui $\phi'(I_x) < 0$ il processo è molto veloce, mantenendosi numericamente stabile.

3.2.4 Scelta del parametro

Rimanendo nel caso $1D$, la funzione di flusso $\phi(s) := sg(s^2)$ soddisfa

$$\phi'(s) = \begin{cases} \geq 0, & \text{se } |s| \leq K \\ < 0, & \text{se } |s| > K \end{cases}$$

Si osserva che l'equazione di Perona-Malik (3.1) può essere riscritta come

$$\partial_t u = \phi'(u_x)u_{xx}, \quad (3.14)$$

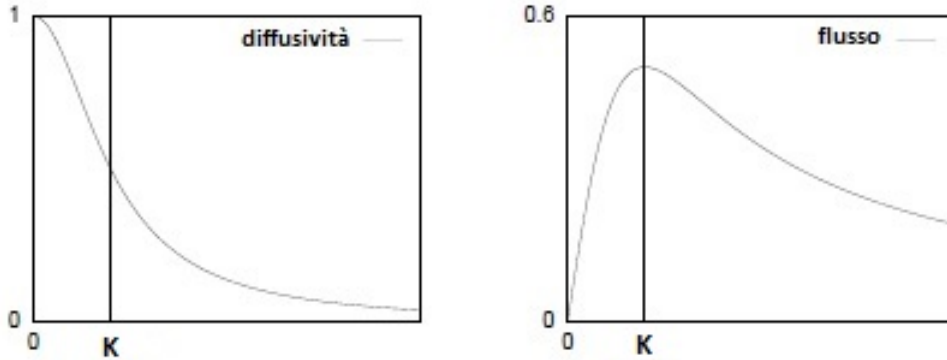


Figura 3.3: A sinistra: diffusività $g(s^2) = \frac{1}{1+s^2/K^2}$. A destra: flusso $\phi(s) = \frac{s}{1+s^2/K^2}$.

dunque il modello è parabolico all'avanti se $|u_x| \leq K$ ed all'indietro se $|u_x| > K$. Di conseguenza, K gioca un ruolo di parametro di contrasto, separando le aree di diffusione all'avanti (basso contrasto) da quelle all'indietro (alto contrasto).

Si può verificare che il filtro di Perona-Malik aumenta la pendenza intorno ai punti di flesso dell'edge entro un'area di diffusione all'indietro. Infatti, se esiste una soluzione u , del modello di Perona-Malik, sufficientemente regolare, allora essa soddisfa:

$$\partial_t(u_x^2) = 2u_x \partial_x(u_t) = 2\phi''(u_x)u_{xx}^2 + 2\phi'(u_x)u_x u_{xxx}. \quad (3.15)$$

In un punto x_0 , ad un certo tempo t , in cui u_x^2 ha massimo, è caratterizzata da $u_x u_{xx} = 0$ ed $u_x u_{xxx} \leq 0$. Di conseguenza,

$$(\partial_t(u_x^2))(x_0, t) \geq 0 \quad \text{se } |u_x(x_0, t)| \geq \lambda, \quad (3.16)$$

con $u_x u_{xxx} < 0$.

3.2.5 Mal-posizione del problema

Per i *problemi di minimo* e le *PDE* si danno le seguenti definizioni di ben-posizione e mal-posizione.

Definizione 3.1. Se un problema di minimo, oppure una PDE, ammette un soluzione, è unica e dipende in modo continuo dai dati, si dice il problema di minimo, o la PDE, è *ben-posto nel senso di Hadamard*.

Se viene violata anche solo una di queste tre condizioni (esistenza, unicità, continuità), si dice che il problema di minimo, o la PDE, è *mal-posto nel senso di Hadamard*.

Si ricorda che la principale scelta di $g(\cdot)$ per esaltare gli edge è

$$g(s) = \frac{1}{1 + \frac{s}{K}}, \quad (3.17)$$

da cui viene la malposizione del problema. Il motivo è il seguente.

Senza perdere di generalità si rimane nel caso $1D$. Sviluppando la divergenza dell'equazione di Perona-Malik si osserva che la derivata del coefficiente di diffusione (3.17) dà origine ad un termine che davanti al Laplaciano, che può essere positivo o negativo, in base a quanto mostrato in (3.2.4). Di conseguenza, l'argomentazione è riconducibile a studiare il problema del calore

$$\begin{cases} \partial_t u(x; t) = -u_{xx}(x; t), & \text{su } \mathbb{R}^2 \times]0, T[\\ u(x; 0) = u_0(x) \end{cases} \quad (3.18)$$

Si applica il cambiamento di variabile $\tau = T - t$, quindi, ogni volta che $u(x; t)$ è soluzione di (3.18), allora $v(x; \tau) = u(x - \tau; t)$ è soluzione di

$$\begin{cases} \partial_t v(x; \tau) = +u_{xx}(x; \tau), & \text{su } \mathbb{R}^2 \times]0, T[\\ v(x; T) = u_0(x) \end{cases} \quad (3.19)$$

che corrisponde all'equazione del calore, con il dato "all'indietro" $v(x; t) = u_0(x)$. Di conseguenza, se (3.18) ammette soluzione, vale lo stesso anche per (3.19). Tuttavia, a causa della regolarità dell'equazione del calore, necessariamente u_0 dovrebbe essere di classe C^∞ ma, non essendo così, si deduce che (3.18) non ha né una soluzione classica, né una debole. Tornando alla definizione (3.1), si ha che il problema di Perona-Malik è mal-posto.

3.3 Implementazione

Per implementare il metodo di Perona-Malik (3.3) si costruisce uno schema alle *differenze finite*. La discretizzazione viene fatta su un reticolo quadrato, con differenze all'avanti e all'indietro, le cui direzioni x ed y vengono suddivise nei sottointervalli rispettivamente Δx e Δy . Ogni pixel viene considerato come un quadrato di lato h e si pone $\Delta x = \Delta y = h = 1$.

I valori di intensità luminosa sono associati ai vertici, mentre i coefficienti di conduzione agli archi che collegano i vertici.

Una prima rappresentazione di (3.1) è la seguente:

$$\frac{I_{i,j}^{t+1} - I_{i,j}^t}{\Delta t} = c_{i,j}^t ((I_{i-1,j}^t - I_{i,j}^t) + I_{i+1,j}^t - I_{i,j}^t) + \quad (3.20)$$

$$+ c_{i,j}^t (I_{i,j-1}^t - I_{i,j}^t) + (I_{i,j+1}^t - I_{i,j}^t) + \quad (3.21)$$

$$+ \frac{\partial c}{\partial x} \left(\frac{I_{i+1,j}^t - I_{i,j}^t}{2} + \frac{I_{i-1,j}^t - I_{i,j}^t}{2} \right) + \quad (3.22)$$

$$+ \frac{\partial c}{\partial y} \left(\frac{I_{i,j+1}^t - I_{i,j}^t}{2} + \frac{I_{i,j-1}^t - I_{i,j}^t}{2} \right), \quad (3.23)$$

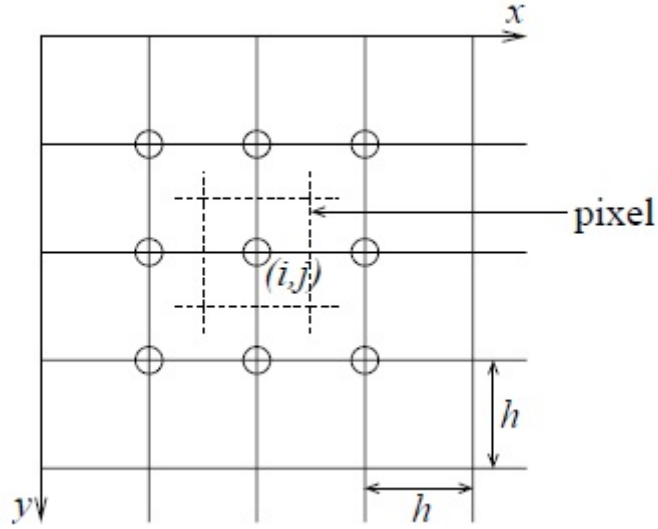


Figura 3.4: Posizionamento del pixel rispetto alla griglia

Considerando le approssimazioni (analogamente per j)

$$c_{i,j} + \frac{1}{2} \frac{\partial c}{\partial x} \approx c_{i+\frac{1}{2},j} \approx g \left(\frac{s_{i,j} + s_{i+1,j}}{2} \right), \quad (3.24)$$

$$c_{i,j} - \frac{1}{2} \frac{\partial c}{\partial x} \approx c_{i-\frac{1}{2},j} \approx g \left(\frac{s_{i,j} + s_{i-1,j}}{2} \right), \quad (3.25)$$

con $s_{i,j} := \|\nabla I(i,j)\|$, si ricava

$$\frac{I_{i,j}^{t+1} - I_{i,j}^t}{\Delta t} = g \left(\frac{s_{i,j} + s_{i-1,j}}{2} \right) (I_{i-1,j}^t - I_{i,j}^t) + \quad (3.26)$$

$$+ g \left(\frac{s_{i,j} + s_{i+1,j}}{2} \right) (I_{i+1,j}^t - I_{i,j}^t) + \quad (3.27)$$

$$+ g \left(\frac{s_{i,j} + s_{i,j-1}}{2} \right) (I_{i,j-1}^t - I_{i,j}^t) + \quad (3.28)$$

$$+ g \left(\frac{s_{i,j} + s_{i,j+1}}{2} \right) (I_{i,j+1}^t - I_{i,j}^t). \quad (3.29)$$

Calcolando $s_{i,j}$ tramite le differenze all'avanti, $s_{i+1,j}$ con le differenze all'indietro (analogamente per j) ed introducendo le notazioni

$$\nabla_N I_{i,j} \equiv I_{i,j-1} - I_{i,j}, \quad c_{N_{i,j}}^t = g \left(\|\nabla I\|_{i,j-1}^t \right), \quad (3.30)$$

$$\nabla_S I_{i,j} \equiv I_{i,j+1} - I_{i,j}, \quad c_{S_{i,j}}^t = g \left(\|\nabla I\|_{i,j+1}^t \right), \quad (3.31)$$

$$\nabla_E I_{i,j} \equiv I_{i-1,j} - I_{i,j}, \quad c_{E_{i,j}}^t = g \left(\|\nabla I\|_{i-1,j}^t \right), \quad (3.32)$$

$$\nabla_W I_{i,j} \equiv I_{i+1,j} - I_{i,j}, \quad c_{W_{i,j}}^t = g \left(\|\nabla I\|_{i+1,j}^t \right), \quad (3.33)$$

dove N, S, E, W rappresentano rispettivamente Nord, Sud, Est, Ovest, si ricava la formula discreta cercata:

$$I_{i,j}^{t+1} = I_{i,j}^t + \Delta t [c_N \cdot \nabla_N I + c_S \cdot \nabla_S I + c_E \cdot \nabla_E I + c_W \cdot \nabla_W I]_{i,j}. \quad (3.34)$$

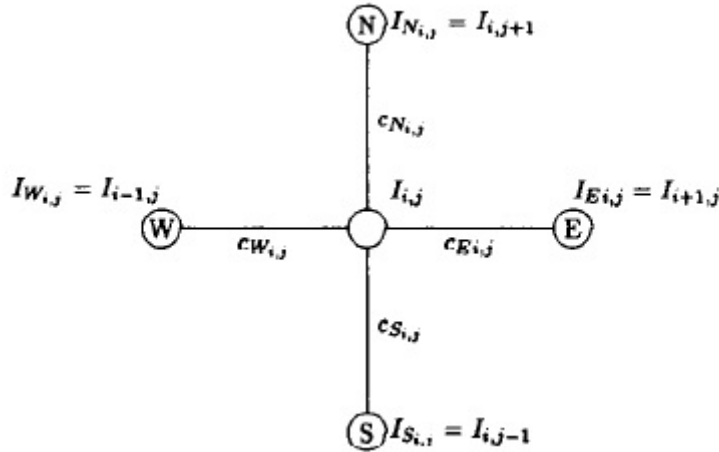


Figura 3.5: Schema di discretizzazione

Per garantire la stabilità numerica si assume $0 \leq \Delta t \leq 1/4$.

Osservazione. I coefficienti di conduzione c_N, c_S, c_E, c_W vengono aggiornati ad ogni iterazione, poichè dipendono dal gradiente della funzione immagine.

Si osservi che il valore del gradiente può essere calcolato su intorni con varie strutture, ottenendo così diversi “compromessi” tra accuratezza e localizzazione. Lo schema utilizzato è la scelta più semplice. Consiste nell'approssimare la norma del gradiente, nella posizione di ogni arco, con il valore assoluto della sua proiezione lungo la direzione

dell'arco. Tuttavia, non è l'esatta discretizzazione di (3.1), ma ha un'equazione di diffusione simile, in cui il tensore di conduzione è diagonale, con entrate $g(\|I_x\|)$ e $g(\|I_y\|)$ invece che $g(\|\nabla I\|)$ e $g(\|\nabla I\|)$. Ha il vantaggio di preservare le proprietà di continuità di (3.1) in modo che la luminosità totale dell'immagine venga mantenuta. Inoltre, il "flusso" di luce attraverso ogni arco dipende solo dal valore di luminosità dei due nodi che lo definiscono. È possibile verificare che, nonostante la scelta per l'approssimazione del gradiente, lo schema di discretizzazione soddisfa il principio del massimo (e del minimo) facendo in modo che la funzione g sia compresa tra 0 ed 1.

Si può dimostrare direttamente da (3.34), sfruttando il fatto che $\lambda \in [0, 1/4]$ e $c \in [0, 1]$, definendo

$$(I_M)_{i,j}^t := \max\{(I, I_N, I_S, I_E, I_W)_{i,j}^t\}, \quad (3.35)$$

$$(I_m)_{i,j}^t := \min\{(I, I_N, I_S, I_E, I_W)_{i,j}^t\} \quad (3.36)$$

il massimo ed il minimo dell'intorno di $I_{i,j}$ all'iterazione t . Si noti che

$$(I_m)_{i,j}^t \leq I_{i,j}^{t+1} \leq (I_M)_{i,j}^t \quad (3.37)$$

ovvero non ci sono massimi o minimi all'interno (of the discretized mult-scale). Infatti:

$$I_{i,j}^{t+1} = I_{i,h}^t + \lambda [c_N \cdot \nabla_N I + c_S \cdot \nabla_S I + c_E \cdot \nabla_E I + c_W \cdot \nabla_W I] \leq \quad (3.38)$$

$$\leq I_{M_{i,j}}^t (1 - \lambda(c_N + c_S + c_E + c_W)_{i,j}^t) + \lambda I_{i,j}^t (c_N + c_S + c_E + c_W)_{i,j}^t = \quad (3.39)$$

$$= I_{M_{i,j}}^t \quad (3.40)$$

ed analogamente

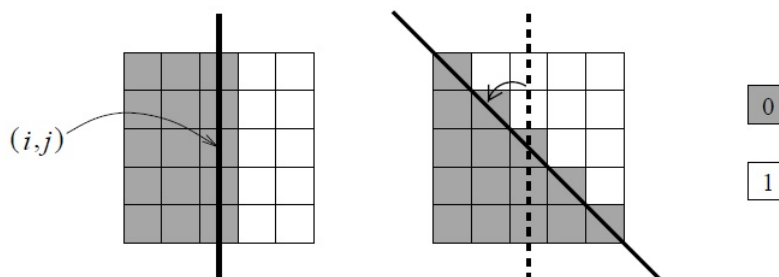
$$I_{i,j}^{t+1} \geq I_{m_{i,j}}^t (1 - \lambda(c_N + c_S + c_E + c_W)_{i,j}^t) + \lambda I_{i,j}^t (c_N + c_S + c_E + c_W)_{i,j}^t = I_{m_{i,j}}^t. \quad (3.41)$$

Lo schema numerico utilizzato da Perona e Malik è dato dalle equazioni numero ... , dove l'immagine originale rappresenta la condizione iniziale e le condizioni al bordo sono adiabatiche, ovvero ponendo il coefficiente di conduzione uguale a zero al bordo dell'immagine.

3.3.1 Miglioramento della griglia

Un limite dello schema alle differenze finite appena descritto e che non è invariante per rotazione, ovvero, ruotando l'edge si ottengono approssimazioni diverse. Considerando la figura seguente, si può verificare questa osservazione con un semplice calcolo algebrico.

Un modo immediato per potenziare lo schema alle differenze finite visto in precedenza è quello di discretizzare l'equazione di Perona-Malik su un intorno 3×3 , così da renderlo invariante per edge inclinati di $\pi/4$.

Figura 3.6: Edge inclinato di $\pi/4$

L'equazione (3.1) viene discretizzata anche nei punti che corrispondono alle diagonali dei pixel dell'intorno 3×3 . Di conseguenza, allo schema precedente vengono aggiunti

$$\nabla_{NE} I_{i,j} \equiv I_{i-1,j-1} - I_{i,j}, \quad c_{NE_{i,j}}^t = g \left(\left\| (\nabla I)_{i-1,j-1}^t \right\| \right), \quad (3.42)$$

$$\nabla_{SE} I_{i,j} \equiv I_{i-1,j+1} - I_{i,j}, \quad c_{SE_{i,j}}^t = g \left(\left\| (\nabla I)_{i-1,j+1}^t \right\| \right), \quad (3.43)$$

$$\nabla_{SW} I_{i,j} \equiv I_{i+1,j+1} - I_{i,j}, \quad c_{SW_{i,j}}^t = g \left(\left\| (\nabla I)_{i+1,j+1}^t \right\| \right), \quad (3.44)$$

$$\nabla_{NW} I_{i,j} \equiv I_{i+1,j-1} - I_{i,j}, \quad c_{NW_{i,j}}^t = g \left(\left\| (\nabla I)_{i+1,j-1}^t \right\| \right), \quad (3.45)$$

Muovendosi lungo le diagonali, gli intervalli $\Delta x = \Delta y = 1$ sono sostituiti da $\sqrt{2}$. Ora, l'equazione discreta (3.34) diventa:

$$I_{i,j}^{t+1} = I_{i,j}^t + \Delta t [c_N \cdot \nabla_N I + c_S \cdot \nabla_S I + c_E \cdot \nabla_E I + c_W \cdot \nabla_W I + \quad (3.46)$$

$$+ c_{NE} \cdot \nabla_{NE} I + c_{SE} \cdot \nabla_{SE} I + c_{SW} \cdot \nabla_{SW} I + c_{NW} \cdot \nabla_{NW} I]_{i,j}. \quad (3.47)$$

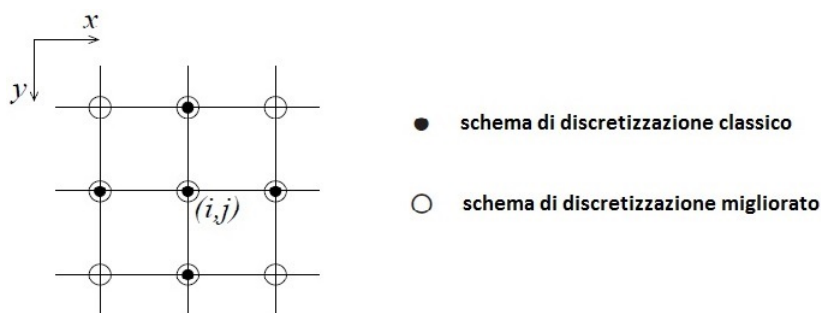


Figura 3.7: Confronto tra le due discretizzazioni

Capitolo 4

Metodi di Deblurring

Lo sfuocamento di un'immagine digitale durante il processo di acquisizione e può avere varie cause, ad esempio: il movimento della telecamera, dell'oggetto in esame, una scorretta "messa a fuoco" delle lenti, o una loro imperfezione. Per questi motivi è molto difficile da ridurre. In generale, produce degli artefatti nell'immagine che la degradano, rendendo più difficile la rilevazione di caratteristiche utili all'analisi.

Ci sono varie tecniche, dette di Deblurring, per ridurre gli effetti di questo fenomeno: il metodo di Richardson-Lucy [18], di Van Cittert [1], o di Landweber [22].

4.1 Esempi di Blur

Un'immagine che viene acquisita può avere un livello di sfuocamento più o meno alto, dovuto a varie cause, come quelle sopra citate. Prima di descrivere alcune importanti tecniche di deblurring, è utile ricordare che il modello di degradazione considerato è il seguente:

$$I = h * I_0 + \eta, \quad (4.1)$$

dove I_0 è l'immagine ideale, I quella degradata, h è l'operatore che causa lo sfuocamento, detto Point Spread Function (PSF), $*$ è un processo di convoluzione ed η è il rumore (solitamente Gaussiano) trattato nel capitolo precedente.

Alcune cause sono le seguenti.

Lo sfuocamento dovuto alla scorretta regolazione delle lenti viene detto "fuori fuoco" ed è rappresentato dalla funzione:

$$f(x, y) = \begin{cases} 0, & \text{se } \sqrt{x^2 + y^2} > r \\ \frac{1}{\pi r^2}, & \text{se } \sqrt{x^2 + y^2} \leq r \end{cases}$$

dove r è il raggio, in relazione con la dimensione della finestra di azione della PSF.

Un altro tipo di blur è dovuto al movimento dell'oggetto in esame o del sistema di acquisizione e viene classificato in due categorie: lineare e circolare.

Il più classico è il blur Gaussiano, descritto dalla convoluzione dell'immagine ideale con una funzione Gaussiana:

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (4.2)$$

dove x ed y rappresentano la distanza dell'oggetto dagli assi e σ la deviazione standard.

4.2 Tecniche di Deblurring

Il processo di deblurring consiste in una deconvoluzione dell'immagine con una PSF.

4.2.1 Metodo di Richardson-Lucy

È uno dei più conosciuti, poichè non dipende dal tipo di rumore ed è iterativo. L'equazione che governa questo metodo è la seguente:

$$I_0^{n+1} = I_0^n h \left(\frac{I}{h I_0^n} \right), \quad (4.3)$$

dove I_0^{n+1} è la nuova approssimazione dell'immagine originale, I è l'immagine affetta da blur, n è il numero di iterazioni ed h è la PSF. Purtroppo questo metodo richiede un tempo sempre maggiore all'aumentare del numero delle iterazioni.

4.2.2 Metodo di Van Cittert

È un metodo iterativo che riduce la differenza tra l'immagine ottenuta sottratta da quella stimata e quella acquisita. L'equazione che la descrive è:

$$I_0^{n+1} = I_0^n + (I - hI_0^n), \quad (4.4)$$

dove I_0^{n+1} , I_0^n , I , h ed n sono definite come prima.

Un punto di forza di questo metodo è che contiene delle operazioni matematiche semplici. Tuttavia, ha gli svantaggi di essere sensibile al rumore ed instabile dopo un certo numero limite di iterazioni; l'immagine potrebbe apparire "mossa".

4.2.3 Metodo di Landweber

Questa è una versione migliorata del metodo di Van Cittert. Rispetto al precedente, guadagna in stabilità e dà buoni risultati anche dopo un alto numero di iterazioni. Viene descritto dall'equazione:

$$I_0^{n+1} = I_0^n + \beta h (I - hI_0^n), \quad (4.5)$$

dove β è una costante che controlla l'aumento della qualità dell'immagine. Un limite è il suo lungo tempo di elaborazione.

4.2.4 Metodo di Poisson

È molto simile al metodo di Richardson-Lucy, con l'unica differenza che in questo caso l'equazione viene usata in forma esponenziale:

$$I_0^{n+1} = I_n \exp \left(h \left(\frac{I}{hI_0^n} - 1 \right) \right). \quad (4.6)$$

Ha un alto costo computazionale ed un lungo tempo di elaborazione a causa dell'esponenziale.

4.3 Metodo proposto

Le immagini che verranno analizzate sono affette da un tipo di blur che può essere modellato dalla convoluzione con una Gaussiana, dunque, si vedrà il fenomeno seguente:

Trattandosi di un'analisi di tipo subpixel, è più importante non perdere dettagli dell'immagine, piuttosto che ricostruirla ed eliminare il blur. La posizione dell'edge in un pixel viene codificata in termini di area sottostante (come mostrato nel primo capitolo), dunque il metodo proposto non compie un'effettiva azione di deblurring, ma tiene conto di questa codifica, calcolando delle aree parziali all'interno della finestra d'azione. A seconda del livello di rumore, può essere cambiata la dimensione della finestra, in modo che comprenda tutto il blur, quindi calcolando la posizione dell'edge più precisamente.

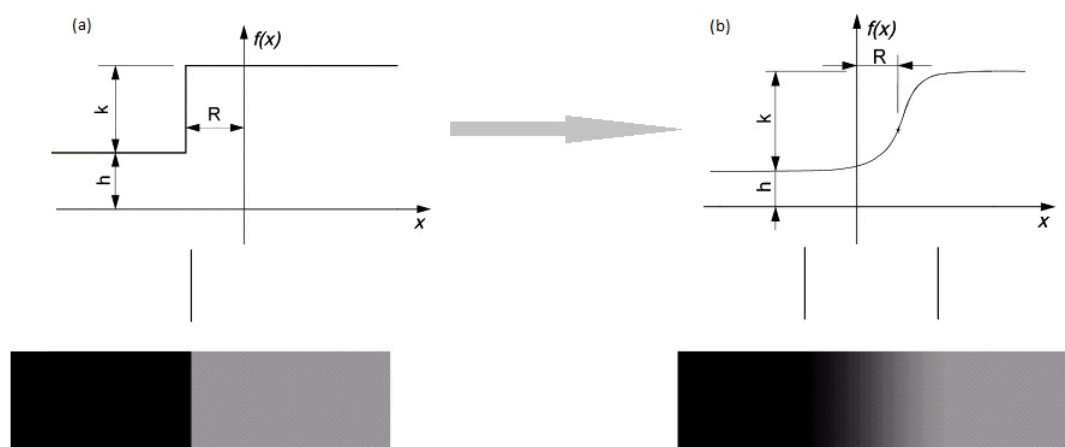


Figura 4.1: (a) edge di tipo gradino, (b) affetto da blur.

Capitolo 5

Subpixel Edge-Detection

I sistemi di Computer Vision vengono largamente usati nell'ambito dell'Image Processing e sfruttano le immagini digitali per analizzare, riconoscere e descrivere vari fenomeni di interesse. Questo tipo di analisi ha lo scopo di separare le informazioni utili da quelle complessive dell'immagine; in questo campo l'Edge-Detection gioca un ruolo fondamentale. In molte applicazioni, uno dei piú importanti criteri di analisi è proprio l'accuratezza della localizzazione dell'edge, ad esempio in campo biomedico, meccanico o nella misurazione delle proprietà di una superficie metallica, anche ad alta temperatura.

Nel corso degli anni sono stati proposti vari metodi di Edge-Detection. I primi si basano sul calcolo delle derivate discrete sull'immagine, ma danno risultati inaccurati e sono molto sensibili a rumore e sfuocamento. Inoltre, la loro precisione non supera quella del pixel. Attualmente, i sistemi di Computer Vision richiedono una precisione piú alta, del decimo, od addirittura del centesimo, di pixel. Di conseguenza, sono state sviluppate delle tecniche di Edge-Detection a livello subpixel, che sono uno degli aspetti principali dell'Image Processing.

5.1 Gruppi principali dei metodi subpixel

La struttura discreta delle immagini limita l'accuratezza dell'analisi. Un caso comune è quello in cui regioni diverse si intersecano all'interno di un pixel, dunque nell'immagine in output si ha un pixel con un'intensità che è la risultante delle intensità in un suo intorno. Questo problema può essere risolto aumentando la risoluzione dell'immagine, sebbene non possa essere completamente eliminato, come si può notare nella figura.

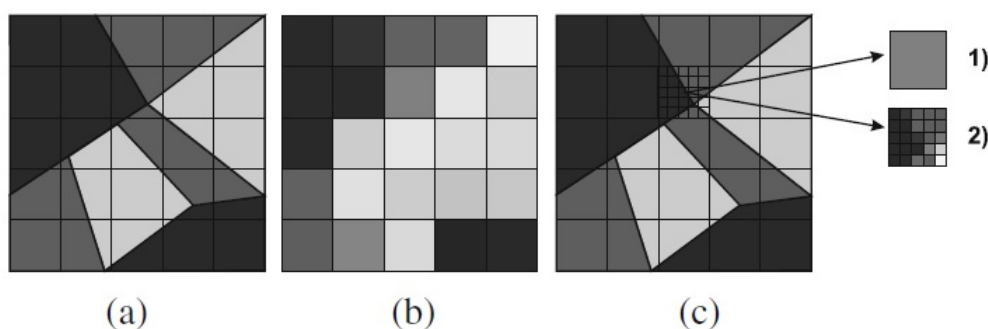


Figura 5.1: (a) immagine originale con una griglia di pixel, (b) immagine acquisita, (c) aumento della risoluzione della griglia.

I tradizionali algoritmi di Image Processing considerano il pixel come l'unità base ed indivisibile di un'immagine, dunque non sono in grado di stabilire se un pixel con intensità eterogenea appartiene a quante e quali regioni. Così nasce l'analisi Subpixel per l'Edge-Detection, con lo scopo di superare questi limiti e poter classificare un pixel. I metodi principali si possono classificare in tre gruppi:

- basati sull'*adattamento di curve*;
- basati sul *calcolo dei momenti*;
- di *ricostruzione*.

5.1.1 Metodi basati sull'adattamento di curve

Hanno lo scopo di costruire un bordo continuo adattando delle curve che passano per punti dell'edge, usando i tradizionali edge-detectors (si veda la figura seguente).

Questo metodo è stato usato da Yao e Ju (2009) [35], interpolando con una spline cubica dei dati spaziali, forniti dall'Operatore di Canny [15], o da Brender (2009) [27], con una B-spline interpolante. Un approccio simile era già stato fornito da Kisworo (1991) [25], ma in modo meno efficace. Tuttavia, l'accuratezza della curva costruita dal metodo dipende fortemente dall'accuratezza del bordo determinato a livello di pixel. Ne

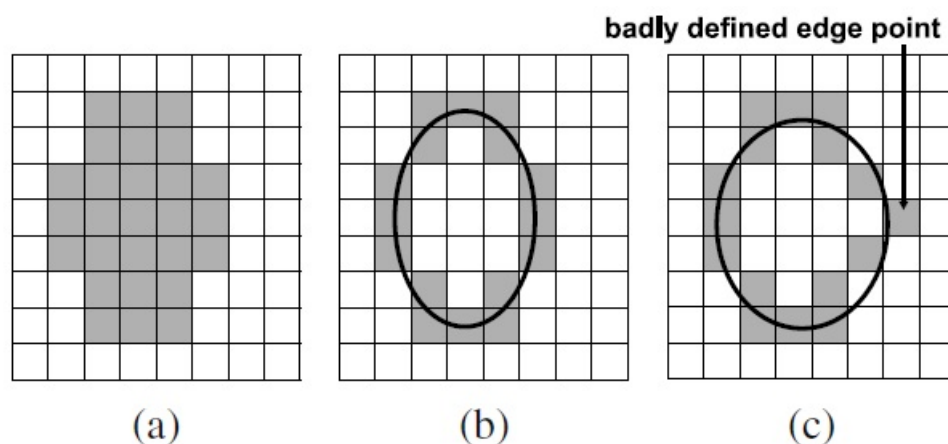


Figura 5.2: (a) immagine in input, (b) curva adattata in caso di una buona localizzazione dell'edge per i pixel, (c) in caso di una grossolana localizzazione dell'edge.

viene che questi metodi hanno successo nelle applicazioni in cui i bordi dell'oggetto in esame hanno forme semplici e si conoscono a priori.

5.1.2 Metodi basati sul calcolo dei momenti

In questo caso vengono sfruttati i momenti, che sono delle medie calcolate sulla base dei dati sperimentali e che possono essere suddivisi in due categorie:

- di *intensità*: basati sui valori di intensità dei pixel;
- *spaziali*: usano le informazioni sulla posizione in un intorno del pixel di interesse.

Il primo caso è stato trattato da Machuca e Gilbert (1981), in [28], integrando la regione che contiene l'edge e la posizione dell'edge viene determinata tramite i momenti calcolati sulla regione di intergrazione.

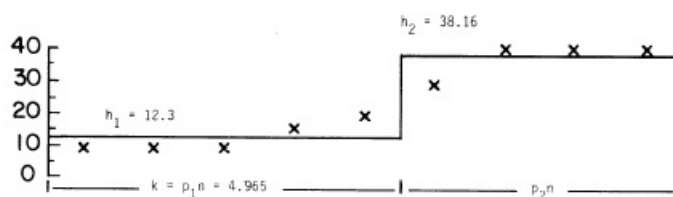


Figura 5.3: (a) immagine in input, (b) curva adattata in caso di una buona localizzazione dell'edge per i pixel, (c) in caso di una grossolana localizzazione dell'edge.

Tabatai e Mitchell (1984), in [5], hanno proposto un metodo di subpixel edge-detection che adatta i primi tre momenti ad un edge ideale di tipo gradino, definito come una

successione di valori di intensità h_1 seguita da un'altra successione con valore h_2 ad un'altezza diversa (si veda la figura).

La definizione dei momenti non tiene conto delle informazioni spaziali dell'immagine. Tuttavia, bisogna dire che questo metodo è in grado di localizzare solo edge definiti da successioni di intensità non decrescenti o non crescenti.

Invece, il metodo proposto da Lyvers (1989), in [?], usa i momenti di tipo spaziale nel modello 2D di edge ideale (si veda la figura). In questo modello l'edge è descritto da quattro parametri che ne determinano la posizione: lo sfondo h , il cambiamento di intensità k lungo l'edge, la transizione l e l'angolo formato rispetto all'asse y .

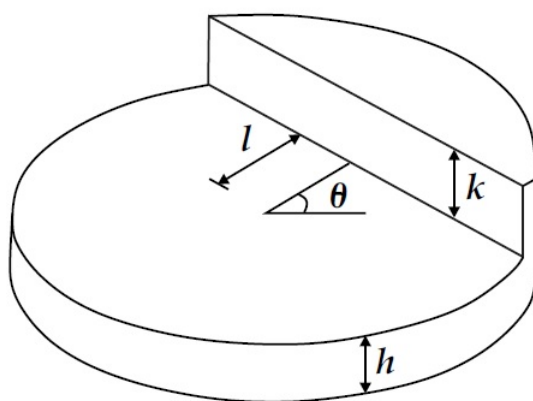


Figura 5.4: modello di edge di Lyvers

Di conseguenza, un edge ideale viene definito come una transizione da un'intensità h ad una $h + k$ e lo si considera all'interno di una circonferenza unitaria a distanza l dal centro. Per determinare la posizione subpixel, i pixel vengono mappati all'interno di una circonferenza unitaria e viene stabilita una relazione tra i momenti spaziali ed i parametri dell'edge.

Il metodo di Lyvers richiede il calcolo di sei momenti, usando una maschera circolare; questo comporta un costo computazionale molto alto. Inoltre, i momenti non sono ortogonali, dunque si verifica un'alta ridondanza nell'utilizzo delle informazioni dell'immagine. Ghosal e Mehrotra (1993), in [31], propongono un metodo che adatta dei momenti ortogonali al metodo di Lyvers, riducendo la complessità computazionale, poiché ne sono richiesti solo tre, ma aumentando la difficoltà nel localizzare oggetti di piccole dimensioni.

Il principale riscontro dei metodi basati sul calcolo dei momenti, di tipo intensità o spaziali, è la mancanza di un criterio per classificare un pixel come contenente, oppure no, un edge. Inoltre, questi metodi producono un nuovo set di parametri per ogni gruppo di pixel che contiene un cambiamento della posizione o nell'intensità ed agiscono solo in un piccolo intorno del pixel per cui passa l'edge.

5.1.3 Metodi basati sulla ricostruzione

Viene ricostruita la funzione che modella le intensità dell'immagine e grazie alle sue proprietà viene determinata la posizione subpixel dell'edge. Ne è un esempio l'approccio di Xu (2009), in [14], che approssima le intensità dell'immagine usando un polinomio di secondo grado. La localizzazione subpixel dell'edge è data dal punto della curva in cui l'intensità dell'immagine è uguale alla media delle intensità dello sfondo dell'oggetto. Tuttavia, i metodi di ricostruzione sono in minoranza rispetto ai due precedenti.

Alcuni di questi cercano di ricostruire la derivata prima dell'immagine in modo da ottenere una funzione continua del gradiente, basandosi sui dati discreti dell'immagine, forniti da operatori come quello di Sobel, Prewitt, o Canny [15]. Ne viene che le coordinate degli estremi della funzione gradiente ricostruita indicano la posizione subpixel dell'edge.

Altri metodi cercano di ricostruire la derivata seconda dell'immagine, con una funzione continua, basandosi sui dati discreti dell'immagine, forniti dal Laplaciano di una Gaussiana (LoG). L'annullamento della funzione delle derivate seconde determina la localizzazione dell'edge, con accuratezza subpixel (si veda la figura).

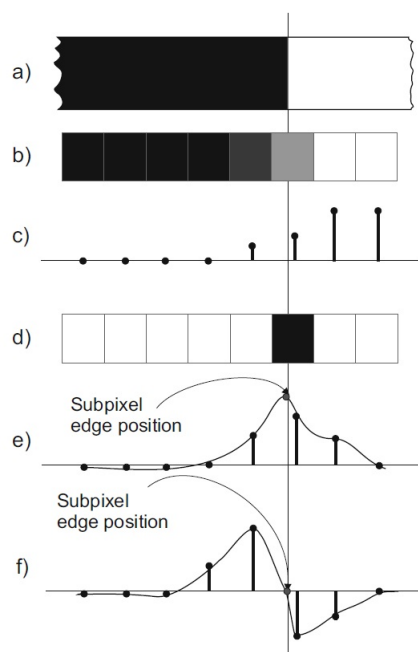


Figura 5.5: (a) immagine originale, (b) immagine acquisita, (c) intensità del profilo dell'edge, (d) posizione grossolana dell'edge, (e) determinazione della posizione subpixel dell'edge usando la derivata prima, (f) usando la derivata seconda.

Solitamente, la funzione delle derivate dell'immagine viene interpolata linearmente nell'intorno in cui le derivate seconde cambiano segno. Al fine di ridurre la complessità

computazionale, la funzione che modella l'immagine viene ricostruita solo in alcuni intornoi, i piú affetti da sfuocamento e rumore. Infatti, viene fatta un'analisi preliminare per una prima localizzazione grossolana degli edge e, solo in seguito, viene raffinata con una localizzazione subpixel dell'edge.

Un aspetto positivo di questi metodi è la loro robustezza alla mal-definizione di un punto dell'edge. Tuttavia, una buona localizzazione dell'edge è assicurata solo quando la funzione di ricostruzione agisce sull'immagine in un intorno dell'edge.

5.1.4 Metodo proposto

L'approccio proposto per la localizzazione subpixel dell'edge differisce da quelli descritti precedentemente nelle assunzione sull'immagine acquisita e dal modo in cui viene cercato l'edge. Invece di cercare il massimo del gradiente, oppure i punti in cui la derivata seconda si annulla, di un'ipotetica funzione continua che dovrebbe descrivere l'immagine, si assume una discontinuità del segnale in input, che viene identificata con l'edge.

L'algoritmo permette di localizzare l'edge con accuratezza subpixel, calcolando delle aree parziali prodotte dal processo di acquisizione dell'immagine.

Il capitolo successivo viene dedicato interamente alla descrizione di questo metodo.

Capitolo 6

Subpixel Edge-Detector basato sul calcolo di aree parziali

La stima delle caratteristiche dell'edge, come la posizione subpixel, l'orientazione, la curvatura ed il cambiamento di intensità da una parte all'altra dell'edge, tramite il calcolo del vettore gradiente, spesso sono inesatti.

Il metodo di *Edge-Detection* che verrà presentato si basa sul calcolo di aree parziali e sulla non assunzione di continuità tra i valori dell'immagine. Vengono considerati casi di edge rettilinei e curvilinei (ad es. semicirconferenze) in presenza di blur e/o noise, con lo scopo di estrarre la posizione dell'edge con una subpixel.

6.1 Acquisizione dell'immagine

In letteratura, spesso si assume che un'immagine digitalizzata F sia una versione campionata di una funzione $f = f(x, y)$ su una griglia rettangolare di pixel e, per poter applicare il calcolo differenziale sull'immagine, si considera f regolare.

Tuttavia, in questo caso le ipotesi saranno diverse, si assume che un edge sia una discontinuità nei valori di intensità della funzione f . Perciò, se un edge attraversa il pixel (i, j) , allora l'intensità di quel pixel sarà:

$$F_{i,j} = \frac{AS_A + BS_B}{h^2}, \quad (6.1)$$

dove A e B rappresentano l'intensità da una parte e dall'altra dell'edge, S_A ed S_B sono le aree che nel pixel (i, j) sono coperte rispettivamente dalle due intensità, ed h è la lunghezza del lato del pixel. I pixel verranno considerati quadrati di lato $h = 1$.

Come si vede dalla figura 6.1, il valore di intensità acquisito dal sensore luminoso è proporzionato all'area all'interno del pixel in entrambi i lati dell'edge.

Con lo scopo di stimare le caratteristiche dell'edge all'interno di un pixel, la maggior

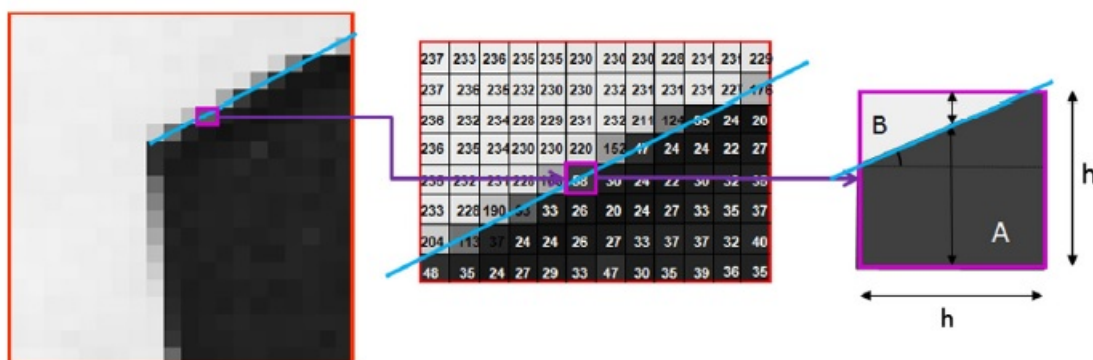


Figura 6.1: Valore di intensità in un pixel

parte dei metodi usano i valori del vettore gradiente calcolato nel pixel, come nei casi di Sobel e Prewitt. Questa analisi risulta insufficiente, poichè non è possibile raggiungere la precisione richiesta riguardo alla posizione dell'edge.

6.2 Analisi dell'errore nel calcolo del gradiente

Si consideri un'immagine ideale in cui è presente un edge con pendenza $1/2$, ovvero che attraversa ogni pixel dall'angolo in basso a sinistra a quello in alto a destra, come viene mostrato in figura.

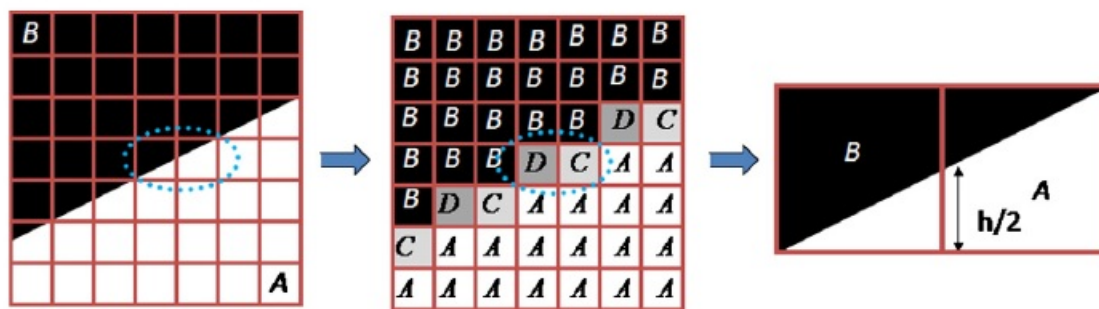


Figura 6.2: Intensità dei pixel rispetto alla posizione dell'edge

I pixel sottostanti e soprastanti l'edge hanno intensità rispettivamente A e B , mentre quelli per quali passa l'edge hanno intensità:

$$C = \frac{3A + B}{4} \quad \text{e} \quad D = \frac{A + 3B}{4}.$$

Si considerino le seguenti maschere come operatori per calcolare le derivate parziali rispetto ad x ed y :

$$M_x = \frac{1}{2h} \begin{pmatrix} -\alpha/2 & 0 & \alpha/2 \\ -(1-\alpha) & 0 & 1-\alpha \\ -\alpha/2 & 0 & \alpha/2 \end{pmatrix} \quad \text{ed} \quad M_y = \frac{1}{2h} \begin{pmatrix} -\alpha/2 & -(1-\alpha) & -\alpha/2 \\ 0 & 0 & 0 \\ \alpha/2 & (1-\alpha) & \alpha/2 \end{pmatrix},$$

dove il parametro α ha lo scopo di dare piú importanza alla colonna (nel caso di M_x) oppure alla riga (nel caso di M_y) centrale.

I valori delle derivate parziali per i pixel di tipo D sono i seguenti:

$$f_x = \frac{6-5\alpha}{16h}(A-B) \quad \text{ed} \quad f_y = \frac{8-\alpha}{16h}(A-B),$$

Si nota che l'orientazione del gradiente non è accurata e dipende fortemente dal valore di α , infatti:

$$\frac{f_x}{f_y} = \frac{6-5\alpha}{8-\alpha} \neq \frac{1}{2}.$$

Inoltre, la norma del vettore gradiente dovrebbe indicare la differenza tra le intensità A e B , ma non è così:

$$\|G\| = \frac{A-B}{16h} \sqrt[2]{26\alpha^2 - 76\alpha + 100} \neq A-B.$$

6.3 Subpixel-Detection di un edge di rettilineo

Si cominci considerando il caso piú semplice, in cui l'edge è rettilineo e con pendenza tra 0 ed 1. Per rappresentarlo si usa l'equazione $y = a + bx$, dove il sistema di coordinate è centrato nel pixel (i, j) , assumendo di non conoscere i valori di a, b, A, B .

Dall'ipotesi iniziale (6.1), il valore di intensità del pixel è dato da:

$$F_{i,j} = B + \frac{A-B}{h^2} P_{i,j}, \quad (6.2)$$

dove $P_{i,j}$ è l'area del pixel (i, j) sotto l'edge ($0 \leq P_{i,j} \leq h^2$).

Si consideri una finestra 5×3 centrata in (i, j) ; 5 righe sono necessarie per assicurarsi che l'edge attraversi la finestra da sinistra a destra.

Siano S_L, S_M ed S_R la somma delle intensità dei pixel nelle colonne di sinistra, quella

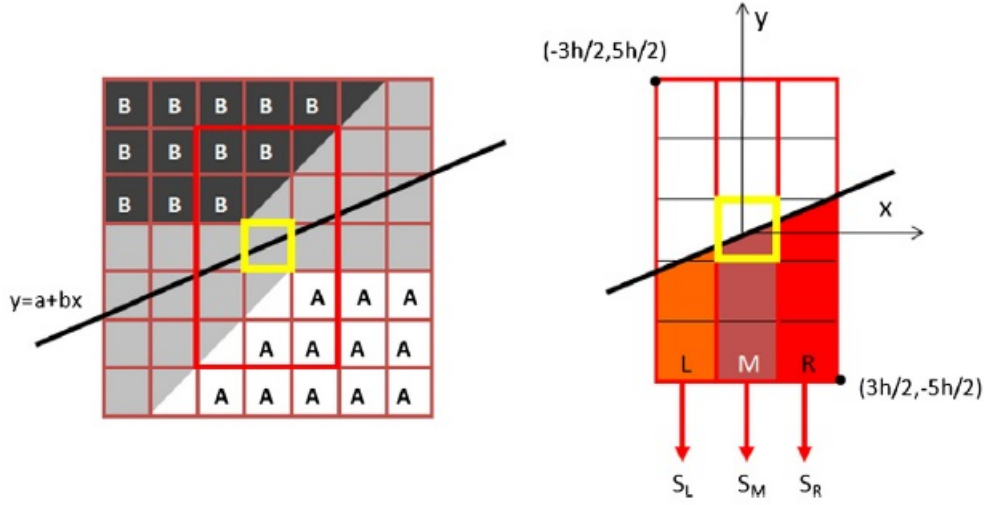


Figura 6.3: Edge rettilineo

centrale e di destra sono rispettivamente:

$$S_L = \sum_{n=j-2}^{j+2} F_{i-1,n} = 5B + \frac{A-B}{h^2}L, \quad (6.3)$$

$$S_M = \sum_{n=j-2}^{j+2} F_{i,n} = 5B + \frac{A-B}{h^2}M, \quad (6.4)$$

$$S_R = \sum_{n=j-2}^{j+2} F_{i+1,n} = 5B + \frac{A-B}{h^2}R. \quad (6.5)$$

Siano L , M ed R le aree sottese dall'edge all'interno delle tre colonne della finestra:

$$L = \int_{-3h/2}^{-h/2} (a + bx + 5h/2) dx = ah - bh^2 + \frac{5}{2}h^2, \quad (6.6)$$

$$M = \int_{-h/2}^{+h/2} (a + bx + 5h/2) dx = ah + \frac{5}{2}h^2, \quad (6.7)$$

$$R = \int_{h/2}^{3h/2} (a + bx + 5h/2) dx = ah + bh^2 + \frac{5}{2}h^2. \quad (6.8)$$

Dalle espressioni di S_M e di S_L, S_R si ottengono rispettivamente i coefficienti

$$a = \frac{2S_M - 5(A+B)}{2(A-B)}h, \quad b = \frac{S_R - S_L}{2(A-B)}.$$

Per stimare le intensità A e B si possono usare tre pixel negli angoli opposti alla finestra:

$$A = \frac{1}{3} (F_{i,j+2} + F_{i+1,j+2} + F_{i+1,j+1}), \quad B = \frac{1}{3} (F_{i-1,j-1} + F_{i-1,j-2} + F_{i,j-2}) \quad (6.9)$$

A questo punto, se ne determina la posizione subpixel dell'edge, misurando la distanza verticale dal centro del pixel: $(0, a)$.

Invece, il vettore normale all'edge è dato da:

$$N = \frac{A - B}{\sqrt{1 + b^2}} [b, -1], \quad (6.10)$$

la norma rappresenta il cambiamento di intensità da una parte all'altra dell'edge.

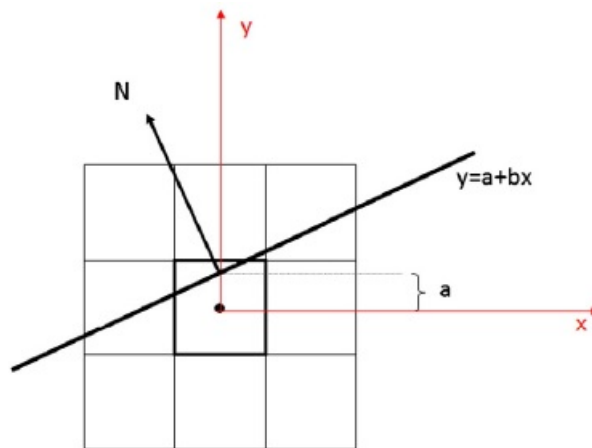


Figura 6.4: Localizzazione subpixel dell'edge

6.4 Subpixel-Detection di un edge di curvilineo

Spesso gli edge da trattare sono delle curve, di conseguenza per approssimarli si usa una funzione del tipo $f(x) = a + bx + cx^2$.

Si consideri ancora il caso in cui la pendenza in $x = 0$ è compresa tra 0 ed 1.

Come si è visto nella sezione precedente, si lavora con una finestra di dimensioni 5×3 , nonostante in alcuni casi non si abbia la certezza che l'edge attraversi la finestra da sinistra a destra. Il valore delle aree sottese dall'edge, nelle colonne di sinistra, quella

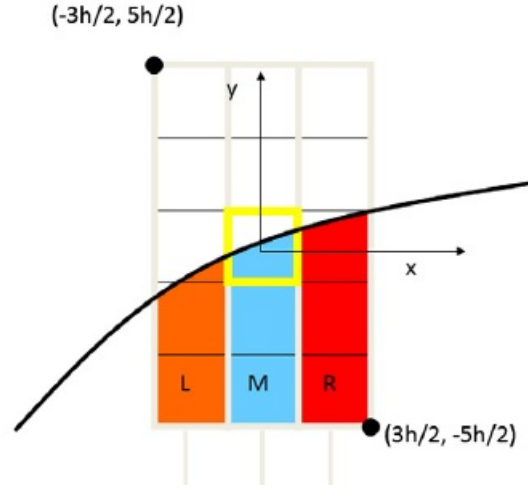


Figura 6.5: Localizzazione subpixel dell'edge

centrale e di destra sono rispettivamente:

$$L = \int_{-3h/2}^{-h/2} (a + bx + cx^2 + 5h/2) dx = ah - bh^2 + \frac{13}{12}ch^3 + \frac{5}{2}h^2 \quad (6.11)$$

$$M = \int_{-h/2}^{h/2} (a + bx + cx^2 + 5h/2) dx = ah + \frac{1}{12}ch^3 + \frac{5}{2}h^2 \quad (6.12)$$

$$R = \int_{h/2}^{+3h/2} (a + bx + cx^2 + 5h/2) dx = ah + bh^2 + \frac{13}{12}ch^3 + \frac{5}{2}h^2 \quad (6.13)$$

$$(6.14)$$

Analogamente al caso dell'edge rettilineo, grazie ad S_L , S_M ed S_R si ottiene un sistema di 3 equazioni in 3 incognite, le cui soluzioni sono i coefficienti della curva:

$$a = \frac{2S_M - 5(A + B)}{2(A - B)} - \frac{1}{12}c, \quad (6.15)$$

$$b = \frac{S_R - S_L}{2(A - B)}, \quad (6.16)$$

$$c = \frac{S_L + S_R - 2S_M}{2(A - B)}. \quad (6.17)$$

Si osservi che il cambiamento di intensità da una parte all'altra dell'edge ed il vettore normale, calcolati rispetto alla retta centrale verticale che passa per il pixel ($x = 0$), sono gli stessi del caso dell'edge lineare. Invece, la curvatura si stima:

$$K = \frac{2c}{(1 + b^2)^{3/2}}. \quad (6.18)$$

Il metodo descritto è piú generale di quello trattato nella sezione precedente. Può sempre essere applicato, nel caso di un edge lineare dà come risultato $c = 0$.

6.4.1 Generalizzazione a tutti gli ottanti

Se l'edge ha pendenza compresa tra -1 e 0 , l'espressioni di N rimane la stessa, ma S e B si calcolano usando angolo opposti della finestra.

In generale, le stime di A e B sono:

$$A = \frac{1}{3} (F_{i,j+2} + F_{i+m,j+2} + F_{i+m,j+1}), \quad (6.19)$$

$$B = \frac{1}{3} (F_{i-1,j-1} + F_{i-m,j-2} + F_{i,j-2}), \quad (6.20)$$

dove

$$m = \begin{cases} 1, & \text{se } f_x(i, j)f_y(i, j) > 0 \\ -1, & \text{se } f_x(i, j)f_y(i, j) < 0 \end{cases}$$

La curvatura K ha lo stesso segno del coefficiente c della parabola, ma è diverso quando la curva passa per i valori piú alti o piú bassi. Per esempio, data una circonferenza, con intensità A all'interno e B all'esterno, la curvatura è negativa in ogni punto dell'edge. Infatti, l'espressione generale della curvatura è:

$$K = \frac{2cn}{(1+b^2)^{3/2}}, \quad \text{con } n = \begin{cases} 1, & \text{se } f_y(i, j) > 0 \\ -1, & \text{se } f_y(i, j) < 0 \end{cases}$$

Se il valore assoluto della pendenza è maggiore di 1 , l'espressione è equivalente, ma verrà usata una finestra ruotata, cioè di dimensioni 3×5 , invece che 5×3 .

In questo caso la curva del secondo ordine che approssima l'edge è $f(y) = a + y + cy^2$, dunque la posizione subpixel viene misurata orizzontalmente dal centro del pixel e le coordinate saranno: $(a, 0)$.

Il vettore normale rimane lo stesso, ma le stime di A e B diventano:

$$A = \frac{1}{3} (F_{i+2,j} + F_{i+2,j+m} + F_{i+1,j+m})$$

$$B = \frac{1}{3} (F_{i-1,j-m} + F_{i-2,j-m} + F_{i-2,j}),$$

dove m è il medesimo del caso verticale.

Per poter applicare questo metodo, è necessario determinare per quali pixel passa l'edge ed il modo piú semplice è quello di calcolare il gradiente usando le tradizionali maschere di derivazione.

Si può dimostrare che in un'immagine ideale con un singolo edge, esso attraversa il pixel (i, j) con pendenza tra -1 ed 1 se

$$\begin{cases} |f_y(i, j)| > |f_x(i, j)| \\ |f_y(i, j-1)| \leq |f_y(i, j)| \geq |f_y(i, j+1)| \end{cases}$$

invece con pendenza maggiore di 1 se

$$\begin{cases} |f_x(i, j)| > |f_y(i, j)| \\ |f_x(i-1, j)| \leq |f_x(i, j)| \geq |f_x(i+1, j)| \end{cases}$$

6.5 Edge-Detection in condizioni di rumore

Il processo di acquisizione delle immagini introduce rumore, che altera i valori di intensità. Il modo tradizionale di ridurlo è quello di convolvere l'immagine con una maschera regolarizzante, ad esempio Gaussiano:

$$K = \begin{pmatrix} a_{11} & a_{01} & a_{11} \\ a_{01} & a_{00} & a_{01} \\ a_{11} & a_{01} & a_{11} \end{pmatrix}, \quad \text{con } a_{00} > a_{01} > a_{11} > 0 \text{ ed } a_{00} + 4a_{01} + 4a_{11} = 1.$$

Si pone G l'immagine ottenuta dalla convoluzione di quella acquisita F con K . Un problema di questo metodo è che la convoluzione con un kernel Gaussiano aumenta l'area coperta da pixel con valori intermedi tra A e B .

Si supponga che la pendenza dell'edge nel pixel (i, j) sia tra -1 ed 1 , dunque $|G_y(i, j)| > |G_x(i, j)|$, e sia m il segno della pendenza, dato da:

$$m = \begin{cases} 1, & \text{se } G_x(i, j)G_y(i, j) > 0 \\ -1, & \text{se } G_x(i, j)G_y(i, j) < 0 \end{cases}$$

Siano S_L , S_M ed S_R la somma delle intensità delle colonne di pixel nell'immagine G :

$$\begin{aligned} S_L &= \sum_{k=-3+m}^{3+m} G_{i-1, j+k} = (a_{00} + 2a_{01})(L_{-1} + A + B) + (a_{01} + 2a_{11})(L_{-2} + L_0 + 2(A + B)), \\ S_M &= \sum_{k=-3}^3 G_{i, j+k} = (a_{00} + 2a_{01})(M_0 + A + B) + (a_{01} + 2a_{11})(M_{-1} + M_1 + 2(A + B)), \\ S_R &= \sum_{k=-3-m}^{3-m} G_{i+1, j+k} = (a_{00} + 2a_{01})(R_1 + A + B) + (a_{01} + 2a_{11})(R_0 + R_2 + 2(A + B)), \end{aligned}$$

dove

$$L_i = \sum_{-2+m}^{2+m} F_{i-1,j+k} = A \frac{Q_i}{h^2} + B \left(5 - \frac{Q_i}{h^2} \right), \quad \forall i \in \{-2, -1, 0\}$$

$$M_i = \sum_{-2}^2 F_{i,j+k} = A \frac{P_i}{h^2} + B \left(5 - \frac{P_i}{h^2} \right), \quad \forall i \in \{-1, 0, 1\}$$

$$R_i = \sum_{-2-m}^{2-m} F_{i+1,j+k} = A \frac{T_i}{h^2} + B \left(5 - \frac{T_i}{h^2} \right), \quad \forall i \in \{0, 1, 2\}$$

e

$$Q_i = \int_{(i-\frac{1}{2})h}^{(i+\frac{1}{2})h} \left(a + bx + cx^2 + \frac{5+2m}{2}h \right) dx = ah + bih^2 + ch^3 \left(\frac{1}{12} + i^2 \right) + \frac{5+2m}{2}h^2$$

$$P_i = \int_{(i-\frac{1}{2})h}^{(i+\frac{1}{2})h} \left(a + bx + cx^2 + \frac{5}{2}h \right) dx = ah + bih^2 + ch^3 \left(\frac{1}{12} + i^2 \right) + \frac{5}{2}h^2$$

$$T_i = \int_{(i-\frac{1}{2})h}^{(i+\frac{1}{2})h} \left(a + bx + cx^2 + \frac{5-2m}{2}h \right) dx = ah + bih^2 + ch^3 \left(\frac{1}{12} + i^2 \right) + \frac{5-2m}{2}h^2.$$

Inoltre, la stima delle intensità dei pixel ai lati dell'edge è:

$$A = \frac{G_{i-1,j-3} + G_{i-m,j-4} + G_{i,j-4}}{3}, \quad B = \frac{G_{i,j+4} + G_{i+m,j+4} + G_{i+m,j+3}}{3}.$$

Si ricava un sistema di tre equazioni in tre incogne, le cui soluzioni sono i coefficienti della curva approssimante l'edge:

$$\begin{cases} a = \frac{2S_M - 7(A+B)}{2(A-B)} - \frac{1+24a_{01}+48a_{11}}{12}c, \\ b = m + \frac{S_R - S_L}{2(A-B)} \\ c = \begin{cases} 0, & \text{nel caso di un edge di I grado} \\ \frac{S_L + S_R - 2S_M}{2(A-B)} & \text{nel caso di un edge di II grado} \end{cases} \end{cases}$$

da cui si ottengono le caratteristiche subpixel dell'edge.

Osservazione 1. Nel caso in cui $|G_y(i, j)| < |G_x(i, j)|$ lo schema sarebbe simmetrico.

Vantaggi principali

Questo metodo ha vari punti di forza. Il principale è che in assenza di blur e noise le caratteristiche dell'edge vengono determinate con precisione. Inoltre, il metodo è molto robusto a differenti livelli di rumore.

Complessità computazionale

Siano N il numero totale dei pixel dell'immagine ed N_p il numero dei pixel in cui è stato individuato l'edge. Nella seguente tabella viene mostrata la complessità computazionale del metodo di edge-detection:

Tabella

Compl. comput. della fase di Edge-Detection

Fase di Edge-Detection	Addizioni	Moltiplicazioni
Regolarizzazione con una maschera 3×3	$8N$	$9N$
Stima delle derivate parziali	$2N$	$4N$
Individuazione del pixel dell'edge	$4N$	0
Calcolo delle finestre con dimensione variabile	$18N_p$	0
Somma di colonne	$21N_p$	0
Calcolo dei coefficienti di una curva	$18N_p$	$12N_p$
Totale	$14N + 53N_p$	$13N + 12N_p$

In conclusione, nella fase di Edge-Detection, il numero di addizioni e di moltiplicazioni è rispettivamente:

$$A = 14N + 53N_p \quad \text{ed} \quad M = 13N + 12N_p.$$

Capitolo 7

Il modello di Edge-Detection

In quest'ultimo capitolo viene presentato un *modello di subpixel edge-detection* in presenza di noise e blur e testato su alcune immagini sintetiche e reali.

7.1 Il modello di subpixel edge-detection

Viene presentato il modello di edge-detection. È stato scelto l'operatore di Perona e Malik per la parte di denoising. Esso è in grado di eliminare il rumore in maniera estremamente mirata, ovvero non altera le caratteristiche degli edge, al contrario le mantiene e le risalta. Inoltre, regolarizza le regioni interne. Si è preferito non effettuare un vero e proprio deblurring, per non alterare le caratteristiche dell'edge. Infine, è stato scelto un metodo di edge-detection, che opera calcolando delle aree parziali intorno all'edge, anche in presenza di blur.

Il modello è stato testato su immagini degradate, affette da blur e noise Gaussiani. Sono state analizzate delle immagini sintetiche con edge rettilinei e curvilinei. L'obiettivo è stato scegliere dei parametri sulle immagini sintetiche per applicarli a dei casi reali, di cui non si conosce la posizione esatta dell'edge, che presentato un livello di degradazione simile a quello applicato alle immagini sintetiche. In questo caso è stata studiata la ripetibilità del metodo scelto, conducendo degli esperimenti su un certo numero di campioni della stessa immagine. Ricavando la deviazione standard della posizione stimata per ogni pixel dell'edge sui campioni, si deduce il grado di concordanza tra le varie misure.

7.1.1 Scelta del coefficiente di diffusione del Perona-Malik

Una prima analisi è stata condotta, riguardo all'azione di denoising dell'operatore di Perona e Malik in presenza di noise, con lo scopo di studiare il parametro K del coefficiente di diffusione $g(s) = 1 / (1 + |s|^2 / K^2)$ dell'operatore.

Osservazione. Si danno alcune precisazioni sul significato di *noise* e *blur* Gaussiani. Dicendo che un'immagine è affetta da noise si intende che ai suoi valori di intensità è stata aggiunta una successione di numeri casuale e con distribuzione Gaussiana, con media μ e varianza σ^2 . Si dirà che il livello di noise è al 50% quando la deviazione standard è $\sigma = 50$.

Invece, si dice che un'immagine è affetta da blur Gaussiano di livello σ , quando è stata convoluta con una funzione Gaussiana

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{\|x\|^2}{2\sigma^2}}.$$

dove σ rappresenta la deviazione standard.

Si considera il cosiddetto *Fantoccio di Shepp-Logan* (fig. 7.1(a)) e vengono effettuati vari test in presenza di rumore (fig. 7.1(b)). Negli esperimenti i valori assegnati al parametro K sono stati $K = 5$ per l'immagine in (fig. 7.1(c)) e $K = 30$ per quella in (fig. 7.1(d)).

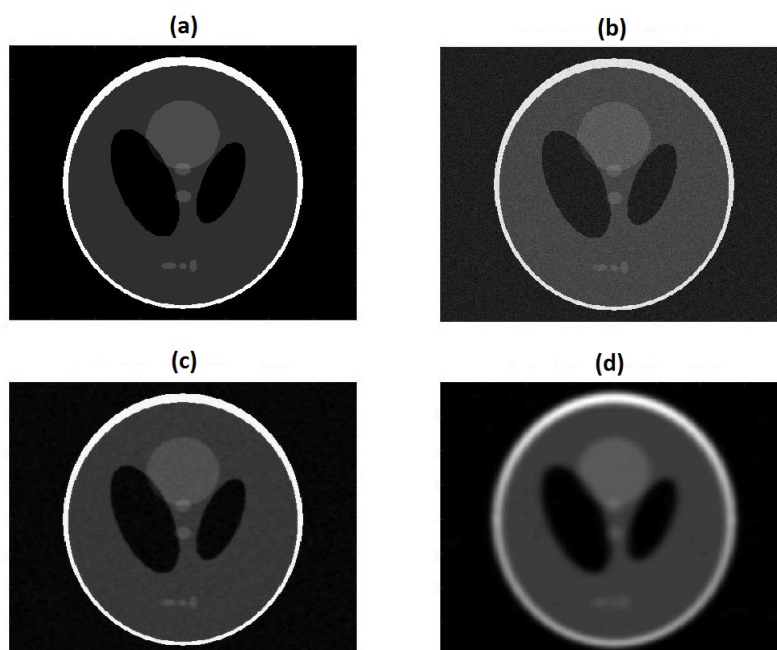


Figura 7.1: (a) fantoccio, (b) aggiunta di rumore al 30%, (c) denoised con parametro $K = 5$, (d) denoised con parametro $K = 30$

Questa prima analisi mostra l'importanza della scelta di questo parametro, poichè “pesa” la regolarizzazione. Infatti, a parità del numero delle iterazioni, che è stato fissato a 500 si nota che nel caso della figura (7.1(c)) ha un valore troppo alto rispetto al numero delle iterazioni effettuate, con la conseguenza della perdita delle caratteristiche dell'edge. Si intuisce così la relazione presente tra il valore del parametro, il numero di iterazioni e la quantità di rumore di cui è affetta l'immagine. Invece, dalla figura (7.1(d)) si nota che il valore di K era adeguato al numero di iterazioni e rispetto al rumore. Inoltre, si osservano le proprietà dell'operatore di Perona e Malik, descritte nei capitoli precedenti, ovvero viene ridotto il rumore, ma allo stesso tempo preservati gli edge. Si osservi inoltre che non sempre è sufficiente solo cambiare il numero di iterazioni, nel caso in cui K non sia adeguato.

Come è stato rilevato sul test del fantoccio di Shepp-Logan, la scelta del parametro K del coefficiente di diffusione dell'operatore di denoising è fondamentale. In particolare, nel corso della regolarizzazione, il livello di rumore diminuisce ad ogni iterazione, dunque è una buona idea fare in modo che K venga aggiornato.

Riprendendo il coefficiente di diffusione

$$g(s) = \frac{1}{1 + \frac{|s|^2}{K^2}}, \quad (7.1)$$

si osserva che il flusso $s \cdot g(s)$ ha derivata

$$\frac{d(s \cdot g(s))}{ds} = K^2 \frac{(K^2 - s^2)}{(K^2 + s^2)^2} \quad (7.2)$$

che si annulla in $s = K$, ovvero K è una stima del valore dell'immagine piú affetto da rumore. Dunque si è ritenuto consono considerare K come una stima della variabilità dei valori del gradiente dell'immagine, ovvero la deviazione standard.

In questo modo, nel corso della fase di denoising, il parametro si aggiorna automaticamente “pesando” l'azione della regolarizzazione lungo l'edge ad ogni iterazione, ovvero per ogni rappresentazione dell'immagine nell'analisi Multi-Scala. Di conseguenza, anche se il numero delle iterazioni non è quello ottimale, non si hanno degli effetti così degradanti come si è visto nella figura (7.1(d)).

7.1.2 Subpixel analysis

Verrà utilizzato il metodo di Subpixel Edge-Detector basato sul calcolo di aree parziali presentato nel capitolo VI. Questo è fondato sull'osservazione che se l'edge cade all'interno di un pixel, la sua posizione viene codificata tramite l'intensità dell'immagine, che è proporzionale alla misura dell'area all'interno del pixel in entrambi i lati dell'edge. Si verifica su alcune immagini, sia sintetiche sia reali, che il metodo è efficiente.

7.1.3 Sistema di acquisizione

Le immagini reali su cui sarà testato il modello sono state fornite da Marposs. Il loro sistema di acquisizione dell'immagine è formato da una sorgente che si ipotizza essere puntiforme da cui viene emessa la luce, la quale passa per una lente che la riflette sull'oggetto in esame. Dietro all'oggetto vi è un'altra lente che cattura la luce e la trasmette ad un apparecchio digitale che registra le informazioni che verranno elaborate (si veda la figura (7.2)).

Supponendo che le lenti siano ideali (ad es. non distorgano la direzione dei raggi luminosi) nella parte dell'immagine in cui vi è l'ingrandimento si dovrebbe ottenere un profilo netto, a forma di gradino, poichè questo è il tipo di edge che verrà studiato.

7.2 Test su edge rettilinei

In presenza di noise

Si presentano qui alcune prove di denoising con il funzionale di Perona-Malik. Una delle tecniche piú classiche di minimizzazione è la *Steepest Descent* del gradiente, ovvero si risolve un problema del tipo (2.1) mostrato nel capitolo III. Si ricorda che il metodo

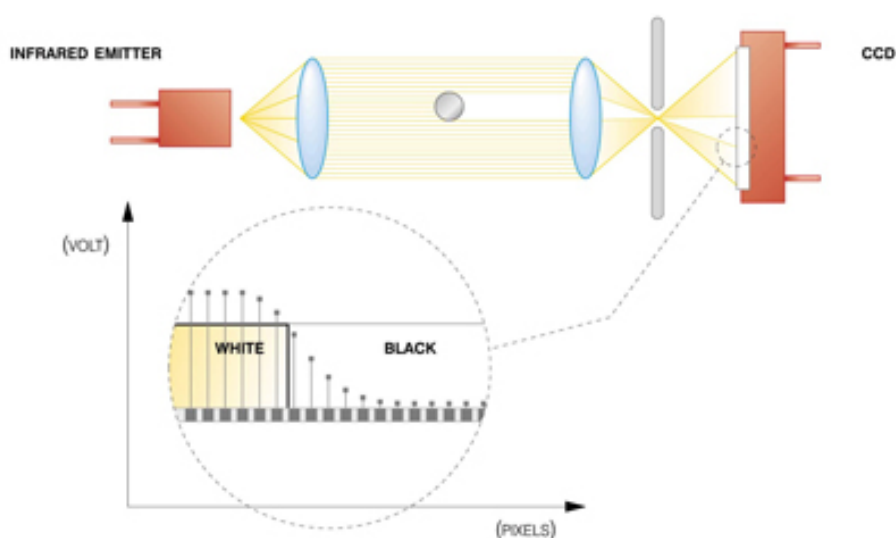


Figura 7.2: sistema di acquisizione

converge, ma per funzionare in modo efficiente, è necessario stabilire dei parametri di soglia. Inoltre, è un metodo sensibile alla convessità dell'operatore. Questi sono alcuni dei motivi per cui è stato necessario condurre un'analisi dei parametri sulla variazione dell'accuratezza e della ripetibilità del metodo proposto.

Gli edge studiati in questa sezione sono rettilinei, con inclinazione significative, a 20, 30, 45 e 90 gradi (si veda la figura (7.3) un esempio di immagine per ciascuno di questi angoli). Di seguito vengono mostrati nella situazione in cui sono affetti da rumore Gaussiano al 50%.

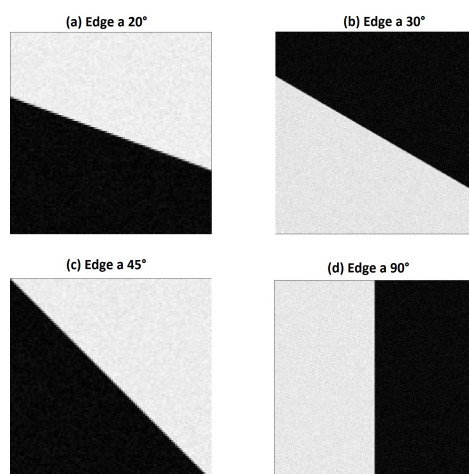


Figura 7.3: edge con rumore Gaussiano: (a) 20, (b) 30, (c) 45, (d) 90 gradi.

Volendo condurre un'analisi di tipo subpixel, è stata variata la posizione dell'edge all'interno del pixel, di un centesimo in centesimo, quindi sono state studiate un centinaio di immagini per ogni inclinazione dell'edge. Di seguito, (in figura 7.4) vengono mostrati gli istogrammi degli *errori massimi e minimi*, sull'asse Y , calcolati al *variare del centesimo della posizione* dell'edge, sull'asse X .

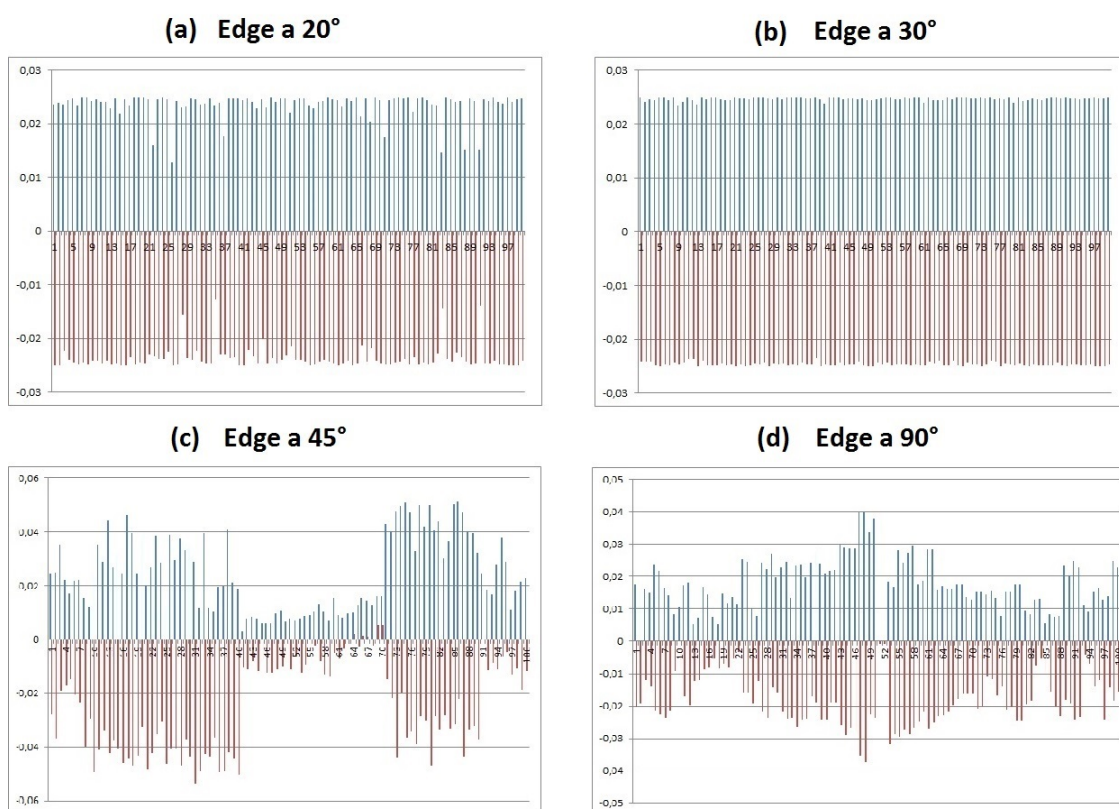


Figura 7.4: errore subpixel della posizione di edge inclinati a 20, 30, 45, 90 gradi, affetti da noise.

Si osserva dai grafici in figura (7.4(c)) e (7.4(d)), che la localizzazione della posizione è condizionata sia dall'inclinazione del bordo, sia dalla sua posizione all'interno del pixel. Infatti, come si può osservare dalla figura, quando l'edge si trova molto vicino al confine tra un pixel e l'altro, è più difficile localizzarlo. Come si vede dai grafici, questo problema non si pone nel caso in cui l'edge sia proprio sul confine tra due pixel. Tuttavia, la localizzazione è abbastanza precisa, poichè l'errore oscilla in un intorno di circa 4 centesimi di pixel ed in condizioni favorevoli anche di 2 centesimi di pixel.

Relazione tra il numero delle iterazioni e la posizione dell'edge

Volendo analizzare sperimentalmente il comportamento del metodo e conoscendo l'immagine ideale di riferimento, per stabilire il numero ideale di iterazioni, sono state imposte le condizioni per cui il valore assoluto del valore medio dell'errore dell'iterazione successiva fosse minore di quella precedente e che diminuisse la deviazione standard. Queste due condizioni sono state considerate simultaneamente, poichè il valore medio dell'errore può rimanere nullo, o molto vicino allo zero, ma aumentare l'oscillazione delle posizioni subpixel calcolate dell'edge. Al contrario, la deviazione standard può diminuire con l'aumentare delle iterazioni, ma cambiare il valore medio dell'errore, che corrisponde ad uno "spostamento" dell'edge, situazione che si verifica quando l'immagine viene regolarizzata eccessivamente. In seguito è stata simulata una condizione reale in cui non conosciamo a priori la posizione dell'edge e l'errore non è stato calcolato rispetto alla posizione ideale dell'edge, ma rispetto alla stima al passo precedente.

Di seguito, in figura (7.5), viene mostrato l'andamento delle iterazioni dell'edge inclinato a 90 gradi al variare dei un centesimo di pixel della posizione:

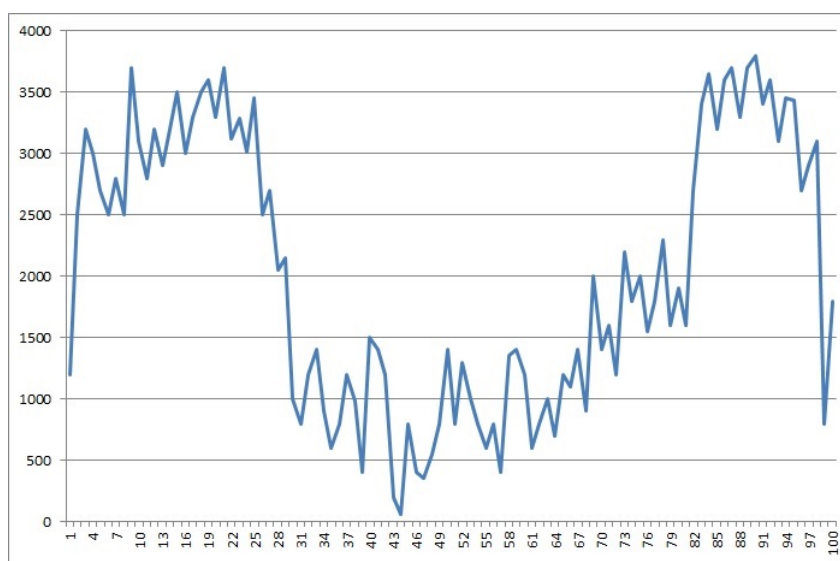


Figura 7.5: andamento del numero di iterazioni, al variare di un centesimo di subpixel della posizione di un edge inclinato a 90 gradi.

Questo grafico corrisponde a (7.3d) ed è quello più interessante da osservare, poichè l'edge ha la stessa posizione in tutti i pixel per cui passa. Questo mostra esplicitamente la dipendenza della localizzazione con la posizione subpixel, poichè l'edge ha la stessa posizione in ogni pixel.

Si nota che quando l'edge si trova in prossimità del confine tra due pixel, diminuisce il numero di iterazioni ottimale, ma l'edge viene localizzato con minore precisione rispetto

alle altre posizioni. È anche interessante notare che nel caso in cui l'edge sia esattamente sul confine tra due pixel, allora viene localizzato con precisione alta ed un numero molto basso di iterazioni. Quest'ultimo fenomeno probabilmente è dovuto al fatto che, siccome l'edge non si trova effettivamente all'interno di un pixel, dunque i valori dei pixel circostanti sono condizionati solo dal rumore, rendendo più facile l'edge-detection.

In presenza di noise e blur

In figura (7.6) è sono state considerate delle immagini con le stesse inclinazioni di bordi a 20° , 30° , 45° , 90° che nella figura (??), e con lo stesso livello di noise al 50%, ma affette da blur di livello $\sigma = 3$.

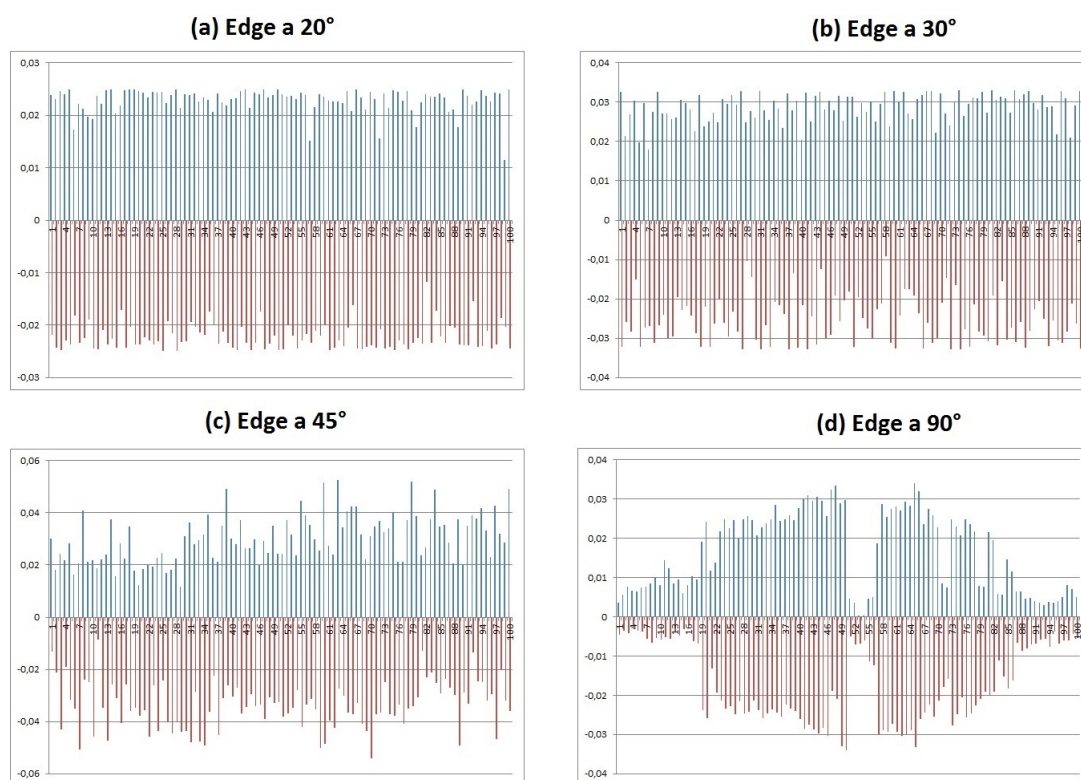


Figura 7.6: errore subpixel della posizione di edge inclinati a 20, 30, 45, 90 gradi, affette noise e blur.

È interessante osservare che, essendo l'immagine affetta da blur Gaussiano, l'edge viene localizzato con una precisione subpixel inferiore, ma molto vicina al caso precedente in cui era presente solo il noise. Questo risultato mostra che il metodo reagisce bene anche in situazioni di edge affetto da blur.

Analisi di edge rettilinei senza denoising

I grafici in figura (7.7) e (7.7), sono il risultato dell'applicazione del modello di deblurring ed edge-detection, senza il preprocessing mediante l'operatore di Perona Malik. In figura (7.7) si considerano edge con inclinazioni 20° , 30° , 45° , 90° , come nelle due sezioni precedenti e con il solo noise, mentre in figura 7.8 le stesse inclinazioni sono studiate con noise e blur.

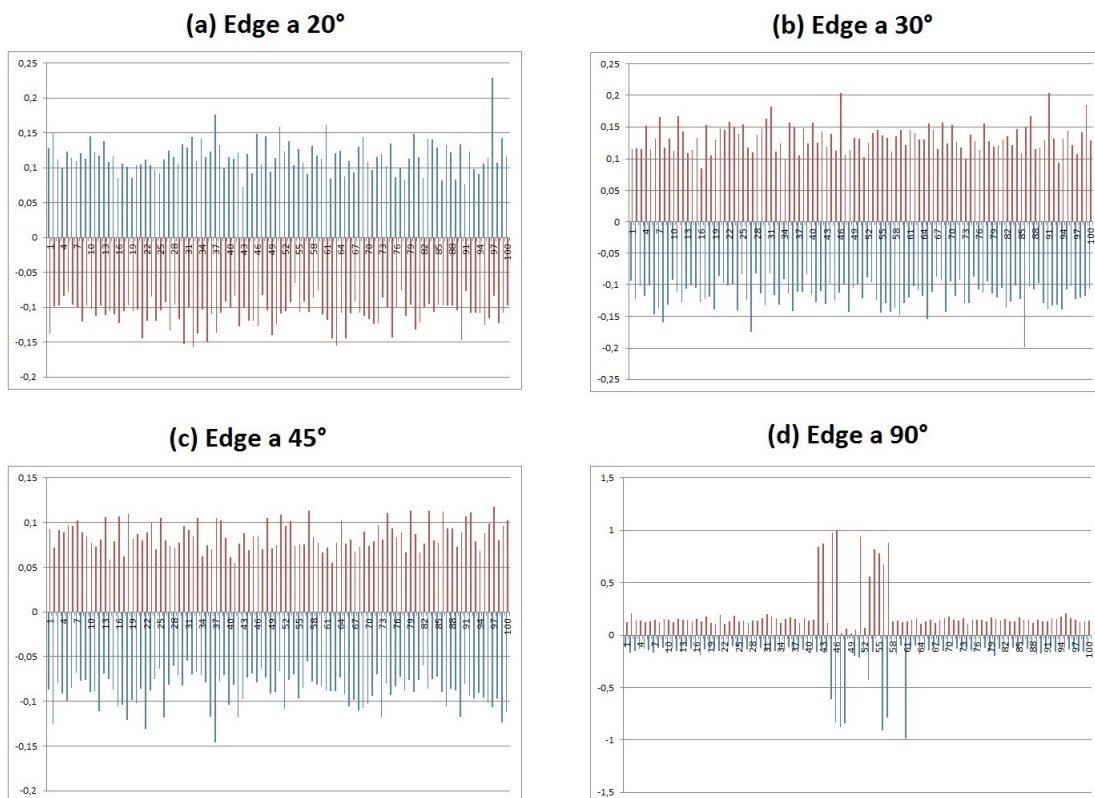


Figura 7.7: (a)-(d) edge detection, senza la fase di denoising, di edge inclinati a 20° , 30° , 45° , 90° , in presenza di noise.

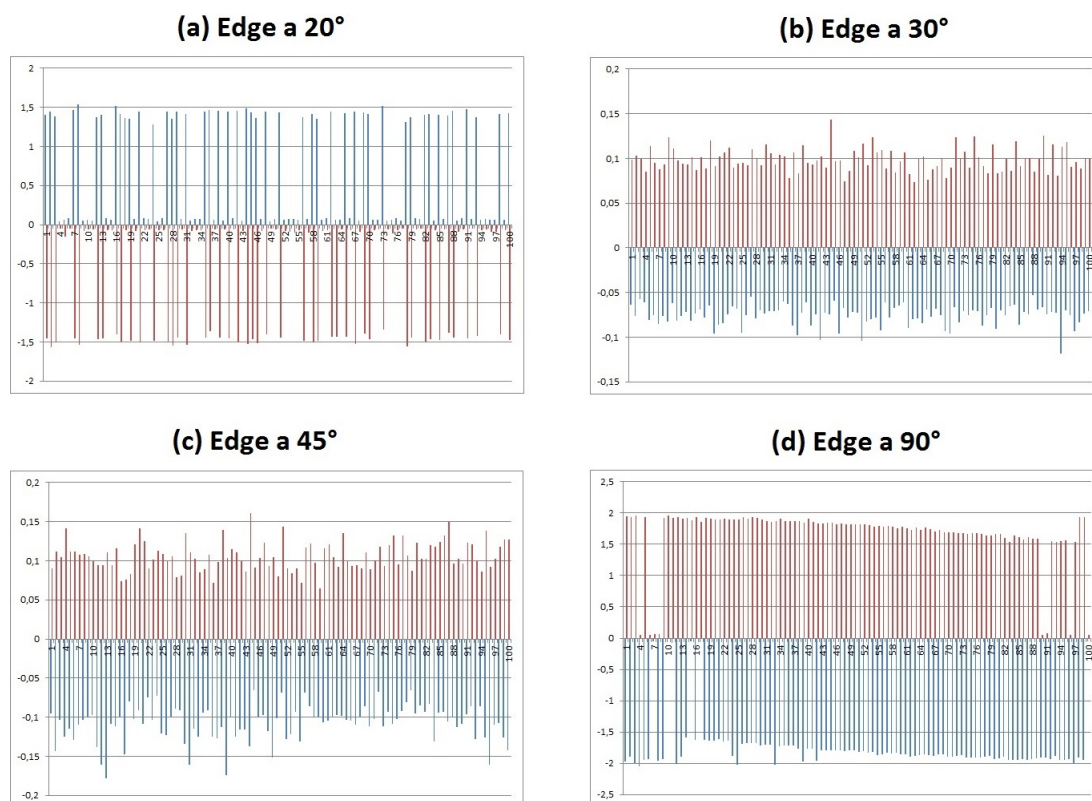


Figura 7.8: (a)-(d) edge detection, senza la fase di denoising, di edge inclinati a 20° , 30° , 45° , 90° , in presenza di noise e blur.

I risultati sono evidentemente peggiori. Infatti, non a caso è stato scelto proprio questo l'operatore di Perona e Malik, poichè fortemente dipendente dal gradiente, dunque dotato di una grande capacità non solo di mantenere inalterate le caratteristiche dei contorni, ma anche di esaltarle. Questo migliora notevolmente l'edge-detection subpixel.

7.2.1 Test su immagini reali

Il test su immagini reali state eseguite su un'immagine che rappresenta il contorno di un utensile posizionato verticalmente, probabilmente inclinato di qualche grado rispetto all'asse verticale, si veda la figura (7.9).

Dalla figura (7.10) si deduce che l'edge si trova nel 14esimo pixel lungo l'asse X e la posizione stimata è circa 14.9 con l'incertezza di un decimo di pixel.

Poichè le immagini reali possono presentare caratteristiche diverse e una minore regolarità rispetto a quelle ideali, è stato testato ulteriormente il metodo verificandone la ripetibilità su 100 immagini diverse del medesimo oggetto. Il metodo è stato applicato senza preprocessing con il Perona-Malik e la deviazione standard per ogni pixel della

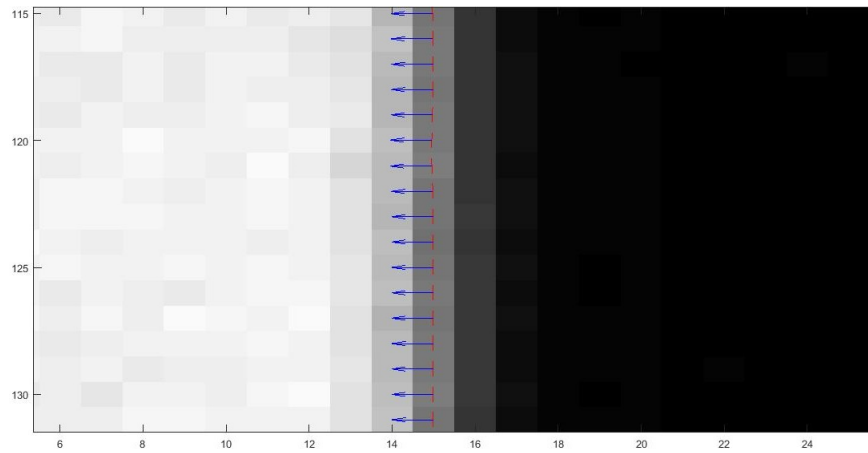


Figura 7.9: edge detected (le frecce di colore blu indicano la direzione ortogonale al bordo)

posizione stimata dell'edge sui cento campioni è mostrato in figura 7.11 (a). La deviazione standard è stata poi ricalcolata utilizzando l'algoritmo proposto contenente anche il denoising, come mostra la figura (7.11(b)).

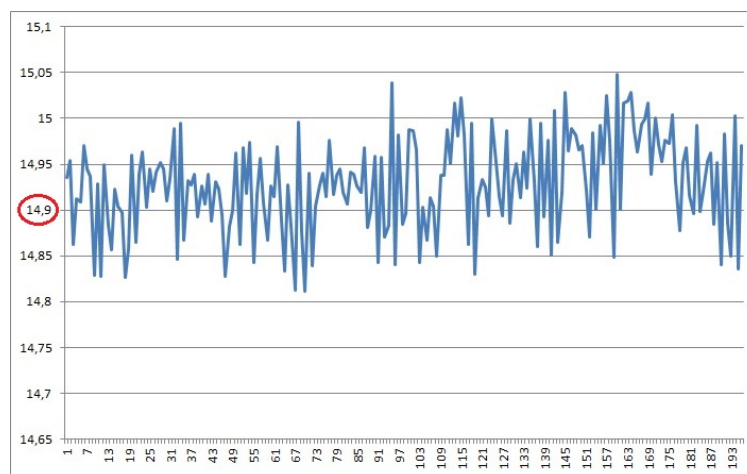


Figura 7.10: posizione stimata dell'edge con precisione del decimo di subpixel. A sinistra il metodo è applicato senza algoritmo di denoising, mentre a destra il denoising viene applicato.

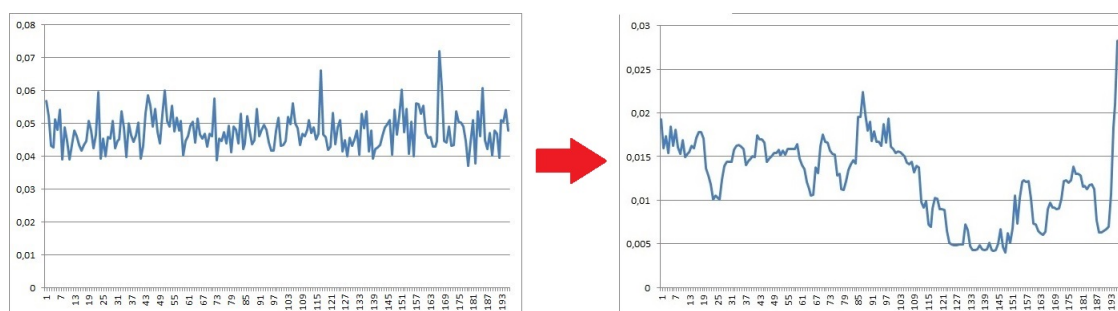


Figura 7.11: deviazione standard per ogni pixel della posizione stimata dell'edge sui cento campioni.

Si osserva che la fase di denoising è molto rilevante per ridurre l'intervallo di incertezza con cui viene localizzato l'edge.

7.2.2 In caso di angoli

Dopo aver analizzato immagini con edge rettilinei, si vuole trattare il caso in cui l'immagine raffiguri un oggetto, in presenza di noise e blur, con degli angoli.

Si osservi la figura (7.12):

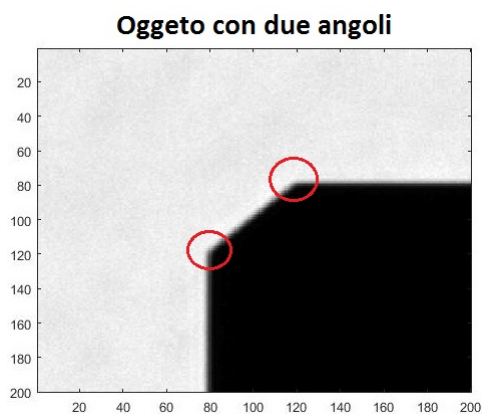


Figura 7.12: oggetto con due angoli, affetta da rumore e blur

Il livello di rumore è ancora al 50%, dunque si tiene conto dell'analisi dei parametri fatta in precedenza riguardo al numero di iterazioni della fase di denoising dell'operatore di Perona-Malik. Si considerano 3000 iterazioni, coefficiente di stabilità $\lambda = 1/4$, intervallo temporale $\Delta t = 0.1$. In seguito, è stata eseguita l'edge-detection per verificare la localizzazione negli spigoli. La figura (7.13) ne mostra il risultato:

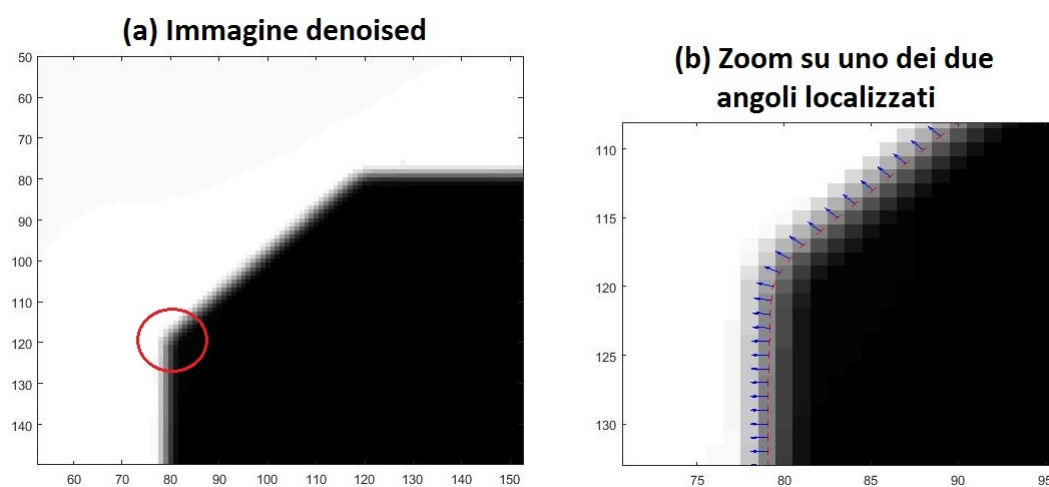


Figura 7.13: L'oggetto con due angoli che compare in figura (7.13) viene qui ingrandito (a), e vengono indicate con un vettore le direzioni determinate del contorno.

Si può osservare che dopo aver ridotto il rumore dell'immagine, la presenza degli angoli non complica la localizzazione del contorno. Purtroppo essendo un'immagine reale, non si può confrontare la posizione subpixel calcolata con quella effettiva.

7.3 Test su circonferenze

Analisi preliminare

In questa sezione vengono mostrati alcuni risultati dei test eseguiti su delle circonferenze, con lo scopo di comprendere il comportamento del metodo su edge curvilinei.

In prima analisi è stato testato il metodo di edge-detection su una circonferenza sintetica di raggio 50, in assenza sia di noise che di blur, e ne è stato localizzato il contorno, come si vede dalla figura (7.14):

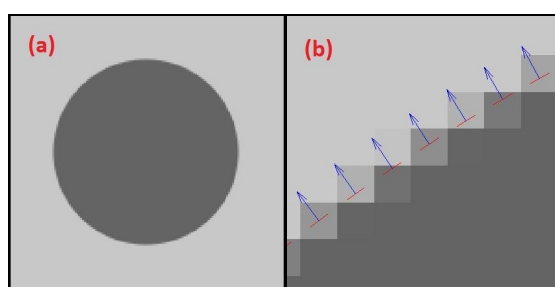


Figura 7.14: Edge-Detection in assenza di rumore (le frecce blu rappresentano la direzione ortogonale all'edge).

Per avere un riscontro pratico dell'accuratezza subpixel, sono stati calcolati raggi per ogni pixel facendo riferimento alla posizione stimata della circonferenza e confrontati con il raggio esatto. Si può vedere nella figura (7.15) che la stima cambia al variare dell'angolo lungo il quale il raggio è stimato. In particolare, per la simmetria dell'immagine la stima dei raggi su ogni quadrante è la medesima, mentre l'errore nella stima del raggio è minima quando il raggio è orizzontale o verticale.

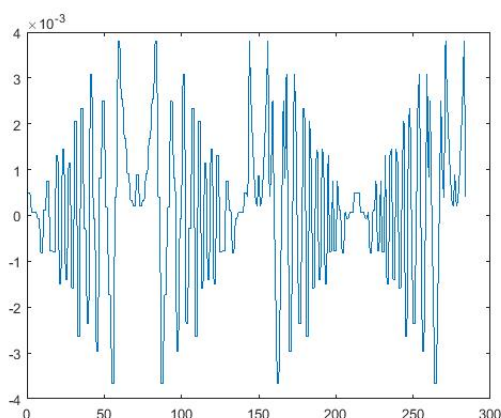


Figura 7.15: raggi calcolati di una circonferenza in assenza di rumore e sfuocamento.

In seguito, sono state esaminate 100 circonferenze con raggio che varia da 100 a 100.99 con un intervallo del centesimo di pixel.

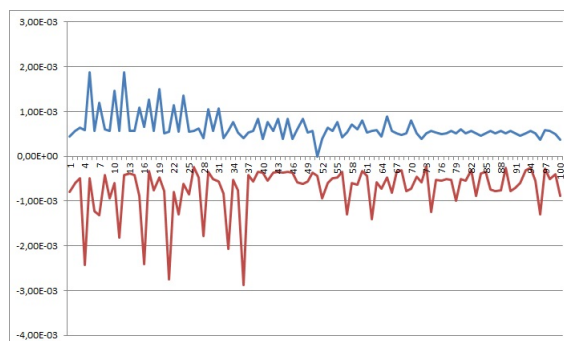


Figura 7.16: errore sui raggi che variano di un centesimo di pixel , da 100 a 100.99, in assenza di noise e blur.

Il grafico (7.16) mostra sull'asse Y l'*errore massimo* (in blu) e l'*errore minimo* (in rosso) dei raggi calcolati, al variare della lunghezza del raggio per ciascuno dei 100 cerchi considerati, sull'asse X . Dal grafico si osserva che vi è una zona in cui l'errore è maggiore. Questo probabilmente è dovuto al fatto che con certe lunghezze del raggio, l'edge si trova in posizioni più difficili da stimare, ad esempio in prossimità del confine tra due pixel. Tuttavia, gli errori sono molto bassi, del millesimo su subpixel.

Analisi del blur

La stessa analisi è stata condotta nel caso di circonferenze con blur di livello $\sigma = 1$ e $\sigma = 3$, da cui si sono ricavati i risultati seguenti:

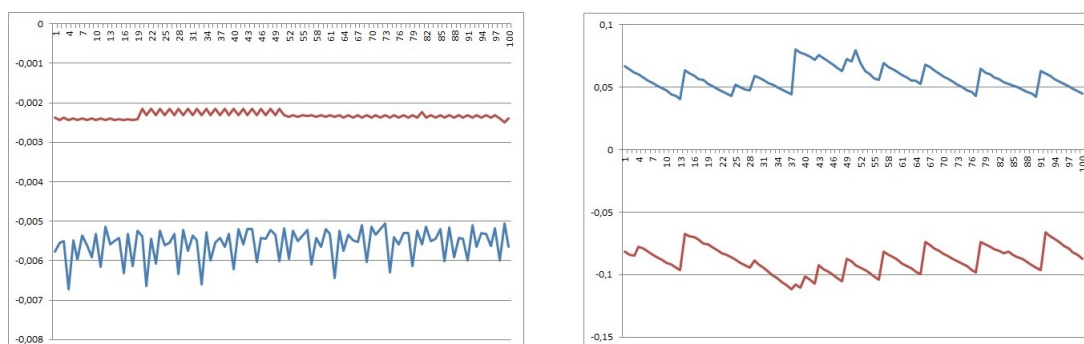


Figura 7.17: errore sui raggi al variare della posizione dell'edge di un centesimo di pixel, in assenza di noise, ma con blur di parametro $\sigma = 1$ a sinistra e di parametro $\sigma = 3$ a destra.

Nella figura (7.17) di sinistra sia gli errori massimi che quelli minimi del calcolo dei raggi sono nel quadrante negativo, rispettivamente in rosso e blu sull'asse Y. Tuttavia, l'errore sui raggi è molto basso, poichè nell'ordine del millesimo di pixel. Nella figura (7.17) di destra aumenta il blur, dunque l'incertezza della posizione dell'edge. Questo significa che la fase di edge-detection del metodo è sensibile al blur.

Analisi del solo noise e di noise e blur

Purtroppo, in presenza di noise, il caso delle circonferenze i risultati ottenuti non hanno la stessa accuratezza nel caso degli edge rettilinei. Sono state considerate una circonferenza con noise al 50% nella figura (7.18(a)) ed altre due con lo stesso noise, ma blur di parametro rispettivamente $\sigma = 1$ e $\sigma = 3$ nella figura (7.18(b)) e (7.18(c)).

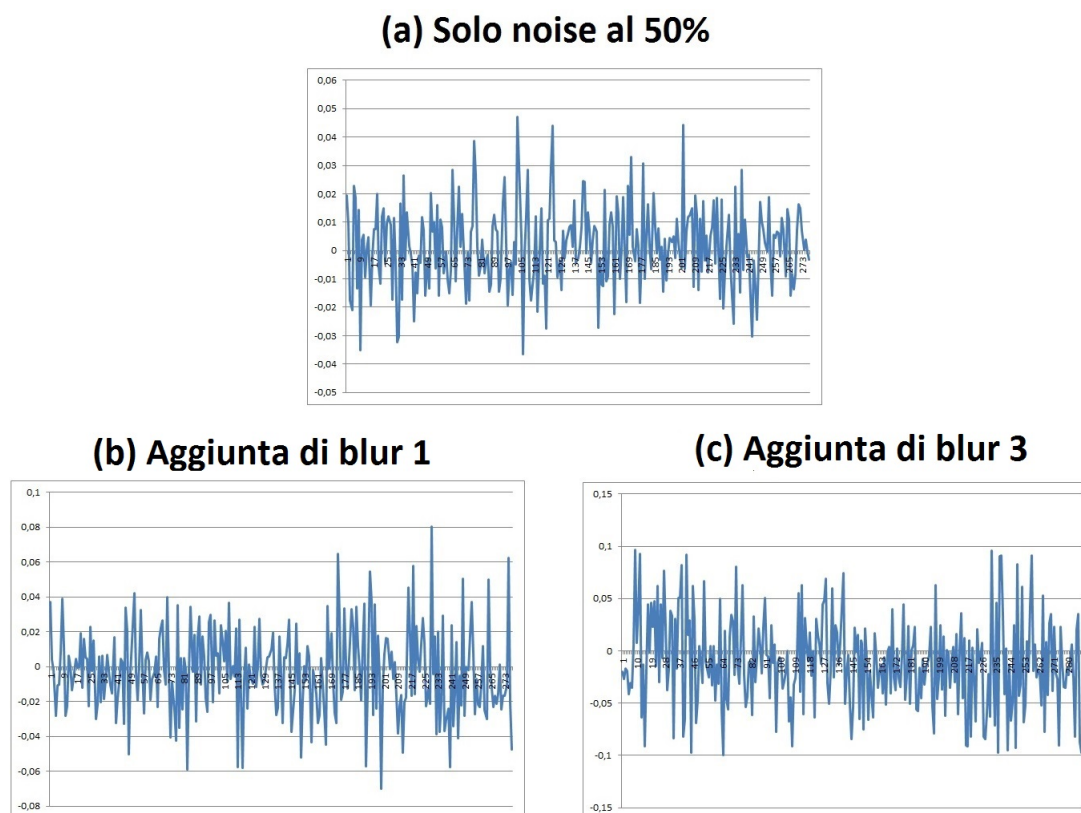


Figura 7.18: (a) errore calcolato sui raggi, pixel per pixel, di una circonferenza, (a) immagine affetta da noise al 50%, (b) aggiunta un blur di parametro $\sigma = 1$, (c) di parametro $\sigma = 3$.

Questi risultati sono stati ottenuti impostando sperimentalmente i parametri dell'operatore di denoising: coefficiente di stabilità $\lambda = 1/4$, intervallo temporale $\Delta t = 0.1$.

Dalle immagini si nota che in caso di noise e blur con parametro $\sigma = 1$ si hanno delle stime simili, mentre aumentando il blur a $\sigma = 3$ aumenta notevolmente l'intervallo di incertezza sulla posizione dell'edge.

Si precisa che, mentre per gli edge rettilinei l'accuratezza dell'edge-detection dipende dalla posizione all'interno del pixel, in questo caso l'accuratezza dipende in modo significativo dalla direzione lungo la quale è valutato il raggio.

Anche in questo caso è stata verificata la ripetibilità del metodo, su una circonferenza con blur di parametro $\sigma = 1$ e noise al 50%. Nella figura (7.19) viene mostrata la deviazione standard della localizzazione dell'edge fatta su cento campioni di circonferenze:

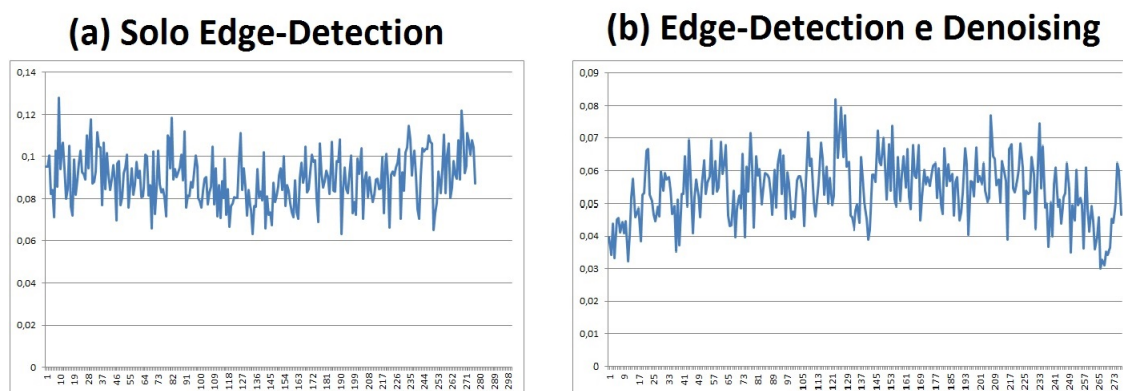


Figura 7.19: (a) senza denoising, (b) con denoising

Sull'asse Y si trova la deviazione standard, mentre sull'asse X si hanno i singoli pixel in cui è stato localizzato l'edge, a cui corrisponde un valore calcolato su cento campioni dell'immagine. Si osserva come l'operatore di denoising migliora le prestazioni del metodo, diminuendo la deviazione standard.

Bibliografia

- [1] A. Bennis, S. M. Riad - Filtering Capabilities and Convergence of the Van-Cittert Deconvolution Technique - IEEE Transactions on Instrumentation and Measurement, vol. 41, No. 2, 1992.
- [2] A. Chambolle - Image Segmentation by Variational Methods: Mumford and Shah Functional and the Discrete Approximations - Siam J. Appl. Math., Vol. 55, No. 3, pp. 827-863, June 1995.
- [3] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, T. Pock - An introduction to Total Variation for Image Analysis - CMAP, Ecole Polytechnique, CNRS, 91128, Palaiseau, France, 2009.
- [4] A. Fabijanska - A survey of Subpixel Edge Detection methods for Images of heat-emitting metal specimens - Applied Mathematics and Computer Science - 2012.
- [5] A. J. Tabatai, O. R. Mitchell - Edge Location to Subpixel Values in Digital Imagery - IEEE Transaction on Pattern Analysis and Machine Intelligence - Vol. Pami-6 No. 2, March 1984.
- [6] A. Marquina, S.J. Osher - Image Super-Resolution by TV-Regularization and Bregman Iteration - Journal of Scientific Computing, pp. 367-382, July 2008.
- [7] A. Trujillo-Pino, K. Krissian, M. Alemán-Flores, D. Santana-Cedrés - Accurate subpixel edge location based on partial area effect - Image and Vision Computing, No. 31 - pp. 72-90, 2013.
- [8] B. Pauland, D. Ghoshal - An Improved Study on Edge Based Image Segmentation and Subsequent Edge Thinning - Indian Journal of Science and Technology - DOI published, Tripura, Vol 9, No. 44, November 2016.
- [9] D. Marr, E. Hildreth - Theory of edge detection - M.I.T. Psychology Department and Artificial Intelligence Laboratory - Cambridge, Massachusetts, U.S.A., Proc. R. Soc. Lond. B 207, pp. 187-217, 1980.

-
- [10] E. Ardizzone, R. Pirrone, R. Gallea, O. Gambino - Noise Filtering Using Edge-Driven Adaptive Anisotropic Diffusion - IEEE Computer Society - Palermo, Italy, 2008.
 - [11] E. P. Lyvers, O. R. Mitchell, M. L. Akey, A. P. Reeves - Subpixel Measurements Using a Moment-Based Edge Operator - Vol. 11, No. 12, December 1989.
 - [12] F. Vankawala, A. Ganatra, A. Patel - International Journal of Computer Applications - A Survey on different Image Deblurring Techniques - Volume 116, No. 13, April 2015, 15.
 - [13] G. Aubert, P. Kornprobst - Mathematical problems in Image Processing - Applied Mathematical Sciences, vol. 147, 1996.
 - [14] G. S. Xu - Sub-pixel Edge Detection Based on Curve Fitting - Proceedings of the 2nd International Conference on Information and Computing Science, Manchester, UK, pp. 373- 375, 2009a.
 - [15] J. Canny - A computational approach to edge detection - IEEE Trans. Pattern Anal. Mach. Intell, Vol. 8, 679-698, 1986.
 - [16] Jean-Michel Morel, S. Solimini - Variational Methods in Image Segmentation, with seven image processing experiments - Birkhauser, Boston, 1995.
 - [17] J. Weickert - Anisotropic Diffusion in Image Processing - ECMI Series, Teubner, Stuttgart, 1998.
 - [18] J.- L. Wu, C.-F. Chang, C.-S. Chen - An Improved Richardson-Lucy Algorithm for Single Image Deblurring Using Local Extrema Filtering - IEEE International Symposium on Intelligent Signal Processing.
 - [19] K. Jensen, D. Anastassiou - Subpixel Edge Localization and the Interpolation of Still Images - IEEE Transactions on Image Processing - Vol. 4, No. 3, March 1995.
 - [20] L. Alvarez, F. Guichard, P. L. Lions, J. M. Morel - Axioms and Fundamental Equations of Image Processing - Archive for Rat. Mech. Anal., Vol. 123, pp. 200 - 257, 1993.
 - [21] L. Alvarez, P.-L. Lions, J. M. Morel - Image selective smoothing and edge detection by nonlinear diffusion (II) - SIAM Journal on numerical analysis 29, pp. 845-866, 1992.
 - [22] L. Lang, Y. Xu - Adaptive Landweber method to deblur images - IEEE Signal Processing Letters, vol. 10, No. 5, 2003.

-
- [23] L. Rudin - Images, numerical analysis of singularities and shock filters - Caltech, C.S. Dept. Report TR, 1987.
- [24] M. Bertero, P. Boccacci - Introduction to Inverse Problems in Imaging - IOP publishing, Bristol, 1998.
- [25] M. Kisworo, S. Venkatesh, G. West - 2D Edge Feature Extraction to Subpixel Accuracy using the Generalized Energy Approach - Proceedings of the IEEE Region 10 International Conference on EC3-Energy, Computer, Communication and Control Systems, New Delhi, India, pp. 344-348, 1991.
- [26] N.K. Nordstrom - Variational edge detection - PhD dissertation, Department of electrical engineering - University of California at Berkeley, 1990.
- [27] R. Breder, V.-V. Estrela V.-V., J. T- De Assis - Sub-pixel Accuracy Edge Fitting by Means of B-spline - Proceedings of the IEEE International Workshop on Multimedia Signal Processing, Rio De Janeiro, Brazil, pp. 1-5, 2009.
- [28] R. Machuca, A. L. Gilbert - Finding Edges in Noisy Scenes - IEEE Transactions on Pattern Analysis and Machine Intelligence 3(1), 103-111, 1981.
- [29] P. Perona, J. Malik - Scale Space and Edge Detection using Anisotropic Diffusion - Proc. IEEE Computer Society Workshop on Computer Vision, 1987.
- [30] P. Perona, J. Malik - A Network for Multiscale Image Segmentation - Department of Electrical Engineering and Computer Science - July 1988.
- [31] S. Ghosal, R. Mehrotra - Orthogonal Moment Operators for Subpixel Edge Detection - Pattern Recognition Letters 26(2), 295-305, 1993.
- [32] V. Caselles, R. Kimmel, G. Sapiro - Geodesic Active Countours - International Journal of Computer Vision, Vol. 22, pp. 61-79, 1997.
- [33] Y. Mo Jung, S. H. Kang, and J. Shen - Multiphase Image Segmentation via Modica-Mortola Phase transition - SIAM J. Appl. Math., 67(5), 1213-1232, 2007.
- [34] Yu-Wing, T. Wai-Shun, T. Chi-Keung Tang - Perceptually-Inspired and Edge-Directed Color Image Super-Resolution - IEEE Computer Society Conference on Computer Vision and Pattern Recognition - 2006.
- [35] Y. Yao, H. Ju - A Sub-Pixel Edge Detection Method Based on Canny Operator - Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China, pp. 97-100, 2009.