

*Metodologie di Morfologia Matematica per il filtraggio
di immagini*

Ierardi Maria Grazia

Spagnuolo Michela

Rapporto tecnico n° 4/2002

Indice

1	Introduzione	2
2	Morfologia Matematica	4
2.1	Operatori di insieme applicati alle immagini	5
2.2	Relazioni d'ordine	7
2.3	Trasformazioni da immagine a immagine	8
2.4	Trasformazioni morfologiche	9
2.5	Elemento di struttura	10
2.6	Erosione	11
2.7	Dilatazione	14
2.8	Opening e closing	15
3	Grafi topologici per la rappresentazione di immagini	16
3.1	Il <i>component tree</i>	17
3.2	Un algoritmo per costruire il <i>component tree</i>	19
3.3	Il grafo di Reeb	21
3.4	Algoritmo per il calcolo del grafo di Reeb per immagini	23
4	Risultati ottenuti e sviluppi futuri	24
	<i>Bibliografia</i>	27

Metodologie di Morfologia Matematica per il filtraggio di immagini

1 Introduzione

Comprendere e rappresentare strutture di forma è di fondamentale importanza in tutti gli ambiti scientifici che necessitano di una caratterizzazione delle proprietà geometriche e morfologiche di oggetti. La Computer Vision e Computer Graphics sono tipici contesti in cui la conoscenza della forma, trattata in base ad ipotesi molto differenti, è necessaria per risolvere problemi di analisi, riconoscimento o semplificazione, con scopi e requisiti molto simili.

I diversi campi di applicazione hanno indotto un'ampia varietà di problemi di analisi di immagini, che a sua volta è all'origine dei molti approcci che sono stati e tuttora sono sviluppati ed oggetto di studio. Ad esempio, la Morfologia Matematica (MM), argomento trattato in questo rapporto tecnico, affronta l'analisi delle immagini con una formalizzazione matematica molto precisa di operatori di base, detti morfologici, che possono essere usati per costruire strumenti più complessi molto utili per diverse classi di problemi.

Strumenti di analisi e rappresentazione della forma sono utili nella Computer Graphics, ad esempio come base per la compressione di modelli geometrici molto complessi in termini di primitive geometriche utilizzate. In questo contesto, infatti, la conoscenza delle caratteristiche principali della forma, o *features*, e la loro configurazione, è importante per una strategia di semplificazione che lasci da parte aspetti irrilevanti del modello senza perdere l'aspetto completo.

La ricerca basata sul contenuto nei database ad oggetti e il processing basato sulla forma, inoltre, stanno convogliando un'attenzione crescente alla formalizzazione della forma come combinazione di geometria e semantica, e come strumento per valutare la somiglianza tra modelli differenti.

In entrambi i casi, la topologia, come studio delle proprietà che sono invarianti per trasformazioni continue di un oggetto, sembra uno strumento appropriato per affrontare questo tipo di problemi.

Negli ultimi anni, nel contesto della grafica, particolare attenzione è stata rivolta alla cosiddetta topologia computazionale, una branca della topologia che ha l'obiettivo di affrontare i problemi topologici senza perdere di vista il problema della fattibilità computazionale. Oltre a strutture a linea, come la *medial axis* per le immagini, sono state introdotte diverse altre strutture a grafo di tipo lineare, denominate *skeleton*, che riescono a descrivere oggetti anche molto complessi tramite uno schizzo che mantiene le proprietà topologiche del modello di partenza. Il grafo di Reeb è un esempio significativo di struttura sia topologica che morfologica, che può essere usato come punto di partenza per la costruzione di modelli avanzati di oggetti geometrici. Il modello di forma ideale dovrebbe infatti preservare le caratteristiche topologiche e garantire un buon rapporto tra efficienza computazionale e contenuto informativo ed efficacia della rappresentazione. In alcune applicazioni, inoltre, sarebbe molto più utile avere una descrizione della superficie basata sulle caratteristiche principali di forma, piuttosto che sulla specifica di tutte le entità geometriche che ne definiscono la rappresentazione. Per questo motivo è utile studiare modelli organizzati gerarchicamente rispetto al contenuto descrittivo della forma o della topologia della superficie.

In questo rapporto viene presentato in particolare uno strumento della Matematica Morfologica per la rappresentazione di immagini, il Component Tree, e un algoritmo per la sua costruzione. Obiettivo di questo rapporto, inoltre, è confrontare due strutture topologiche concettualmente simili ma che sono state applicate a contesti differenti: il Component Tree, utilizzato per la rappresentazione di immagini digitali e il Grafo di Reeb, prevalentemente applicato a dati vettoriali.

In particolare nel prossimo paragrafo viene introdotto un breve studio degli strumenti di analisi di immagine propri della Morfologia. Nel terzo paragrafo viene presentata la struttura digitale detta *component tree* e lo sviluppo di un algoritmo per costruire tale

struttura, utilizzando strumenti della Morfologia Matematica. Nel quarto paragrafo, infine, vengono messi a confronto alcuni esempi di applicazione del Component Tree e del Grafo di Reeb ad alcune immagini e vengono presentati sviluppi futuri.

2 Morfologia Matematica

La Morfologia Matematica è stata definita inizialmente per lo spazio Euclideo e la teoria corrispondente era chiamata morfologia euclidea. Tuttavia, la maggior parte delle tecnologie di analisi d'immagine usa dati digitali. La Morfologia fu perciò estesa allo spazio discreto. La versione discreta dello spazio euclideo bidimensionale \mathbb{R}^2 è \mathbb{Z}^2 . Il processo di digitalizzazione o il passaggio dallo spazio continuo \mathbb{R}^2 allo spazio discreto \mathbb{Z}^2 è ottenuto campionando \mathbb{R}^2 . In generale si considera una rete di punti distribuiti uniformemente detti pixel (elementi della figura). Le immagini discrete sono definite associando un valore numerico ad ogni punto della rete di digitalizzazione.

Le immagini binarie si distinguono dalle immagini a livello di grigio per l'intervallo di valori dati ai punti o ai pixel della rete di digitalizzazione.

Quando le operazioni di calcolo coinvolgono alcune relazioni di vicinanza, l'uso di una rete di digitalizzazione non è sufficiente. Il concetto di *grafo* permette la definizione di relazioni di vicinanza tra punti della rete di digitalizzazione.

Uno dei campi di applicazione in cui si è rivelato particolarmente efficace l'utilizzo della Morfologia Matematica è il filtraggio di immagini. Introduciamo il concetto di filtro di immagini presentando innanzitutto gli operatori *point* e *neighbourhood image*. Gli operatori *point image* modificano il valore di un punto dell'immagine indipendentemente dai valori dei punti vicini ad esso. Un tipico esempio è l'operazione che consiste nel calcolare la visione negativa di un'immagine, cioè un'immagine che abbia luci ed ombre invertite. Il valore di output di un operatore *neighbourhood image* applicato ad un punto dell'immagine dato è ottenuto combinando i valori dell'immagine che giacciono in vicinanza del punto considerato. Per esempio, si può sostituire il livello di grigio di ogni punto dell'immagine con il valore medio dei punti vicini ad esso. Definiamo un filtro di immagini come un operatore *neighbourhood*. In

questo senso, i filtri di immagini possono eseguire un'ampia varietà di tasks, come la riduzione del rumore, la rilevazione dei bordi, la compensazione per messa a fuoco scorretta, e blurring dovuto al moto. La Morfologia Matematica fornisce filtri di immagini per compiere i primi due *task*. Inoltre, i filtri morfologici sono adatti per l'estrazione o la soppressione di oggetti o strutture dell'immagine. In pratica, la maggioranza dei filtri morfologici è realizzata utilizzando alcune conoscenze sulla forma e sulle proprietà geometriche degli oggetti dell'immagine.

L'idea che sta alla base della maggior parte degli operatori morfologici è la seguente: le strutture dell'immagine sono osservate attraverso un piccolo campione e sono o mantenute o alterate a seconda se il campione le ricopre oppure no.

2.1 Operatori di insieme applicati alle immagini

Gli operatori di insieme di base sono l'*unione* \cup e l'*intersezione* \cap . Per le immagini a livelli di grigio, l'unione diventa l'operatore *point-wise maximum* e l'intersezione è sostituita dall'operatore *point-wise minimum*. Il point-wise maximum \vee e il point-wise minimum \wedge tra due immagini f e g , che hanno lo stesso dominio di definizione, sono definiti come segue per ogni punto x :

$$(f \vee g)(x) = \max[f(x), g(x)],$$

$$(f \wedge g)(x) = \min[f(x), g(x)].$$

Un altro operatore di insieme di base è il *complemento*. Il complemento di un'immagine f , indicato con f^c , è definito per ogni pixel x come il massimo valore del tipo di dato utilizzato per memorizzare l'immagine meno il valore dell'immagine f nella posizione x . Il complemento di un'immagine binaria, un segnale monodimensionale, e di un'immagine a livelli di grigio è mostrato nella figura 1.

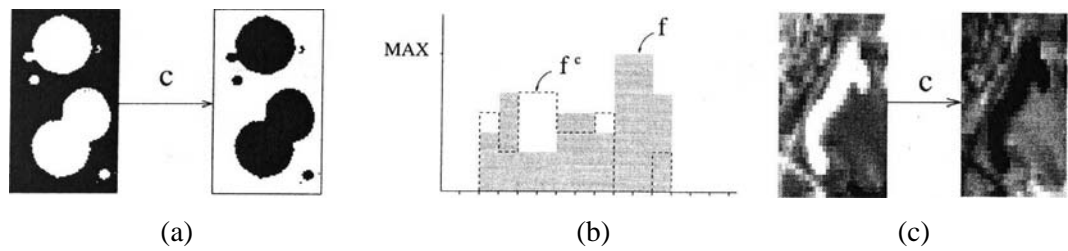


Figura 1: Complemento di un'immagine binaria (a), di un segnale uno-dimensionale (b) e di un'immagine 2-D a livelli di grigio (c)

Dati due insiemi X e Y , la loro differenza, $X \setminus Y$, è definita come l'intersezione tra X e il complemento di Y .

La differenza di insiemi si applica solo ad immagini binarie. La generalizzazione alle immagini a livelli di grigio come $f \wedge g^c$ ha la proprietà indesiderabile che i valori di output dipenderanno dal valore t_{\max} utilizzato per complementare g .

La traslazione di un'immagine, f , mediante un vettore, b , è indicata con f_b . Il valore dell'immagine traslata in un dato pixel x è uguale al valore al valore dell'immagine originale nella posizione traslata con il vettore opposto: $f_b(x) = f(x - b)$.

Per definizione, la traslazione di un vettore b può essere calcolata se e solo se la rete di digitalizzazione traslata coincide con la rete di digitalizzazione originale. Di conseguenza, le traslazioni sono applicate solo ad immagini definite su una rete di punti uniformemente distribuiti. L'operatore di traslazione è illustrato nella figura 2 per un segnale monodimensionale e per il dominio di definizione di un'immagine 2-D.

Tutti gli operatori morfologici si basano sulla sola combinazione degli operatori di intersezione (point-wise minimum), unione (point-wise maximum), complemento e traslazione.

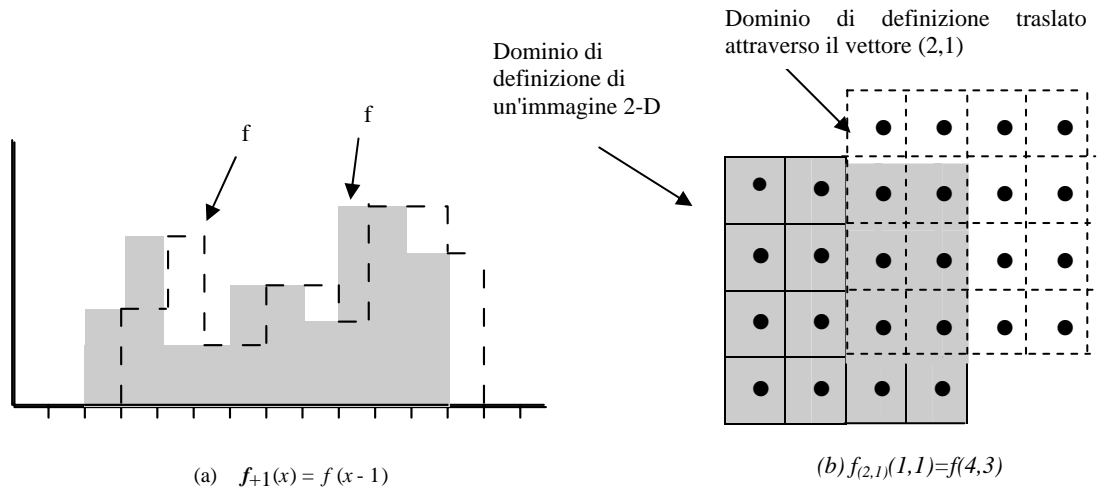


Figura 2: Traslazione di un segnale 1-D (a) e di un'immagine 2-D (b)

2.2 Relazioni d'ordine

Un'immagine f è minore o uguale a un'immagine g con lo stesso dominio di definizione se il valore di f è minore o uguale al valore di g per tutti i pixel x del dominio di definizione comune. Analogamente, se si definisce la *cross-section* a livello t di un'immagine a livelli di grigio f come l'insieme dei pixel dell'immagine i cui valori sono maggiori o uguali a t e si denota questo insieme con $CS_t(f)$, si può dire che per ogni livello di grigio t la cross-section di f a livello t è inclusa nella cross-section di g a livello t :

$$f \leq g \Leftrightarrow \forall x, f(x) \leq g(x) \Leftrightarrow \forall t, CS_t(f) \subseteq CS_t(g).$$

In altri termini, f è minore o uguale a g se il sottografo di f è incluso in quello di g :

$$f \leq g \Leftrightarrow SG(f) \subseteq SG(g).$$

L'ordinamento è un concetto chiave in morfologia matematica e verrà utilizzato in seguito per la costruzione di un grafo topologico dell'immagine. La maggior parte delle trasformazioni morfologiche preserva l'ordinamento sulle immagini.

L'insieme di tutte le immagini con la relazione d'ordine \leq non è *totalmente* ordinato, dal momento che esistono coppie di immagini che non sono ordinate, cioè coppie $\{f, h\}$ tale che né $f \leq h$ né $f \geq h$. Comunque, l'insieme di tutte le immagini soddisfa tutte le proprietà di un insieme *parzialmente* ordinato:

- i) $f \leq f$
- ii) $f \leq g, g \leq f \Leftrightarrow f = g,$
- iii) $f \leq g, g \leq h \Rightarrow f \leq h.$

2.3 Trasformazioni da immagine a immagine

Le trasformazioni morfologiche tra immagini sono *trasformazioni da immagine ad immagine*, cioè l'immagine trasformata ha lo stesso dominio di definizione dell'immagine di input ed è ancora un mapping di questo dominio di definizione nell'insieme degli interi non negativi.

Una trasformazione da immagine ad immagine ampiamente utilizzata è l'*operatore di threshold*, che fissa tutti i pixel dell'immagine di input, che giacciono in un dato intervallo di valori di livello di grigio, al valore 1 ed i rimanenti al valore 0. Questa tecnica talvolta è chiamata *level slicing*.

In termini matematici, l'operatore di threshold T assegna ad ogni pixel x dell'immagine di input f , il cui valore sia compreso nell'intervallo $[t_i, t_j]$, il valore 1 e a tutti gli altri pixel il valore 0:

$$\left[T_{[t_i, t_j]}(f) \right](x) = \begin{cases} 1, & \text{se } t_i \leq f(x) \leq t_j, \\ 0, & \text{altrimenti} \end{cases}$$

Segue che l'operatore di threshold mappa ogni immagine a livelli di grigio in un'immagine binaria.

Una trasformazione da immagine ad immagine Ψ è *invariante per decomposizione threshold* se può essere scritta come somma delle trasformazioni delle cross-sections. Molte trasformazioni morfologiche sono invarianti per decomposizione threshold.

Il *dominio di invarianza* della trasformazione di una data immagine è l'insieme di immagini che non sono modificate da questa trasformazione. Per esempio, il dominio di invarianza della trasformazione identità è l'insieme di tutte le immagini.

In alcuni casi è necessario iterare le trasformazioni di immagini: l'immagine trasformata è utilizzata come input per una seconda trasformazione, ecc..

Tali trasformazioni sono *trasformazioni dei punti della immagine*, cioè il valore di output di un dato pixel dipende solo dal valore di questo pixel nell'immagine di input. Le trasformazioni morfologiche, invece, sono *trasformazioni degli intorni dell'immagine*. In questo caso, il valore di output di un dato pixel è funzione dei valori dei pixel che si trovano in una regione contigua centrata nel pixel considerato.

2.4 Trasformazioni morfologiche

Gli operatori morfologici mirano ad estrarre strutture rilevanti dell'immagine, considerata attraverso il suo sottografo di rappresentazione. Questo obiettivo é ottenuto esplorando l'immagine con un altro insieme di forma nota detto *elemento di struttura* (ES). La forma dell' ES di solito é scelta in accordo con alcune conoscenze a priori della geometria di strutture dell'immagine rilevanti ed irrilevanti. Per strutture irrilevanti intendiamo o il rumore o gli oggetti che vorremmo sopprimere. L'approccio morfologico al processing di immagini é illustrato nella figura 3. Le figure 3a e 3b sono due immagini, binaria e a livelli di grigio rispettivamente. La figura 3c é utilizzata solo per scopi di rendering: rappresenta una visione ombreggiata della figura 3b per arricchire la sua rappresentazione topografica. Nella figura 3d sono mostrati alcuni esempi di elementi di struttura: un cerchio, un esagono, un quadrato, un segmento di linea orizzontale ed una coppia di punti. Se utilizziamo come sonda un ES verticale, possiamo estrarre tutte le strutture verticali delle immagini binarie e a livelli di grigio (figure 3e e 3f). Le trasformazioni che sono state usate in questo esempio si basano su combinazioni di due operatori morfologici di base: l'*erosione* e la *dilatazione*. Questi

operatori duali sono le due lettere dell'alfabeto morfologico, dal momento che tutti gli altri operatori sono espressi in termini di erosioni e dilatazioni.

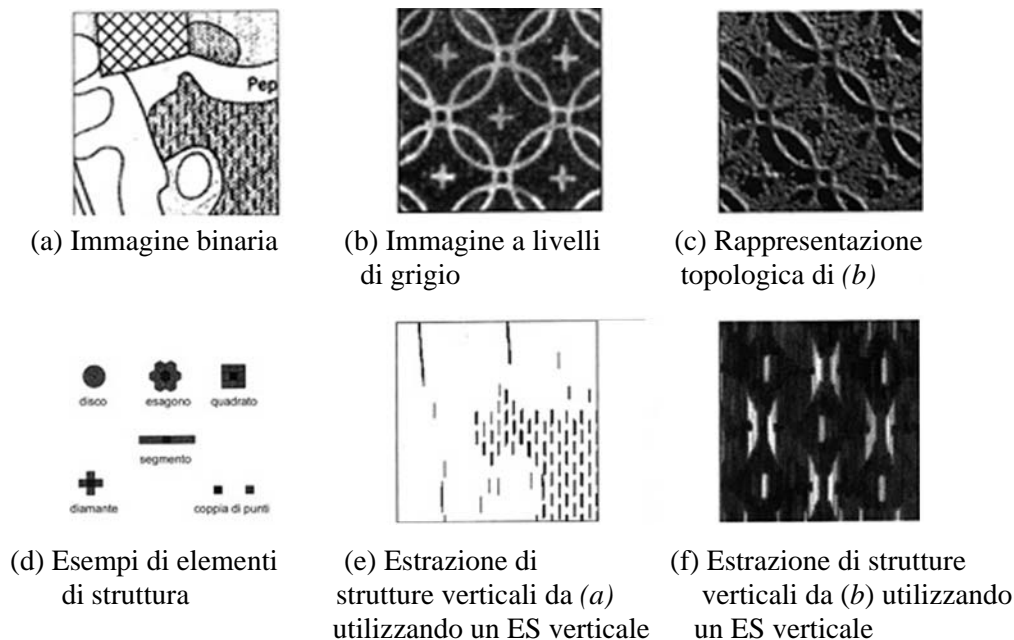


Figura 3: Approccio morfologico all'*image processing*: sia le immagini binarie sia le immagini a livelli di grigio sono considerate come insiemi; un altro insieme di forma nota e chiamato *elemento di struttura*, é utilizzato per studiare l'immagine e perciò per evidenziare le strutture che abbiano forma simile.

2.5 Elemento di struttura

Un elemento di struttura (ES) non é altro che un piccolo insieme utilizzato per indagare l'immagine sotto studio. Analogamente, ES di dimensione $n+1$ sono chiamati *volumetrici, nonflat o a livelli di grigio* e corrispondono a piccole immagini a livelli di grigio. La forma degli elementi di struttura flat non dipende dalla rappresentazione in scala dell'immagine a livelli di grigio. Quando si utilizzano ES nonflat, i loro valori di livello di grigio dovrebbero avere le stesse unità di misura e di scala dell'immagine di input.

Gli operatori morfologici fondamentali richiedono la definizione di un'origine per ogni ES. Questa origine permette il posizionamento dell'ES in un certo punto o pixel: un ES nel punto x significa che la sua origine coincide con x . Concentreremo la nostra attenzione sulla morfologia utilizzando ES flat, cioè elementi di struttura che hanno la stessa dimensione topologica del dominio dell'immagine.

La forma e la dimensione dell'elemento di struttura deve essere adatta alle proprietà geometriche degli oggetti dell'immagine che devono essere processati. Per esempio, elementi di struttura lineari sono adatti per l'estrazione di oggetti lineari.

Qualsiasi forma può essere scelta come elemento di struttura. Tuttavia, nelle applicazioni pratiche si utilizza solo un'insieme limitato di elementi di struttura. A parte la forma, si può anche definire la dimensione e, per alcuni elementi di struttura, una specifica orientazione.

Quando si processano i grafi, il concetto di forma e di orientazione scompare, dal momento che l'ES è definito semplicemente in base alle relazioni di vicinanza di ogni vertice. Perciò non è necessario scegliere un ES.

2.6 Erosione

La prima domanda che può nascere quando si esplora un insieme con un elemento di struttura è “*L'elemento di struttura è incluso nell'insieme?*”. L'insieme eroso è il luogo dei punti dove la risposta a questa domanda è affermativa. L'insieme studiato rappresenta o gli oggetti di un'immagine binaria o il sottografo di un'immagine a livelli di grigio.

Definizione (erosione)

L'operatore di erosione di un insieme X attraverso un elemento di struttura B è indicato con $\varepsilon_B(X)$ ed è definito come il luogo dei punti x tali che B è incluso in X quando la sua origine è in x :

$$\varepsilon_B(X) = \{ x \mid B_x \subseteq X \}. \quad (1)$$

Analogamente l'erosione può essere riscritta in termini di intersezione di traslazioni di insiemi e può essere estesa direttamente alle immagini binarie e a livelli di grigio. L'erosione in termini di intersezioni di traslazioni di immagini è illustrata nella figura 4. In questa figura l'ES è un segmento di tre pixel e l'origine dell'ES è nel pixel centrale.

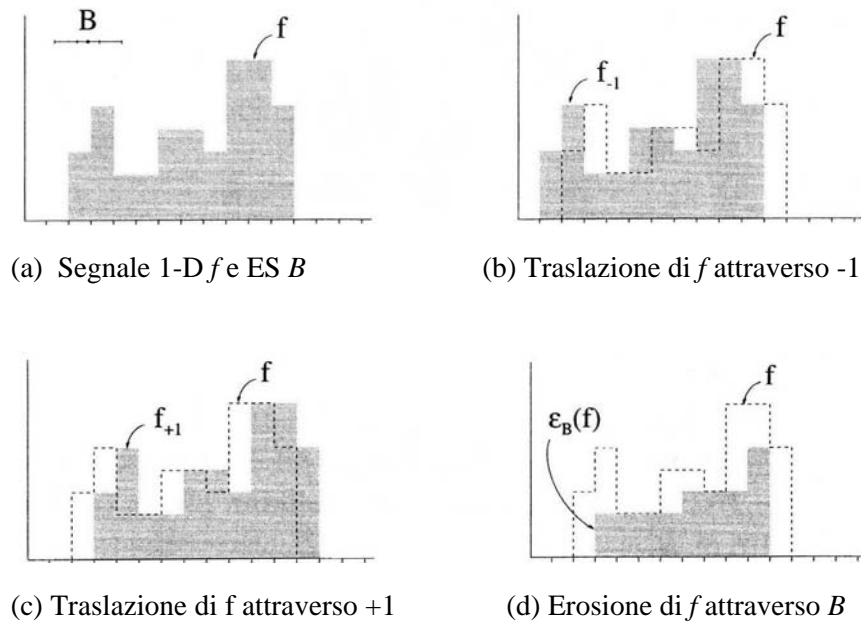


Figura 4: Erosione di un segnale 1-D. L'elemento di struttura B definisce 3 traslazioni di immagine: f , f_{-1} e f_{+1} . L'erosione ϵ di f attraverso B è uguale al minimo calcolato puntualmente fra le tre immagini. Il valore eroso di un certo pixel corrisponde anche al valore minimo dell'immagine nell'insieme dei pixel vicini definito dall'ES quando è centrato in tale pixel

Nella figura 5 sono mostrati alcuni esempi di erosioni su immagini binarie e a livelli di grigio.

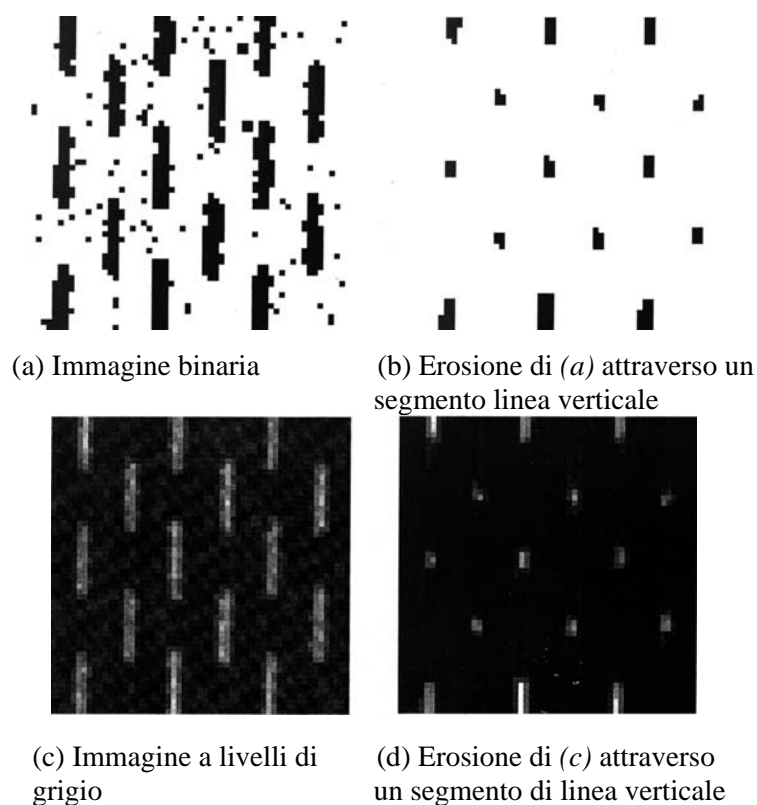


Figura 5: Erosione di un'immagine binaria e di un'immagine a livelli di grigio attraverso un segmento di linea verticale di 13 pixel

In questa figura le barre verticali erose sono ridotte a pochi punti, perché l'elemento di struttura poteva ricoprire le barre solo quando era centrato su questi punti. Le barre connesse ai bordi in alto e in basso non sono così erose come quelle che giacciono all'interno dell'immagine, perché si è assunto che i valori fuori dal dominio di definizione dell'immagine siano tutti uguali a t_{\max} . Questo è un modo comune per trattare gli effetti di bordo quando si erode un'immagine.

Elementi di struttura nonflat, ad esempio volumetrici, hanno valori in scala di grigio per ogni punto del loro dominio di definizione.

I pesi in scala di grigio di un ES volumetrico dovrebbero essere scelti in accordo con i valori di intensità dell'immagine. A causa della difficoltà di avere questa condizione e a causa dei calcoli extra richiesti dalla sottrazione, ES in scala di grigio sono utilizzati raramente nelle applicazioni pratiche.

2.7 Dilatazione

La dilatazione é l'operatore duale dell'erosione e si basa sulla domanda seguente: "L'elemento di struttura **interseca** l'insieme?". L'insieme dilatato é il luogo dei punti dove la risposta a questa domanda é affermativa.

Definizione (dilatazione)

La dilatazione di un insieme X attraverso un elemento di struttura B si indica con $\delta_B(X)$ ed é definito come il luogo dei punti x tali che B interseca X quando la sua origine coincide con x :

$$\delta_B(X) = \{x | B_x \cap X \neq \emptyset\}. \quad (2)$$

Analogamente la dilatazione può essere scritta in termini di unione di traslazioni di insiemi e può essere direttamente estesa alle immagini binarie e a livelli di grigio.

In altri termini, il valore dilatato di un pixel x é il massimo valore dell'immagine nella finestra definita dall'elemento di struttura quando la sua origine é in x .

La dilatazione di un segnale monodimensionale é mostrata nella figura 6.

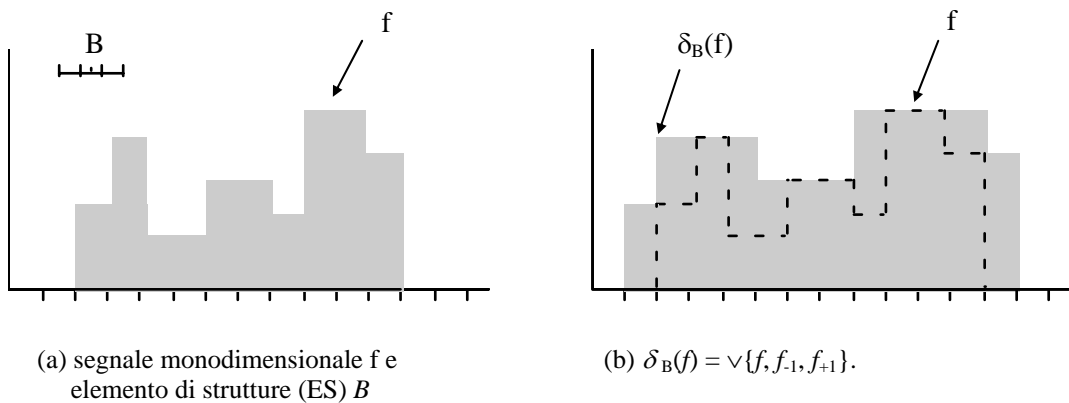


Figura 6 Dilatazione rappresentata come point-wise maximum di una serie di traslazioni di immagine Il valore dilatato di un certo pixel corrisponde al valore massimo dell'immagine all'interno della regione di vicinanza definita dall'ES quando é centrato in quel pixel. Notare che l'equazione che ne segue comprende anche: $\delta_B[SG(f)] = \{(x, t) | B_{(x,t)} \cap SG(f) \neq \emptyset\}$.

Esempi di dilatazioni di immagini binarie e a livelli di grigio sono mostrate nella figura 7. Notare che l'ES è abbastanza lungo da coprire i segmenti di linea della linea tratteggiata quando esso si trova tra due segmenti. Questo spiega perché le lacune sono completamente riempite nell'immagine dilatata.

A differenza delle immagini dilatate o erose attraverso ES flat, i valori dei livelli di grigio delle immagini dilatate o erose attraverso ES nonflat non sono limitate dal valore minimo e massimo dei livelli di grigio dell'immagine di input. In particolare le erosioni con ES nonflat possono restituire come output interi negativi.

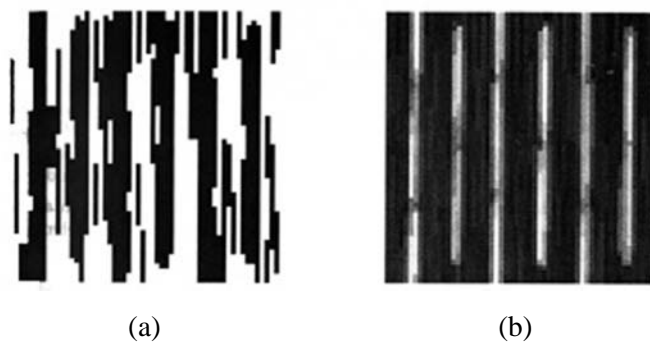


Figura 7: Dilatazioni di immagine binaria (a) e di immagine a livelli di grigio (b) attraverso un segmento verticale di 13 pixel (le immagini originali sono mostrate in fig. 5a e 5c rispettivamente).

2.8 Opening e closing

L'erosione di un'immagine non solo rimuove tutte le strutture che non possono contenere l'elemento di struttura, ma riduce anche le altre. La ricerca di un operatore che recuperi la maggior parte delle strutture perse attraverso l'erosione conduce alla definizione dell'operatore morfologico *opening*. Il principio consiste nel dilatare l'immagine precedentemente erosa utilizzando lo stesso elemento di struttura. In generale, non tutte le strutture sono recuperate. Per esempio, oggetti completamente distrutti dall'erosione non possono essere affatto recuperati. Questo é il comportamento alla base delle proprietà di filtraggio dell'operatore opening: le strutture dell'immagine sono selettivamente filtrate e la selezione dipende dalla forma e dalla dimensione

dell'ES. L'operatore duale dell'opening morfologico é il *closing morfologico*. Entrambi questi operatori sono alla base dell'approccio morfologico al filtraggio di immagini.

3 Grafi topologici per la rappresentazione di immagini

La topologia, intesa come lo studio delle proprietà di oggetti che sono invarianti per trasformazioni bicontinue, è un uno strumento molto efficiente per l'analisi e la codifica di oggetti. Nell'*image processing* sono state introdotte molte strutture per rappresentare immagini e catturare i collegamenti tra le componenti principali dell'immagine [S82]. L'obiettivo è costruire descrizioni ad alto livello della scena rappresentata dall'immagine a partire da *features* a basso livello, per la segmentazione o per la comprensione della scena. Conoscere la topologia che sta alla base di una particolare immagine è utile, inoltre, per controllare tipiche operazioni di immagini, come, ad esempio, la skeletonizzazione per *thinning*, in cui punti critici che rendono sconnesso lo skeleton devono essere identificati per assicurare un risultato corretto [BS2000].

Nel campo dell'analisi di immagini, i grafi di struttura sono stati utilizzati principalmente per ottimizzare la codifica e la manipolazione di immagini e delle loro componenti più significative. Ci sono, comunque, molte somiglianze con l'uso dei grafi topologici per l'analisi di modelli di superficie, ed è interessante puntualizzarli. Prima di tutto, le immagini a livello di grigio possono essere rappresentate come funzioni che hanno \mathbb{R}^2 come dominio, e si può associare ai livelli di grigio una catena di interi. In pratica, si può associare una quota ad ogni valore di livello di grigio. La topologia delle superfici dell'immagine è ovviamente semplice, mentre la configurazione dei punti critici, per esempio livelli di grigio massimo e minimo, fornisce un supporto interessante per molte operazioni di filtraggio. I punti critici, comunque, non sono le uniche componenti dell'immagine usate per definire grafi di struttura, ma sono state proposte altre soluzioni con significati leggermente differenti.

In questo contesto, abbiamo preso in considerazione in particolare il *component tree*, anche noto come *confinement tree* o *dendrone* [BC97, J99, DM2000], e il grafo di Reeb [B98,BFS2000,ABS2001].

3.1 Il *component tree*

Il *component tree* è una rappresentazione dell'immagine a livelli di grigio che contiene informazioni su ciascuna componente dell'immagine e sui collegamenti che esistono tra le componenti a livelli di grigio consecutivi nell'immagine [J99]. Data un'immagine f , le sue componenti sono le componenti connesse della scomposizione a soglie di f . Ogni insieme soglia può essere filtrato utilizzando un criterio appropriato, in modo che solo le componenti rilevanti vengano prese in considerazione. Tali componenti connesse definiscono l'insieme dei nodi del *component tree*. Inoltre, si possono associare ad ogni nodo i cosiddetti *attribute openings*. Essi memorizzano l'informazione sia su proprietà geometriche, come area e perimetro, sia su proprietà topologiche, per esempio se la componente è stata mantenuta o rimossa dal processo di filtraggio. L'adiacenza tra le componenti, cioè l'insieme degli archi, viene definita considerando le relazioni che derivano dal considerare i vari livelli di grigio impilati uno sull'altro.

Prima di formalizzare il concetto di *component tree* è necessario estendere il concetto di componente per includere i legami tra componenti a livelli di grigio consecutivi.

Definiamo l'insieme di tutte le componenti in un immagine a livelli di grigio f come

$$\{C_{T,N}: C_{T,N} \text{ è l}'n\text{-esima componente in } X_t(f); t \text{ è un livello di grigio qualsiasi in } f\}$$

L'immagine originale f può essere ricostruita come uno *stack* di queste componenti:

$$f(x) = \max \left\{ t : x \in \bigcup_n C_{t,n} \right\}$$

Una proprietà importante degli insiemi soglia è che $X_{t+1}(f) \subseteq X_t(f)$ [CGW86], che implica che per ogni componente $C_{t+1,n}$, esista una componente $C_{t,m}$ tale che $C_{t+1,n} \subseteq C_{t,m}$. In questo caso diciamo che le due componenti $C_{t+1,n}$ e $C_{t,m}$ sono collegate nel grafo e la componente $C_{t+1,n}$ è una *discendente* di $C_{t,m}$. La componente $C_{\min,n}$ al minimo livello di

grigio min nell'immagine è chiamata *radice*, perché è il dominio dell'intera immagine ed è un soprainsieme per tutte le componenti dell'immagine. La componente radice è collegata a tutte le componenti a livello di grigio $min + 1$, che sono a loro volta collegate con tutte le componenti a livello di grigio $min + 2$, e così via. Una componente che non è collegata ad un'altra componente a livello di grigio superiore è chiamata *massimo regionale* (un massimo regionale è un insieme connesso di pixel di livello di grigio costante che è circondato da pixel di livello di grigio inferiore).

Quindi, l'insieme delle componenti dell'immagine ed i collegamenti tra di esse formano una struttura ad albero. Si definisce *component tree* T l'insieme dei nodi N connessi da un insieme di archi. Ogni nodo dell'albero rappresenta una particolare componente $C_{t,n}$ nell'immagine e sarà indicata con $n(C_{t,n})$ o semplicemente con n . Due nodi sono collegati da un arco quando le corrispondenti componenti sono collegate. Il nodo è una rappresentazione astratta della componente che può essere qualsiasi cosa, da una lista di tutti i pixel della componente al valore di un singolo attributo come l'area della componente. Per usare il component tree per il filtraggio di immagini può essere necessario memorizzare in ciascun nodo anche altri tipi di informazione, come ad esempio:

- (i) il livello di grigio della componente rappresentata;
- (ii) la localizzazione nell'immagine di un pixel rappresentativo della componente;
- (iii) il valore di alcuni attributi della componente, come l'area, il perimetro o la lunghezza della componente principale;
- (iv) se la componente è stata mantenuta o rimossa dal filtro (deciso durante il passo di filtraggio dell'albero, come discusso nella sezione seguente).

Il nodo $n(C_{min,n})$ appartenente a N che corrisponde alla componente radice $C_{min,n}$ è la radice dell'albero. Quei nodi nell'albero che corrispondono a massimi regionali sono le foglie dell'albero e saranno indicate con n_l . Un ramo indicato con $B(n_l)$ è definito come la sequenza più corta di nodi collegati da una qualsiasi foglia data n_l fino alla radice dell'albero.

In figura 8, è mostrata un'immagine e la sua rappresentazione come component tree.

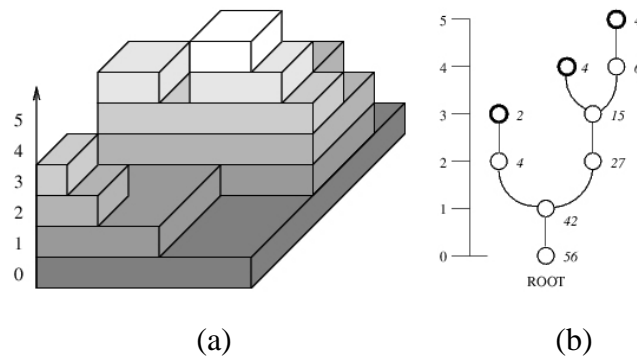


Figura 8: Il component tree: visione prospettiva di un'immagine (a) e il suo component tree (b).

3.2 Un algoritmo per costruire il *component tree*

Nell'implementazione di un algoritmo per la costruzione del component tree é stato seguito un approccio top-down. L'immagine è rappresentata attraverso una struttura a matrice e si suppone che i livelli di grigio che la compongono siano compresi nell'intervallo $[\min, \max] \subset \mathbb{R}^2$, dove $\min \geq 0$ e $\max \leq 255$. L'algoritmo richiede come input l'immagine originale e due valori interi scelti dall'utente come passo (*step*) di campionamento e soglia (*threshold*) sulle dimensioni delle componenti. Tale passo determina quanto fitta si vuole che sia la scomposizione dell'immagine e la soglia viene utilizzata come parametro per la funzione morfologica che corrisponde all'operatore di opening, applicato alle componenti della scomposizione. Una volta acquisita l'immagine, si inizializza una matrice di lavoro, formata dalle maschere delle componenti dell'immagine originale che abbiano un valore di grigio maggiore strettamente di *max-step*. In questa matrice di lavoro ad ogni componente connessa viene associato un nodo dell'albero. La costruzione del component tree prosegue quindi determinando le componenti a livello di grigio inferiore. Gli archi che collegano le componenti sono determinati applicando la proprietà degli insiemi soglia, espressa nel paragrafo 3.1, per cui data una componente connessa n a livello di grigio t , $C_{t,n}$, esiste un'unica componente m al livello di grigio inferiore $t\text{-step}$, $C_{t\text{-step},m}$, tale che $C_{t,n} \subseteq C_{t\text{-step},m}$.

In pratica, ad ogni livello le componenti dell'insieme soglia $X_{t\text{-step}}$ sono memorizzate in una matrice ausiliaria e sono etichettate in modo coerente rispetto al livello precedente. Gli archi che collegano le componenti sono determinati semplicemente confrontando le inclusioni tra la maschera della matrice di lavoro e quella della matrice ausiliaria. Tale procedimento é iterato finché il livello di grigio t é maggiore o uguale a zero. L'ultimo passo dell'algorithm determina la radice dell'albero, cioè restituisce un'unica componente connessa che ha la stessa dimensione dell'immagine originale. L'albero così ottenuto viene memorizzato attraverso una matrice di incidenza e può essere visualizzato con la funzione ausiliaria *draw_graph*.

L'approccio seguito per costruire il component tree è illustrato graficamente in figura 9, nel caso di un segnale unidimensionale.

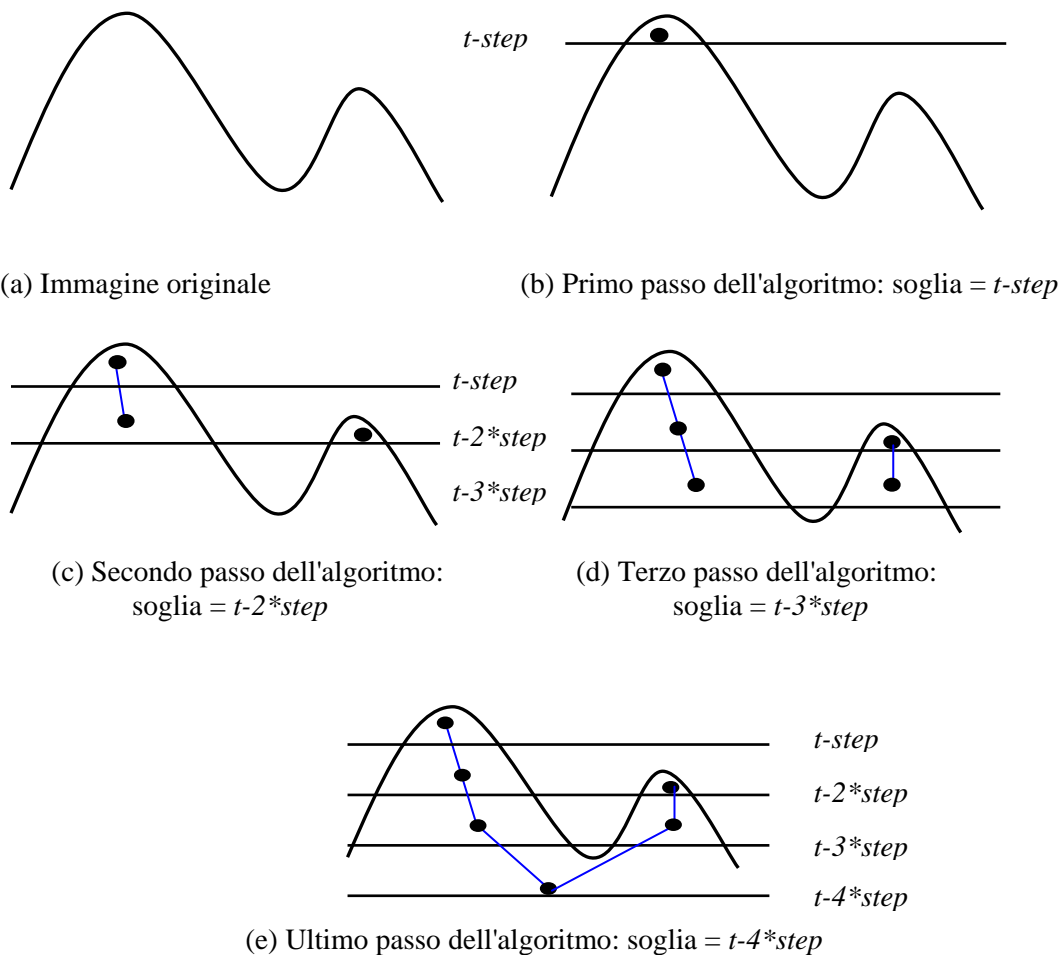


Figura 9: Costruzione del component tree

3.3 Il grafo di Reeb

Nella modellazione di superfici, siano esse naturali come i terreni o appartenenti a manufatti, è di fondamentale importanza definire un modello geometrico che permetta la ricostruzione precisa della superficie a partire da un insieme di dati rilevati direttamente dal mondo reale. Esempio tipico è dato dall'applicazione della teoria di Morse a superfici rappresentate da triangolazioni. Secondo tale teoria, la topologia di una superficie liscia (differenziabile) e compatta può essere studiata efficacemente studiando la configurazione dei punti critici di una funzione continua definita sulla superficie, detta anche funzione di mapping. Reeb, nel 1946, ha formalizzato una codifica dei risultati di un'analisi di questo tipo, proponendo di descrivere un oggetto con un grafo, detto grafo di Reeb, che rappresenta l'evoluzione delle curve di livello di una funzione continua definita sulla superficie [R46]. Il grafo di Reeb è più precisamente definito come lo spazio quoziente ottenuto "collassando" in un punto rappresentativo ogni componente connessa delle curve di livello della funzione di mapping. Il grafo di Reeb è formalmente definito come segue:

Definizione(grafo di Reeb)

Sia $f: M \rightarrow \mathbb{R}$ una funzione a valori reali definita su una varietà compatta M , ovvero su una superficie 2-manifold senza bordo. Un **grafo di Reeb** è lo spazio quoziente di M ottenuto dalla relazione di equivalenza \sim definita da

$$x_1 \sim x_2 \Leftrightarrow f(x_1) = f(x_2) \text{ e } x_1 \text{ e } x_2 \text{ appartengono alla stessa componente connessa di } f^{-1}(f(x_1)) .$$

Si può osservare che i due punti $(x_1, f(x_1))$ e $(x_2, f(x_2))$ nel grafo di Reeb sono rappresentati dallo stesso punto se il valore di f è lo stesso e appartengono alla stessa componente connessa dell'immagine inversa di $f(x_1)$ (o di $f(x_2)$). Il grafo rappresenta tutti i punti appartenenti alla stessa classe di equivalenza della superficie M come un

unico punto nello spazio quoziente. Per esempio se consideriamo la funzione altezza, h definita su una superficie S , immersa regolarmente in \mathbb{R}^3 , che mappa ogni suo punto nella terza coordinata, ossia $h(x, y, z) = z$, allora il grafo di Reeb è lo spazio quoziente dato dalla relazione di equivalenza che identifica i punti (x_1, y_1, z) e (x_2, y_2, z) se essi giacciono sulla stessa componente connessa della sezione orizzontale effettuata sulla superficie ad altezza z .

Ogni contorno connesso sul piano orizzontale nel grafo di Reeb è rappresentato da un punto. Nella figura 10 è rappresentato il grafo di Reeb rispetto alla funzione altezza di un bi-toro.

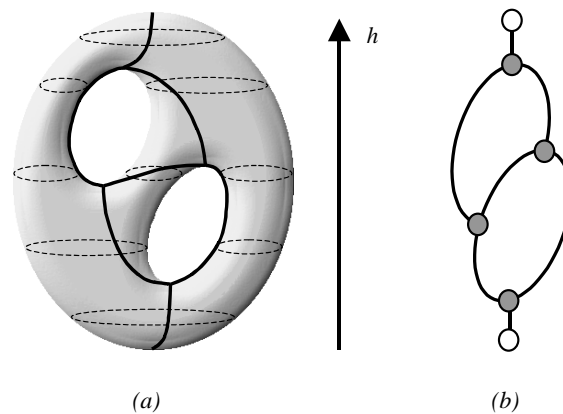


Figura 10 Grafo di Reeb di un bi-toro effettuato rispetto alla funzione altezza h : le classi di equivalenza sono evidenziate in grassetto sulla superficie (a) mentre in (b) è mostrato il grafo corrispondente.

I grafi di Reeb risultano utili nella visualizzazione di alcune caratteristiche delle superfici di oggetti di grosse dimensioni. Inizialmente alcune applicazioni dei grafi di Reeb sono state sviluppate da T. L. Kunii e Y. Shinagawa e un raffinamento di tali grafi è stato introdotto da A. Fomenko con lo scopo iniziale di classificare topologicamente i sistemi Hamiltoniani integrabili. Ulteriori riferimenti bibliografici per i grafi di Reeb possono essere trovati in [SKK91], [SHK91], [SH91], [FK97], [CSA2000], [TISKU95], [KOBPS97].

3.4 Algoritmo per il calcolo del grafo di Reeb per immagini

Analogamente all'algoritmo per la costruzione del component tree, il calcolo del grafo di Reeb per immagini segue un approccio top-down. L'immagine è rappresentata attraverso una struttura a matrice e si suppone che i livelli di grigio che la compongono siano compresi nell'intervallo $[\min, \max] \subset \mathbb{R}^2$, dove $\min \geq 0$ e $\max \leq 255$. L'algoritmo richiede come input l'immagine originale, il valore intero (*step*) scelto come passo di campionamento e la soglia (*threshold*) di filtraggio. Una volta acquisita l'immagine, si inizializza una matrice di lavoro, formata dalle maschere delle componenti dell'immagine originale che abbiano un valore di grigio maggiore strettamente di *max-step*. In questa matrice di lavoro ad ogni componente connessa viene associato un nodo dell'albero. La costruzione del grafo di Reeb prosegue quindi determinando le componenti a livello di grigio inferiore. In modo analogo alla costruzione del component tree anche in questo caso gli archi che collegano le componenti sono determinati applicando la proprietà degli insiemi soglia, per cui data una componente connessa n a livello di grigio t , $C_{t,n}$, esiste un'unica componente m al livello di grigio inferiore $t\text{-step}$, $C_{t\text{-step},m}$, tale che $C_{t,n} \subseteq C_{t\text{-step},m}$. In pratica, ad ogni livello sono ricavate le componenti dell'immagine a valori di grigio compresi nell'intervallo $[t, t+\text{step}]$, tali componenti sono memorizzate in una matrice ausiliaria e sono etichettate in modo coerente rispetto al livello precedente. Dal confronto delle inclusioni tra la maschera della matrice di lavoro e quella della matrice ausiliaria si definiscono gli archi che collegano le componenti. Tale procedimento è iterato finché il livello di grigio t è maggiore o uguale a zero. L'ultimo passo dell'algoritmo determina la radice dell'albero, cioè restituisce un'unica componente connessa che ha la stessa dimensione dell'immagine originale. L'albero così ottenuto viene memorizzato attraverso una matrice di incidenza e può essere visualizzato con la funzione ausiliaria *draw_graph*.

4 Risultati ottenuti e sviluppi futuri

In questo paragrafo presentiamo alcuni commenti sui risultati ottenuti, un confronto tra component tree e grafo di Reeb per triangolazioni e per immagini e possibili sviluppi futuri.

Il component tree é una struttura dati efficiente e accessibile per implementare filtri connessi a livelli di grigio. Purtroppo, uno degli svantaggi principali dei filtri connessi, é che possono essere sensibili al rumore. Anche una piccola quantità di rumore può collegare insieme regioni normalmente distinte, perché il vincolo di connettività é stretto e non tiene conto della "forza" della connessione. Una prima osservazione, dunque, è che il numero di componenti connesse ad ogni livello di grigio dipende dalla scelta della grandezza dell'elemento di struttura e, quindi, il component tree può essere sensibile al noise dell'immagine. Il successo del filtraggio basato su component tree e *attribute signature* dipende molto dal tipo di attributo utilizzato; tale attributo deve essere scelto con cura, se lo scopo é discriminare adeguatamente l'oggetto desiderato dal resto dell'immagine.

Come si può vedere in figura 10, se si sceglie un elemento di struttura sufficientemente piccolo, il numero di componenti connesse dell'insieme soglia é molto elevato e, quindi, i nodi del grafo perdono di significatività.

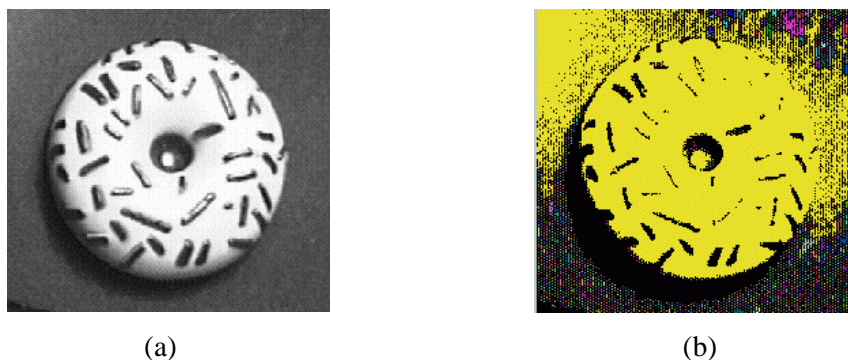


Figura 10: (a) Immagine originale; (b) Una fase di elaborazione del component tree per l'immagine (a)

Per quanto riguarda la relazione del component tree con il grafo di Reeb estratto per *mesh* triangolari si può osservare che entrambi i grafi riconoscono i massimi regionali ma, come si vede confrontando le figure 11*b* e 11*d*, nel component tree non vengono riconosciute le regioni di quota minima, cioè alle regioni rappresentate in blu in figura 11*a* non corrisponde nessun nodo del component tree.

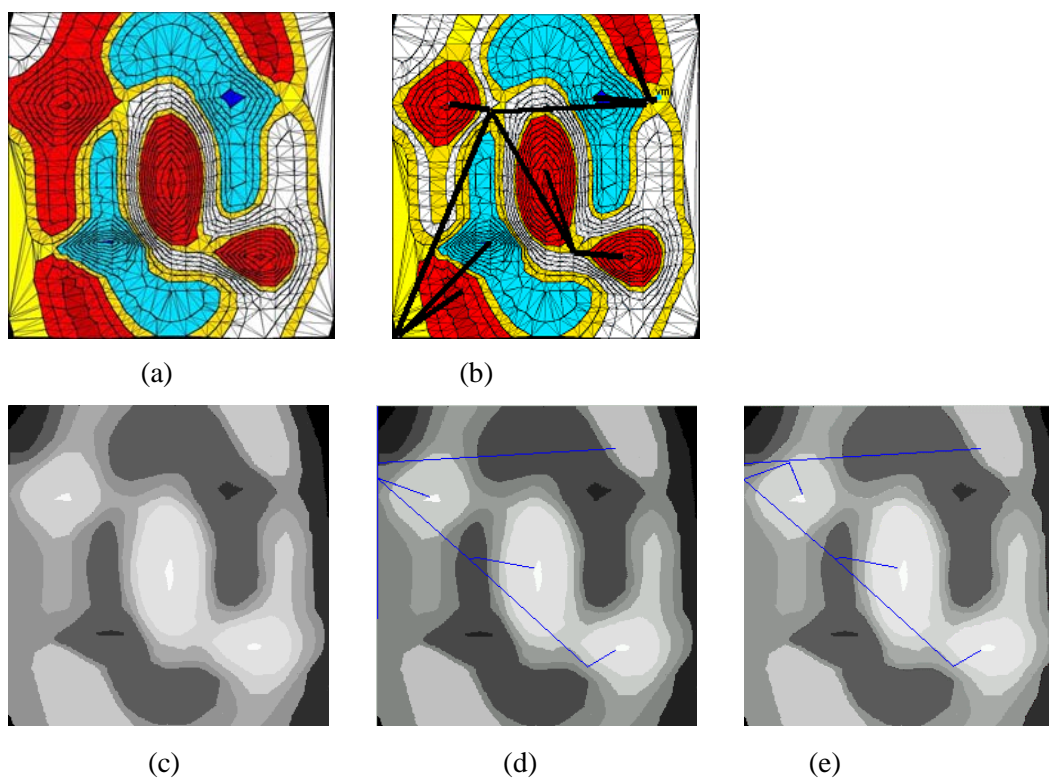
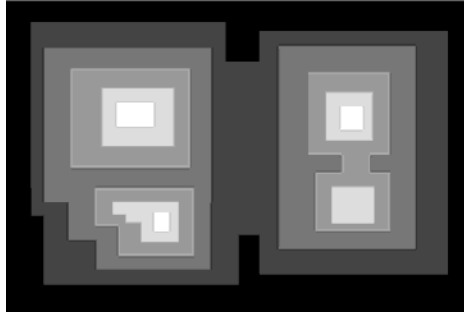


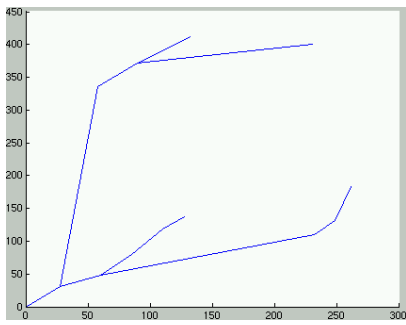
Figura 11: (a) Triangolazione; (b) Grafo di Reeb della triangolazione (a); (c) rappresentazione a livelli di grigio dell'immagine (a); (d) component tree dell'immagine (c); (e) grafo di Reeb per immagini di (c)

Come si può osservare confrontando le figure 11*d* e 11*e*, il grafo di Reeb calcolato per le immagini con l'algoritmo presentato in questo rapporto é praticamente equivalente al component tree. Tale similarità può essere riscontrata anche negli esempi di figura 12.

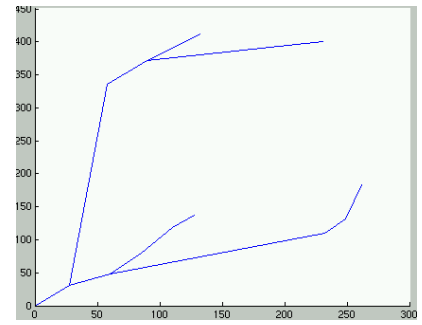
Figura 12: alcuni esempi di component tree e di grafi di reeb per immagini



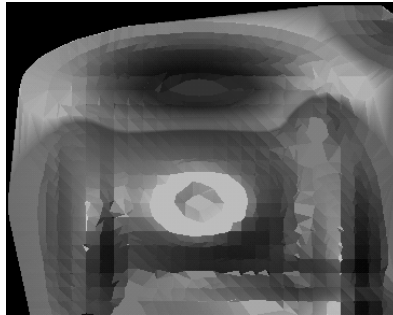
(a) Immagine a livelli di grigio



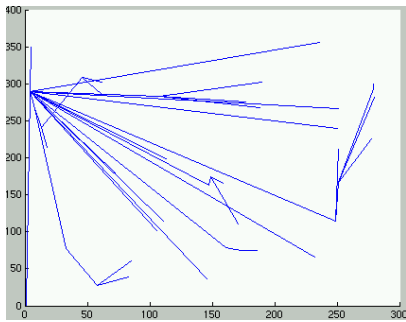
(b) Component tree di (a)
con *step* 20 e *threshold* 30



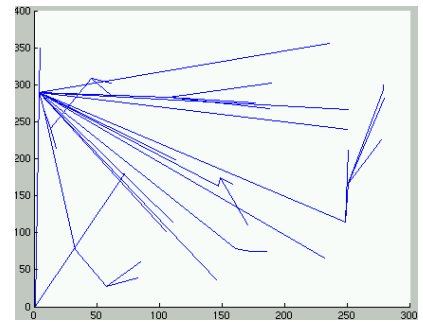
(c) Grafo di Reeb di (a)
con *step* 20 e *threshold* 30



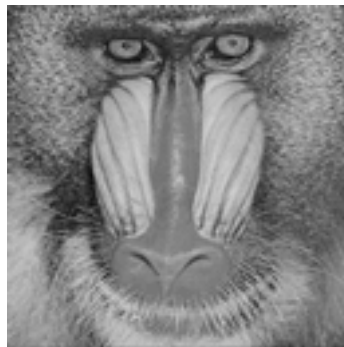
(d) immagine a livelli di grigio



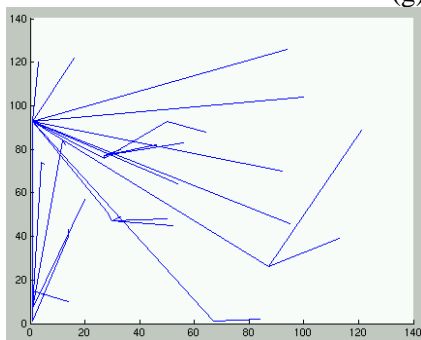
(e) Component tree di (d)
con *step* 20 e *threshold* 30



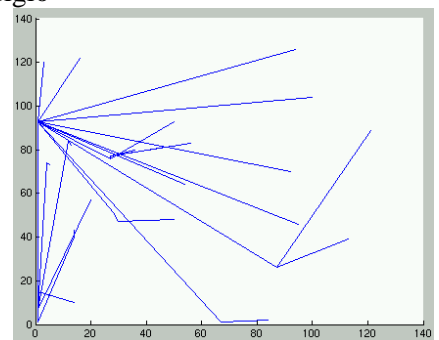
(f) Grafo di Reeb di (d)
con *step* 20 e *threshold* 30



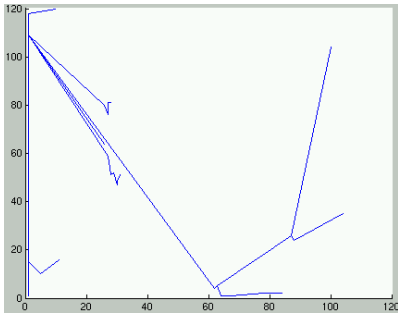
(g) Immagine a livelli di grigio



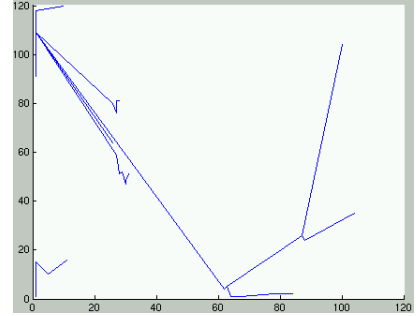
(h) Component tree di (g)
con *step* 20 e *threshold* 30



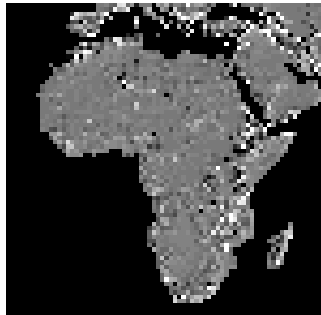
(l) Grafo di Reeb di (g)
con *step* 20 e *threshold* 30



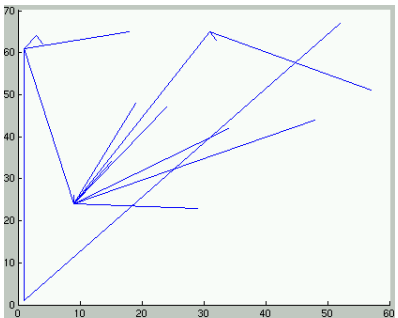
(m) Component tree di (g)
con *step* 1 e *threshold* 200



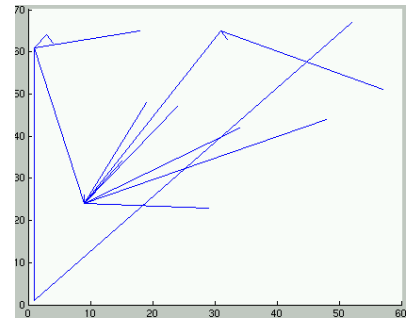
(n) Grafo di Reeb di (g)
con *step* 1 e *threshold* 200



(p) Immagine a livelli di grigio



(q) Component tree di (p)
con *step* 10 e *threshold* 20



(r) Grafo di Reeb di (p)
con *step* 10 e *threshold* 20

Come possibili ulteriori sviluppi della ricerca, ci sono molte aree della Matematica Morfologica che necessitano di attenzione. Per qualsiasi problema di filtraggio di immagine dato, l'attributo utilizzato nella *signature* é scelto sostanzialmente per tentativo o per errore, dal momento che per ora non abbiamo meccanismi formali che permettano di selezionare l'attributo migliore da utilizzare. Al momento si può solo determinare se questo é vero testando tali attributi su semplici immagini. I mezzi di classificazione di *signature* potrebbero essere ulteriormente sviluppati. Nel metodo attuale, le *signature* delle *features* desiderate dell'immagine sono paragonate visivamente con quelle dello sfondo e le *features* preminenti sono identificate nella *signature* che potrebbe essere utilizzata per scopi di discriminazione.

BIBLIOGRAFIA

- [ABS2001] Attene M., Biasotti S., Spagnuolo M.: *Remeshing Techniques for Topological Analysis*. In Proc. of Shape Modeling International 2001, Genova, Italy, 2001
- [BFS2000] Biasotti S., Falcidieno B., Spagnuolo M.: *Extended Reeb Graphs for Surface Understanding and Description*. In Proc. of 9th Discrete Geometry for Computer Imagery Conference, LNCS, Springer Verlag, Uppsala 2000
- [BS2000] Sanniti di Baja G., Svensson S.: *Detecting Centres of Maximal Geodesic Discs on the Distance Transform of Surface in 3D Images*. In Proc. of 9th Discrete Geometry for Computer Imagery Conference, LNCS, Springer Verlag, Uppsala 2000
- [CSA2000] Carr H., Snoeyink J., Axen U.: *Computing contour trees in All Dimensions*. In Proc. SODA 2000 (Symposium in Discrete Algorithms), 2000
- [DM2000] Demongeot J., Mattes J. G.: *Efficient Algorithms to implement the Confinement Tree*. In Proc. of 9th Discrete Geometry for Computer Imagery Conference, LNCS, Springer Verlag, Uppsala 2000

- [J99] Jones R.: *Connected Filtering and Segmentation Using Component Trees*. Computer Vision and Image Understanding, Vol. 75, No. 3, 1999
- [B98] Biasotti S.: *Rappresentazione di superfici naturali mediante grafi di Reeb*. Thesis for the Laurea Degree, Department of Mathematics, University of Genoa, September 1998
- [BC97] Bertrand G., Couprie M.: *Topological Grayscale Watershed Transformation*. In Proc. of SPIE, Vision Geometry V, Vol. 3168, Bellingham, WA, 1997
- [FK97] Fomenko A., Kunii T. L.: *Topological Modelling for Visualization*, Springer-Verlag, 1997
- [KOBPS97] Van Kreveld M., Van Ostrum R., Bajaj C., Pascucci V., Schikore D.: *Contour trees and small seed sets for isosurfaces traversal*. In Proc. 13th Anu. ACM Sympos. Comput. Geom, 1997
- [TISKU95] Takahashi S., Ikeda T., Shinagawa Y., Kunii T. L., Ueda M.: *Algorithms for Extracting Correct Critical Points and Construction Topological Graphs from Discrete geographical Elevation Data*, Eurographics '95, Vol. 14, Number 3, 1995
- [SKK91] Shinagawa Y, Kunii T. L., Kergosien Y. L.: *Surface Coding Based on Morse Theory*, IEEE Computer Graphics & Applications, pp 66-78, September 1991.
- [SHK91] Shinagawa Y., Kunii T.L.: *The Homotopy model: A generalized model for smooth surface generation from cross sectional data*, The Visual Computer, 7(2-3), pp 72-86, 1991
- [SH91] Shinagawa Y., Kunii T. L.: *Constructing a Reeb graph automatically from cross sections*, IEEE Computer Graphics and Applications, 11(6), pp 44-51, 1991
- [CGW86] Coyle E. J., Gallagher N. C., Wendt P. D.: *Stack filters*, IEEE Trans. Acoust. Speech Signal Process. 34, No. 4, 1986, pp. 898-911, 1986
- [S82] Serra J.: *Image Analysis and Mathematical Morphology*. Academic Press, 1982
- [R46] Reeb G.: *Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique*. Comptes Rendus Acad. Sciences, Paris, 222: 847-849, 1946