

UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"



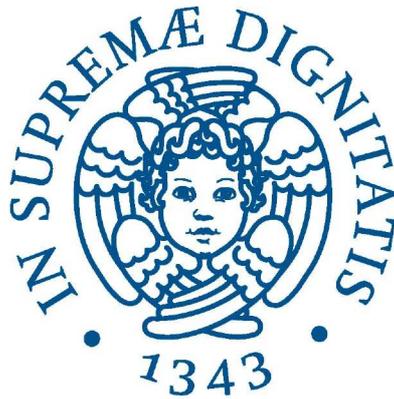
Corso di Dottorato di Ricerca in
INGEGNERIA DELL'INFORMAZIONE

Tesi di Dottorato di Ricerca

ANALISI DI IMMAGINI MEDIANTE LA
COMBINAZIONE DI METODI DI
PATTERN-RECOGNITION E
SUPER-RISOLUZIONE

Anno 2015
SSD ING-INF/05

UNIVERSITÀ DI PISA
Scuola di Dottorato in Ingegneria "Leonardo da Vinci"



Corso di Dottorato di Ricerca in
INGEGNERIA DELL'INFORMAZIONE

Tesi di Dottorato di Ricerca

ANALISI DI IMMAGINI MEDIANTE LA COMBINAZIONE DI METODI DI
PATTERN-RECOGNITION E SUPER-RISOLUZIONE

Autore:
Marco Righi

Relatori:
Prof. Francesco Marcelloni
Dott. Ovidio Salvetti
Dott. Mario D'Acunto

Anno 2015
SSD ING-INF/05

Sommario

L'ambito di ricerca di questa tesi di dottorato è quello dell'*image analysis*. In particolare ci si è occupati di migliorare la qualità e la risoluzione di immagini come quelle provenienti da microscopia a scansione di sonda o da sistemi di *sonar sub-water detection*.

Il contributo originale del lavoro è quello combinare metodi di super-risoluzione con metodi di pattern-recognition. Per capire meglio il contesto e il possibile impatto dei risultati ottenuti basti pensare al ruolo che le immagini e la *image analysis* hanno nelle più disparate tecniche di indagine della natura o in campo diagnostico sia in ambito biomedico, sia in ambito di testing non distruttivi dei materiali. Infatti, la maggior parte delle tecniche di indagine della natura, dai microscopi ottici e di sonda, alla risonanza magnetica, o le tecniche di rilevamento, solo per citarne qualcuna, condensano l'informazione acquisita in un'immagine, che descrive una rappresentazione visiva, non solida della realtà, dalle funzioni del cervello, a porzioni di territorio, alle molecole, agli atomi. Un comune aspetto di queste tecniche è che spesso la risoluzione delle immagini non sempre è ottimale, sia per limiti strumentali (limitato numero di detectors per l'acquisizione dei fotoni nella risonanza magnetica, ad esempio), sia per il metodo di acquisizione dei dispositivi usati (come nel caso delle microscopie di sonda), sia per il rumore di varia natura, sempre presente, indipendentemente dalla tecnica di acquisizione dei segnali.

I metodi di ampliamento della risoluzione sono comunemente noti come metodi di super-risoluzione. Tuttavia l'ampliamento è un'operazione piuttosto complicata, che richiede spesso una conoscenza a priori dell'oggetto da super-risolvere, non sempre disponibile. In questo lavoro di tesi, abbiamo cercato di accoppiare metodi di super-risoluzione con metodi di pattern-recognition in grado di offrire una conoscenza a priori dell'oggetto da super-risolvere attraverso un modello di riferimento quanto più possibile prossimo all'oggetto reale. L'innovativa combinazione di metodi di pattern-recognition e di super-risoluzione hanno portato allo sviluppo di un nuovo algoritmo, il PRIAR (*Pattern Recognition Image Augmented Resolution*), e alla sua implementazione, il *Tool PRIAR*. Il tool è stato applicato con risultati, a nostro avviso estremamente interessanti, a immagini di cellule staminali acquisite con Microscopi a scansione di sonda, di tipo a forza atomica chiamati *Atomic Force Microscope* (AFM) e immagini di sub-water identification da sonar.

I primi risultati da questo lavoro sono incoraggianti e fanno ipotizzare a un impiego della nostra metodologia in vari ambiti che vanno dal biomedicale, alla scienza dei materiali, all'object oriented identification, e persino alla scienza forense.

Abstract

Scope of this dissertation is Image Analysis, applied to images coming from the probe microscopes and sonar sub-water detection. In particular, the contribution of the original work has been the combination of methods of super-resolution methods of pattern-recognition. We can better understand the possible impact of the results obtained, by considering the role that images and computer vision have in several different investigation and diagnostic techniques.

Very often indeed, the resolution of the images is not optimal, due to the instrumental limits (for example, limited number of detectors for the acquisition of the photons in the magnetic resonance imaging), or due to acquisition method (as in the case of the probe microscopies), or to the noise of various kinds, present in all acquisition techniques. The methods for enhancing the resolution are commonly known as super-resolution methods. They are not trivial, since they require *a priori* knowledge of the object to super-resolve, not always available.

In this thesis, we have tried to develop methods of super-resolution coupled with methods of pattern-recognition that offered a priori knowledge of the object to super-resolve in the form of a more faithful model of the real object. The innovative combination of pattern-recognition methods and super-resolution led to the development of a new algorithm, called PRIAR (Pattern Recognition Augmented Image Resolution), and to its implementation, the *Tool PRIAR*. The tool has been used with images of stem cells acquired with the Atomic Force Microscope (AFM), a kind of scanning probe microscope, and with images of sub-water identification by sonar.

The obtained results are encouraging and we hope they can be fruitfully applied to various areas ranging from biomedical, materials science, to object oriented identification, and also to forensic science.

La sapienza è figliola della speranza.
Leonardo Da Vinci

Questa tesi è dedicata alla mia famiglia: alla piccola Anna per la sua tenera età e un mondo da scoprire, a Giulia che muove i primi passi nel mondo delle scienze, a Antonella, il focolare della nostra casa.

Questa tesi è dedicata a chi ha saputo portare luce nella mia vita e rimanere nel mio cuore. Questa tesi è dedicata a chi persevera incessantemente nella ricerca della verità del mondo e delle sue leggi.

∴

Indice

1	Introduzione	1
1.1	Motivazione	1
1.2	Contributi della tesi	3
1.3	Organizzazione della tesi	4
2	Super-risoluzione	5
2.1	Introduzione	5
2.2	Definizione della risoluzione di una immagine	6
2.3	Perché è importante migliorare la risoluzione di un'immagine?	7
2.4	Acquisizione ottica delle immagini e principali artefatti	8
2.4.1	Aliasing	8
2.4.2	Blurring	8
	Distorsione	9
	Aberrazione sferica	9
	Coma	10
	Astigmatismo	12
	Campo di curvatura Petzval	14
	Sfocatura	15
2.4.3	Rumore	16
	Il rumore generato dalla carica elettrica	16
	Il rumore termico	17
2.5	Acquisizione AFM delle immagini e principali artefatti	18
2.5.1	Scansione in un AFM	18
2.5.2	Artefatti in un AFM	19
	Scaling	20
	Crosstalk	21
	Non linearità, isteresi, creep e inarcatura	21
	Overshoot	24
	Convoluzione	24
	Rumore	25

2.6	Lo stato dell'arte della super-risoluzione	26
2.6.1	I metodi di base: bilineare, bicubico e spline	27
2.6.2	Il metodo di Jensen	28
2.6.3	Il metodo di Allebach	30
2.6.4	il metodo di Carey	32
2.6.5	il metodo Muresan	34
2.6.6	il metodo NEDI	35
2.6.7	il metodo Freeman	37
2.6.8	il metodo Kim	39
2.7	Considerazioni	39
3	Pattern-recognition	43
3.1	Introduzione	43
3.2	La segmentazione	45
3.2.1	Connessione tra pixel	45
3.2.2	Omogeneità	46
3.2.3	Distanza e misura della similarità	46
3.3	Analisi delle caratteristiche e segmentazione	49
3.3.1	La distribuzione del colore	50
3.3.2	Le trame dell'immagine	51
3.3.3	Uso dei frattali per l'analisi delle caratteristiche	52
3.3.4	Metodo piramidale	55
3.3.5	Metodo spartiacque	55
3.3.6	Contorni attivi	56
3.3.7	Metodo basato sui grafi	58
3.3.8	Metodo Spectral Embedding MinCut	62
3.4	Metodi per il riconoscimento di oggetti nelle immagini	64
3.4.1	Analisi statistica	65
3.4.2	Analisi sintattica	65
3.4.3	Template matching	66
3.4.4	Reti neurali	66
3.4.5	Schema generali per un sistema di riconoscimento per le immagini	67
3.5	Considerazioni	69
4	PRIAR: Pattern Recognition Image Augmented Resolution	73
4.1	Introduzione	73
4.2	La soluzione proposta da PRIAR	74
4.3	Pattern-recognition	77
4.3.1	Analisi single-point	77
4.3.2	Analisi multi-point	78
4.4	Estensione dell'area riconosciuta	79
4.5	Calcolo del modello	80
4.6	Tool PRIAR Graphic User Interface	81

4.7	Considerazioni	83
5	Studio di alcuni casi: applicazioni a immagini AFM e subacquee	85
5.1	Introduzione	85
5.2	Immagine AFM "A"	86
5.3	Immagine AFM "B"	91
5.4	Immagine AFM "C"	96
5.5	Immagine AFM "D"	101
5.6	Immagini di oggetti subacquei	106
5.7	Considerazioni	109
6	Conclusioni	111
	Indice analitico	113
	Riferimenti bibliografici	115
A	Implementazione	127
A.1	La funzione di esplorazione	127
A.2	Generazione della mappa di probabilità	130

Elenco delle figure

2.1	Il prototipo di Frederick Collier Bakewell	6
2.2	Una immagine in bassa risoluzione e una super-risolta	7
2.3	L'aliasing e problema della scalettatura	9
2.4	L'aliasing e effetto Moiré	9
2.5	Esempi di distorsioni di un'immagine	10
2.6	Aberrazione sferica	11
2.7	Aberrazione sferica: un esempio	12
2.8	La coma	12
2.9	La coma: un esempio	13
2.10	Astigmatismo	13
2.11	Astigmatismo: un esempio	14
2.12	Il campo di curvatura Petzval	15
2.13	Sfuocatura	16
2.14	Rumore elettrico	17
2.15	La testina in un AFM	18
2.16	L'acquisizione topografica di un AFM	19
2.17	Le direzioni di scansione di un AFM	19
2.18	AFM: la distorsione in una struttura piramidale	21
2.19	AFM: la distorsione cross-talk	21
2.20	AFM: l'isteresi	22
2.21	AFM: il creep	23
2.22	AFM: artefatto della curvatura a arco	23
2.23	AFM: artefatto da overshoot	24
2.24	AFM: artefatto da convoluzione	25
2.25	AFM: la convoluzione tra punta e superficie	26
2.26	Un esempio di risoluzione aumentata	27
2.27	Metodo di Jessen: un esempio	30
2.28	Schema generale dell'algoritmo di Allebach	31
2.29	Schema dell'interpolazione guidata dell'algoritmo di Allebach	31

2.30	Filtro COSO usato nell'algoritmo di Allebach	32
2.31	Metodo di Allebach: un esempio	33
2.32	Diagramma a blocchi del metodo Carey	33
2.33	Metodo di Carey: un esempio	34
2.34	Diagramma a blocchi del metodo Muresan	35
2.35	Metodo di Muresan: un esempio	36
2.36	Covarianza locale utilizzata nel metodo Li	36
2.37	Ricostruzione di un'immagine in alta risoluzione con il metodo Li	37
2.38	Metodo di Li: un esempio	38
2.39	Catena di Markov in Freeman	38
2.40	Metodo di Freeman: un esempio	39
2.41	Metodo di Kim [111]	40
2.42	Metodo di Kim: un esempio	40
3.1	Connettività tra pixel	46
3.2	Divergenza di Kullback–Leibler	49
3.3	Distanza di Kolmogorov–Smirnov	50
3.4	Esempio di calcolo del fattore J	52
3.5	Esempio di calcolo frattale	54
3.6	Segmentazione con il metodo piramidale	56
3.7	Metodo watershed	57
3.8	Metodo dei contorni attivi	58
3.9	Segmentazione con l'uso di un grafo	59
3.10	Metodo del taglio del grafo	60
3.11	Esempio di segmentazione live-wire	62
3.12	Esemplificazione dell'algoritmo di Zahn [208]	63
3.13	Algoritmo Min-Cut	64
3.14	Applicazione dell'algoritmo SE Min-Cut	65
3.15	Esempio di descrizione sintattica	66
3.16	Esempio di template matching	67
3.17	Esempio di rete neurale artificiale	68
3.18	Uno schema generale per il riconoscimento delle immagini	69
3.19	valutazione degli algoritmi di pattern-recognition	71
4.1	Schema a blocchi dell'algoritmo PRIAR	75
4.2	costruzione dell'ellissoide minimo	78
4.3	mappa di probabilità	79
4.4	PRIAR GUI	81
4.5	PRIAR GUI	83
5.1	Immagine AFM "A", analisi di una superficie a gradino	87
5.2	Immagine AFM "A", risposta del classificatore	87
5.3	Immagine AFM "A": il calcolo di O_1	88

5.4	Immagine AFM "A": il calcolo di O_2	89
5.5	Immagine AFM "A": il calcolo di O_3	90
5.6	Immagine AFM "A": modello calcolato	91
5.7	Immagine AFM "B": analisi di una superficie a gradino	92
5.8	Immagine AFM "B": risposta del classificatore	92
5.9	Immagine AFM "B": O_1	93
5.10	Immagine AFM "B": O_2	94
5.11	Immagine AFM "B": O_3	95
5.12	Immagine AFM "B": modello calcolato	96
5.13	Immagine AFM "C": analisi di una superficie a gradino	97
5.14	Immagine AFM "C": risposta del classificatore	97
5.15	Immagine AFM "C": O_1	98
5.16	Immagine AFM "C": O_2	99
5.17	Immagine AFM "C": O_3	100
5.18	Immagine AFM "C": modello calcolato	101
5.19	Immagine AFM "D": analisi di una superficie a gradino	102
5.20	Immagine AFM "D": risposta del classificatore	102
5.21	Immagine AFM "D": O'_2	103
5.22	Immagine AFM "D": O_3	104
5.23	Immagine AFM "D": modello calcolato	105
5.24	Immagine AFM "D": immagine 3D prima dell'elaborazione	105
5.25	Immagine AFM "D": immagine 3D dopo l'elaborazione	106
5.26	Anfora 1: test in laboratorio	107
5.27	Anfora 2: test in laboratorio	107
5.28	Sito archeologico della Meloria	108
5.29	Sito archeologico della Meloria	109
A.1	mappa di probabilità	132

Elenco delle tabelle

3.1 definizione dei pixel recuperati, della precisione, del silenzio e del rumore . . 70

Introduzione

1.1 Motivazione

I metodi di *image analysis* affrontano oggi il problema di come migliorare la qualità delle immagini a bassa risoluzione e di come realizzarne un riconoscimento semantico. Le crescenti risorse di calcolo oggi disponibili permettono di ottenere i risultati descritti in questo lavoro quando sono coadiuvate con opportuni metodi per l'elaborazione delle immagini. L'individuazione di questi metodi è un campo di ricerca ancora aperto.

Ottenere immagini di qualità è necessario per poter meglio eseguire successive analisi, però per ottenere direttamente immagini o video ad alta risoluzione (*High Resolution HR*) è necessario utilizzare di dispositivi particolari. Il legame che esiste con il tipo di dispositivo di acquisizione diviene eccessivamente vincolante soprattutto in prospettiva di una diffusione sempre maggiore di dispositivi ubiqui, sempre più economici e con caratteristiche che permettono essenzialmente la produzione di immagini di qualità scadente. Una soluzione di compromesso è quella di accettare immagini a bassa risoluzione (*Low Resolution, LR*) e di elaborarle per ottenere immagini ad alta risoluzione. Con alta risoluzione intendiamo qui sia il miglioramento della fedeltà rispetto all'oggetto rappresentato, attraverso l'eliminazione del rumore (*denoising*) e il miglioramento del contenuto, sia l'aumento della densità dei pixel e dei bit per pixel.

Le tecniche per il miglioramento della risoluzione e l'estrazione delle informazioni dalle immagini si chiamano di *super-risoluzione* (Super-Resolution). Intuitivamente, l'immagine ad alta risoluzione si ottiene combinando l'informazione non ridondante contenuta in più immagini a bassa risoluzione. Da un punto di vista algoritmico, i metodi di SR generano una *denoised hyper-resoluted image* (o un insieme di immagini) a partire da una o più immagini LR. Il che non è semplice, dato che le immagini a bassa risoluzione sono spesso degradate dalla *point spread function*¹ (PSF) e dai sistemi di digitalizzazione. Inoltre presentano scarsa informazione spaziale e colorimetrica. La necessità di migliorare la qualità delle immagini a bassa risoluzione spiega perché i metodi

¹ La point spread function è la funzione di trasferimento che descrive il comportamento del dispositivo di acquisizione.

di elaborazione della super-risoluzione siano diventati un'area di ricerca rilevante. Particolarmente difficile e quindi interessante è la super-risoluzione *single-frame*, ovvero la super-risoluzione nel caso in cui l'informazione per il calcolo della nuova immagine ad alta risoluzione proviene da una sola immagine. In generale, gli algoritmi di super-risoluzione si possono infatti classificare in base al numero di immagini in input e in output coinvolte nel processo. Quando si ricava una singola immagine ad alta risoluzione a partire da una singola immagine a bassa risoluzione si parla di super-risoluzione *single-image single-output (SISO)* o *single frame*. I problemi in cui viene impiegata questa metodologia sono relativi alle situazioni in cui è possibile ottenere una sola immagine descrittiva. Quando si ricava una singola immagine ad alta risoluzione a partire da varie immagini a bassa risoluzione si parla di *multiple-image single-output (MISO)* super-risoluzione o di algoritmi di super-risoluzione *statici*. Un esempio tipico è il riconoscimento di una targa in un filmato in bassa risoluzione. Quando si ricavano varie immagini ad alta risoluzione a partire da varie immagini a bassa risoluzione si parla infine di super-risoluzione *multiple-image multiple-output (MIMO)* o *video-to-video*. In ogni caso, si tratta di super-risoluzione dinamica. I casi MISO e MIMO possono essere genericamente descritti come super-risoluzione *multi-frame*.

In questo lavoro è studiato il caso della super-risoluzione *single-frame*. Questa classe di problemi, che permette di lavorare solo su un ristretto numero di informazioni, fornisce la possibilità di entrare in ambiti di applicazione molto estesi: gli scenari infatti spaziano dalle fotografie fatte con un economico smartphone fino a arrivare a immagini acquisite con costosi microscopi a forza atomica (AFM).

In generale, il problema della super-risoluzione è un problema mal posto, nel senso della definizione degli anni '20 (del 900) data da Hadamard [86, 131, 180]. Egli afferma che un modello matematico per un problema fisico (si pensi ad esempio ad un problema al bordo per un'equazione differenziale) deve essere ben posto, vale a dire, deve avere le seguenti proprietà:

- esistenza: il problema ammette una soluzione;
- unicità: esiste al più una soluzione;
- stabilità: la soluzione dipende con continuità dai dati.

I problemi legati alla super-risoluzione sono generalmente problemi mal posti. Questo è vero, in particolare, per la super-risoluzione SISO in quanto l'informazione di partenza è minore rispetto agli altri casi. In altri termini, data un'immagine a bassa risoluzione I esistono molte soluzioni ad alta risoluzione, ovvero, molte immagini \tilde{I}_j che, ridotte, possono generare I .

Per questo motivo in letteratura troviamo solamente alcuni casi specifici in cui si cercano di estendere gli algoritmi di super-risoluzione SISO con reti neurali [214], ad esempio, per il riconoscimento dei volti. Tuttavia la reale efficacia generale di questi algoritmi non è a oggi largamente comprovata eccetto che in casi particolari. Una possibile alternativa è quella di integrare i metodi SR di tipo SISO con i metodi di *pattern-recognition* (PR) e modellazione in grado di calcolare informazioni aggiuntive sugli oggetti rappresentati

per mezzo di una base di conoscenza o *Knowledge-base* (KB). La risoluzione può essere infatti migliorata se si conosce l'oggetto ripreso dall'immagine e se ne può ricavare un modello. Con pattern-recognition ci si riferisce al processo di classificazione dei dati in oggetti e classi, basato sulla sua capacità di rilevare e identificare le strutture distintive degli oggetti rappresentati. In generale, si tratta di una metodologia molto usata in image analysis per affrontare vari problemi tra i quali: il riconoscimento dei caratteri su carta stampata (detti sistemi OCR), il rilevamento e il riconoscimento dei volti, il rilevamento degli oggetti, il loro riconoscimento e la loro classificazione. Dato che dipendono fortemente dalla qualità delle immagini analizzate, le tecniche PR, a loro volta, possono beneficiare dalla super-risoluzione che consente loro di avere maggiore informazione su cui lavorare. Questa combinazione di tecniche si rivela particolarmente utile negli ambienti con un elevato grado di automazione, nei quali i database sono popolati con immagini di grandi dimensioni che precludono un'elaborazione manuale. Infatti un'elaborazione esaustiva accettabile può avvenire solo per mezzo del calcolo automatico. Inoltre questa analisi, finalizzata a arricchire l'informazione di una singola immagine, può essere proficuamente utilizzata nei sistemi retro-azionati per migliorarne l'efficacia.

1.2 Contributi della tesi

In questo lavoro di tesi, che ha permesso lo sviluppo del modello teorico *PRIAR* (Pattern-Recognition Image Augmented Resolution) e della sua implementazione battezzata *tool PRIAR*, sono stati approfonditi ed estesi gli studi teorici e le metodologie di super-risoluzione affrontando il problema da un nuovo punto di vista che porta ad integrare i metodi di super-risoluzione SISO, con quelli di classificazione, pattern-recognition e modellazione. L'uso dei metodi di SR permette di incrementare la densità di informazione delle immagini così da potervi applicare proficuamente metodi di PR per poi arricchire ulteriormente l'informazione delle immagini con i dati presi dalla KB. Con le informazioni contenute nella KB si migliora la risoluzione spaziale, la profondità del colore ed eventuali altri errori causati dal processo di acquisizione: la KB contiene infatti l'informazione del modello dell'oggetto riconosciuto dal PR che viene utilizzata per migliorarne ulteriormente la risoluzione. Il risultato ottenuto è un'immagine ad alta risoluzione nella quale sono infine inserite meta-informazioni utili per fasi di analisi successive.

La metodologia proposta in *PRIAR* viene qui introdotta, descritta e discussa sulla base di diversi casi di studio con lo scopo di dimostrare la generalità del nostro approccio. Lo studio dei casi riguarda principalmente le immagini provenienti da microscopi a scansione di sonda (*Scanning Probe Microscopy*, SPM)². Questi microscopi generalmente acquisiscono immagini topografiche di molecole che giacciono nello spazio di pochi nanometri. Tipicamente la risoluzione di queste immagini è di 256×256 pixel con 8bpp (bit per pixel). Le immagini ottenute, per la natura dei materiali e del funzionamento del

² L'AFM è un particolare microscopio SPM inventato da Gerd Binnig, Calvin Quate e Christoph Gerber nel 1986.

sistema di acquisizione, sono uniche e quindi i risultati teorici di questa tesi sono una soluzione naturale al problema relativo al miglioramento di queste immagini. Queste topografie generalmente sono rappresentate utilizzando immagini monocromatiche. Inoltre l'introduzione dei microscopi SPM ha aperto negli anni '80 del '90 la possibilità di osservare e manipolare la materia su scala atomica, rendendo possibile il miglioramento della conoscenza e delle tecnologie per oggetti della dimensione di pochi nanometri (nell'ambito quindi delle nanotecnologie). In questo lavoro ci concentriamo infatti su immagini acquisite tramite microscopi a forza atomica (*Atomic Force Microscopy*, AFM), un tipo particolare di SPM. Per il corretto funzionamento dei microscopi SPM, ad oggi, è necessaria una procedura di messa a punto che viene eseguita solo manualmente. Tuttavia, i risultati di questa tesi sono stati studiati anche in funzione di ulteriori sviluppi per l'automazione del processo di acquisizione.

Un secondo banco di prova per il tool PRIAR è stato il test con le immagini subacquee. Queste immagini sono riprese in tempo reale da veicoli sottomarini autonomi (*Autonomous Underwater Vehicle*, AUV)³, utilizzando telecamere subacquee. Dalle immagini subacquee sono state ricavate delle immagini topografiche sulle quali è stato applicato efficacemente il metodo PRIAR. La PSF delle telecamere subacquee è fortemente influenzata dalle interfacce acqua-vetro e vetro-aria che deve attraversare la luce prima di essere acquisita. A questo problema si vanno a sommare la scarsa illuminazione che troviamo negli ambienti marini, l'aberrazione cromatica verso il colore blu e i disturbi intrinseci degli ambienti marini (ad esempio bolle d'aria, corpuscoli in sospensione, ecc.).

1.3 Organizzazione della tesi

Il **capitolo 2** introduce i metodi di super-risoluzione, evidenziando quali sono le problematiche classiche affrontate con le acquisizioni ottenute con strumenti ottici e estendendo questa analisi ai microscopi a forza atomica in quanto oggetto del lavoro sperimentale. Il capitolo prosegue mettendo a confronto i metodi single-frame attualmente in uso.

Il **capitolo 3** presenta lo stato dell'arte dei metodi di pattern-recognition, con particolare attenzione ai risultati fino a oggi ottenuti nel segmentare un'immagine e nel riconoscere oggetti specifici all'interno dell'immagine.

Il **capitolo 4** presenta il frutto dell'analisi teorica dello stato dell'arte presentato e le idee che hanno portato a comporre la pipeline multi-modulare del PRIAR dove i metodi di PR e SR lavorano congiuntamente. La presentazione si conclude con la descrizione del tool PRIAR. I risultati di questo capitolo sono basati sulle idee sviluppate in [42, 34, 43, 40, 41].

Il **capitolo 5** presenta alcuni risultati sperimentali ottenuti con PRIAR in base ai risultati teorici raggiunti presentando l'analisi di diversi casi testati. La sperimentazione è stata eseguita su immagini da AFM e immagini da riprese subacquee. I risultati di questo capitolo sono basati sulle idee sviluppate in [164, 156, 155].

³ Gli AUV sono sottomarini robotici in grado di muoversi e compiere azioni autonomamente nell'ambiente marino.

Super-risoluzione

Il presente capitolo introduce i metodi di super-risoluzione, evidenziandone lo stato dell'arte e le problematiche.

2.1 Introduzione

La storia della codifica elettronica delle immagini risale al 1851 [98]. Nel 1851 il britannico Frederick Collier Bakewell mostrò al *World's Fair* a Londra un prototipo che trasmetteva un'immagine per mezzo una linea che sfruttando cavi telegrafici. Questo dispositivo era sostanzialmente la prima macchina per FAX: il trasmettitore convertiva l'immagine in un segnale elettrico e che il ricevitore stampava su uno speciale tipo di carta la cui superficie era cosparsa di uno reagente chimico sensibile all'applicazione di una differenza di potenziale. Per la scansione veniva utilizzato un dispositivo simile a uno scanner a tamburo. Sulla macchina trasmittente il disegno era impresso intorno al cilindro dello scanner mentre sulla macchina ricevente era collegato una penna di materiale conduttore utilizzato per *copiare* il disegno sorgente. L'intensità di corrente scuriva con diversa intensità la carta elettro-sensibile. In figura 2.1 una riproduzione del prototipo.

La prima immagine digitale fu creata nel 1920 con il metodo Bartlane. Il metodo prese il nome dai due inventori britannici, Harry G. Bartholomew and Maynard D. McFarlane. Questo metodo utilizzava una serie di negativi su lastre di zinco. Le lastre vengono esposte per specifici intervalli di tempo imprimendole in diversa misura. La prima versione di questo metodo utilizzava cinque lastre, corrispondenti a cinque livelli di quantizzazione. La prima immagine digitale creata con una tecnologia confrontabile con quella moderna risale al 1957 quando Russel Kirsch sviluppò il suo dispositivo al *National Bureau of Standards*. Il dispositivo consisteva essenzialmente in uno scanner a tamburo con un tubo fotomoltiplicatore che produceva immagini digitali memorizzabili in un calcolatore. Le immagini digitali non sono generate solo da dispositivi di acquisizione ottici, in particolare menzioniamo le immagini da microscopi a forza atomica (AFM) [38] che sono trattate in questo lavoro di ricerca.

Gli elementi che costituiscono un sistema per l'elaborazione delle immagini sono:

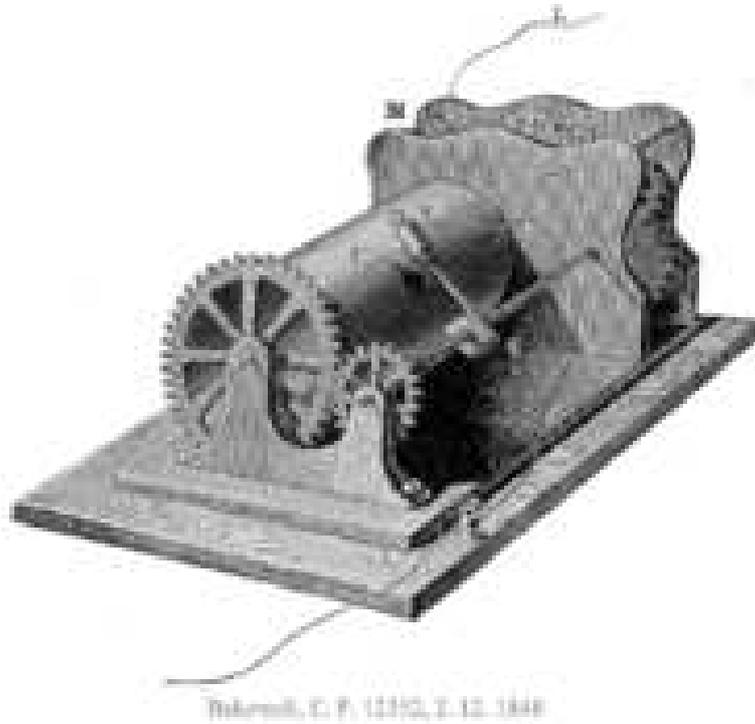


Figura 2.1: il prototipo di Frederick Collier Bakewell

- un sistema di acquisizione (uno scanner, una macchina fotografica, una videocamera, un microscopio AFM, ecc.)
- un sistema di elaborazione e memorizzazione dell'immagine

Alcune immagini, come quelle acquisite da certe tipologie di videocamere o microscopi, sono per loro natura di bassa qualità. Un campo di ricerca molto attivo riguarda lo studio di come immagini di bassa qualità (es. figura 2.2 a) possono essere trasformate in immagini di qualità migliore (es. figura 2.2 b).

Il termine super-risoluzione nasce proprio per etichettare quel processo in cui un'immagine a bassa risoluzione viene migliorata. Talvolta il processo prende il nome di miglioramento della risoluzione delle immagini.

2.2 Definizione della risoluzione di una immagine

Poniamo di avere una scena reale s e la sua rappresentazione digitale I . Della rappresentazione digitale definiamo:

- *la risoluzione spaziale*: l'informazione di un'immagine I nello spazio viene rappresentata tramite oggetti indivisibili detti pixel. Un pixel è quindi la più piccola informazione

2.3. PERCHÉ È IMPORTANTE MIGLIORARE LA RISOLUZIONE DI UN'IMMAGINE?

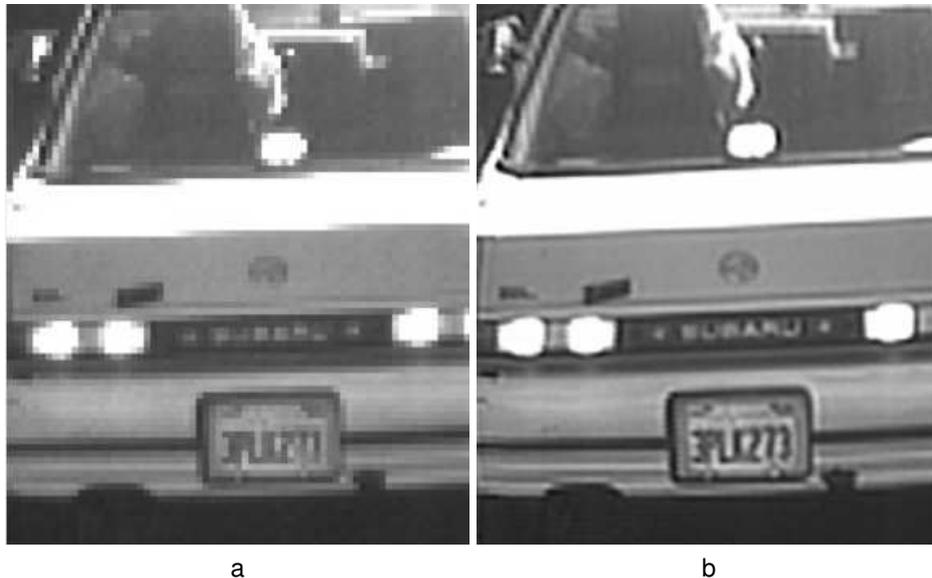


Figura 2.2: una immagine in bassa risoluzione e una super-risolta, in (a) l'immagine è in bassa risoluzione, in (b) la medesima immagine super-risolta dopo il processo di elaborazione [116]

spaziale che possiamo avere di un'immagine. Il numero dei pixel che compongono un'immagine è una quantità intera ed è data dal prodotto dei pixel distribuiti sulle colonne C e il numero dei pixel distribuiti sulle righe R . Una misura caratteristica di risoluzione spaziale è 640×480 , ciò vuol dire che l'immagine è scomposta in una griglia di 640 righe e 480 colonne.

- *la quantizzazione o profondità di colore*: la profondità di colore in un'immagine I descrive con quanti bit viene codificata la gamma dinamica dell'immagine registrata. Esempi tipici di profondità di colore sono 8 bit per immagini generate da particolari microscopi oppure 24 bit (8 bit per rappresentare il rosso, 8 bit per rappresentare il verde, 8 bit per rappresentare il blu) per le immagini utilizzate per la produzione della grafica.

2.3 Perché è importante migliorare la risoluzione di un'immagine?

L'aumento della risoluzione di un'immagine può portare a diversi vantaggi, ad esempio per la compressione dell'informazione oppure per l'estrazione di informazione dalle immagini. Infatti gli strumenti di acquisizione dell'immagine permettono una risoluzione spaziale e una quantizzazione insufficiente alla fruizione delle immagini. Questo avviene soprattutto in campo scientifico, non dimentichiamoci però dei vantaggi che tale tecnologia porta anche in altri campi, come ad esempio nella cinematografia, permettendo una

migliore visione di film registrati con risoluzioni più basse rispetto a quelle che si possono raggiungere oggi.

Si tenga presente anche che limitarsi ad aumentare il numero di pixel o i sensori a parità di dimensioni non solo migliora la risoluzione dell'immagine, ma contemporaneamente causa un incremento del rumore elettrico [174] che a sua volta inficia la risoluzione. Oltre ai limiti imposti dai sensori ottici troviamo altri limiti imposti da altre tecnologie come l'AFM.

2.4 Acquisizione ottica delle immagini e principali artefatti

Le immagini digitalizzate sono affette principalmente da tre tipi di disturbi che ne riducono la qualità [173] :

- aliasing,
- blurring,
- rumore.

Questi tipi di disturbi danno origine a quella che si chiama PSF (Point Spread Function).

2.4.1 Aliasing

Il fenomeno dell'aliasing [110, 146, 75, 130] è legato alla frequenza di campionamento [165, 104], si presenta quando la frequenza di campionamento è bassa rispetto al fenomeno che viene acquisito. La conseguenza della presenza dell'aliasing è la perdita di dettagli tali da poter ricostruire un'immagine della scena priva di aberrazioni geometriche.

In dettaglio, il problema dell'aliasing può dar luogo:

- al problema della scalettatura (figura 2.3);
- all'effetto Moiré (figura 2.4).

2.4.2 Blurring

Il blurring consiste principalmente nella scorretta ricezione del digitalizzatore delle inter-distanze tra gli oggetti presenti sulla scena. La conseguenza di questo tipo di aberrazione è la deformazione dell'immagine. Il blurring di un'immagine è suddiviso nelle seguenti tipologie generali:

- distorsione
- aberrazione sferica
- astigmatismo
- campo di curvatura Petzval
- sfocatura

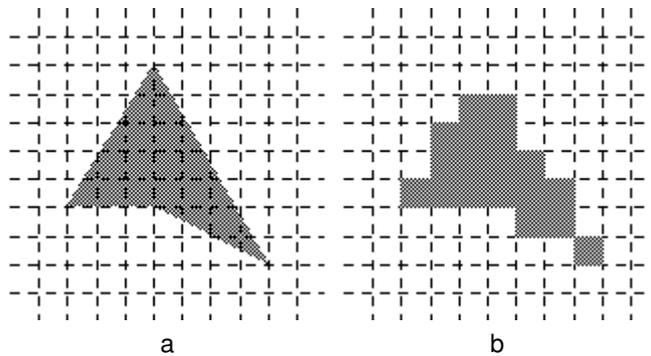


Figura 2.3: l'aliasing e problema della scalettatura [110, 146, 75, 130], in (a) l'immagine originale, in (b) l'immagine affetta da aliasing



Figura 2.4: l'aliasing e effetto Moiré, in (a) un'immagine con un lieve effetto Moiré e in (b) un'immagine con un effetto Moiré accentuato [168]

Distorsione

Il problema della distorsione riguarda la proiezione non regolare delle curve sul sensore che digitalizza la scena. La tipica distorsione è di due tipi (o una composizione tra i tipi): a barile (figura 2.5 b) o a cuscino (figura 2.5 c).

Aberrazione sferica

L'aberrazione sferica [29] è un difetto di natura geometrica derivante dalla forma delle superfici rifrangenti o riflettenti e dalla posizione, rispetto a esse, degli oggetti. Data una sorgente puntiforme monocromatica sull'asse di un sistema ottico, se l'apertura angolare

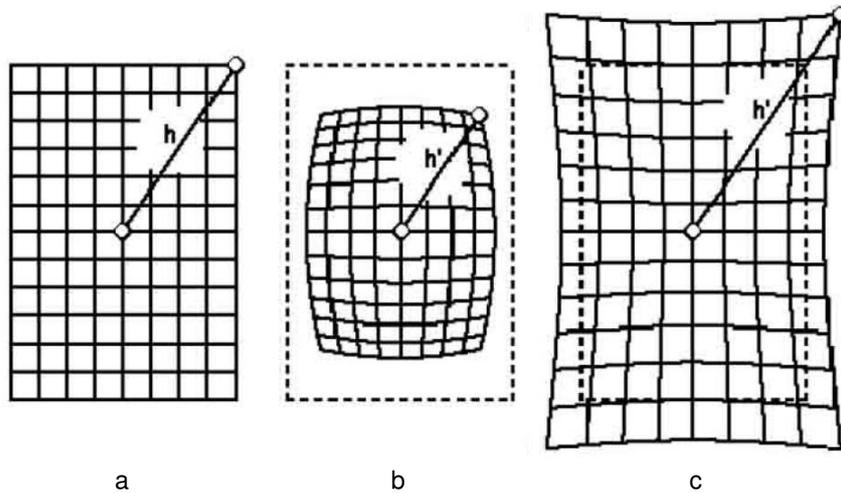


Figura 2.5: esempi di distorsioni di un'immagine, (a) immagine originale (b) distorsione a botte (c) distorsione a cuscino [15]

del sistema rispetto alla sorgente non è sufficientemente piccola, i raggi emergenti non concorrono più in un punto, ma involuppano una superficie (la caustica), che in ogni caso, per motivi di simmetria, è una superficie di rotazione intorno all'asse ottico (figura 2.6).

La caustica prende il nome di catacaustica e diacaustica per sistemi, rispettivamente, catottrici, ossia riflettenti, e diottrici, ossia rifrangenti. Si distinguono un'aberrazione sferica longitudinale (o assiale) e un'aberrazione sferica trasversale (all'asse). Per realizzare un sistema esente da questa aberrazione, cioè un sistema aplanatico, si sfrutta il fatto che le aberrazioni di sfericità di una lente divergente sono in senso opposto a quelle di una lente convergente, per cui è possibile correggere l'aberrazione di sfericità per un dato punto dell'asse mediante un'opportuna combinazione di una lente di un tipo con una lente dell'altro tipo. Condizioni necessarie e sufficienti per conseguire tale risultato sono la condizione di Herschel per l'aberrazione sferica assiale e la condizione di Abbe, o condizione dei seni, per l'aberrazione sferica trasversale. Il punto per il quale conviene eliminare l'aberrazione di sfericità varia da strumento a strumento: per es., negli obiettivi per telescopi rifrattori (cannocchiali) è il punto all'infinito dell'asse, mentre in quelli per microscopi è un punto molto vicino alla prima lente dell'obiettivo. Un aplanatismo parziale, cioè una diminuzione dell'aberrazione sferica, si consegue in ogni caso con l'uso di un diaframma che escluda i raggi marginali; la diminuzione riesce tanto più rilevante quanto più piccolo è il foro del diaframma, a scapito però della luminosità delle immagini. In figura 2.7 un esempio di aberrazione sferica.

Coma

La coma [91] è un tipo di distorsione che si presenta quando in un sistema ottico convergente i raggi di luce extra-assiali sono focalizzati su differenti piani: questo crea sul-

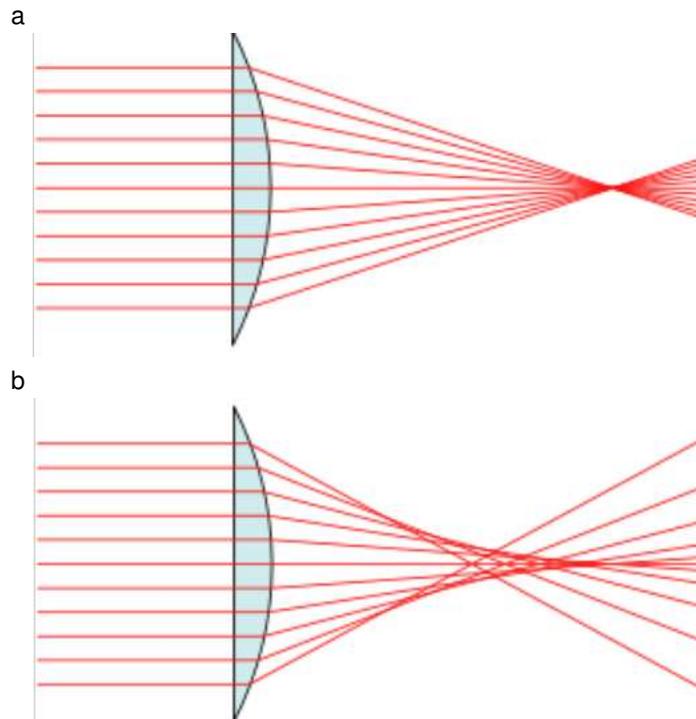


Figura 2.6: aberrazione sferica, in (a) la rappresentazione di un obiettivo perfetto senza aberrazione sferica dove tutti i raggi entranti sono concentrate nel punto focale e in (b) un obiettivo con aberrazione sferica dove i diversi raggi non sono focalizzati dopo la lente in un punto focale

l'immagine degli anelli luminosi. Questi anelli tendono in parte a sovrapporsi creando di conseguenza un artefatto luminoso che ricorda alla lontana una cometa, da cui il termine coma (coma è il termine greco che indica la coda della cometa) . La coda della cometa può essere rivolta verso l'asse ottico (coma interna) o verso i bordi della lente (coma esterna).

Nella figura 2.8 è rappresentato questo fenomeno con i tre raggi (p' , p'' , p''') che generano tre cerchi luminosi leggermente sovrapposti: la forma è quella di una cometa.

La coma tenderà a diventare più marcata quanto più l'oggetto che intendiamo riprendere è spostato lateralmente rispetto all'asse del sistema. Indicando con θ l'angolo che il raggio di luce proveniente dall'oggetto forma con detto asse, i raggi che passano per il centro dell'ottica che ha una distanza focale pari ad f verranno focalizzati alla distanza $f \cdot \tan(\theta)$. Maggiore è l'angolo θ maggiore è l'importanza di questo problema.

La coma, essendo legata alla curvatura della lente, si riduce impiegando lenti asferiche anche se mai sarà possibile giungere ad una soluzione definitiva: esistono delle coma particolarmente complesse da eliminare come la coma ellittica che genera degli anelli non circolari.

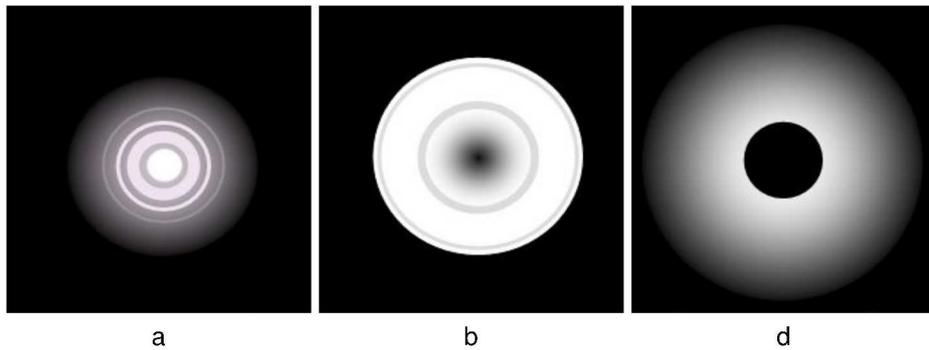


Figura 2.7: un esempio di aberrazione sferica, (a) l'immagine a fuoco (b) aberrazione extrafocale (c) aberrazione intrafocale

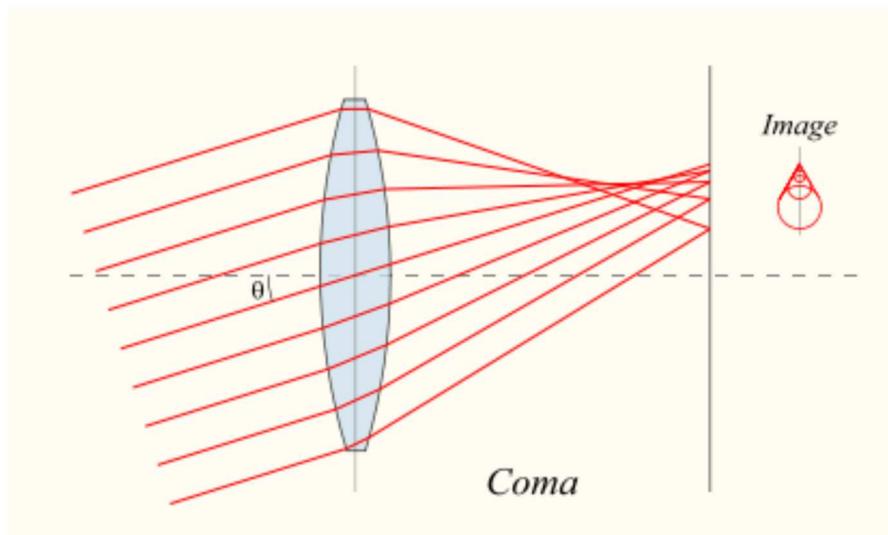


Figura 2.8: la coma [91].

In figura 2.9 a l'immagine non affetta dal fenomeno, in figura 2.9 b l'immagine affetta dal fenomeno.

Astigmatismo

L'astigmatismo [87], dal greco α -(a-) che significa "senza" e $\sigma\tau\iota\gamma\mu\alpha$ (stigma), segno, punto, è un'aberrazione ottica presente in un sistema singolo o composto di lenti, dagli obbiettivi all'occhio; i raggi che si propagano in due piani intersecanti l'asse ottico ad angoli diversi, ad esempio perpendicolari tra loro, hanno fuochi differenti e proiettando l'immagine di un punto, lo stesso risulta deformato. Se un sistema ottico con astigmatismo viene utilizzato per formare l'immagine di una croce, le linee orizzontali e verticali saranno a fuoco a due distanze diverse (figura 2.10).



Figura 2.9: la coma [91], in (a) l'immagine non affetta da coma e in (b) l'immagine affetta da coma

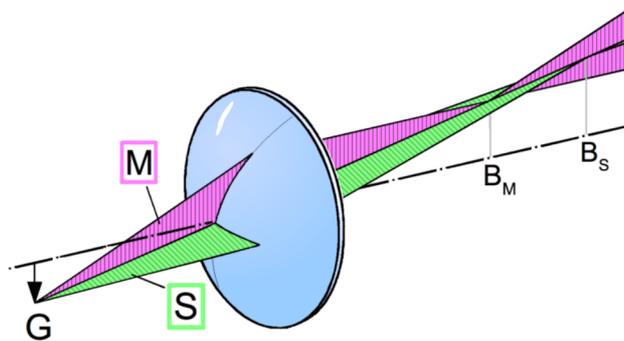


Figura 2.10: il problema dell'astigmatismo ovvero una lente convessa con raggi emessi da un oggetto fuori asse, i piani sagittale e meridionale sono contrassegnati come rispettivamente "M" e "S"; a causa dell'astigmatismo (fortemente esagerato), i piani di immagine "B" sono diversi per i raggi nei piani sagittale e meridionale [87].

Visivamente (figura 2.11) un'immagine prodotta da un sistema astigmatico apparirà più o meno evidentemente deformata (un punto risulterà ellittico, con parte del bordo sfocato) o sdoppiata, con un'immagine principale a fuoco ed una secondaria, meno incisa, sfocata e traslata rispetto a quella principale, a renderne meno intelligibile la lettura. L'aberrazione colpisce potenzialmente tutti i sistemi ottici. L'entità dell'aberrazione è correlata a diversi fattori: dalla distanza del diaframma dal diottro, alla potenza (focale) della lente, a un fattore indice della curvatura di campo e a un fattore che tiene conto della coma.

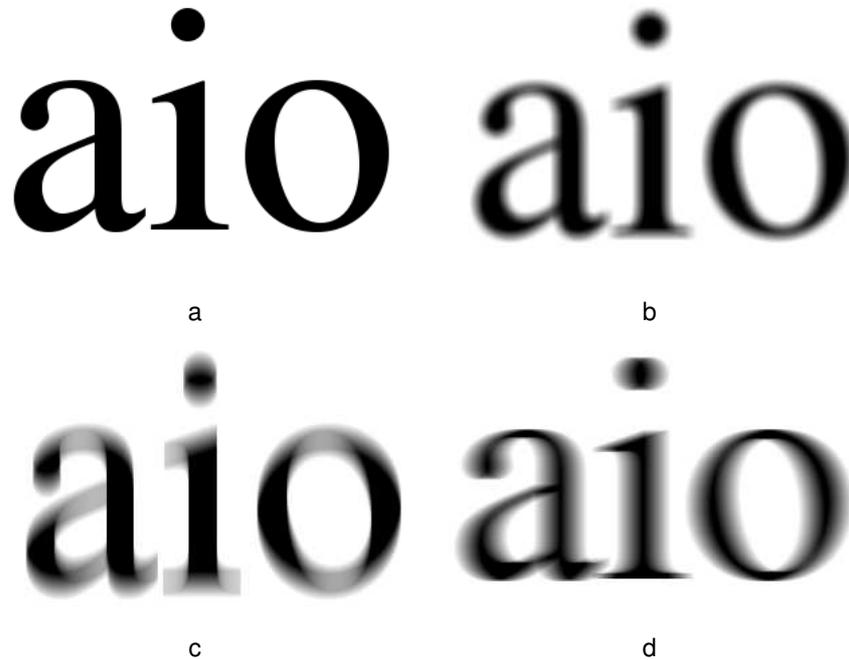


Figura 2.11: esempio di astigmatismo [91], (a) l'immagine originale (b) astigmatismo orizzontale e verticale (c) astigmatismo verticale (d) astigmatismo orizzontale

Campo di curvatura Petzval

Il problema del campo di curvatura Petzval è legato alla struttura sferica della lente: i raggi luminosi (sempre per le differenti lunghezze d'onda) si focalizzano su di una superficie sferica piuttosto che piana. Questa particolare focalizzazione è un'aberrazione di tipo monocromatico chiamata appunto curvatura di campo e che di solito si accoppia con l'astigmatismo.

Per misurare la curvatura di campo si usa il valore denominato somma di Petzval che altro non è se non il raggio di curvatura della superficie sferica (figura 2.12). Questo valore può essere positivo o negativo a seconda che la superficie sia concava o convessa. Più la somma di Petzval è piccola migliore è l'obiettivo. Purtroppo non è possibile utilizzare delle lenti totalmente diritte (quindi con valore della somma pari a zero) in quanto andremmo ad introdurre un ulteriore difetto, l'astigmatismo: in fase di progettazione di una lente bisognerà trovare un compromesso tra i due difetti che, purtroppo, non potranno mai contemporaneamente essere annullati del tutto.

La curvatura di campo si può correggere introducendo astigmatismo.

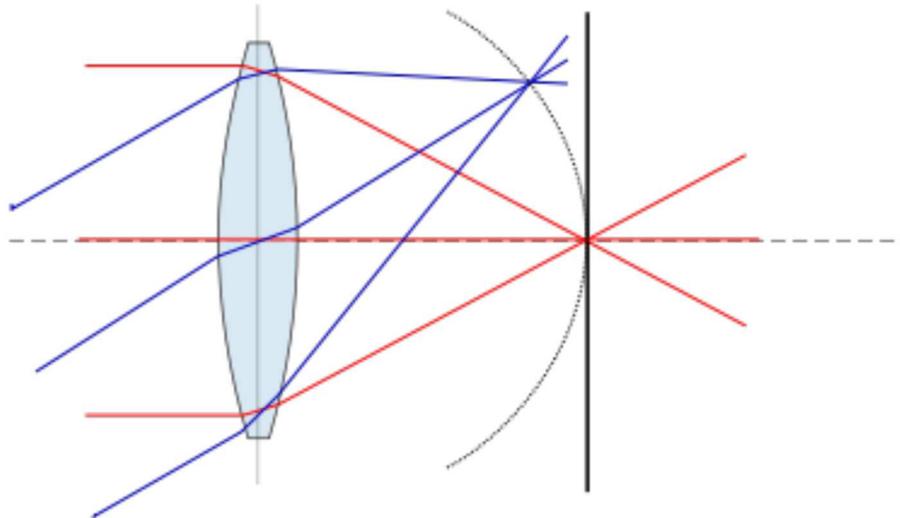


Figura 2.12: campo di curvatura Petzval, il piano dell'immagine (l'arco) si discosta da una superficie piana (la linea verticale) [87].

Sfocatura

La sfocatura è un fenomeno presente nei sistemi di digitalizzazione un cui è l'immagine attraversa un sistema di lenti (o con modello di funzionamento a esso associabile come l'occhio). Generalmente si manifesta con una bassa nitidezza e un basso contrasto dell'immagine. Il termine messa a fuoco (di una lente) deriva probabilmente dall'effetto che produce l'immagine del Sole focalizzato ad esempio su di un foglio di carta, tramite una lente di ingrandimento o altro rifrattore positivo: la concentrazione dei raggi solari riscaldano il foglio e gli danno letteralmente fuoco. Nei casi in cui la messa a fuoco del Sole sul foglio fosse "errata", la concentrazione inferiore dei raggi solari ridurrebbero il potere di riscaldamento, annullando l'effetto (fuori fuoco o non a fuoco).

La sfocatura [91] si manifesta quando si verifica una traslazione lungo l'asse ottico lontano dal piano o superficie di miglior fuoco. In generale la sfocatura riduce la nitidezza e il contrasto dell'immagine. Quando si riduce il contrasto dell'immagine, i bordi che dovrebbero essere rappresentati in modo netto sono invece rappresentati da transizioni graduali di colore: ad esempio, alcuni dettagli della scena potrebbero apparire offuscati a causa della sfocatura divenendo illeggibili. Quasi tutti i dispositivi ottici per l'acquisizione di immagini incorporano una qualche forma di regolazione della messa a fuoco per ridurre al minimo defocus e massimizzare la qualità dell'immagine. In figura 2.13 (a) abbiamo una immagine a fuoco, in figura 2.13 (b) la medesima immagine catturata con una errata messa a fuoco.



Figura 2.13: sfuocatura, in (a) una immagine a fuoco e in (b) una immagine sfuocata [91].

2.4.3 Rumore

Il rumore registrato durante la digitalizzazione di un'immagine [82, 172, 87, 80, 94, 47, 100] è un rumore causato principalmente:

- dalla natura discreta della carica elettrica
- dal surriscaldamento dei circuiti elettrici

Il rumore generato dalla carica elettrica

La natura discreta della carica elettrica genera un rumore con distribuzione di tipo di Poisson. Questo tipo di rumore può essere registrato quando un numero finito di particelle che trasportano energia (ad esempio elettroni in un circuito elettronico o fotoni in un dispositivo ottico) è abbastanza piccolo per dare origine a fluttuazioni statistiche rilevabili in una misurazione.

Dal momento che quando il numero degli eventi (N) è elevato la distribuzione di Poisson si avvicina a una distribuzione normale, nel caso di un rivelatore ottico questo rumore si avvicinerà alla distribuzione normale per un gran numero di fotoni raccolti. Il rapporto segnale-rumore è funzione di N e quando N diviene piccolo deve essere controllato con attenzione (eq. 2.1).

$$SNR = \sqrt{N} \quad (2.1)$$

In figura 2.14 una simulazione del rumore generato dalla carica elettrica al verificarsi di un basso numero di eventi. Nella simulazione abbiamo:

- (a) figura campione, 100000 fotoni per pixel
- (b) immagine con 10000 fotoni per pixel
- (c) immagine con 1000 fotoni per pixel

2.4. ACQUISIZIONE OTTICA DELLE IMMAGINI E PRINCIPALI ARTEFATTI

- (d) immagine con 100 fotoni per pixel
- (e) immagine con 10 fotoni per pixel
- (f) immagine con 1 fotone per pixel
- (g) immagine con 0.1 fotoni per pixel
- (h) immagine con 0.01 fotoni per pixel
- (i) immagine con 0.001 fotoni per pixel

Nella simulazione notiamo la rapida perdita di qualità dell'immagine a partire dalla figura 2.14 (e).

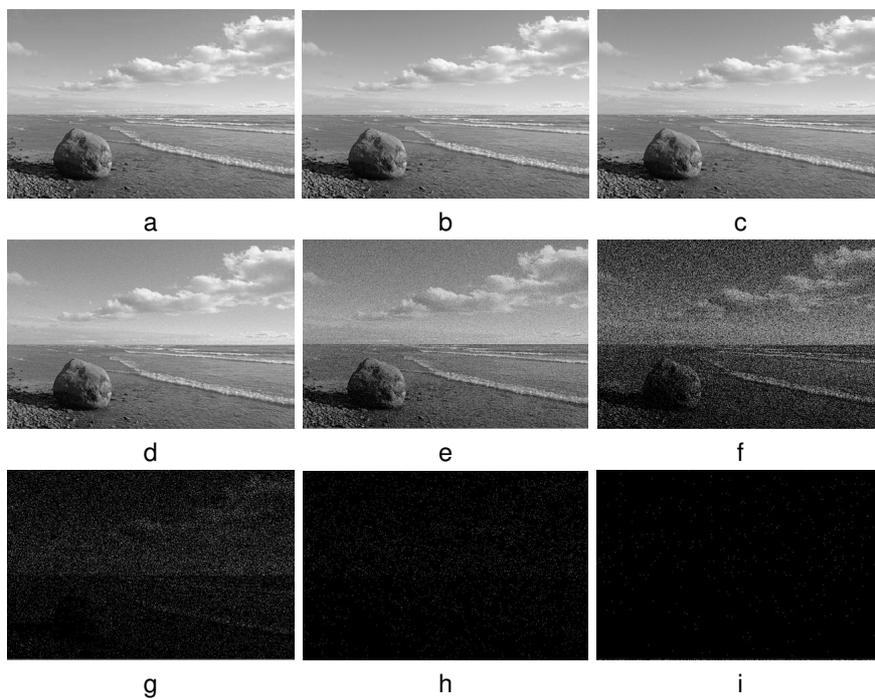


Figura 2.14: rumore elettrico, da (a) a (i) un'immagine in cui aumenta il rumore generato dalla carica elettrica

Il rumore termico

Il rumore termico (detto anche rumore di Johnson o rumore di Nyquist) è il rumore più comune nei dispositivi elettronici. Nasce dal calore scaturito dall'attrito delle particelle cariche in un elemento che offre una resistenza al loro flusso. L'equazione 2.2 descrive la potenza del rumore termico

$$P_T(rms) = k \times T \times \Delta f \quad (2.2)$$

dove:

- k è la costante di Boltzman
- T è la temperatura
- Δf è il range di funzionamento in frequenza

L'uso di materiali super-conduttori o il funzionamento dei dispositivi a bassa temperatura può considerevolmente attenuare il problema del rumore termico.

2.5 Acquisizione AFM delle immagini e principali artefatti

Le immagini da AFM [44, 38, 193, 151, 95, 79, 128] sono acquisite sfruttando un principio diverso rispetto a quelle acquisite da dispositivi ottici.

2.5.1 Scansione in un AFM

Per trattare problemi legati agli AFM vengono qui descritti i nomi delle componenti più importanti e i principi di funzionamento, per un maggiore approfondimento consultare la bibliografia.. In figura 2.15 vediamo le parti principali della testina del microscopio. La

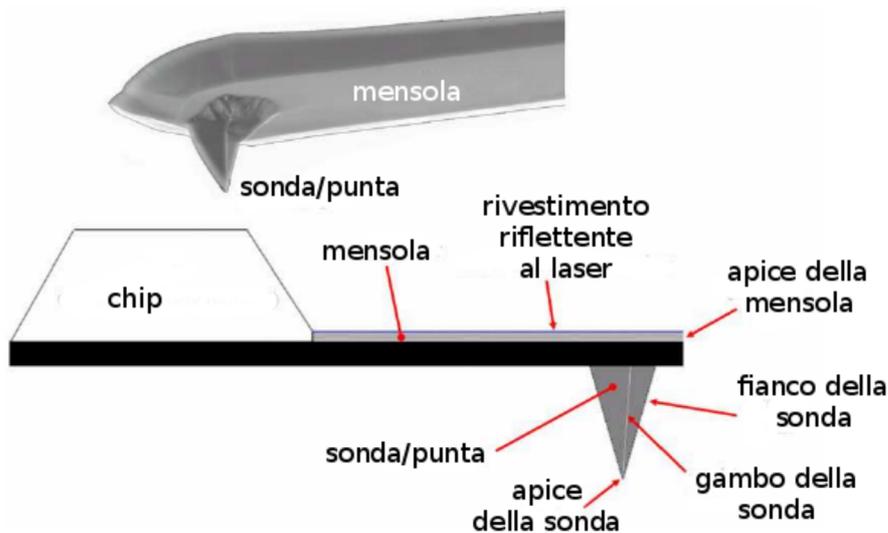


Figura 2.15: la testina in un AFM [128]

testina esplora una superficie cercando di prendere con regolarità la topografia del campione esaminato. In figura 2.16 un esempio. Generalmente la scansione di una superficie avviene lungo l'asse x e l'asse y che prendono il nome di direzione di scansione veloce e direzione di scansione lenta. Ogni profilo generalmente viene scansionato due volta: una in avanti e l'altra nella direzione opposta. Questi due movimenti sono generalmente indicati con scansione o tracciamento e scansione all'indietro o tracciamento all'indietro.

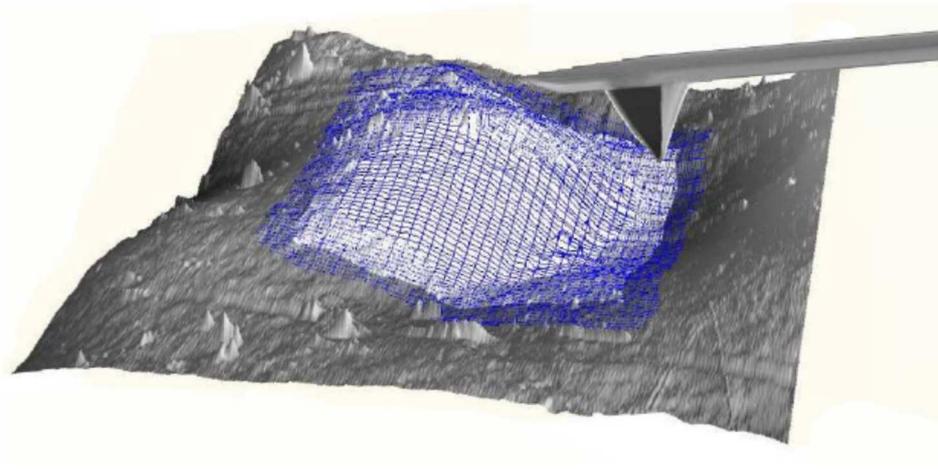


Figura 2.16: l'acquisizione topografica di un AFM [128]

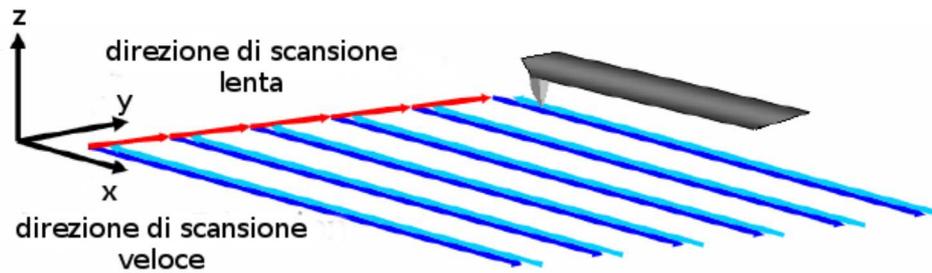


Figura 2.17: le direzioni di scansione di un AFM [128]

In figura 2.17 viene illustrata la tecnica di scansione. La superficie viene caratterizzata da una matrice di punti (generalmente rappresentati come pixel). La posizione (i, j) nella matrice ha una corrispondenza biunivoca con le coordinate (x, y) dell'immagine. Il valore memorizzato in $P(i, j)$ corrisponde all'altezza del campione nel punto corrispondente.

2.5.2 Artefatti in un AFM

I principali artefatti in un microscopio a forza atomica causati dal sistema di scansione sono:

- lo scaling

- il crosstalk
- la non linearità, isteresi, creep e inarcatura
- l'overshoot

Il principale tipo di artefatto in un microscopio a forza atomica causato dalla sonda si verifica quando le dimensioni della sonda non sono compatibili con il campione acquisito: in questo caso il risultato è una convoluzione tra il movimento della sonda e la superficie del campione.

Durante l'acquisizione di una immagine con un AFM anche il rumore ambientale, come il semplice camminare nelle vicinanze del microscopio, causa artefatti. In generale anche il più piccolo rumore nell'ambiente, genera rumore nell'informazione acquisita, per arginare questo problema gli AFM devono essere installati in apposite strutture.

Scaling

L'AFM come visto sposta la testina a destra e sinistra lungo due assi principali. I movimenti sono controllati elettronicamente: a una certa tensione applicata corrisponde una certa posizione nello spazio della testina. I sistemi non retro-azionati per portare la testina in una posizione le applicano una tensione. I sistemi retro-azionati sono dotati di un controllo in retroazione sul piano (x, y) e la posizione della punta è monitorato rispetto ad un sistema di riferimento. In [45] Danzebrink valuta la precisione di questi sistemi: nei sistemi non retro-azionati la linearità dello spostamento della testina può arrivare al massimo al 10% mentre nei sistemi retro-azionati è inferiore all'1%. Lo stesso autore [45] menziona anche i microscopi metrologici la cui linearità di spostamento della testina è di circa lo 0,01%.

Per tutti questi strumenti, in maggior misura per quelli non retro-azionati, la posizione misurata non corrisponde alla misura della testina. Questo porta a una distorsione che è generalmente proporzionale all'ampiezza dell'area misurata e può essere descritta da un fattore di scaling lineare. La distorsione risulta essere dell'ordine di qualche punto percentuale per gli strumenti a anello aperto¹ e inferiore all'1% negli strumenti a anello chiuso. In figura 2.18 sono simulati un pattern piramidale e l'acquisizione di uno strumento con una distorsione del +9,0%, -4,6% e del +0,1% per gli assi x , y e z . I segni positivo e negativo indicano un allargamento o un restringimento lungo gli assi di riferimento. Con una opportuna calibrazione dello strumento è possibile approssimare con maggiore efficacia le corrette posizioni come descritto nell'equazione 2.3.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_{xx'} & 0 & 0 \\ 0 & c_{yy'} & 0 \\ 0 & 0 & c_{zz'} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (2.3)$$

¹ la locuzione *anello aperto* può essere usata come sinonimo di sistema non retro-azionato come la locuzione *anello chiuso* può essere usata come sinonimo di sistema retro-azionato.

2.5. ACQUISIZIONE AFM DELLE IMMAGINI E PRINCIPALI ARTEFATTI

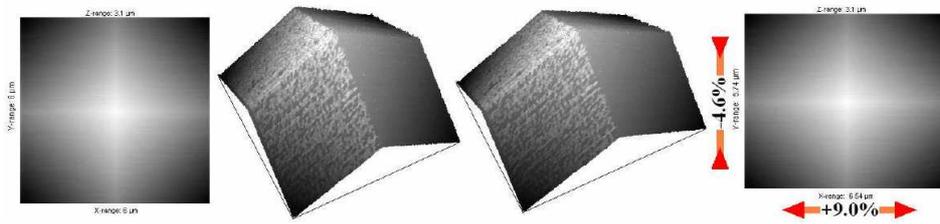


Figura 2.18: la distorsione in una struttura piramidale [128]

Crosstalk

La punta è soggetta a errori che portano a movimento non ortogonali rispetto agli assi x e y . Questo accade quando i movimenti lungo i tre assi non sono correlati e il movimento lungo uno di questi assi provoca uno spostamento lungo un asse non interessato al movimento. Questo tipo di problema si manifesta come un errato allineamento angolare tra il sistema di riferimento dello strumento e il sistema di riferimento ideale. Anche in questo caso, gli strumenti a anello aperto sono più soggetti a questo tipo di errore mentre gli strumenti a anello chiuso lo sono di meno. In figura 2.19 viene mostrato un esempio di artefatto di tipo cross-talk sfruttando la piramide di figura 2.18. Il crosstalk si manifesta con un errore sugli angoli. Con una opportuna calibrazione dello strumento è possibile

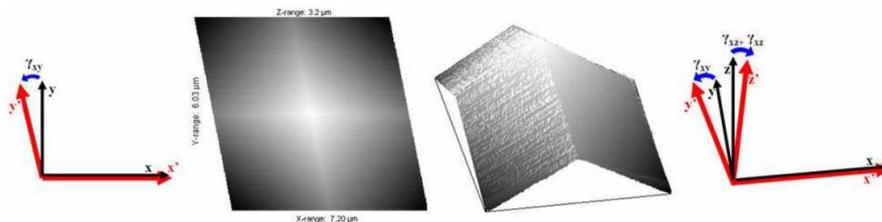


Figura 2.19: disallineamento angolare tra gli assi cartesiani dello strumento e il riferimento ideale, gli angoli γ_{xy} , γ_{xz} e γ_{yz} rappresentano rispettivamente la deviazione dall'ortogonale dai piani x, y, x, z e y, z [128]

approssimare con maggiore efficacia le corrette posizioni come descritto nell'equazione 2.4.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & c_{xy'} & c_{xz'} \\ 0 & 1 & c_{yz'} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (2.4)$$

Non linearità, isteresi, creep e inarcatura

Il problema della non linearità colpisce soprattutto i microscopi a anello aperto in quanto il loro sistema di controllo è basato su sistemi non lineari. I sistemi retro-azionati,

diversamente, sono più immuni a questo tipo di artefatti.

I sistemi ad anello aperto sono generalmente utilizzati per il controllo dei dispositivi piezoelettrici che soffrono del problema dell'isteresi e il creep.

A causa della differenza nelle proprietà dei materiali e delle dimensioni di ogni elemento piezoelettrico, gli scanner AFM rispondono in modo diverso quando viene applicata una differenza di potenziale alle testine. Questa diversa risposta può essere misurata in termini di sensibilità, definita come il rapporto tra il movimento del dispositivo piezoelettrico e la tensione applicata. In altre parole, il dispositivo piezoelettrico si distende o si contrae (come un muscolo) a seconda della tensione che gli si applica. La sensibilità è non lineare, infatti gli scanner basati su dispositivi piezoelettrici mostrano una maggiore sensibilità su valori di fondo scala piuttosto che con valori di inizio scala. Questo fenomeno fa sì che i materiali manifestino una isteresi. In figura 2.20 viene mostrata la relazione tra la tensione con cui si pilota la testina e la posizione della medesima. Inoltre la sensibilità del materiale piezo-elettrico decresce con il tempo e l'uso. Il creep può essere visto

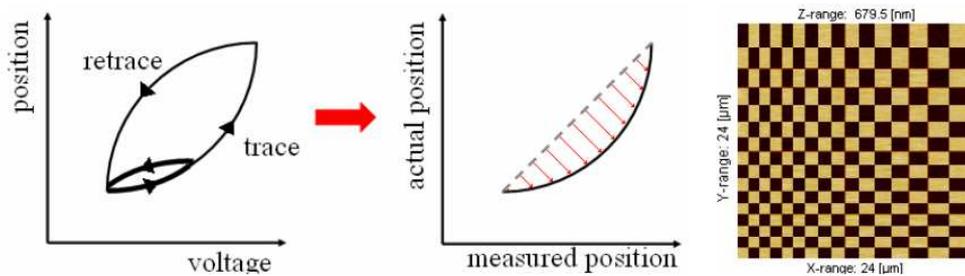


Figura 2.20: curva caratteristica dell'isteresi negli attuatori piezo-elettrici degli AFM dove la non-linearità influenza le misurazioni dello strumento, a destra è simulata che fornisce un'idea dei possibili effetti dell'isteresi durante l'acquisizione di un modello regolare a scacchiera [128]

con una deriva dell'attuatore piezoelettrico dopo che gli è stata applicata una tensione. Questo generalmente accade quando viene richiesto un grande spostamento lungo gli assi x o y . Per esempio, quando viene richiesto uno spostamento su tutta la superficie per effettuare una nuova scansione oppure quando lo scanner effettua una scansione all'indietro. Quando viene richiesto un grande spostamento, la reazione dello scanner non è immediata e per coprire tutta la distanza la testina si sposta inizialmente molto velocemente e infine più lentamente: la tensione applicata viene variata in modo graduale. Il creep è un fenomeno caratteristico della seconda fase dello spostamento sul piano orizzontale e provoca una variazione sulla scala della scansione e la posizione del centro di riferimento fisico della testina scanner. Il creep compare nell'immagine come un allungamento e uno stiramento della superficie analizzata. Un esempio è illustrato in figura 2.21. Spesso questi microscopi sono afflitti da problemi di non linearità che legano l'asse z al piano x, y . Da questo nasce il problema della distorsione a arco. Questo artefatto si



Figura 2.21: una tipica immagine affetta da creep [128]

manifesta graficamente come la proiezione di una superficie curva su un piano. L'artefatto nasce dalle caratteristiche costruttive degli strumenti: infatti la testina nello spostarsi sulla superficie, nelle zone laterali, subisce una flessione. Questo artefatto è trascurabile nei casi che l'acquisizione avvenga per aree che si trovano molto vicine al centro del piano di acquisizione dello strumento e diventa sempre più accentuata man mano che la superficie si estende. In figura 2.22 viene messo in evidenza il fenomeno della curvatura a arco acquisendo una superficie piana.

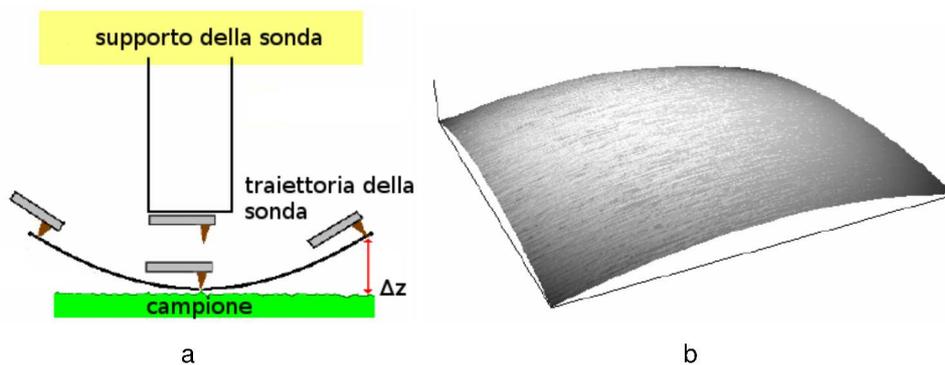


Figura 2.22: in evidenza il problema della curvatura a arco, in (a) il comportamento della sonda sulla superficie, in (b) un risultato dopo l'acquisizione di una superficie piana [128]

Overshoot

Una compensazione errata dell'isteresi e del creep nelle ceramiche piezoelettriche che spostano la mensola nel moto perpendicolare alla superficie possono causare problemi di tracciamento dei bordi. Il problema è osservabile più facilmente durante la misurazione di strutture di micro-fibre di produzione sintetica: viene messo in evidenza il problema facendo sì che la punta si estenda maggiormente rispetto allo spigolo da tracciare. In figura 2.23 vediamo esempi di overshooting in un sistema non retro-azionato, in un sistema con guadagno eccessivo e in un sistema con guadagno debole.

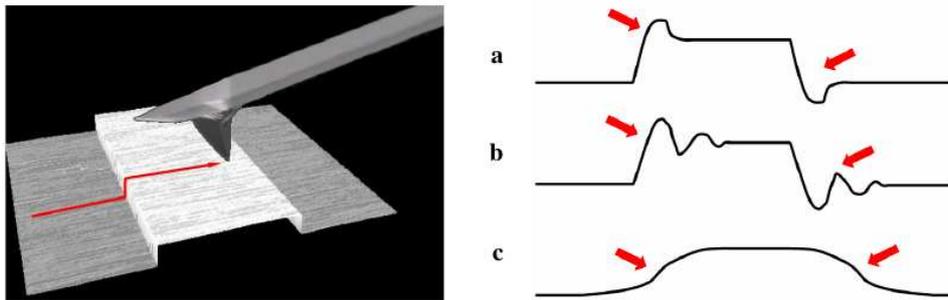


Figura 2.23: in evidenza il problema dell'overshoot, in (a) come il problema si manifesta senza retroazione, in (b) come il problema si manifesta quando il guadagno è eccessivo e in (c) quando il guadagno è debole [128]

Convoluzione

Poiché le acquisizioni che sono effettuate con un AFM avvengono per mezzo di una testina che si muove lungo un piano e la testina ha delle dimensioni fisiche che possono essere superiori alle forme geometriche che presenta la superficie che viene acquisita, necessariamente l'informazione ottenuta è una convoluzione tra la testina e la superficie analizzata. Uno degli effetti di questo artefatto, è la dilatazione di alcune forme geometriche presenti sul campione analizzato. In figura 2.24. Nell'esempio l'estremità della sonda non tocca la superficie del campione che, invece, a causa delle geometrie del campione e della sonda, viene toccata da un bordo laterale della sonda. Poiché ciò che viene registrata è l'estremità della sonda, la dimensione delle "cuspidi" del campione vengono dilatate. Chiamiamo con P la punta, con S la superficie e con I la topografia ottenuta. Una stima dell'andamento di I può essere calcolato con l'equazione 2.5 dove \oplus indica l'operatore di convoluzione.

$$I = S \oplus P \quad (2.5)$$

In dettaglio, l'operatore \oplus è descritto nell'equazione 2.6.

$$i(x, y) = \max_{(u,v)} [s(x-u, y-v) + p(u, v)] \quad (2.6)$$

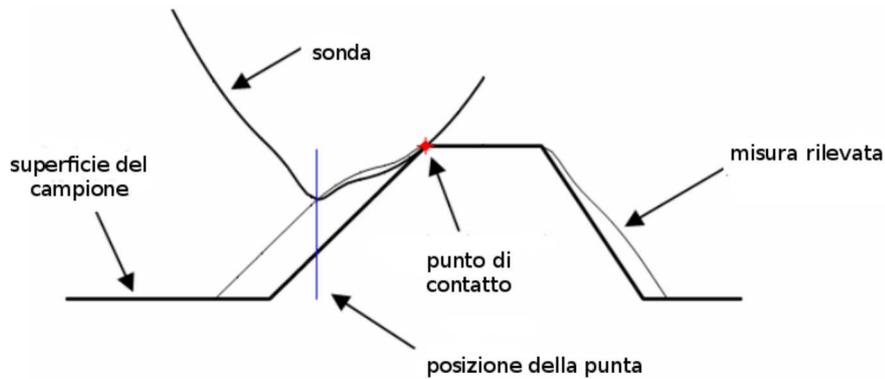


Figura 2.24: in evidenza il problema della convoluzione [128]

dove u e v sono le componenti di un vettore che definisce la traslazione relativamente al fenomeno della convoluzione.

Durante l'acquisizione il fenomeno della dilatazione è parzialmente compensato fenomeno dell'erosione. L'erosione è il duale della dilatazione. Anche se non è possibile definire con precisione tutta la superficie analizzata a causa di un errato contatto tra la sonda e la superficie del campione, è possibile calcolare una approssimazione S_r della medesima come descritto dall'equazione 2.7 dove Θ è l'operatore di de-convoluzione.

$$S_r = I\Theta P \quad (2.7)$$

In dettaglio, l'operatore Θ è descritto nell'equazione 2.8.

$$s_r(x, y) = \min_{(u, v)} [i(x + u, y + v) - p(u, v)] \quad (2.8)$$

La forma della punta è sempre "convoluta" con la superficie, mettiamo in evidenza [46]:

- la dilatazione restringe buche e pozzi mentre amplia le cuspidi [54] (Figura 2.25 a);
- una sonda sottile è in grado di rilevare le strutture più piccole rispetto a una punta di grandi dimensioni (Figura 2.25 b e Figura 2.25 c);

Il problema della distorsione è più significativo quando le dimensioni della testina sono vicine alle dimensioni degli oggetti che sono sulla superficie del campione che vogliamo misurare [23].

Rumore

Gli AFM sono affetti da vari tipi di rumore: il rumore elettrico, il rumore acustico, le vibrazioni e il rumore meccanico. Il risultato di tutti questi rumori che influenzano tipicamente le alte frequenze può essere modellato con un processo stocastico.

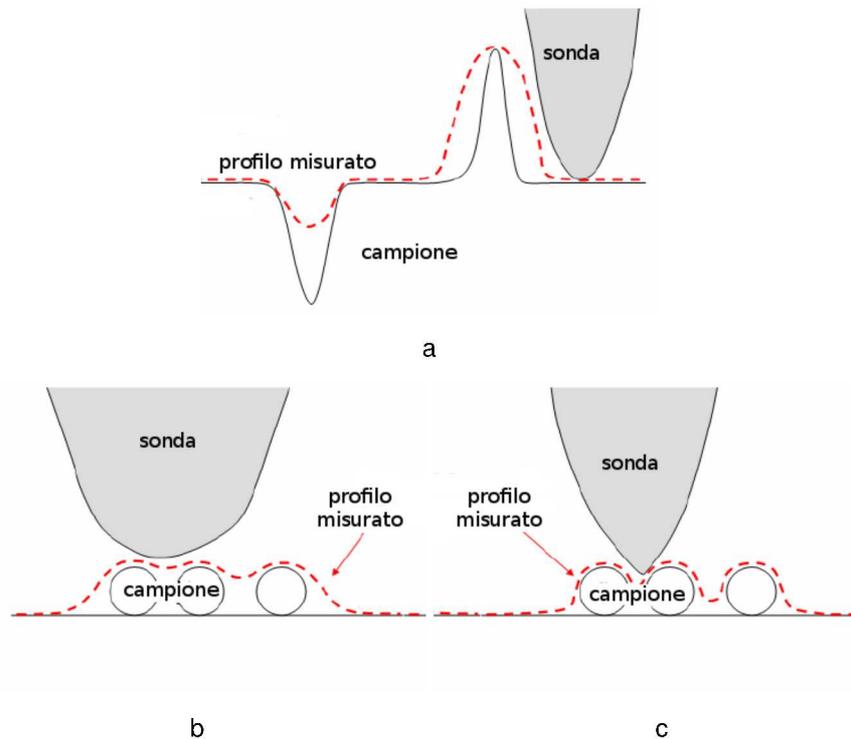


Figura 2.25: la convoluzione tra punta e superficie, in (a) la buca è sottodimensionata e il picco è sovradimensionato, in (b) e in (c) è messo in evidenza il profilo misurato con due diverse punte [128]

2.6 Lo stato dell'arte della super-risoluzione

Esistono principalmente tre famiglie di algoritmi di super-risoluzione basati su [16, 64, 145, 185, 125]:

- metodi basati sull'interpolazione;
- metodi che simulano la formazione dell'immagine;
- metodi basati sull'apprendimento.

Gli algoritmi presentati analizzano le immagini:

- nel dominio delle frequenze;
- nel dominio dello spazio;
- con metodologia mista nel dominio dello spazio e nel dominio delle frequenze;
- nel dominio delle wavelet.

I metodi trattati calcolano un'immagine super-risolta partendo da una versione a bassa risoluzione e stimando il valore dei pixel mancanti utilizzando una griglia sovra-

campionata. In altri termini, un software incrementa la frequenza di campionamento spaziale (aumento della risoluzione spaziale) calcolando il miglior valore (incrementando la risoluzione colorimetrica) dei nuovi campioni.

Supponiamo di avere un'immagine a bassa risoluzione $I_{i,j}$ la cui dimensione è $W \times H$ e la sua immagine corrispondente ad alta risoluzione sia $I'_{i,j}$ di dimensione $aW \times bH$ (a e b sono i fattori di zoom rispettivamente della larghezza e dell'altezza). Senza perdita di generalità assumiamo che $a = b = 2$. Possiamo calcolare alcuni valori dei pixel ad esempio in posizione $I'_{2i,2j}$ partendo dalle informazioni dell'immagine in bassa risoluzione. Infatti abbiamo che $I_{i,j} = I'_{2i,2j}$ con $i = 0, 1, \dots, H$ e $j = 0, 1, \dots, W$. Per il calcolo del valore degli altri pixel ($I'_{2i+1,2j}$ o $I'_{2i,2j+1}$).

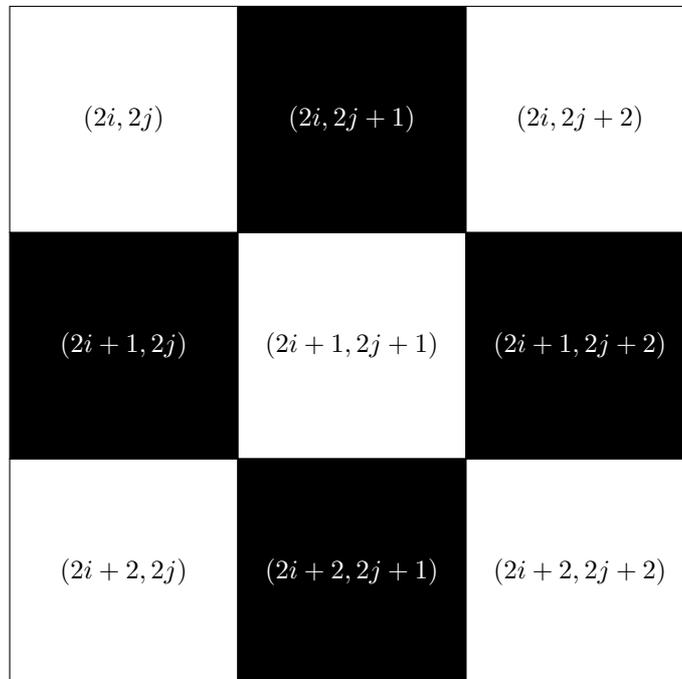


Figura 2.26: un esempio di risoluzione aumentata, le aree bianche sono quelle dell'immagine originale I in bassa risoluzione mentre le aree nere sono quelle da calcolare nell'immagine I' elaborata

In figura 2.26 vediamo a confronto i pixel dell'immagine originale (aree bianche) e i pixel che devono essere calcolati per aumentare la risoluzione dell'immagine (aree nere).

2.6.1 I metodi di base: bilineare, bicubico e spline

i primi esperimenti con la super-risoluzione sono stati effettuati con i metodi di interpolazione bilineare e bicubico [189, 179]. Il vantaggio di questi metodi consiste nel poter essere eseguiti con risorse di calcolo esigue, lo svantaggio nel risultato che offrono. Infatti, il

modello che utilizzano per ricostruire l'immagine I' si basa su un modello estremamente semplificato della relazione tra le immagini I e I' . I problemi più evidenti all'occhio che nascono con questo tipo di elaborazione sono l'aliasing e il blurring, ovvero, scalettature e oggetti non ben definiti, sfuocati.

Questi problemi vengono parzialmente risolti per mezzo dei metodi di interpolazione basati su spline [117].

2.6.2 Il metodo di Jensen

Per evitare questo tipo di problemi su I' sono state studiate varie soluzioni che attenuano gli artefatti che si creano lungo gli spigoli. Nel lavoro di Jensen [102] viene proposta una stima dell'orientazione di ogni spigolo presente nell'immagine. La stima dell'orientazione degli spigoli avviene usando una proiezione su una base ortonormale nella quale il processo di interpolazione è ottimizzato per evitare di effettuare interpolazione nella zona degli spigoli. Per raggiungere questo scopo viene utilizzato un modello in cui il calcolo da effettuare in prossimità degli spigoli avviene in funzione di quattro parametri schematizzati nell'equazione 2.9.

$$S(i, j, A, B, \rho, \Theta) = \begin{cases} A & \text{se } i \cos \Theta + j \sin \Theta \geq \rho \\ B & \text{se } i \cos \Theta + j \sin \Theta < \rho \end{cases} \quad (2.9)$$

dove:

- $i \cos \Theta + j \sin \Theta = \rho$ è la linea di separazione tra due regioni dello spazio
- Θ è l'orientamento degli spigolo
- i e j sono le coordinate sulle quali viene eseguito il calcolo
- A e B sono i valori dei pixel mancanti

L'equazione 2.9 può essere interpretata come una funzione periodica con argomento ϕ . Utilizzando la base ortogonale $B_n(\phi)$:

$$B_1(\phi) = \frac{1}{\sqrt{\pi}} \cos(\phi) \quad (2.10)$$

$$B_2(\phi) = \frac{1}{\sqrt{\pi}} \sin(\phi) \quad (2.11)$$

$$B_3(\phi) = \frac{1}{\sqrt{\pi}} \cos(2\phi) \quad (2.12)$$

$$B_4(\phi) = \frac{1}{\sqrt{\pi}} \sin(2\phi) \quad (2.13)$$

la proiezione del modello S (eq. 2.9) in $B_n\phi$ è dato da:

$$a_n \langle S(\phi, A, B, \rho, \Theta), B_n(\phi) \rangle = \int_a^b S(\phi, A, B, \rho, \Theta), B_n(\phi) d\phi \quad (2.14)$$

2.6. LO STATO DELL'ARTE DELLA SUPER-RISOLUZIONE

L'equazione 2.14 permette il calcolo dei coefficienti a_n tramite i quali è possibile calcolare l'orientamento degli spigoli Θ :

$$\Theta = \tan^{-1} \left(\frac{a_1}{a_2} \right) \quad (2.15)$$

Per effettuare un'interpolazione su aree selezionate (ad esempio non sfumando gli spigoli degli angoli vivi) l'algoritmo esaminato analizza anche l'altezza dei medesimi con l'obiettivo di mantenere la regolarità degli spigoli.

Per il calcolo di questo valore consideriamo una matrice di punti $W[i, j]$ di dimensione 3×3 . Il più piccolo errore quadratico medio calcolato con la base generatrice data è:

$$f(\phi) = \sum_{i=0}^4 \lambda_i B_n(\phi) \quad (2.16)$$

dove il coefficiente λ_i è quello che maggiormente adatta nel discreto il valore a_i . Per il computo dei coefficienti λ_i , per ogni W contenuto in I calcoliamo

$$\lambda_k = \sum_{m=1}^3 \sum_{n=1}^3 W[m, n] M_k[m, n] \quad (2.17)$$

e lo associamo al punto $(1, 1)$ di W . M_k è l'operatore mostrato delle equazioni 2.18, 2.19, 2.20, 2.21 e 2.22 con parametri $\alpha = \frac{\sqrt{\pi}}{4\sqrt{2}}$ e $\beta = \frac{\sqrt{\pi}}{4}$.

$$M_0 = \begin{bmatrix} \alpha & \alpha & \alpha \\ \alpha & 0 & \alpha \\ \alpha & \alpha & \alpha \end{bmatrix} \quad (2.18)$$

$$M_1 = \begin{bmatrix} -\alpha & 0 & \alpha \\ -\beta & 0 & \beta \\ -\alpha & 0 & \alpha \end{bmatrix} \quad (2.19)$$

$$M_2 = \begin{bmatrix} \alpha & \beta & \alpha \\ 0 & 0 & 0 \\ -\alpha & -\beta & -\alpha \end{bmatrix} \quad (2.20)$$

$$M_3 = \begin{bmatrix} 0 & -\beta & 0 \\ \beta & 0 & \beta \\ 0 & -\beta & 0 \end{bmatrix} \quad (2.21)$$

$$M_4 = \begin{bmatrix} -\beta & 0 & \beta \\ 0 & 0 & 0 \\ \beta & 0 & -\beta \end{bmatrix} \quad (2.22)$$

Il valore dei pixel mancanti A e B su ogni lato dello spigolo è calcolato come:

$$B = \frac{1}{2\pi} \left(\sqrt{2\pi} \lambda_0 - 2\sigma\delta \right) \quad (2.23)$$

$$A = \delta + B \quad (2.24)$$

dove

$$\delta = \frac{\sqrt{\pi} [\lambda_1 \cos \theta + \lambda_2 \sin \theta]}{2 \sin \sigma} \quad (2.25)$$

$$\sigma = \cos^{-1} \left(\frac{\lambda_3 \cos 2\theta + \lambda_4 \sin 2\theta}{\lambda_1 \cos \theta + \lambda_2 \sin \theta} \right) \quad (2.26)$$

In figura 2.27 un'immagine di un faro (in risoluzione originale e degradato) e l'immagine super-risolta con l'algorithmo di Jensen.

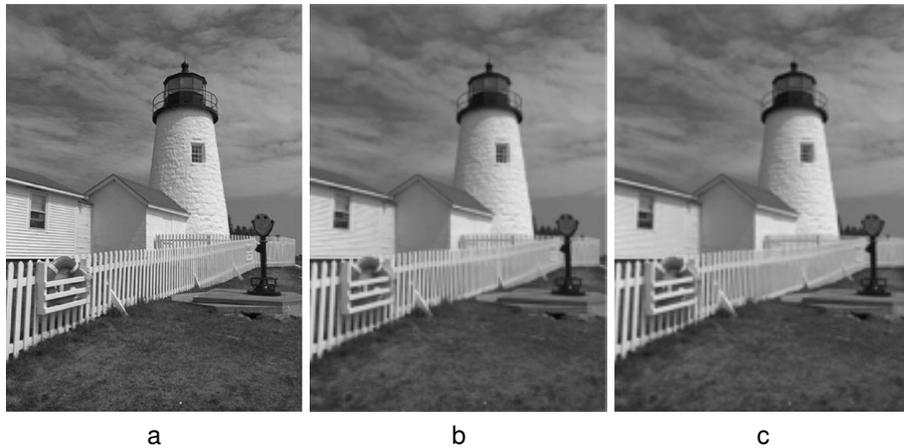


Figura 2.27: immagine ottenuta con il metodo di Jessen, in (a) l'immagine originale, in (b) l'immagine degradata e in (c) l'immagine elaborata

Come vediamo l'algorithmo di Jensen introduce una forte sfocatura specialmente dove gli spigoli sono molto vicini.

2.6.3 Il metodo di Allebach

Allebach nell'articolo [4] propone un metodo diverso per generare una mappa degli spigoli per guidare l'incremento di risoluzione una immagine in bassa risoluzione. La figura 2.28 mostra l'idea alla base di questo metodo.

Il primo calcolo che questo algorithmo compie consiste nello calcolate l'estimatore a livello di sub-pixel per generare mappa degli spigoli in risoluzione più elevata. Successivamente la mappa calcolata viene utilizzata per guidate l'incremento di risoluzione dell'immagine in bassa risoluzione I nell'immagine a risoluzione più elevata I' . La figura 2.29 mostra l'architettura della tecnica l'interpolazione guidata.

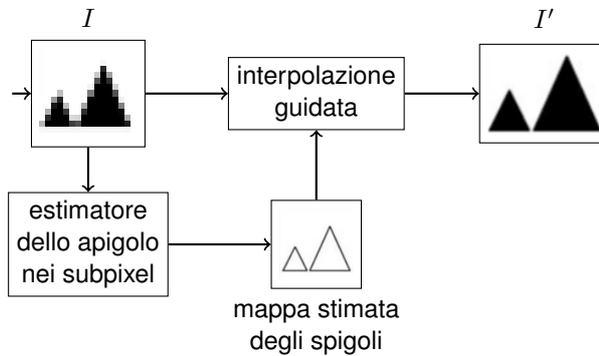


Figura 2.28: schema generale dell'algoritmo di Allebach [4]

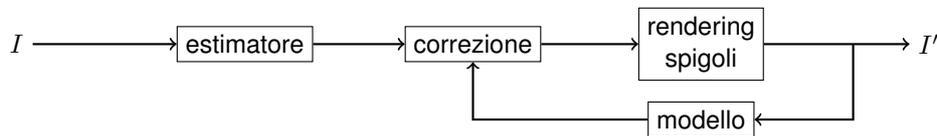


Figura 2.29: schema dell'interpolazione guidata dell'algoritmo di Allebach [4]

L'algoritmo è composto da due blocchi principali: l'esecuzione del rendering e la correzione dei dati. Il rendering esegue semplicemente una interpolazione bilineare dell'immagine in bassa risoluzione. Da ciò deriva l'implicita assunzione che l'immagine in alta risoluzione viene calcolata dai campioni dell'immagine in bassa risoluzione. Il passo successivo dell'algoritmo Allebach [4] consiste nel compensare iterativamente gli errori grazie all'interpolazione ottenuta dal modello dell'immagine (viene sfruttata la differenza tra l'immagine calcolata dal modello e l'ultima immagine calcolata).

Per il calcolo della mappa degli spigoli con grana del sub-pixel, l'immagine in bassa risoluzione è filtrata con un filtro COSO (Center On Surround Off) calibrato per avere una costante positiva nel centro e una costante negativa nelle regioni esterne. Le altezze relative sono scelte per mantenere a zero la componente continua. I coefficienti del filtro COSO sono dati dall'equazione 2.27 dove h_c e h_s sono calcolate rispettivamente nella zona centrale e nella zona periferica del filtro usando la PSF (Point Spread Function) della *Laplacian-of-Gaussian* (LoG) che è data dall'equazione 2.28.

$$h^{COSO}(i, j) = \begin{cases} h_c & |i|, |j| \leq N_c \\ h_s & N_c < |i| \leq N_s \text{ e } |i| \leq N_s \\ & N_c < |j| \leq N_s \text{ e } |j| \leq N_s \\ 0 & \text{altrimenti} \end{cases} \quad (2.27)$$

$$h^{Log}(i, j) = \frac{2}{\sigma^2} \left[1 - \frac{i^2 + j^2}{2\sigma^2} \right] e^{-\frac{i^2 + j^2}{2\sigma^2}} \quad (2.28)$$

N_c e N_s sono le dimensioni dell'area centrale e dell'area periferica, σ la varianza calcolata sul valore dei pixel.

Il risultato del filtro COSO non viene utilizzato direttamente come mappa degli spigoli ma ne è una buona approssimazione. Per determinare la mappa degli spigoli ad alta risoluzione viene interpolata l'uscita del filtro COSO tra i punti dell'immagine in bassa risoluzione per stimare le aree di zero-crossing nell'immagine in alta risoluzione. La procedura adottata è la seguente: per ogni gruppo di 4 pixel dell'immagine in bassa risoluzione (i punti a,b,c e d della figura 2.30) vengono esaminati per ogni punto i segni del filtro COSO. Questi segni possono comparire in diverse formazioni geometriche. Per il caso mostrato in figura 2.30 viene eseguita l'interpolazione lungo i segmenti $a - b$, $b - c$ e $b - d$ per ottenere la stima dei punti di attraversamento dello zero (*zero-crossing*) zc_1 , zc_2 e zc_3 . Lo spigolo è approssimato utilizzando la cella in bassa risoluzione racchiusa dai punti a, b, c e d e i due segmenti $zc_1 - zc_3$ e $zc_3 - zc_2$, discretizzandoli sulla griglia in alta risoluzione.

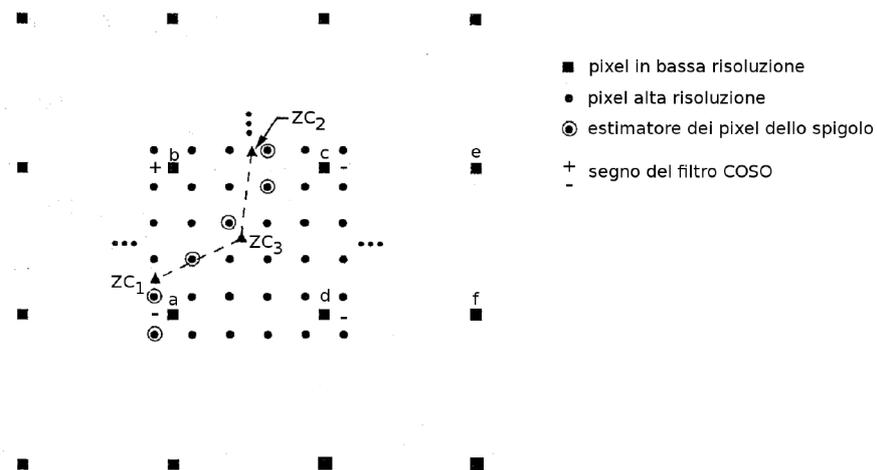


Figura 2.30: il filtro COSO usato nell'algorithm di Allebach [4]

In figura 2.31 a l'immagine originale e in figura 2.31 b l'immagine elaborata a confronto.

2.6.4 il metodo di Carey

Analizziamo adesso i metodi [139, 27] che lavorano nel dominio delle wavelet. Questi algoritmi considerano l'immagine in bassa risoluzione come il prodotto di un filtro passa-basso di una trasformata wavelet e utilizzano la dipendenza proporzionale tra le wavelet per predire l'informazione mancante.

In [27] Carey la trasformata wavelet è sfruttata considerando l'immagine in bassa risoluzione come il risultato di una immagine in alta risoluzione filtrata con un filtro passa

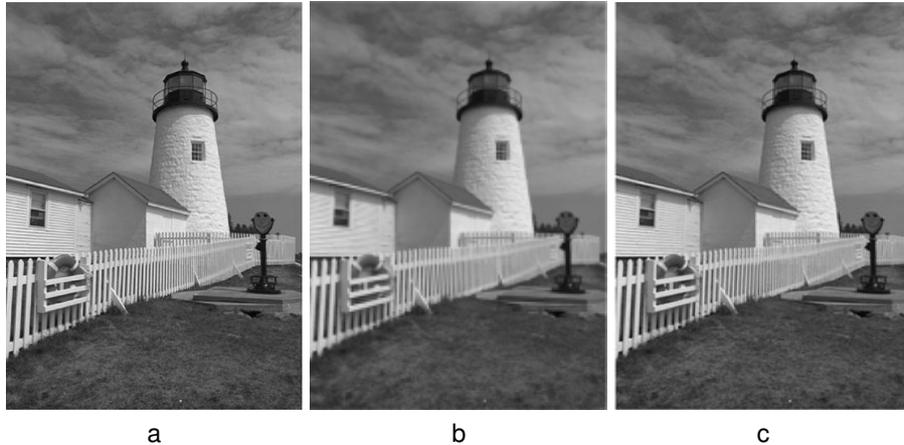


Figura 2.31: risultati ottenuti con metodo di Allebach [4], in (a) l'immagine originale, in (b) l'immagine degradata e in (c) l'immagine elaborata

basso nel dominio delle wavelet. Tramite la trasformata wavelet vengono generati i dettagli in alta frequenza che mancano, successivamente viene interpolata una immagine a dimensione doppia interpolando i valori dei pixel lungo le righe e le colonne. Successivamente viene calcolato un altro gruppo di bande ad alta frequenza utilizzando due stadi wavelet indipendenti. Il primo identifica gli spigoli con un elevato fattore di correlazione tra le righe. Successivamente viene calcolato l'andamento della wavelet in prossimità di questi spigoli per approssimarne la ricostruzione a risoluzione doppia. La stessa procedura è successivamente applicata alle colonne. Il sistema utilizzato per questa elaborazione è rappresentato in figura 2.32. Secondo Carey [27] l'estrapolazione dei coefficienti locali permette di evitare i problemi di sfocatura mantenendo nitidi gli spigoli. In figura 2.33 il risultato di questo algoritmo.

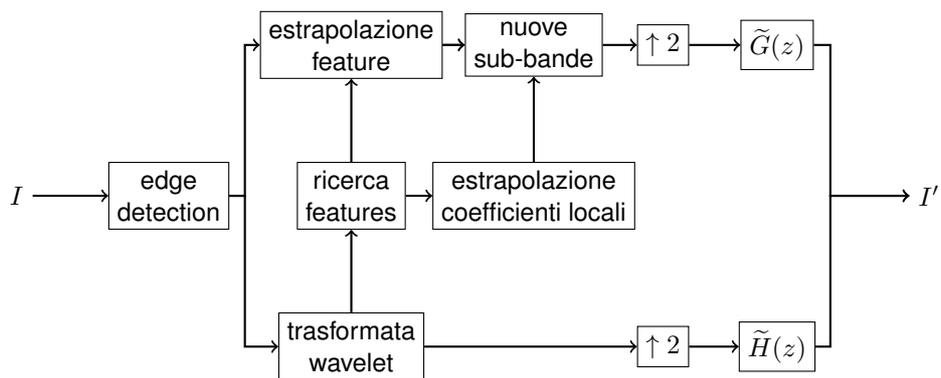


Figura 2.32: diagramma a blocchi del metodo Carey [27], in ingresso l'immagine in bassa risoluzione I , in uscita la legge super-risolta I'



Figura 2.33: risultati ottenuti con metodo di Carey [27], in (a) l'immagine originale, in (b) l'immagine degradata e in (c) l'immagine elaborata

In [27] essenzialmente viene sfruttata la proprietà di Lipschitz [163] (definita in 2.6.4) degli spigoli vivi nelle wavelet: viene sfruttata l'informazione del valore massimo del modulo nell'immagine in bassa risoluzione per predire i coefficienti della wavelet dell'immagine in alta risoluzione (così da antitrasformare e ricostruire l'immagine in alta risoluzione).

Definizione (funzione lipschitziana). *Sia I un intervallo limitato o illimitato in \mathbb{R} . Una funzione*

$$f : I \subseteq \mathbb{R} \rightarrow \mathbb{R} \quad (2.29)$$

si dice lipschitziana se esiste una costante $L > 0$ per cui si ha:

$$\|f(x_1) - f(x_2)\| \leq L \|x_1 - x_2\| \quad \forall x_1, x_2 \in I \quad (2.30)$$

o equivalentemente

$$\frac{\|f(x_1) - f(x_2)\|}{\|x_1 - x_2\|} \leq L \quad \forall x_1, x_2 \in I, x_1 \neq x_2 \quad (2.31)$$

Il rapporto tra variazione di ordinata $|f(x_1) - f(x_2)|$ e variazione di ascissa $|x_1 - x_2|$ non può superare un valore costante fissato L . In termini intuitivi potremmo dire che una funzione è lipschitziana in un intervallo se il suo grafico, in tale intervallo, *ha una pendenza limitata*.

La più piccola costante $L > 0$ per cui vale l'equazione 2.30 prende il nome di costante di Lipschitz.

2.6.5 il metodo Muresan

Il lavoro [139] estende questa strategia estendendo la proprietà di Lipschitz. Infatti mentre in [27] viene sfruttato il massimo locale di una wavelet a una data scala in [139]

viene utilizzato il cono di influenza. Il cono di influenza, definito in [139], effettua una valutazione degli spigoli per le varie scale della wavelet.

L'approccio, chiamato nell'articolo *Prediction of Image Detail* può essere spiegato con l'aiuto della figura 2.34. Nel diagramma l'immagine in alta risoluzione I' è posta in ingresso ai filtri. Muresan [139] ipotizza che l'immagine in bassa risoluzione I possa essere simulata prendendo l'immagine in alta risoluzione I' , filtrandola e sotto-campionandola. Nelle ipotesi, il filtro passa basso L rappresenta l'effetto del sistema di acquisizione delle immagini. Nell'articolo viene illustrato che è possibile ricostruire l'immagine in alta risoluzione nell'ipotesi di conoscere il segnale D mancante (che può essere calcolato con esattezza come il segnale in uscita al filtro H). Poiché nel sistema non c'è conoscenza di D , esso può solo essere stimato o predetto. Nell'approccio [139] le informazioni che mancano per ricostruire un'immagine è basato sull'analisi delle wavelet nel dominio scala-spazio in prossimità degli spigoli. In particolare, viene utilizzata l'energia dei coefficienti delle wavelet relativi agli spigoli che sono dentro il cono di influenza. In figura 2.35 il risultato di un esperimento. Questo metodo, come possiamo vedere dalle figure, è migliore del Carey [27] introducendo un errore minore nella ricostruzione degli spigoli.

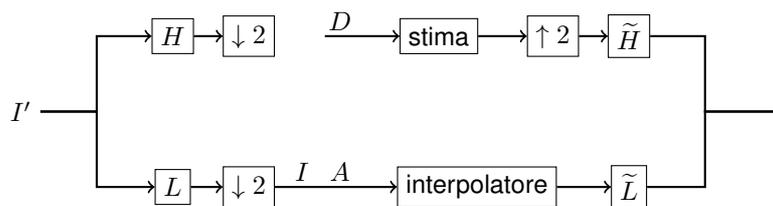


Figura 2.34: diagramma a blocchi del metodo Muresan [139]

2.6.6 il metodo NEDI

Il metodo NEDI è presentato in [121] da Li. Il metodo analizza la direzionalità degli spigoli considerando la covarianza tra l'immagine in bassa risoluzione e l'immagine in alta risoluzione. Presa una finestra, viene calcolata localmente la covarianza tra pixel vicini nell'immagine in bassa risoluzione per stimare la covarianza tra i pixel nell'immagine in alta risoluzione. In figura 2.36 un esempio.

La covarianza della relazione b_0 è stimata dalla covarianza del vicino a_0 in funzione della finestra locale. I cerchi vuoti nella figura rappresentano i pixel dell'immagine in bassa risoluzione e i cerchi pieni i pixel che devono essere stimati nell'immagine in alta risoluzione. La covarianza è sfruttata tra i pixel in bassa risoluzione lungo le loro 4 diagonali in una finestra di dimensioni $m \times m$. Per delineare il metodo con maggiore precisione consideriamo il vettore A . Il vettore A contiene le diagonali dei pixel in bassa risoluzione che andranno a determinare il valore del pixel in alta risoluzione o . Sia poi X un vettore di dimensione m^2 della finestra $m \times m$ e C una matrice di dimensione $4 \times m^2$ contenente



Figura 2.35: risultati ottenuti con metodo di Muresan [139], in (a) l'immagine originale, in (b) l'immagine degradata e in (c) l'immagine elaborata

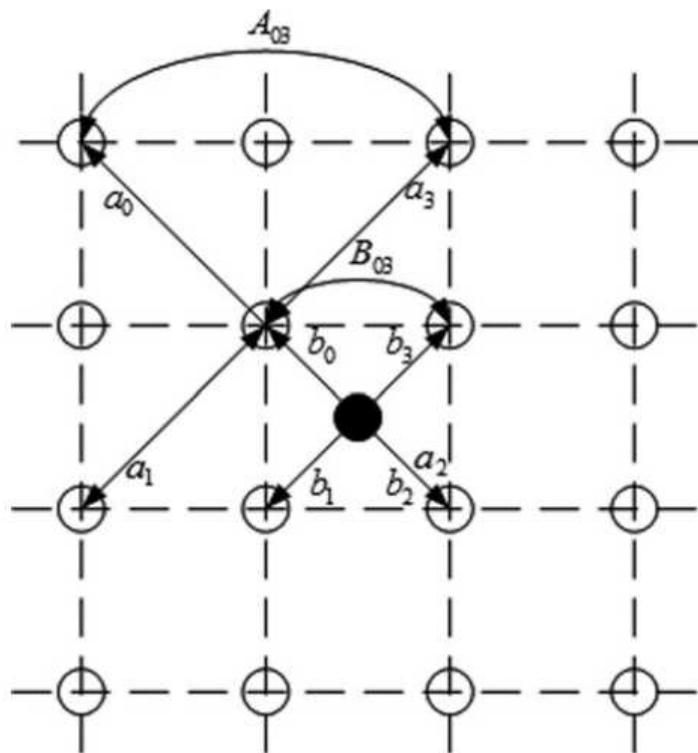


Figura 2.36: covarianza locale utilizzata nel metodo Li [121]

le diagonali dei vicini dei pixel in X . L'equazione utilizzata per aumentare la risoluzione è descritta in 2.32.

$$o = (C^T \cdot C)^{-1}(C^T \cdot X) \cdot A^T \quad (2.32)$$

L'algoritmo NEDI sfrutta due passaggi per determinare tutti i pixel in alta risoluzione. La prima passata sfrutta i vicini lungo le diagonali per interpolare tutti i pixel in alta risoluzione che hanno entrambe le coordinate pari, la seconda passata sfrutta i vicini orizzontali e verticali per interpolare i rimanenti pixel nell'immagine in alta risoluzione. In figura 2.37 un esempio di calcolo dei pixel nell'immagine super-risolta. In figura 2.38 un esempio di immagine super-risolta.

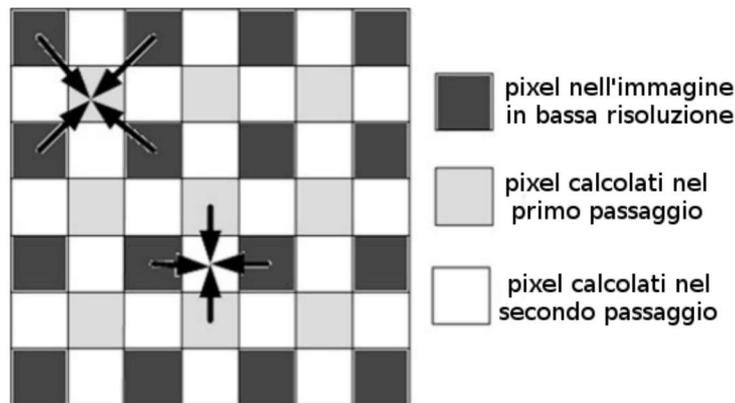


Figura 2.37: ricostruzione di un'immagine in alta risoluzione con il metodo Li [121]

2.6.7 il metodo Freeman

Il metodo proposto da Freeman [71] migliora la risoluzione di una immagine per mezzo di una stima iniziale e tramite un meccanismo di apprendimento automatico realizzato per mezzo di immagini artificiali. L'algoritmo sviluppato prende in esame le macro-relazioni che sussistono tra le regioni di un'immagine e le micro-relazioni che sussistono tra pixel vicini. La macro-relazione permette di effettuare delle stime iniziali della scena, la micro-relazione permette il calcolo puntuale dei pixel nelle immagini super-risolte. Data un'immagine in bassa risoluzione I , Freeman [71] cerca di identificare l'immagine in alta risoluzione I' utilizzando una probabilità a posteriori $P(I'|I) = cP(I', I)$ dove la costante di normalizzazione $c = \frac{1}{P(y)}$. Lo stimatore usato da Freeman [71] può essere o il calcolo dell'errore quadratico medio oppure il calcolo del MAP (maximum a posteriori). Questo calcolo viene eseguito per mezzo delle reti di Markov. La scena viene divisa in



Figura 2.38: risultati ottenuti con metodo di Li [139]: in (a) l'immagine originale, in (b) l'immagine degradata e in (c) l'immagine elaborata

aree e ogni area viene assegnata a un nodo di una rete di Markov. Ogni nodo è connesso alla corrispondente area e alle aree vicine. In figura 2.39 vediamo come ogni nodo nella rete descrive un'area dell'immagine: le osservazioni y_i nell'immagine I corrispondono a una ripresa della scena x_i che può essere rappresentata in una immagine I' .

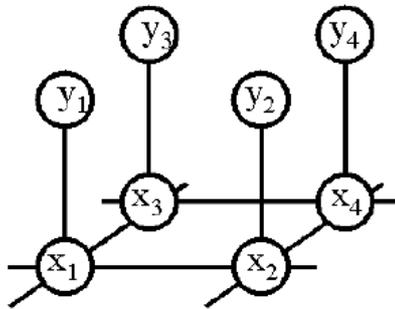


Figura 2.39: modello usato da Freeman [71] per super-risolvere una immagine; ogni nodo nella rete descrive un'area dell'immagine dove le osservazioni y_i nell'immagine I corrispondono a una ripresa della scena x_i che può essere rappresentata in una immagine I'

Per un *random Markov field* la probabilità congiunta tra una x e le osservazioni y può essere scritta come in equazione 2.33 [76].

$$P(x_1, \dots, x_N, y_1, \dots, y_N) = \prod_{(i,j)} \Psi(x_i, x_j) \prod_j \Phi(x_k, y_k) \quad (2.33)$$

L'equazione 2.33 introduce la coppia di funzioni Ψ e Φ , Ψ e Φ sono le funzioni di apprendimento. In figura 2.40 un esempio di calcolo del metodo di Freeman.

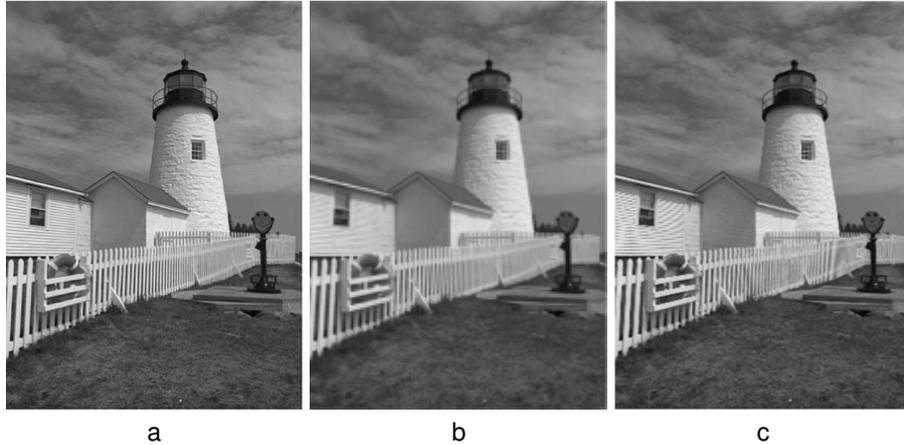


Figura 2.40: risultati ottenuti con metodo di Freeman [71]: in (a) l'immagine originale, in (b) l'immagine degradata e in (c) l'immagine elaborata

2.6.8 il metodo Kim

Il metodo proposto da Kim [111] utilizza il framework proposto in [71]. I punti dell'immagine I' sono calcolati combinando più metodi. Il primo passo che viene compiuto consiste nel calcolo delle alte frequenze nell'immagine I'_1 utilizzando una spline. La risoluzione spaziale e colorimetrica dell'immagine I'_1 corrisponde alla risoluzione dell'immagine I' che desideriamo calcolare. Inoltre, utilizzando le assunzioni tra le componenti in alta frequenza e in bassa frequenza di [70], viene effettuata una ulteriore stima I'_2 delle alte frequenze utilizzando l'interpolazione della Laplaciana di I'_1 . I'_1 e I'_2 sono sommate in modo opportuno per creare l'immagine super-risolta. Oltre a I'_2 vengono generati molte immagini candidate Z_i di dimensione pari a I' . Le immagini Z_i sono una stima di parte dell'immagine considerando solo le informazioni locali di un'area dell'immagine globale. Inoltre, per mettere in risalto gli spigoli, viene effettuato un ulteriore post-processing con il metodo proposto da Tappen [176]. La figura 2.41 illustra un esempio di calcolo.

In figura 2.42 un esempio di risoluzione aumentata con il metodo di Kim [111].

2.7 Considerazioni

Per scegliere il metodo di super-risoluzione migliore con il quale confrontare il nostro lavoro abbiamo utilizzato come immagini campione un'immagine di un faro (512×768), alcune immagini provenienti da microscopio a forza atomica e altre immagini suggerite

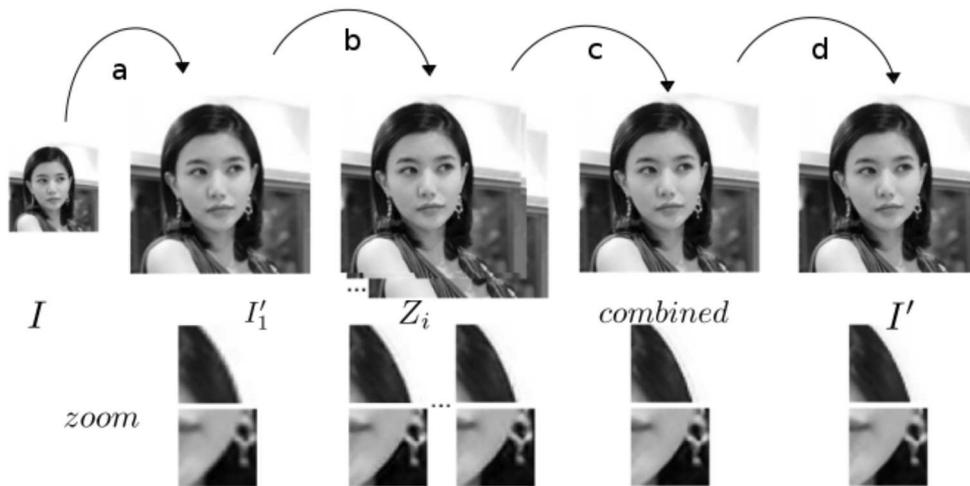


Figura 2.41: i passi principali del metodo di Kim [111], in (a) l'immagine originale viene interpolata alla scala desiderata (b) sono generate alcune immagini candidate come risultato del calcolo di regressione (c) le immagini candidate sono combinate (d) sono rimossi gli artefatti



Figura 2.42: risultati ottenuti con metodo di Kim [111]: in (a) l'immagine originale, in (b) l'immagine degradata e in (c) l'immagine elaborata

negli articoli esaminati. Tutte le immagini testate sono in livelli di grigio con una profondità di colore di 8 bit. Le immagini sono state sotto-campionate di un fattore 2 per l'altezza e di un fattore 2 per la larghezza. Successivamente sono state date in input agli algoritmi di super-risoluzione. Successivamente le immagini super-risolve sono state confrontate con le immagini originali. Nel confronto è stata calcolato l'errore tra l'immagine originale \bar{I} e l'immagine I'_m calcolata con il metodo m . In altre parole per i metodi elencati il punteggio S assegnato è stato:

$$S = \frac{\sum_x \sum_y \bar{I}'_{(x,y)} - I'_{(x,y)}}{X \cdot Y} \quad (2.34)$$

dove

- \bar{I}' è l'immagine sorgente
- I' è l'immagine super-risolta con uno dei metodi presentati
- X e Y sono le dimensioni \bar{I}' dell'immagine elaborata

Tra i metodi elencati quello che ha fornito le risposte migliori è il metodo di Kim [111].

Pattern-recognition

Questo capitolo presenta lo stato dell'arte dei metodi di pattern-recognition, con particolare attenzione ai risultati fino a oggi ottenuti nel segmentare un'immagine e nel riconoscere oggetti specifici all'interno dell'immagine stessa.

3.1 Introduzione

I termini machine-learning, pattern-recognition e altri, come data-mining e KDD (Knowledge Discovery in Databases) sono molto correlati [140, 178]. Con machine-learning si intendono generalmente quei metodi di apprendimento supervisionato che derivano dallo studio dell'intelligenza artificiale. Queste tecniche sono molto usate per risolvere problemi di ingegneria e si sono diffuse soprattutto nel contesto della computer vision, ad esempio nel campo della robotica dove il rilevamento di profili geometrici, dati da linee e cerchi, permette di individuare volti e altre entità. Con il termine pattern-recognition si identificano sia metodi supervisionati che metodi non supervisionati.

I metodi di pattern-recognition supervisionati generalizzano insiemi di regole apprendendo da dati (opportunosamente preparati) per i quali si conosce il risultato desiderato. Questo insieme di dati prende il nome di training-set. L'apprendimento avviene tramite dei modelli dipendenti da parametri che vengono determinati dall'algoritmo. L'obiettivo degli algoritmi di apprendimento è l'assegnamento di valori a queste variabili in modo da cercare di approssimare nel modo migliore possibile i dati del training-set così da ottenere un modello generale capace di risolvere correttamente le classi di problemi ai quali viene applicato.

I metodi non supervisionati necessitano di un modello *a priori* del problema che devono affrontare in quanto effettuano una elaborazione senza nessuna fase di apprendimento.

Quindi la sostanziale differenza tra i metodi supervisionati ed i metodi non supervisionati consiste nel modello teorico usato per risolvere un problema: i metodi non supervisionati possono essere immediatamente utilizzati per effettuare calcoli, mentre

quelli supervisionati necessitano di procedure di apprendimento (che possono richiedere anche giorni).

Recentemente sono stati esaminati metodi di apprendimento semi-supervisionati [30, 213].

I metodi di pattern-recognition in generale vanno ad arricchire i dati elaborati con informazioni utili alla loro caratterizzazione. In statistica con lo stesso scopo nel 1936 venne introdotta l'analisi discriminante.

Un esempio di pattern-recognition è la classificazione che serve, ad esempio, a distinguere lo *spam* dalla posta buona. Un altro esempio che esula dalla classificazione lo troviamo nel processo di riconoscimento vocale nel quale devono essere correttamente identificate sequenze di suoni.

Riteniamo importante effettuare un distinguo tra i metodi di pattern-recognition e i metodi di pattern-matching: dai metodi di pattern-recognition ci possiamo aspettare anche una risposta approssimata mentre dai metodi di pattern-matching ci aspettiamo sempre una risposta esatta in funzione dei dati di input. Un esempio di pattern-matching sono le espressioni regolari [72].

In questa ricerca il pattern-recognition è stato studiato con l'obiettivo di effettuare la segmentazione di immagini, classificarle e riconoscerne le componenti di interesse.

La segmentazione delle immagini è effettuata per individuare in un'immagine una o più parti aventi un loro significato e un loro scopo nella figura complessiva.

La classificazione è eseguita con lo scopo di attribuire una classe di appartenenza a un'intera figura o a parti di essa, l'attribuzione di una classe "finalizza" il lavoro iniziato con la segmentazione andando a attribuire un significato all'immagine o a parte di essa.

Il pattern-recognition in questo lavoro di ricerca mostra il suo primo impiego nel processo di segmentazione di un'immagine. La segmentazione è, in breve, il processo che separa un'immagine in regioni dove la separazione avviene seguendo uno specifico criterio. Queste regioni prendono il nome di segmenti.

La segmentazione rappresenta spesso la base per l'analisi delle immagini in quanto una errata segmentazione può propagare un errore non sistematico nei successivi calcoli [28]. Da ciò nasce l'esigenza di utilizzare un metodo per la segmentazione che sia il più preciso possibile.

Il processo di segmentazione risulta ancora oggi estremamente difficile da realizzare se non per alcune applicazioni specifiche e anche in questo caso il risultato lascia spazio a una certa approssimazione. In [148] a pagina 579 è scritto testualmente (trad.): "*non esiste una teoria della segmentazione dell'immagine. Di conseguenza, non è emerso nessun metodo standard per la segmentazione delle immagini. Piuttosto, sono stati individuati una serie di metodi ad hoc che hanno ricevuto un certo grado di popolarità. Dato che i metodi sono ad hoc sarebbe importante avere gli strumenti per la valutazione delle loro prestazioni. Haralick e Shapiro [89] hanno fornito la seguente linea guida per identificare qualitativamente una buona segmentazione delle immagini: 'le regioni di una immagine segmentata devono essere uniformi e omogenee rispetto a qualche caratteristica come i toni di grigio o la texture. L'area di ogni regione deve essere semplice e*

senza buchi. Le regioni segmentate adiacenti dovrebbero avere valori significativamente differenti rispetto alla caratteristica rispetto alla quale sono uniformi. I confini di ciascun segmento dovrebbero essere semplici, senza lacerazioni, e devono essere spazialmente accurati". Purtroppo, non è stata sviluppata nessuna metrica quantitativa finalizzata alla misura della bontà di una segmentazione".

Poichè in questa ricerca vengono analizzate immagini monocromatiche, lo stato dell'arte si concentrerà sugli aspetti legati a questo tipo di immagini. Per le immagini monocromatiche le tecniche di segmentazione sono divise in due tipologie classi:

- orientate ai bordi e sulle regioni;
- orientate sulle caratteristiche.

Alcune volte segmentare significa individuare certi punti di interesse, come gli angoli [90, 136]. La recensione di Fu [73], cui può essere ricondotta anche [143], suddivide le tecniche di segmentazione in tre classi:

- caratterizzazione della funzione soglia;
- rilevamento degli spigoli;
- estrazione delle regioni.

Le tecniche di segmentazione si sono evolute negli ultimi due decenni [80, 100] riuscendo a migliorare le performance dei riconoscitori. Spesso le tecniche di patter-recognition sono applicate a immagini pre-elaborate, ad esempio viene trattato il problema della sfocatura e viene tolto il rumore di fondo. Questo rende più semplice l'applicazione degli algoritmi di pattern-recognition.

Una tecnica ampiamente utilizzata utilizza l'accrescimento di una regione e la rilevazione degli spigoli [26].

3.2 La segmentazione

3.2.1 Connessione tra pixel

La segmentazione I è un'operazione che decompone l'immagine in N_R regioni R_i con $i = 1 \dots N_R$. Una regione è un insieme di pixel connessi. Castleman enfatizza la necessità di introdurre il concetto di pixel connessi come un concetto fondamentale nei metodi di segmentazione dell'immagine. Ci sono essenzialmente 3 tipo di di connettività tra pixel e si parla di:

- pixel 4 connessi
- pixel 6 connessi
- pixel 8 connessi

In figura 3.1 esempi esplicativi delle connessioni.

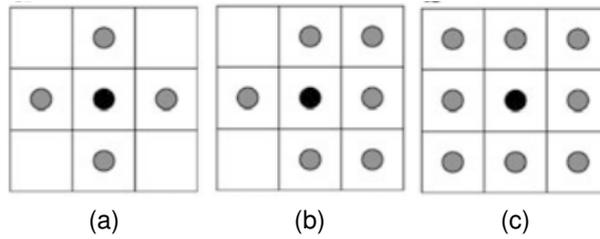


Figura 3.1: connettività tra pixel, in (a) pixel 4 connessi (b) una delle quattro permutazioni dei pixel 6 connessi (c) pixel 8 connessi

3.2.2 Omogeneità

Dal lavoro [88] la nozione di omogeneità tra pixel è inseparabile quando si parla di segmentazione. Apparentemente il concetto di omogeneità può essere interpretato semplicemente come l'insieme dei pixel vicini che visivamente rappresentano la stessa entità. In realtà l'omogeneità intesa come proprietà coerente significativa per la segmentazione deve essere definita in funzione del fenomeno analizzato [65, 182, 161].

Nell'equazione 3.1 definiamo un criterio binario $\gamma(R_i)$ di omogeneità per una regione R_i . L'equazione 3.1 definisce le regioni omogenee e quelle non omogenee.

$$\gamma(R_i) = \begin{cases} TRUE & \text{se } P \in R_i : \|I(P) - \mu(R_i)\| \leq \Theta \\ FALSE & \text{altrimenti} \end{cases} \quad (3.1)$$

dove $\mu(R) = \frac{1}{\|R\|} \sum_{P' \in R} I(P')$ è il valore dei pixel nella regione e Θ è una soglia.

Questa formulazione del problema descrive le regioni in cui i pixel non differiscono dalla media del valore dei pixel oltre una certa soglia. questo tipo di operazione richiede che \mathbb{V} sia un'algebra dotata di proprietà additiva e norma. Questa definizione non solo può non portare a una segmentazione unica dell'immagine, risulta spesso troppo semplice per fornire risultati su casi reali.

3.2.3 Distanza e misura della similarità

Durante il processo di segmentazione generalmente si cerca di massimizzare l'eterogeneità tra le regioni e l'omogeneità all'interno di una regione con una penalizzazione di una o dell'altra funzione obiettivo che dipende dal numero di regioni segmentate. Per calcolare la proprietà di omogeneità si può ricorrere alla descrizione della proprietà di disomogeneità o distanza. Successivamente questa proprietà può essere minimizzata all'interno di una regione. A tale scopo introduciamo la distanza di Minkowski [133] di ordine p tra due punti $X = (x_1, \dots, x_n)$ e $Y = (y_1, \dots, y_n)$ è definita in equazione 3.2.

$$d(X, Y) = \left(\sum_{i=1}^n \|x_i - y_i\|^p \right)^{\frac{1}{p}} \quad (3.2)$$

La distanza di Minkowski generalmente è utilizzata per valori di p da 1 a 2. il secondo coincide con la distanza euclidea, il primo prende il nome anche di distanza Manhattan. Al caso limite otteniamo la norma ∞ (o distanza di Chebyshev) [184]:

$$\lim_{p \rightarrow \infty} \left(\sum_{i=1}^n \|x_i - y_i\|^p \right)^{\frac{1}{p}} = \max_{i=1}^n \|x_i - y_i\| \quad (3.3)$$

La distanza di Hamming (molto utilizzata nella teoria dell'informazione) tra due vettori è uguale al numero di posizioni nelle quali corrispondono simboli diversi. In altre parole misura il numero minimo di sostituzioni necessarie a trasformare un vettore in un altro.

La distanza di Mahalanobis è utile per determinare la somiglianza tra un insieme di campioni non ben noti e un insieme di campioni noti. La differenza fondamentale dalla distanza euclidea è che viene considerata la correlazione tra le variabili, rendendo l'analisi invariante rispetto alla scala. Formalmente la distanza di Mahalanobis di un vettore multivariato $X = (x_1, \dots, x_n)^T$ da un gruppo di valori con media $\mu = (\mu_1, \dots, \mu_n)^T$ e matrice di covarianza C è definito come:

$$D_M(X, \mu) = \sqrt{(X - \mu)^T C^{-1} (X - \mu)} \quad (3.4)$$

La distanza di Mahalanobis può essere definita anche come la misura della diversità tra due vettori casuali X e Y che hanno la stessa distribuzione con matrice di covarianza C :

$$d_M(X, Y) = \sqrt{(X - Y)^T C^{-1} (X - Y)} \quad (3.5)$$

Se la matrice di covarianza è la matrice identità, la distanza di Mahalanobis si riduce alla distanza euclidea. Se la matrice è diagonale, allora la distanza è chiamata distanza euclidea normalizzata:

$$d_M(X, Y) = \sqrt{\sum_{i=1}^N \frac{(x_i - y_i)^2}{\sigma_i^2}} \quad (3.6)$$

dove σ_i è la deviazione standard di x_i sul set di campioni.

Le diverse misure introdotte dalle equazioni 3.2 3.3 3.4 3.5 3.6 possono essere utilizzate in diversi contesti nei quali è necessario misurare la distanza tra i pixel. Talvolta però è utile confrontare le distribuzioni dei pixel anziché i pixel, per questo scopo sono utilizzati gli istogrammi. Poiché un istogramma può essere interpretato come una distribuzione di probabilità empirica, agli istogrammi possono essere applicate misure per calcolare la distanza tra le distribuzioni di probabilità. Le metriche per il calcolo delle distanze che verranno illustrate sono applicabili tra due funzioni di distribuzione e quindi possono essere utilizzate nei casi in esame.

La distanza di Bhattacharyya misura la similarità di due distribuzioni di probabilità discrete o continue [11]. Siano $p(x)$ e $q(x)$ due distribuzioni di probabilità, la distanza di Bhattacharyya è generalmente utilizzata nei metodi di classificazione per misurare il grado di separabilità di due classi. Essa è strettamente connessa al coefficiente di

Bhattacharyya che misura la sovrapposizione tra campioni statistici o popolazioni, ed è utilizzato per misurare quanto sono vicini dei due campioni:

$$D_B(p(x), q(x)) = -\ln [BC(p(x), q(x))] \quad (3.7)$$

dove:

$$BC(p(x), q(x)) = \int_{-\infty}^{+\infty} \sqrt{p(x)q(x)} dx \quad (3.8)$$

Anche la distanza di Hellinger è utilizzata per quantificare la somiglianza tra due distribuzioni di probabilità. Essa è definita dall'integrale di Hellinger [92] (un caso particolare dell'integrale di Kolmogorov¹) che è un tipo di *f-divergenza*.

Definizione (funzione *f*-divergente). Siano $(\mathfrak{X}, \mathfrak{B})$ uno spazio misurabile e $\{P_\theta\}$ dove $\theta \in \Theta$ e Θ un insieme tale che una famiglia di misure di probabilità continue relative a una misura μ su \mathfrak{B} che è σ -finita, siano P_{θ_1} e P_{θ_2} due distribuzioni di probabilità su uno spazio \mathfrak{X} tale che P_{θ_1} sia continuo rispetto a P_{θ_2} e . Allora, per una funzione convessa f tale che $f(1) = 0$, la *f*-divergenza di θ_2 da θ_1 è definita come

$$D_f(P_{\theta_1} || P_{\theta_2}) = \int_{\mathfrak{X}} f\left(\frac{dP_{\theta_1}}{dP_{\theta_2}}\right) dP_{\theta_2} \quad (3.9)$$

Se P_{θ_1} e P_{θ_2} sono entrambe continue rispetto a una distribuzione di riferimento φ su \mathfrak{X} allora le loro densità di probabilità soddisfano le equazioni $dP_{\theta_1} = p_{\theta_1} d\varphi$ e $dP_{\theta_2} = p_{\theta_2} d\varphi$. In questo caso la *f*-divergenza può essere scritta come

$$D_f(P_{\theta_1} || P_{\theta_2}) = \int_{\mathfrak{X}} f\left(\frac{p_{\theta_1}(x)}{p_{\theta_2}(x)}\right) p_{\theta_2}(x) d\varphi(x) \quad (3.10)$$

La distanza di Hellinger [141] tra P_{θ_1} e P_{θ_2} con $(\theta_1, \theta_2) \in \Theta$ è definita come:

$$D_H(\theta_1, \theta_2) = \sqrt{\int_{\mathfrak{X}} \left[\sqrt{\frac{dP_{\theta_1}}{d\mu}} - \sqrt{\frac{dP_{\theta_2}}{d\mu}} \right]^2 d\mu} \quad (3.11)$$

La distanza (o divergenza) di Kullback–Leibler è una misura non simmetrica che calcola la differenza tra due distribuzioni di probabilità $p(x)$ e $q(x)$. Se le distribuzioni di probabilità sono continue, la divergenza di Kullback–Leibler è definita come:

$$D_{KL}(p(x), q(x)) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx \quad (3.12)$$

La figura 3.2 esemplifica illustrando il calcolo della distanza tra una gaussiana $p(x)$ e sue varianti $\tilde{p}(x)$ con media maggiore di 0 ($D_{KL}(\tilde{p}(x), p(x)) = \int_{-\infty}^{\infty} p(x) \ln \frac{\tilde{p}(x)}{p(x)} dx$).

La distanza di Kolmogorov–Smirnov misura la massima discrepanza tra le somme di due distribuzioni cumulative [11]. È data dalla formula:

$$D(X, Y) = \max_i \|F(i, X) - F(i, Y)\| \quad (3.13)$$

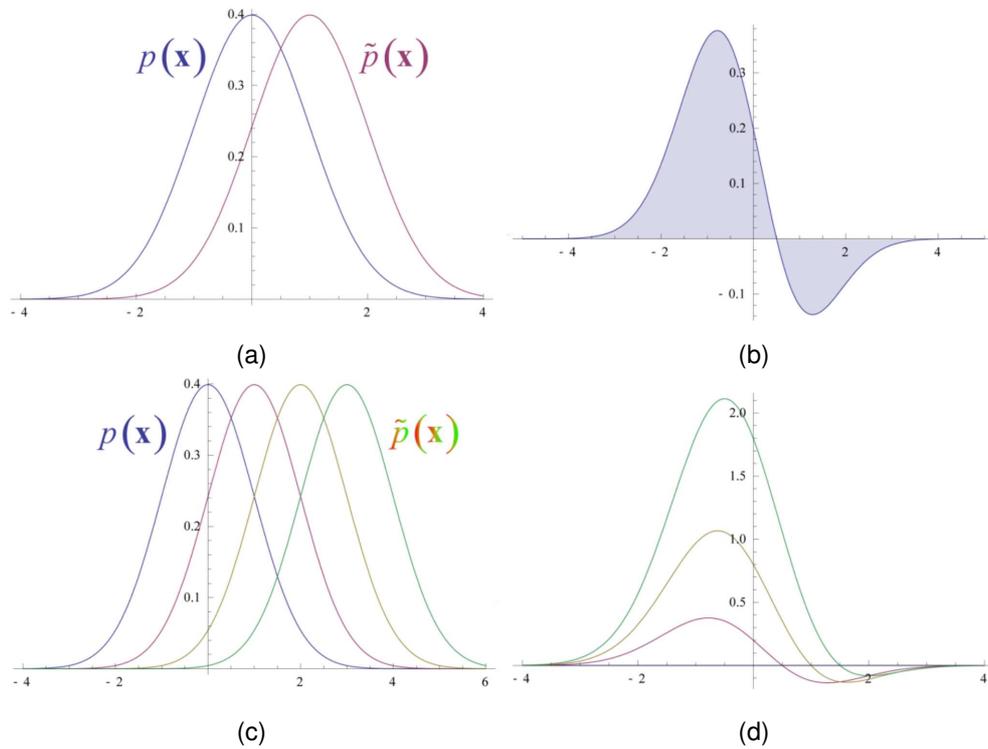


Figura 3.2: divergenza di Kullback–Leibler $D_{KL}(\tilde{p}(x), p(x))$, in (a) le distribuzioni di probabilità gaussiane $\tilde{p}(x)$ e $p(x)$, (b) il calcolo della distanza di Kullback–Leibler tra le due funzioni, (c) (d) a confronto il calcolo della distanza di Kullback–Leibler tra distribuzioni di probabilità gaussiane [138]

dove $F(i, X)$ è l'*i*-esimo valore della distribuzione cumulativa di X . In figura 3.3 un esempio che illustra il calcolo della distanza di Kolmogorov–Smirnov. La distanza di Kullback–Leibler generalmente è usata quando si effettuano i calcoli in presenza di probabilità condizionata [11] per misurare come cambia l'informazione nelle distribuzioni prima e dopo aver applicato delle ipotesi inferenziali. La distanza di Kolmogorov–Smirnov viene tipicamente utilizzata nel contesto di un test non parametrico.

3.3 Analisi delle caratteristiche e segmentazione

L'elaborazione di segnali (o immagini) dipende molto dalla definizione delle caratteristiche che si fanno emergere nel segnale e dal metodo di misura utilizzato per misurarle. Utilizzando queste definizioni si cerca di estrapolare le informazioni che sono nelle immagini immerse sotto uno strato di complessità. L'estrazione dell'informazione avviene utilizzando strutture dati complesse o più semplicemente vettori multidimensionali. Inoltre, al fine

¹ http://www.encyclopediaofmath.org/index.php/Kolmogorov_integral

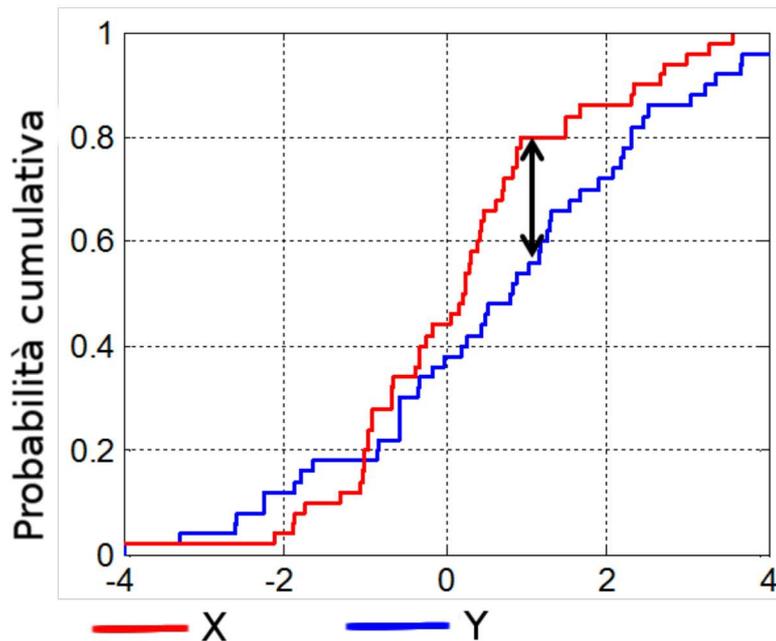


Figura 3.3: distanza di Kolmogorov–Smirnov, in rosso e in blu sono illustrate due probabilità cumulative, la freccia doppia indica la distanza massima tra le due curve

di segmentare, la finalità delle caratteristiche far risaltare le proprietà di omogeneità delle aree o marcare gli elementi distintivi tra le aree. La domanda principale diviene quindi come stimare questi criteri.

3.3.1 La distribuzione del colore

La distribuzione del colore è un elemento chiave nell'analisi delle immagini, offrendo la possibilità di effettuare sia una caratterizzazione locale che una caratterizzazione globale. Le caratteristiche legate al colore sono una caratteristica importante, infatti, gli algoritmi che le utilizzano sono robusti rispetto alle dimensioni dell'immagine (fattore di scala), la risoluzione e l'orientamento.

Molte caratteristiche sono costruite sull'assunzione che l'immagine analizzata sia una particolare istanza di un *random field* [1], ovvero che possa essere modellata per mezzo di una variabile multivariata casuale nello spazio. L'istogramma di una immagine in questo caso è una stima della funzione della densità di probabilità. Tutti gli algoritmi basati su una soglia nel dominio degli istogrammi, astraggono il calcolo dall'informazione spaziale che è persa nell'istogramma. Talvolta gli algoritmi che utilizzano la funzione istogramma, riducono il numero dei colori presenti nell'immagine, ad esempio, in [74] viene messo in evidenza come la riduzione del numero dei colori renda più robusta la segmentazione rispetto alle variazioni di luce.

3.3.2 Le trame dell'immagine

Data una area di un'immagine, il fattore J è definito da una misura di similarità della trama. In [49] Deng assume che anche in una immagine in scala di grigi, in ogni regione siano presenti solo pochi colori significativi. Da questo punto di vista il fattore J esprime una variazione normalizzata delle distanze spaziali dei centri delle aree omogenee. Nel caso di una finestra centrata su una regione omogenea, il fattore J ha valore 1.

Dato un insieme Q di N pixel ($Q = \{P_1, \dots, P_N\}$) la posizione media m è data dall'equazione:

$$m = \frac{1}{N} \sum_{i=1}^N P_i \quad (3.14)$$

Ponendo C le classi di Q , chiamiamo Q_i le regioni cui afferiscono le classi. Poniamo m_i il valore medio di M punti ($M < N$ e $M = \{P_j, \dots, P_k\}$) della classe Q_i , il valore medio per ciascuna classe è dato da:

$$m_i = \frac{1}{Q_i} \sum_{P \in Q_i} P_i \quad (3.15)$$

Definiamo in 3.16 e 3.17 le varianze spaziale totale e la varianza spaziale locale di una classe Q_i :

$$S_T = \sum_{q \in Q} \|q - m\|^2 \quad (3.16)$$

$$S_W = \sum_{i=1}^C \sum_{q \in Q} \|q - m\|^2 \quad (3.17)$$

La misura di J è definita come:

$$J = \frac{S_T - S_W}{S_W} \quad (3.18)$$

J essenzialmente misura la distanza tra le classi: un elevato valore di J indica che le classi sono più separate rispetto a bassi valori di J . In [52] l'espressione, a vantaggio di una maggiore stabilità, è modificata in:

$$\tilde{J} = \frac{S_T - S_W}{S_T} \quad (3.19)$$

In figura 3.4 un esempio del calcolo del valore di J . Sperimentalmente [49] mostra che l'efficacia di fattore J aumenta quando il rapporto $\frac{\text{numero dei colori}}{\text{numero di pixel}}$ è più piccolo del numero dei pixel presenti nella finestra e quindi, tipicamente, non è utile nelle immagini in bassa risoluzione.

In [97] il problema viene affrontato introducendo il concetto di correlogramma che, sostanzialmente, calcola la probabilità $\gamma_{v_i, v_j}^{(k)}$ che un pixel in posizione P_1 di colore v_i abbia un pixel P_2 vicino a distanza k di colore v_j .

$$\gamma_{v_i, v_j}^{(k)} \triangleq p(v_i, v_j | \exists X_1 \exists X_2 : (\Upsilon(X_1) = v_i) \text{ and } (\Upsilon(X_2) = v_j)) \text{ and } (\|\Lambda(X_1) - \Lambda(X_2)\| = k) \quad (3.20)$$

dove:

3.3. ANALISI DELLE CARATTERISTICHE E SEGMENTAZIONE

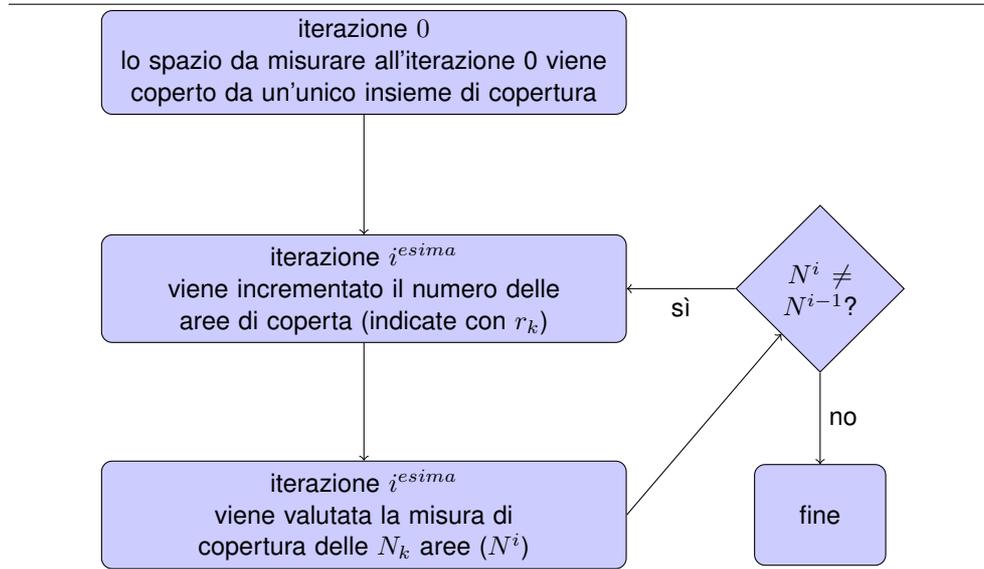
che permettono, rispettivamente, di calcolare la dimensione frattale e la dimensione lacunare. La dimensione frattale è una misura che caratterizza la complessità di un frattale, indicando la percentuale dello spazio che è occupata dal frattale stesso. La dimensione lacunare misura la distribuzione della massa del frattale nello spazio occupato.

La dimensione frattale può essere usata per misurare efficacemente le trame che sono invarianti rispetto alla scala, alla rotazione o alla traslazione. La dimensione lacunare può essere efficacemente usata per la classificazione e la segmentazione come descritto in [31, 196].

Nel 1919 Hausdorff propose una procedura di suddivisione multiscala per misurare gli oggetti irregolari basata sulla copertura del frattale.

La copertura del frattale avviene utilizzando il concetto di vicinanza, ovvero, si possono considerare piccole aree adiacenti di qualsiasi forma (generalmente dei cerchi) che hanno il centro geometrico o sovrapposto o vicino al frattale. Se il frattale può essere racchiuso in uno spazio euclideo (ad esempio la curva di Koch è contenuta in uno spazio $D_E = 2$) viene utilizzata un'area r della stessa dimensione dello spazio. Generalmente il calcolo viene effettuato iterativamente quindi se al passo 1 r coprisse l'intero oggetto da misurare, avremmo una approssimazione eccessiva (al passo 1). L'algoritmo 1 descrive i passi principali [132].

Algoritmo 1 Algoritmo di Hausdorff



La funzione di approssimazione al limite è espressa dall'equazione 3.23.

$$D_H \triangleq \lim_{k \rightarrow \infty} \frac{\ln(N_k)}{\ln\left(\frac{1}{r_k}\right)} \quad (3.23)$$

In figura 3.5 un esempio di calcolo delle dimensioni delle coste della Gran Bretagna con il metodo di Hausdorff.



Figura 3.5: esempio di calcolo della dimensione di Hausdorff per misurare le coste della Gran Bretagna

La dimensione calcolata da Renyi è una estensione della dimensione di Shannon (a sua volta è basata sull'entropia). Prendiamo in esame un frattale che è coperto da N_k regioni (in modo simile a quanto avviene con il calcolo della dimensione di Hausdorff). Consideriamo la densità del frattale D_I calcolata considerando le aree di intersezione delle N_k regioni con il frattale. Se chiamiamo n_{jk} la frequenza con cui il frattale si interseca con la j -esima area di dimensione r_k , allora il rapporto tra il numero totale di intersezioni N_{TK} e il frattale essere stimato dalla probabilità p_{jk} :

$$p_{jk} \triangleq \lim_{k \rightarrow \infty} \left(\frac{n_{jk}}{N_{TK}} \right) \quad (3.24)$$

dove

$$N_{TK} \triangleq \sum_{j=1}^{N_k} n_{jk} \quad (3.25)$$

Con questa distribuzione di probabilità la media attesa del frattale contenuto nelle N_k regioni può essere espresso come funzione dell'entropia:

$$N \triangleq \sum_{j=1}^{N_k} p_{jk} I_{jk} \quad (3.26)$$

Il calcolo della dimensione di Renyi è definito da:

$$H_q \triangleq \frac{1}{1-q} \sum_{j=1}^{N_k} p_{jk} \quad (3.27)$$

dove q è chiamato ordine del momento e $0 \leq p < 1$ e $1 < q \leq \infty$.

L'algoritmo probabilistico definito da Voss [109] considera un'immagine come un insieme di punti S in uno spazio euclideo di dimensione E . I punti sono spazialmente distribuiti con densità di probabilità $\mathbb{P}(m, L)$, ovvero la probabilità di avere m inclusi nella iper-regione di dimensione L centrato su un punto arbitrario di S . Poiché il calcolo è normalizzato, abbiamo che:

$$\sum_{m=1}^{N_p} \mathbb{P}(m, L) = 1 \quad (\forall L) \quad (3.28)$$

dove N_p è il numero dei pixel racchiusi in una regione di dimensione L .

Data un'immagine composta da M punti, il numero di regioni che contengono m punti è $\frac{M}{m} \mathbb{P}(m, L)$. Il numero totale di regioni per coprire l'intera immagine è:

$$\sum_{m=1}^{N_p} \frac{M}{m} \mathbb{P}(m, L) \quad (3.29)$$

3.3.4 Metodo piramidale

Il metodo piramidale riduce lo spazio di ricerca lavorando su un'immagine a risoluzione originale e a risoluzione diminuita [7, 127, 154]. In un'analisi $4 - to - 1$ ogni livello sottocampionato generalmente è ottenuto dividendo la risoluzione dell'immagine di 4 volte (con un fattore di scala 2 per ogni asse) e applicando una sfocatura con distribuzione gaussiana. L'idea fondamentale consiste nel segmentare l'immagine a bassa risoluzione e raffinare la segmentazione man a mano che viene analizzata l'immagine a risoluzione più elevata. La struttura piramidale riduce lo spazio in cui viene effettuato il calcolo: l'uso dell'immagine in bassa risoluzione riduce lo spazio di ricerca, successivamente con le immagini a risoluzione maggiore i punti determinanti per la segmentazione sono ricercati nelle vicinanze di quelli già calcolati. In figura 3.6 un esempio.

3.3.5 Metodo spartiacque

Il metodo spartiacque (*watershed*) modella il problema della segmentazione analizzando l'immagine come se fosse una cartina geografica in rilievo [13, 159]. Storicamente i metodi spartiacque sono stati ideati e utilizzati per lo studio di problemi morfologici; fin dal loro primo utilizzo nell'analisi delle immagini ne sono state sviluppate molte varianti [36]. Il procedimento può essere immaginato come della pioggia che cade su un plastico: l'acqua si raccoglie nei punti più bassi [14]. Gli spartiacque (o dighe) sono determinati dalle aree che rimangono al di sopra della quota dell'acqua. Durante il processo di allagamento di un'immagine in generale vengono eseguite le seguenti azioni:

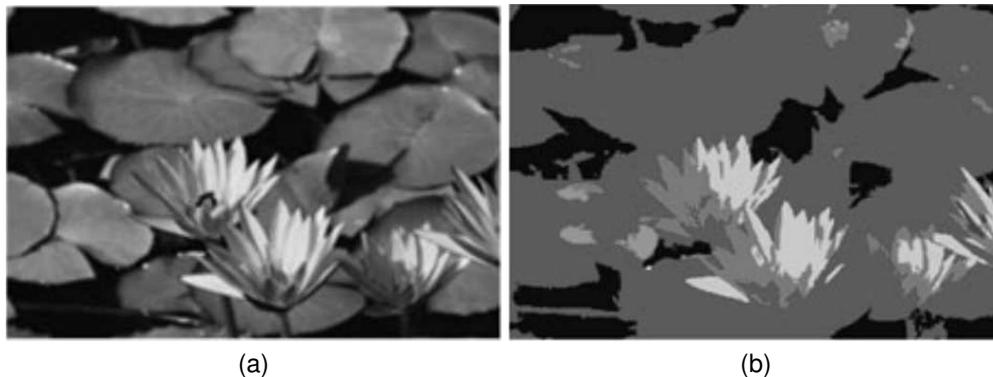


Figura 3.6: segmentazione con il metodo piramidale, in (a) l'immagine originale e in (b) l'immagine segmentata

- man a mano che i pixel sono allagati possono essere etichettate le corrispondenti regioni
- se due regioni si intersecano, devono essere mantenute divise imponendo una diga nel mezzo alle due regioni

L'approccio spartiacque non supervisionato inizia il processo di allagamento partendo dai minimi locali. Per separare le diverse regioni, quando si incontrano, come descritto per il caso generale, vengono costruite delle dighe. Il processo termina quando l'intera immagine è sommersa e rimangono emerse solamente le dighe di separazione tra le regioni. Utilizzando questa visione del problema, un'immagine in toni di grigio può essere vista come una mappa di livelli in cui la tonalità del colore indica l'altezza del livello [35].

Uno svantaggio di questo metodo di segmentazione consiste nel perdere informazioni di tipo globale relative all'immagine dando risalto alle informazioni locali. In figura 3.7 un esempio.

Nel capitolo X di [211] è descritto un approccio che usa il metodo spartiacque per trovare i cluster usando le caratteristiche di uno spazio.

Il punto critico di questo algoritmo è come vengono riempite le dighe. Se utilizziamo ancora l'immagine di una mappa di livelli, è ragionevole aspettarsi che la crescita della regione avvenga sfruttando la distanza euclidea tra pixel, questo potrebbe implicare un eccessivo costo computazionale. Vincent [187] al fine di velocizzare la computazione utilizza la distanza Manhattan.

3.3.6 Contorni attivi

Il metodo dei contorni attivi, conosciuto anche come *snake* fu introdotto da Kass nel 1988 [108]. Nell'articolo veniva definita una spline che minimizza l'energia guidata da forze vincolanti esterne e influenzata dalle forze dell'immagine che la spingono verso gli spigoli. Il metodo è stato efficacemente usato in ambito medico [169], specialmente per la

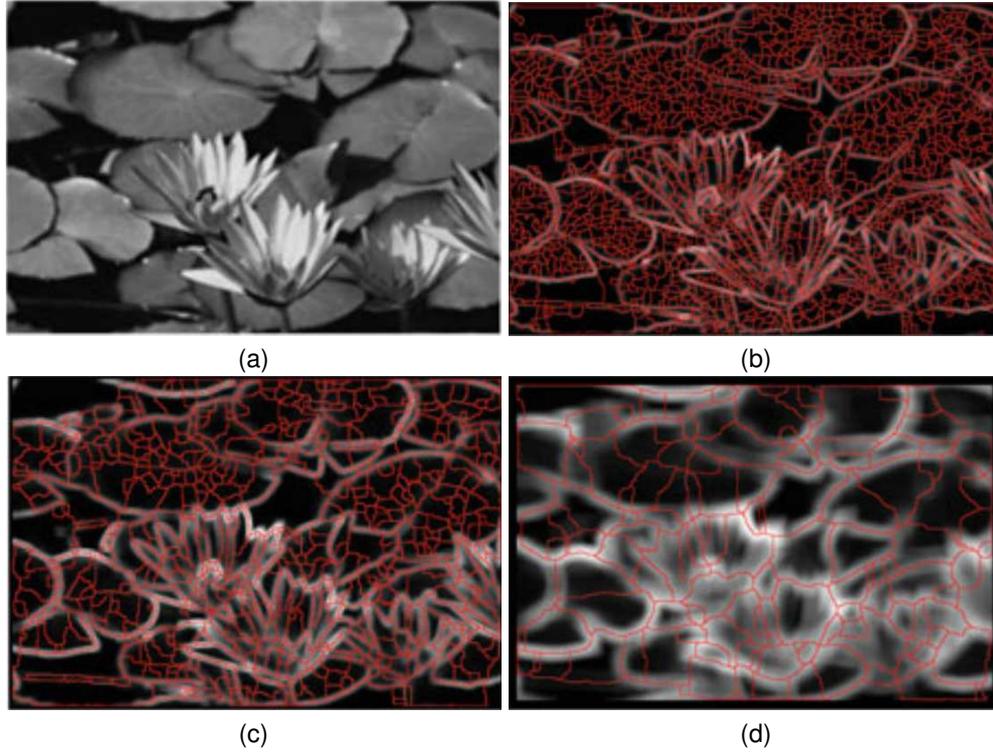


Figura 3.7: metodo watershed, in (a) l'immagine sorgente, da (b) a (d) come si evolve il metodo

tomografia computerizzata e per l'analisi delle immagini del cervello ottenute per mezzo della risonanza magnetica. Il contorno iniziale viene deformato in modo incrementale secondo lo schema energetico calcolato per l'immagine. Il contorno attivo è definito come una spline $c(s) = [x(s), y(s)]$ con $s \in [0, 1]$ che minimizza la funzione energetica 3.30 [177].

$$\epsilon(c) = \epsilon_{int}(c) + \epsilon_{ext}(c) = \int_0^1 [E_{int}(c(s)) + E_{ext}(c(s))] ds \quad (3.30)$$

dove $\epsilon_{int}(c)$ rappresenta l'energia interna e $\epsilon_{ext}(c)$ l'energia esterna. L'energia interna è intrinseca nella spline mentre l'energia esterna può essere calcolata da una pre-elaborazione dell'immagine oppure essere fornita dall'utente (generalmente sotto forma di costante). L'energia interna $\epsilon_{int}(c)$ può essere espressa come in 3.31.

$$\epsilon_{int}(c) = \int_0^1 \frac{1}{2} [\alpha(s) \|c'(s)\|^2 + \beta(s) \|c''(s)\|^2] ds \quad (3.31)$$

dove $c'(s)$ e $c''(s)$ sono la derivata prima e la derivata seconda pesate che nel calcolo vengono pesate da $\alpha(s)$ e $\beta(s)$ che sono a loro volta considerate costanti nella maggior

parte delle implementazioni.

Xu [199] nel suo lavoro mette in evidenza alcune peculiarità del lavoro originale [108]:

- l'inizializzazione dei contorni attivi deve essere vicina agli spigoli
- il metodo converge con difficoltà in presenza di figure concave

Questi problemi erano stati affrontati parzialmente in [108] utilizzando la propagazione nello spazio scalare descritto in [183, 194]. Il modello a mongolfiera [33] introduce forze esterne supplementari aiutare l'identificazione dei contorni dell'oggetto da identificare anche quando questi contorni non sono ben identificabili nella figura. Gli inconvenienti di questa metodologia sono stati corretti da Kapur [106]. Successivamente, per apportare ulteriori miglioramenti, Xu [197, 198] ha introdotto il flusso del vettore gradiente. Il metodo del flusso del vettore gradiente è spesso utilizzato per i risultati che permette di ottenere nonostante sia complesso e computazionalmente oneroso. Esistono altri approcci che affrontano il problema in modo diverso da quanto fatto da Xu [197, 198]: il campo elettrico virtuale [144], il campo dei vettori di confine [175], il campo a gradiente diffuso [113] e il flusso del vettore fluido [191]. In figura 3.8 un esempio.

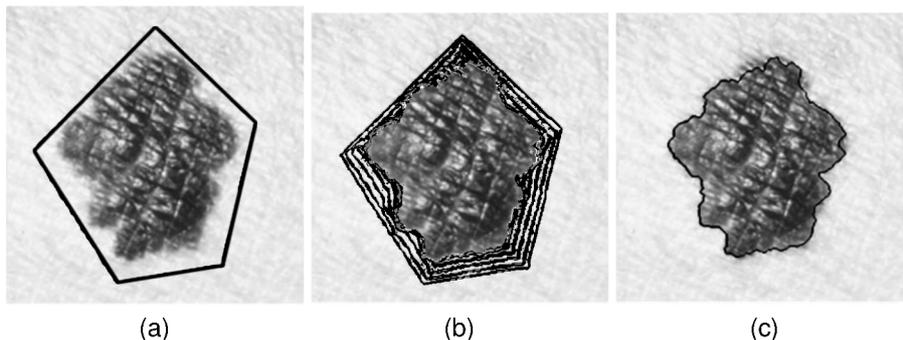


Figura 3.8: metodo dei contorni attivi, in (a) il contorno iniziale, in (b) alcune configurazioni intermedie e in (c) il contorno finale

3.3.7 Metodo basato sui grafi

La segmentazione basata su grafo si sta oggi diffondendo grazie alla caratteristica del metodo di usare caratteristiche locali e globali dell'immagine. I principali metodi basati su grafo sono quelli basati sul taglio degli archi, i metodi interattivi e gli alberi di copertura minima. Nei metodi illustrati in questo paragrafo i nodi del grafo rappresentano diversi aspetti dell'immagine: in [137, 48, 134, 10, 149, 142, 158, 18] rappresentano i pixel, in [62, 60] i vertici dei pixel, in [127] le regioni e in [55] i marcatori indicati dall'utente. I principali metodi per risolvere questo tipo di problemi sui grafi è il matching tra i grafi [55], il random walker [83], l'algoritmo di taglio minimo (o di massimo flusso) [78, 51, 20, 209, 18], l'algoritmo di Dijkstra [137, 62, 60, 61, 134, 10, 149, 158] e l'algoritmo di Kruskal

3.3. ANALISI DELLE CARATTERISTICHE E SEGMENTAZIONE

[67, 66, 59, 210]. Anche gli algoritmi di ricerca sui grafi (come l'algoritmo di Dijkstra) sono utilizzati per l'individuazione delle regioni o dei bordi.

Ogni metodo tratta un'immagine come un grafo G i cui vertici sono messi in relazione con i pixel. Ogni arco ha un peso che generalmente è calcolato in funzione della legge che lega il grafo all'immagine. Da un punto di vista della teoria dei grafi, segmentare significa calcolare un insieme di sottografi $\{SG_1, \dots, SG_n\}$ a partire dal grafo G così che per ogni $k \in \{1, \dots, n\}$, $\forall i, j$ e $i \neq j, v_i, v_j \in SG_k$ con un cammino tra v_i e v_j . La combinazione di tutti i vertici di tutti i sottografi costituisce l'insieme completo di tutti i vertici del grafo. Ogni sottografo rappresenta un insieme di vertici con una forte affinità. In figura 3.9 rappresentiamo graficamente l'idea che sta alla base della segmentazione sfruttando un grafo. I principali metodi utilizzati per ottenere questo risultato possono

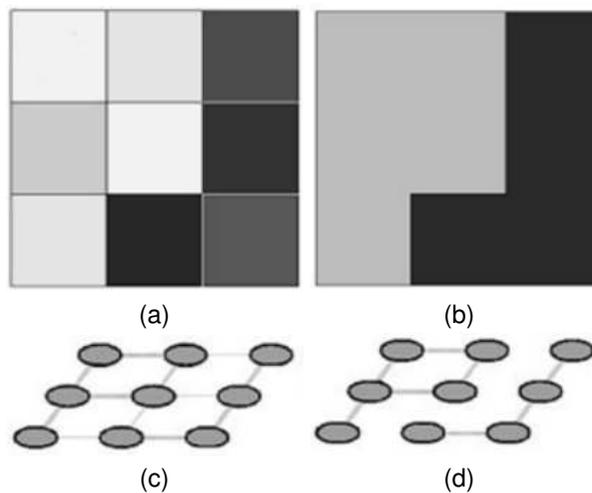


Figura 3.9: segmentazione con l'uso di un grafo, in (a) una immagine esempio, in (b) l'immagine segmentata, in (c) il grafo associato all'immagine esempio, in (d) il grafo associato all'immagine segmentata

essere raggruppati in metodi basati sul taglio del grafo, metodi interattivi e metodi basati sulla copertura minima del grafo.

I metodi basati sul taglio del grafo G nascono con il lavoro di Greig [85]. Il grafo G può essere partizionato in due componenti connesse A e B tali che $A \cup B = V$ e $A \cap B = \{\}$. Il metodo con cui si individuano i vertici assegnati agli insiemi A e B dipende dal criterio utilizzato con cui sono elisi gli archi. La generica funzione di taglio è espressa nell'equazione 3.32.

$$\text{cut}(A, B) = \sum_{\substack{u \in A \\ v \in B}} w(u, v) \quad (3.32)$$

In figura 3.10 un esempio di taglio del grafo. Il taglio ottimo minimizza il valore la funzione

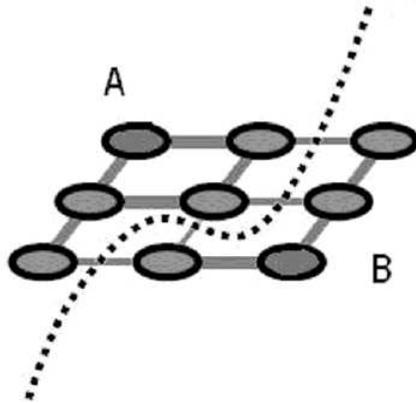


Figura 3.10: un esempio del metodo del taglio del grafo

$cut(A, B)$ dell'equazione 3.32. Questa funzione, utilizzata in modo ricorsivo, può essere utilizzata per suddividere il grafico più volte. I principali riferimenti in questo campo sono [195, 167, 166, 93, 55, 200, 207, 160, 50, 122, 190, 103, 19, 84, 120, 36, 78, 209, 32, 205, 212, 96, 3, 25]. La soluzione [195] favorisce la formazione di regioni molto piccole che portano a una eccessiva segmentazione. Il problema viene parzialmente risolto in [167] in cui il taglio viene normalizzato: il calcolo della funzione di taglio dipende dal numero dei vertici e dal peso delle connessioni tra i vertici. L'equazione 3.33 descrive il criterio utilizzato da quest'ultima soluzione.

$$cut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \quad (3.33)$$

dove

$$assoc(A, V) = \sum_{u \in A, t \in V} w(u, t) \quad (3.34)$$

è il peso totale dei collegamenti da tutti i vertici in A a tutti i vertici del grafo. In [167] viene messo in evidenza che il problema 3.33 è NP-completo. Nell'articolo viene proposta un'approssimazione dell'algorithm per minimizzare il costo della funzione di taglio. Ci sono quattro passi nell'algorithm:

1. il grafo è calcolato a partire da un'immagine assegnando a ogni pixel un vertice e il peso tra due vertici è data da una funzione di similarità tra pixel vicini;
2. sono calcolati gli autovalori dell'equazione 3.35

$$(D - W)x = \lambda Dx \quad (3.35)$$

dove D è la matrice diagonale con

$$d(i) = \sum_j w(i, j) \quad (3.36)$$

e W è la matrice diagonale con $W(i, j) = w_{ij}$. x è l'autovettore mentre λ l'autovalore;

3. il grafo è bipartito utilizzando il più piccolo autovettore;
4. i passi precedenti sono ripetuti fino a quando il valore di taglio normalizzato non è sceso al di sotto della soglia calcolata.

Negli anni sono stati sviluppati molti metodi interattivi, nessuno dei quali ha proprietà generali migliori di un altro ma ognuno esprime le sue potenzialità in casi particolari. I passi principali di un algoritmo iterativo si riassumono in:

1. chiedere all'utente di effettuare le impostazioni iniziali;
2. generare la soluzione ottimale a partire dalle impostazioni ottimali;
3. mostrare il risultato all'utente.

Questi metodi sono utilizzati soprattutto quando gli algoritmi di segmentazione automatici non riescono a segmentare correttamente l'immagine. I principali metodi iterativi sono i metodi live-wire [137, 60, 62], il metodo Braquelair basato su una mappa strutturata dell'immagine [21], un metodo iterativo con il taglio del grafo [18, 17, 69, 118] e metodi che ottimizzano il tempo di esecuzione [124, 114, 105, 119, 162]. Altri contributi in questo campo sono dati da Miranda [48] con l'approccio sul peso degli archi, da Grady con l'algoritmo Grubcut [83], da Xu che raffina i contorni degli oggetti iterativamente [202, 201], da Wang con il metodo di taglio progressivo del grafo [206, 188]. Vediamo brevemente questi metodi di seguito. I metodi live-wire [137, 62] all'utente è richiesto di specificare alcuni punti del contorno dell'oggetto. A partire da questi punti viene applicato l'algoritmo di Dijkstra per calcolare il percorso più breve tra questi punti. Falcao ha introdotto una versione più veloce di questo metodo [60] ideando un nuovo algoritmo di ricerca sul grafico. Il metodo Falcao, su un campione di 492 immagini mediche ha impiegato, rispetto all'implementazione originale dell'algoritmo, un tempo inferiore da 1.3 a 31 volte. In figura 3.11 un esempio.

Il metodo Braquelair [21] effettua operazioni di fusione tra le aree adiacenti del grafo, l'uso della mappa permette una rappresentazione unificata della topologia e della geometria dell'immagine segmentata. Il metodo iterativo basato sul taglio del grafo [18, 17] per segmentare l'immagine sfrutta due vincoli: uno inflessibile e uno flessibile. Per comprendere questo metodo consideriamo $A = (A_1, \dots, A_p, \dots, A_{|\rho|})$ l'insieme dei segmenti, A_p può essere sia lo sfondo o l'oggetto in primo piano. Il vincolo flessibile è costituito dal costo di una funzione $E(A)$ che considera le proprietà dei vincoli e delle regioni dei segmenti.

$$E(A) = \lambda.R(A) + B(A) \quad (3.37)$$

Dove:

$$R(A) = \sum_{p \in \rho} R_p(A_p) \quad (3.38)$$

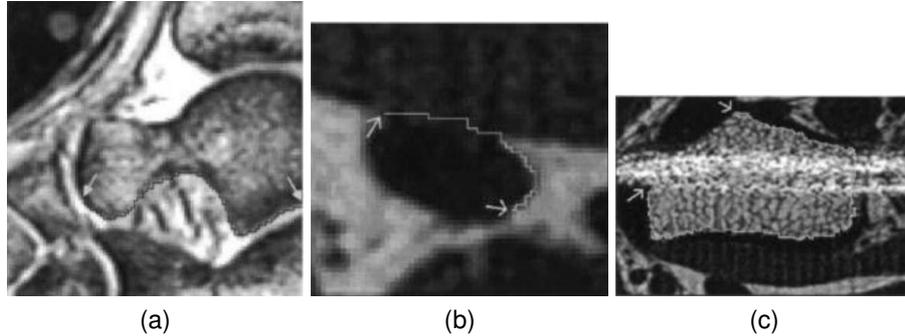


Figura 3.11: esempio di segmentazione live-wire, in (a) il riconoscimento di un astragalo in una risonanza magnetica di un piede, in (b) il riconoscimento di una sacca di sangue in un polso, in (c) il riconoscimento della parte corticale di un osso del polso

$$B(A) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \delta(A_p, A_q) \quad (3.39)$$

Dove N è l'insieme non ordinato di tutte le coppie di pixel vicini e

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{se } A_p \neq A_q \\ 0 & \text{altrimenti} \end{cases} \quad (3.40)$$

Le funzioni $R_p(\cdot)$ e $B_{p,q}$ specificano le penalità nell'associare un pixel all'insieme dei pixel dello sfondo oppure all'oggetto in primo piano. Il coefficiente $\lambda \in \mathbb{R}, \lambda \geq 0$ fornisce una priorità nell'associare un pixel a una regione piuttosto che a un bordo. L'utente fornisce l'indicazione sui vincoli marcando alcuni pixel dello sfondo e dell'oggetto in primo piano. I metodi basati sugli alberi di copertura minima esaminati in questo paragrafo trasformano l'immagine in un grafo connesso non orientato. L'albero di copertura minimo è un sotto-grafo (di peso minimo) che connette tutti i vertici del grafo e ha solamente un percorso per unire due vertici. Nel caso dell'identificazione di più oggetti in un'immagine, l'algoritmo di copertura calcola quando la distanza tra insiemi di nodi supera una funzione di soglia e divide i nodi in più sottoinsiemi. I principali metodi sono stati proposti da Zahn [208], Xu [203, 204] e da Felzenszwalb [67, 66]. Zahn [208] per individuare e definire i cluster affronta il problema utilizzando un metodo basato sulla teoria Gestalt [192]. In figura 3.12 un esempio. In Xu [203, 204] un algoritmo di partizionamento dell'albero divide l'albero di copertura minimo costruito (a partire da un'immagine) in molti sotto-alberi che rappresentano le regioni omogenee. Un vincolo che deve essere rispettato consiste nel mantenere significativamente diverso il valore medio del livello di grigio per ogni coppia di sotto-alberi vicini.

3.3.8 Metodo Spectral Embedding MinCut

Un altro esempio di applicazione del calcolo frattale viene presentato in [57, 56] con l'algoritmo *Spectral Embedding MinCut*. L'idea alla base di questo algoritmo è calcolare

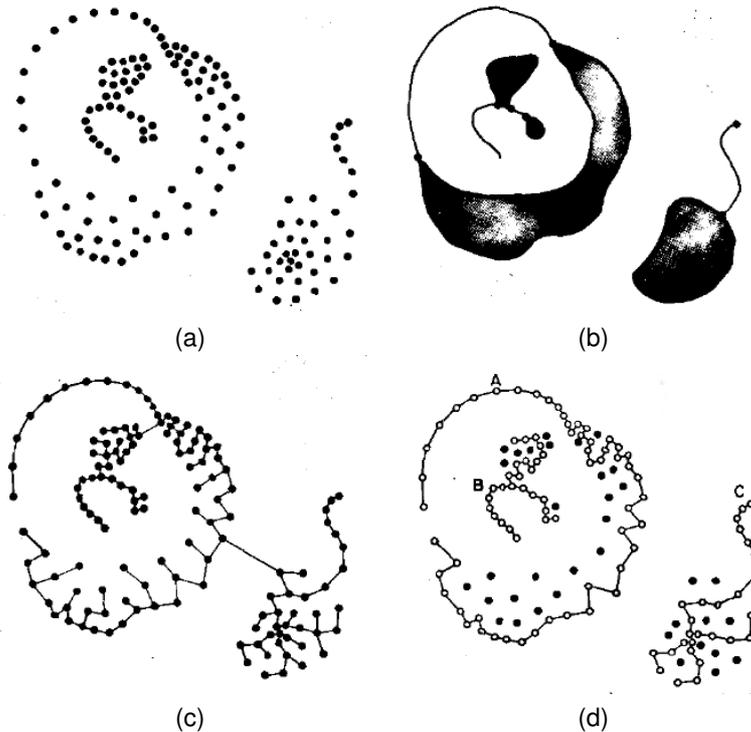


Figura 3.12: esemplificazione dell'algoritmo di Zahn [208], in (a) un insieme di punti calcolati da un'immagine, in (b) la visione gestalt del problema descritta nel lavoro originale, in (c) il grafo calcolato dai punti e in (d) il risultato dell'algoritmo

il costo della separazione tra aree e minimizzarla. La figura 3.13(a) rappresenta una semplice immagine campione e la figura 3.13(b) il grafo utilizzato per separare le aree dell'immagine. Nel grafo ogni pixel è rappresentato da un vertice del grafo e i collegamenti tra i vertici hanno un peso che rappresenta la similarità tra i pixel vicini (più alto è il valore più sono simili). Una misura del costo può essere calcolata sommando il peso dei rami tagliati per ogni singolo pixel. Nella figura, tagliando i rami con pesi 8, 10, 5, 5, 0, 5, 0, 0 otteniamo un costo di 33.

L'algoritmo Spectral Embedding MinCut costruisce una matrice di affinità W usando la misura di affinità tra pixel vicini. Dalla matrice di affinità, assumendo che non vi sia una colonna di tutti 0, viene calcolata una matrice di Markov M normalizzando a 1 ogni colonna di W :

$$M = WD^{-1} \tag{3.41}$$

dove D è la matrice diagonale con $D_{i,j} = \sum_k W_{k,i}$. La matrice M può essere usata per determinare il pattern di diffusione che corrisponde a una camminata casuale che inizia da diversi pixel dell'immagine. In particolare, se la partenza dal pixel \vec{x}_j è associata alla distribuzione di probabilità $p_{0,j}$, lo stato del random walk dopo t passi è dato da:

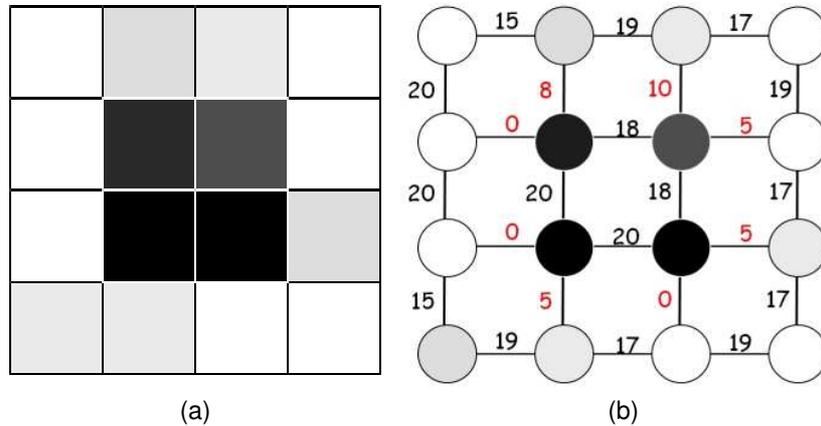


Figura 3.13: algoritmo Min-Cut, in (a) una immagine campione e in (b) i pesi associati a ogni spostamento

$$\vec{p}_{t,j} = M^t \vec{p}_{0,j} \quad (3.42)$$

dove M^t può essere facilmente calcolato sfruttando gli autovalori e gli autovettori M . I vettori $\vec{p}_{k,j}$ calcolati partendo da k diversi pixel dell'immagine sono chiamati kernel di sfocatura (blur kernels). I kernel di sfocatura catturano importanti proprietà delle relazioni tra pixel vicini nell'immagine raggruppando i pixel che appartengono allo stesso cluster. Ogni raggruppamento di pixel generalmente non copre tutta la figura così viene utilizzato come nucleo iniziale da estendere. L'estensione dei nuclei iniziali (detti semi in [57]) utilizza un grafo simile a quello dell'esempio precedente. Il costo del taglio di un grafo in due componenti non connesse R_1 e R_2 viene calcolato sommando il peso dei link che sono rimossi (per tagliare il grafo). Il costo di taglio è definito da:

$$cut(R_1, R_2) = \sum_{v_i \in R_1, v_j \in R_2} w_{i,h} \quad (3.43)$$

Il costo minimo è definito come:

$$MinCut(R_1, R_2) = \min_{R_1, R_2} cut(R_1, R_2) \quad (3.44)$$

In figura 3.14 un esempio completo di segmentazione di una figura.

3.4 Metodi per il riconoscimento di oggetti nelle immagini

Il riconoscimento degli oggetti nelle immagini completa gli argomenti lo stato dell'arte del pattern-recognition relativo a quanto trattato in questa ricerca. In questo paragrafo andiamo a specificare da un diverso punto di vista parte di quanto fin qui già illustrato impegnando i vari metodi per la segmentazione. I metodi per il riconoscimento possono

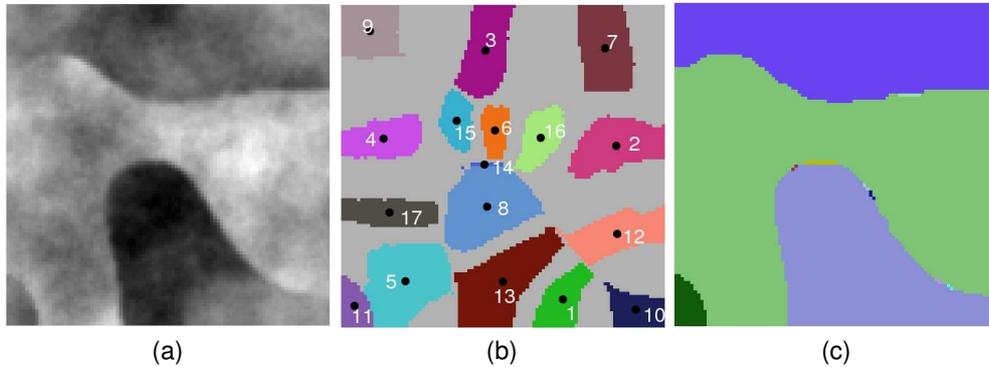


Figura 3.14: algoritmo SE Min-Cut, in (a) una immagine campione, in (b) il raggruppamento in regioni e in (c) il risultato della segmentazione SE Min-Cut]

essere notevolmente avvantaggiati dalla segmentazione dell'immagine: prendono in esame i vari segmenti e cercano di capire che cosa sono. A seguito sono spiegate le principali tecniche di pattern-recognition per il riconoscimento delle immagini.

3.4.1 Analisi statistica

Generalmente le tecniche di pattern-recognition basate su teorie statistiche e probabilistiche modellano i dati da analizzare per mezzo di serie modellano le singole caratteristiche. Questo tipo di modalità operativa è usato nella pratica in quanto è uno dei più semplici da gestire. In questo modo, i pattern che devono essere classificati sono rappresentati da un insieme di caratteristiche che definiscono uno spazio vettoriale multidimensionale dove ogni dimensione rappresenta i dati di una caratteristica. Per confrontare i pattern vengono eseguite statistiche sui punti nello spazio al fine di cercare delle regolarità [101].

3.4.2 Analisi sintattica

Generalmente le tecniche di pattern-recognition basate sull'analisi sintattica, che prende anche il nome di pattern-recognition strutturale in quanto prende in esame la struttura dei dati, elaborano i risultati utilizzando le relazioni tra le caratteristiche. Per tale motivo i pattern sono rappresentati dalle strutture che contengono i dati delle caratteristiche, per mezzo di queste strutture si riescono a rappresentare relazione complesse tra i dati delle caratteristiche [186]. I pattern possono essere descritti in una struttura gerarchica composta da sotto strutture che a loro volta possono contenere sotto strutture. Un primo esempio fu introdotto da Sonka [173], nell'esempio una figura è rappresentata da un insieme predefinito di primitive chiamate parole chiave. Le parole chiave vanno a formare quello che è chiamato il cifrario. La figura 3.15 (a) è un esempio di parole chiave e la figura 3.15 (a) di cifrario. La stringa C del cifrario può essere scritta dall'equazione 3.45.

$$S = d \ b \ a \ b \ c \ b \ a \ b \ d \ b \ a \ b \ c \ b \ a \ b \quad (3.45)$$

Il sistema effettua il parsing del cifrario utilizzando una sorta di grammatica predefinita. Se il cifrario può essere completamente riconosciuto dalla grammatica allora il sistema ha riconosciuto l'oggetto.

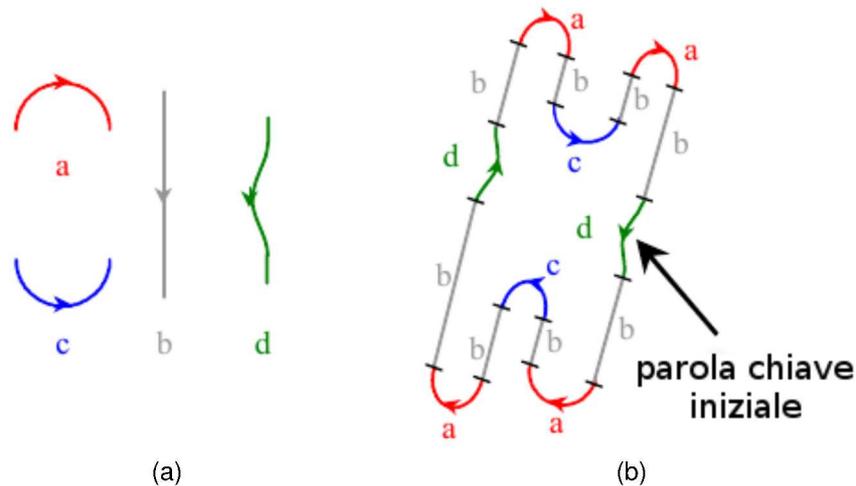


Figura 3.15: esempio di descrizione sintattica, in (a) le parole chiave, in (b) il cifrario

3.4.3 Template matching

L'analisi del template matching è molto utilizzata per identificare delle strutture nelle immagini che non presentano regolarità. con questo tipo di analisi le varie parti dell'immagine sono confrontate con un modello cercando di massimizzare la correlazione (o minimizzando la distanza) tra il modello e una parte dell'immagine.

Esistono varie tecniche per ottenere questo risultato, la più brutale di queste consiste nel confrontare ogni possibile rototraslazione, riduzione e/o deformazione del template con le varie aree dell'immagine. Per ogni confronto viene calcolato un punteggio e quando un punteggio supera una determinata soglia, l'oggetto è riconosciuto nell'immagine. Questa tecnica [22, 157, 9] è molto costosa. La figura 3.16 è un esempio di template matching. In particolare le immagini 3.16 (d) e 3.16 (e) presentano la cross correlazione con le immagini 3.16 (b) e 3.16 (c). Il cerchio indica il punto dove la cross-correlazione raggiunge il massimo. Utilizzando queste coordinate nell'immagine 3.16 (f) sono messi in evidenza i pattern riconosciuti. In questo esempio sono state considerate solamente le traslazioni dei pattern.

3.4.4 Reti neurali

Una rete neurali artificiale è una procedura addestrabile autoadattiva in grado di imparare a risolvere i problemi complessi utilizzando la conoscenza disponibile nel sistema della

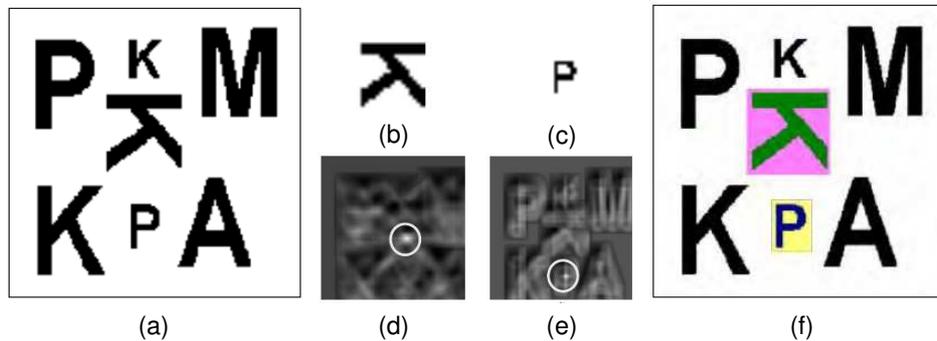


Figura 3.16: esempio di template matching, in (a) l'immagine sorgente, in (b) e in (c) i template, in (d) e in (e) la cross-correlazione per l'assegnamento dei punteggi, in (f) i pattern riconosciuti

stessa rete.

Una rete neurale artificiale cerca di simulare il funzionamento di cervello biologico: è composta da elementi di elaborazione interconnessi che simulano i neuroni. Utilizzando questa interconnessione (o sinapsi), ogni neurone può passare informazioni ad un altro neurone. Come si può vedere nella figura 3.17, queste interconnessioni sono pesate secondo i pesi definiti da una matrice dei pesi W : il peso applicato ad una connessione è il risultato del processo di apprendimento e indica l'importanza del contributo al neurone associato. La figura 3.17 mostra una semplice rete neurale che rappresenta il perceptrone come definito da Frank Rosenblatt nel 1957. Su questo ad esempio, l'uscita O_j , con $j = 1$ o $j = 2$, è definita da una combinazione ponderata dei dati di ingresso. Per maggiori dettagli Abdi [2] nel suo lavoro presenta una buona introduzione alle reti neurali artificiali.

In metodologia esistono altri metodi che utilizzano le reti neurali artificiali come quelli basati sulla logica fuzzy. In alcune applicazioni sono utilizzati metodi ibridi che combinano diversi aspetti di questi metodi al fine di ottenere sistemi risolutivi più precisi. Liu [123] nel suo lavoro presenta una panoramica di questi metodi per effettuare riconoscimenti di immagini.

3.4.5 Schema generali per un sistema di riconoscimento per le immagini

Un sistema per il riconoscimento delle immagini prende in ingresso una o più immagini e restituisce come valore una o più immagini in cui sono fornite espressioni sintattiche come un'immagine nella quale sono etichettate aree riconosciute. In figura 3.18 presentiamo uno schema generico di un sistema di riconoscimento per le immagini. I compiti principali del sistema sono l'analisi dell'immagine (vedi il punto ❶ della figura 3.18) e la classificazione e il riconoscimento dell'immagine (vedi il punto ❷ della figura 3.18).

Dopo l'estrazione delle caratteristiche, le caratteristiche devono essere selezionate per ridurre la complessità del calcolo senza peraltro modificare la risposta del clas-

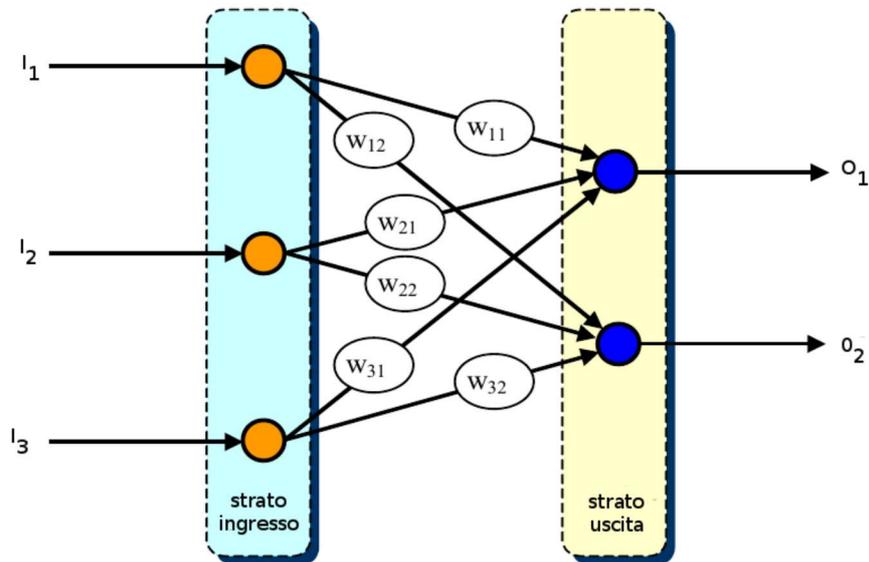


Figura 3.17: esempio di rete neurale artificiale

sificatore. Sostanzialmente si tratta di riconoscere quali caratteristiche non apportano informazione nel calcolo.

Durante questa elaborazione possono essere individuate caratteristiche di livello superiore che sono derivate dalla combinazione e/o trasformazione delle caratteristiche grezze di basso livello). A esempio, queste caratteristiche di alto livello possono essere calcolate applicando la cosiddetta analisi delle componenti indipendenti [157].

Queste caratteristiche devono essere tali da eliminare (o ridurre) i falsi positivi dovuti alla errata classificazione del riconoscitore (vedi il punto ② della figura 3.18). Le proprietà principali di alcune buone caratteristiche possono essere riassunte come:

- invarianza alla traslazione: un oggetto nella figura deve essere riconosciuto indipendentemente dalla sua posizione;
- invarianza alla rotazione: un oggetto nella figura deve essere riconosciuto indipendentemente dal suo orientamento;
- invarianza della scala: un oggetto nella figura deve essere riconosciuto indipendentemente dalla sua scala (ovvero dalla dimensione con cui è rappresentato);
- robustezza ai disturbi: una caratteristica deve essere robusta al rumore, al limite deve essere usabile indipendentemente dall'intensità del rumore;
- indipendenza statistica: due caratteristiche dovrebbero essere statisticamente indipendenti per avere la maggior efficacia;
- compattezza: il numero delle caratteristiche dovrebbe essere l'insieme minimo che permette l'esecuzione del calcolo

- affidabilità: eseguendo il calcolo con uno stesso pattern le caratteristiche estratte dovrebbero rimanere le medesime.

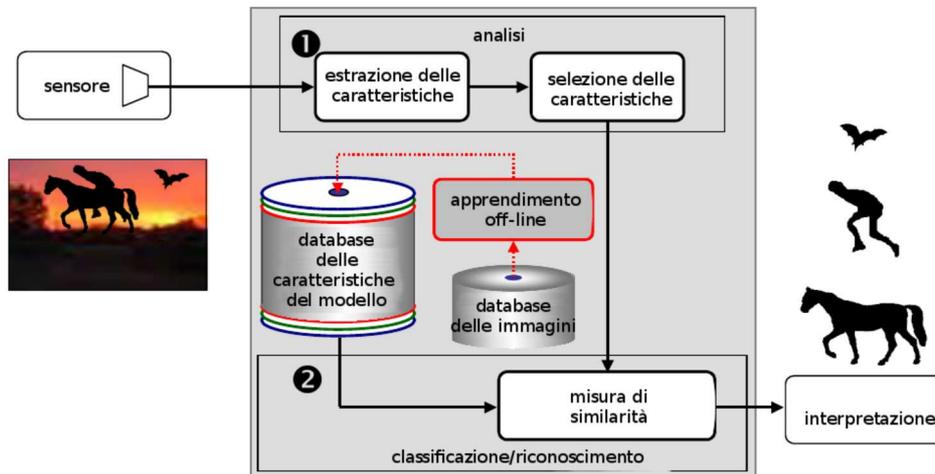


Figura 3.18: uno schema generale per il riconoscimento delle immagini

Durante l'operazione di classificazione, il sistema utilizza le caratteristiche estratte nella fase di analisi dei modelli per confrontare. Come illustrato nella figura 3.18, spesso per facilitare la fruizione di un insieme di caratteristiche si fa uso di una base di dati. A esempio, la base di dati viene sfruttata durante la fase di apprendimento off-line. Ciò consente di creare una base di conoscenza da interrogare con opportune query per confrontare le caratteristiche durante il calcolo on-line. Per confrontare due modelli, il sistema utilizza un parametro che misura una distanza. Questa procedura fornisce un punteggio tra le informazioni contenute nella base di dati e la caratteristica calcolate nell'immagine elaborata. In alcuni casi [115] l'indice di somiglianza è una percentuale. Alcuni parametri comunemente usati in questa analisi sono la distanza di Minkowski, la distanza di Hausdorff, la distanza di Mahalanobis e la distanza euclidea (trattate nei paragrafi 3.2.3 e 3.3.3).

3.5 Considerazioni

La valutazione dei risultati di un algoritmo di pattern-recognition può essere effettuata utilizzando punti di vista diversi, per esempio:

1. confrontando il risultato della macchina con il risultato che avrebbe dato un essere umano guardando l'immagine;
2. valutando la capacità dell'algoritmo nell'individuare particolari oggetti nell'immagine (numeri, lettere, un aereo ecc);
3. valutando l'esecuzione di un compito basato su un sistema di visione.

In generale il risultato della segmentazione dipende, oltre che dall'immagine, da alcuni valori di input che sono forniti all'algoritmo di patter-recognition. Questi valori possono essere assegnati dall'utente, ad esempio utilizzando una informazione a priori sull'immagine, oppure possono essere acquisiti dall'algoritmo stesso ,utilizzando tecniche super-visionate o tecniche non super-visionate.

Una metodologia per valutare la qualità dei risultati ottenuti con i metodi di pattern-recognition consiste nel mettere a confronto, per ogni regione significativa, i pixel $p_i \in P$ che afferiscono a tale regione e i pixel $a_i \in A$ a essa associati da un algoritmo. In figura 3.19 vengono messi a confronto, per una generica regione, quali pixel sono correttamente assegnanti da un algoritmo a tale regione, quali pixel non sono individuati come afferenti a tale regione e quali pixel sono erroneamente assegnati a tale regione. Da questo punto di vista possiamo di seguito definire la precisione, il silenzio e il rumore 3.1.

recuperati	$\frac{P \cap A}{P}$	pixel rilevanti individuati sul totale dei pixel rilevanti presenti nell'insieme
precisione	$\frac{P \cap A}{A}$	pixel rilevanti individuati sul totale dei documenti recuperati
silenzio	$\frac{P \cap A^c}{P}$	pixel rilevanti non recuperati sul totale dei pixel rilevanti presenti nell'insieme
rumore	$\frac{P^c \cap A}{A}$	pixel rilevanti non recuperati sul totale dei pixel recuperati

Tabella 3.1: definizione dei pixel recuperati, della precisione, del silenzio e del rumore

Le metodologie di pattern-recognition qui prese in esame mal si adattano al problema esposto nei seguenti capitoli fornendo risultati imprecisi. Per questo è stato scelto di studiare e sviluppare un metodo specifico per effettuare il pattern-recognition.

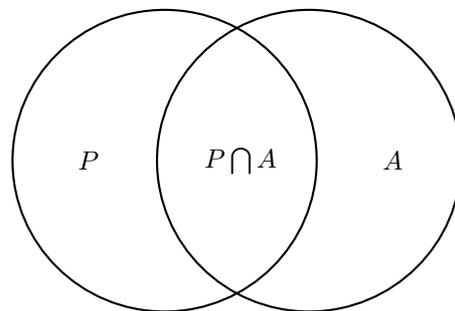


Figura 3.19: valutazione degli algoritmi di pattern-recognition, E_i indica i segmenti individuati da un esperto e A_i i segmenti individuati da un algoritmo

PRIAR: Pattern Recognition Image Augmented Resolution

In questo capitolo vengono introdotti i metodi di super-risoluzione utilizzati in questa tesi, mostrando le tecniche usate per super-risolvere singole immagini a bassa risoluzione.

4.1 Introduzione

I sistemi per la gestione di archivi di dati digitali di grandi dimensione [12, 53] permettono la raccolta di elevate quantità di immagini e sono uno strumento fondamentale per le moderne tecniche di acquisizione automatica o semi-automatica. PRIAR è appunto un algoritmo in grado di effettuare analisi automatiche, essendo specializzato nell'analisi di immagini raster che rappresentano oggetti tridimensionali. L'idea alla base della soluzione adottata consiste nel dividere il problema in più passi:

1. viene presa l'immagine a bassa risoluzione in ingresso;
2. viene super-risolta l'immagine con metodo *blind*;
3. vengono estratte le caratteristiche dell'oggetto rappresentato necessarie per poter applicare i metodi di pattern-recognition;
4. tramite i metodi di pattern-recognition, l'immagine viene riconosciuta e classificata, così da poter utilizzare le informazioni della *knowledge-base* (KB);
5. infine, utilizzando queste informazioni, viene ulteriormente migliorata la risoluzione dell'immagine.

Più in dettaglio, PRIAR riceve in input una immagine di tipo raster, un'area di interesse e un'indicazione relativa all'oggetto da identificare e modellare (un singolo pixel oppure una spezzata). L'immagine di input deve essere raster, monocromatica e a bassa risoluzione [53, 80]. Il colore indica il valore della coordinata Z dell'oggetto rappresentato; la risoluzione tipica è di 256×256 pixel a 8 bpp. Dai dati di input (l'immagine e l'area di interesse) PRIAR calcola un'immagine ad alta risoluzione.

L'aumento di risoluzione dell'immagine ha lo scopo di pre-elaborare l'immagine prima di applicare gli algoritmi di pattern-recognition [140, 178] in PRIAR. Il pattern-recognition è guidato dalla KB.

La struttura identificata per mezzo dell'applicazione delle tecniche di pattern-recognition viene successivamente sostituita con il suo modello (ad alta risoluzione). Quest'ultimo passo di super-risoluzione utilizza le informazioni della KB e il volume tracciato dall'algoritmo di pattern-recognition. Nel nostro caso la KB è stata costruita a partire dall'analisi delle immagini provenienti da AFM (Atomic Force Microscope) [24, 37, 39, 44]¹. Le immagini da AFM, come descritto in maggior dettaglio nel paragrafo 2.5 e nel capitolo 5, sono oggi estremamente importanti per gli studi nel campo delle nanotecnologie e la loro elaborazione è fondamentale per studiare le caratteristiche dei materiali esaminati

La nostra metodologia si differenzia dalle principali tecniche a oggi sviluppate per l'aumento della risoluzione [107, 68, 170] per due motivi:

- l'uso del pattern-recognition a supporto della super-risoluzione; e della super-risoluzione a supporto del pattern recognition;
- l'uso di una knowledge-base per guidare i processi di pattern-recognition e modellazione.

4.2 La soluzione proposta da PRIAR

Le classi di algoritmi di super-risoluzione single-frame, pattern-recognition e modellazione utilizzati nel PRIAR prendono in input un'immagine $I'_{M_1 \times N_1 \times B_1}$, nella quale $M_1 \times N_1$ è la dimensione in righe e colonne dell'immagine in ingresso e B_1 è il numero di bit per rappresentare il colore di ogni pixel. Gli algoritmi processano l'immagine secondo lo schema di figura 4.1.

Sostanzialmente il primo passo consiste nell'effettuare una super-risoluzione cieca (*blind*) finalizzata ad aumentare l'informazione nell'immagine senza fare uso di ipotesi alcuna.

L'immagine così super-risolta risulta essere un'interpolazione dell'immagine a bassa risoluzione. Con PRIAR vogliamo incrementare ulteriormente la risoluzione dell'immagine andando a esaminare gli oggetti che sono in essa presenti. Per questo motivo, la super-risoluzione di PRIAR si distingue da quanto mostrato nello stato dell'arte. Il problema generale è riassunto nell'equazione 4.1

$$S(I'_{M_1 \times N_1 \times B_1}) = I_{M_0 \times N_0 \times B_0} \quad (4.1)$$

dove:

- S è la funzione che definisce il processo di aumento della risoluzione dell'immagine;
- I' è l'immagine discreta in bassa risoluzione;
- I è l'immagine discreta in alta risoluzione;
- M_1 indica il numero di righe dell'immagine a bassa risoluzione, M_0 ad alta risoluzione e $M_1 \leq M_0$;

¹ Le immagini da AFM sono fornite dal dipartimento di medicina e chirurgia dell'Università di Pisa, nell'ambito di una collaborazione con il laboratorio di Segnali e Immagini dell'ISTI-CNR di Pisa.

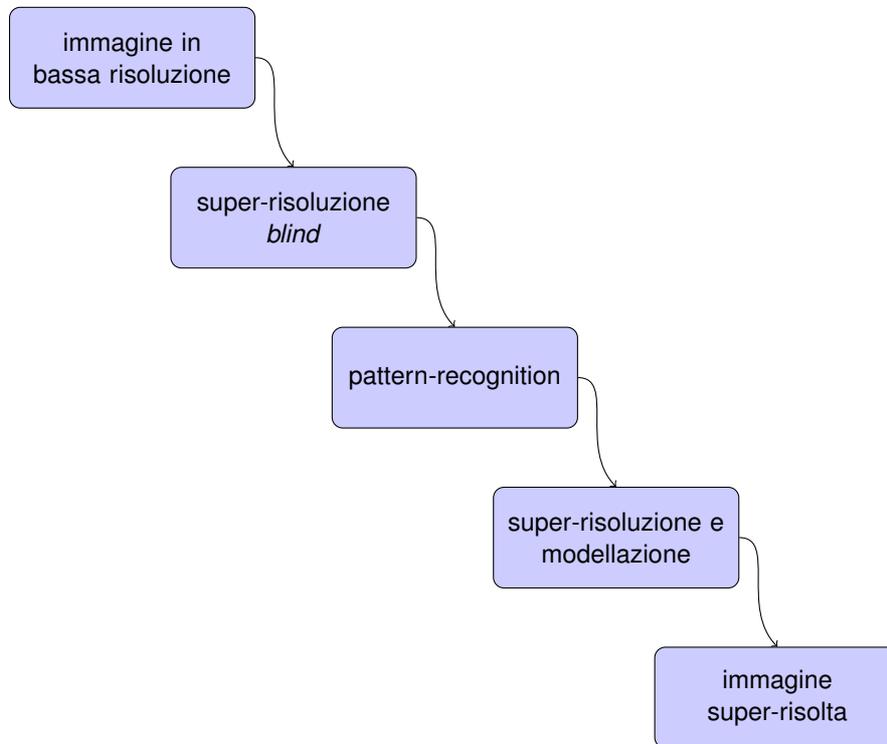


Figura 4.1: Schema a blocchi dell'algoritmo PRIAR

- N_1 indica il numero di colonne dell'immagine a bassa risoluzione, N_0 ad alta risoluzione e $N_1 \leq N_0$;
- B_1 indica il numero di bit utilizzati per rappresentare il colore di ogni pixel nell'immagine a bassa risoluzione, B_0 ad alta risoluzione e $B_1 \leq B_0$.

La funzione S non è una funzione bigettiva in quanto il contenuto informativo dell'immagine I' può essere rappresentato riducendo più immagini I a I' . In generale il problema può essere rappresentato come [181]:

$$i = (h \otimes f(\xi))(x, y) + n(x, y) \quad (4.2)$$

dove:

- $f(\xi)$ è l'immagine reale;
- h è la PSF (Point Spread Function) [135, 82, 172];
- n è il rumore (che assumiamo con distribuzione di probabilità gaussiana);
- i è il segnale associato all'immagine reale;
- \otimes indica il prodotto di convoluzione.

La discretizzazione del segnale i , in condizioni ideali, ci porta a scrivere l'espressione

4.3

$$I = \psi_0 (h \otimes f (\xi)) (x, y) + N_0 (x, y) \quad (4.3)$$

dove:

- I è l'immagine discreta ad alta risoluzione;
- ψ_0 è la funzione di campionamento;
- N_0 è il rumore nell'immagine discreta (che assumiamo con distribuzione di probabilità gaussiana).

Poiché le nostre immagini non sono acquisite in condizioni ideali l'espressione è 4.4.

$$I' = \psi_1 (h_1 \otimes f (\xi_1)) (x, y) + N_1 (x, y) \quad (4.4)$$

dove:

- I' è l'immagine discreta a bassa risoluzione;
- ψ_1 è la funzione di campionamento;
- N_1 è il rumore (che assumiamo con distribuzione di probabilità gaussiana)

Il problema qui trattato consiste nel calcolare la I [147, 185] considerando che ψ_0 è una funzione di campionamento che ha caratteristiche migliori della ψ_1 , ovvero, restituisce un'immagine con una risoluzione spaziale uguale o migliore alla ψ_1 , una profondità di colore uguale o migliore rispetto alla ψ_1 e N_0 una quantità di rumore inferiore a N_1 .

Quindi avendo come input un'immagine I' a bassa risoluzione possiamo modellare il nostro problema come il calcolo della seguente funzione [150]:

$$I' = G_a \cdot I \cdot G_b + \Xi \quad (4.5)$$

dove Ξ è l'apporto del rumore, G_a e G_b sono la distorsione dovuta al PSF e al sotto-campionamento.

In particolare, detta $M \times N$ la dimensione della matrice di input I' , abbiamo che: G_a ha dimensione $M \times j \cdot M$, I ha dimensione $j \cdot M \times k \cdot N$, G_b ha dimensione $k \cdot N \times N$ e Ξ dimensione $M \times N$ con $j, k \in \mathbb{Z}^+$.

Osserviamo che l'equazione 4.5 modella un problema mal posto quindi la funzione non è invertibile e la soluzione I non è unica.

PRIAR, dopo aver calcolato l'immagine super-risolta utilizzando un algoritmo noto [111, 173, 99, 77, 8] (la costruzione modulare dell'algoritmo permette di utilizzare un algoritmo piuttosto che un altro), applica i metodi PRIAR di pattern-recognition per riconoscere gli oggetti presenti nelle immagini. Il riconoscimento avviene per un oggetto O alla volta. La fase del riconoscimento avviene approssimando il problema in 3 fasi partendo dal basso: durante ogni fase vengono riconosciuti i sottoinsiemi di pixel O_1 , O_2 e O_3 dell'oggetto O ($O_1 \subseteq O_2 \subseteq O_3 \subseteq O$). I metodi di pattern-recognition lavorano utilizzando una KB che contiene le informazioni relative agli oggetti che vogliamo super-risolvere. Gli oggetti identificati sono nuovamente super-risolti sfruttando la KB e le informazioni acquisite per mezzo del pattern-recognition.

4.3 Pattern-recognition

Il riconoscimento dell'oggetto avviene per passi successivi che estendono sempre di più l'approssimazione dell'oggetto. PRIAR permette il riconoscimento dell'oggetto effettuando una tra le seguenti analisi:

1. analisi single-point;
2. analisi multi-point.

4.3.1 Analisi single-point

Quando viene selezionato un solo pixel, che giace all'interno del volume della struttura da identificare, PRIAR esegue l'analisi single-point, ovvero, inizialmente traccia un sottinsieme della struttura O_1 . Il pixel viene utilizzato come punto iniziale. Il riconoscimento di O_1 avviene tramite un crawler che inizia la sua navigazione della superficie del volume dalle coordinate indicate dall'utente. L'idea di navigazione adottata dal crawler, dato che sta navigando sulla superficie di un oggetto, è quella di continuare la navigazione fino a quando tra due coordinate adiacenti il divario o *gap* tra i valori dei pixel è entro una certa soglia e il loro valore non supera una certa distanza dal punto iniziale (eq. 4.6).

$$\begin{aligned} p_{i+1} \in O_1 &\iff \\ p_i \in O_1, p_{i+1} \in ad(p_i), \|p_i - p_{i+1}\| < \Delta_R, \|p_0 - p_{i+1}\| < \Delta_A & \end{aligned} \quad (4.6)$$

dove:

- la funzione $ad(p_i)$ restituisce l'insieme dei pixel adiacenti a p_i
- Δ_R è il massimo valore delta tra due pixel adiacenti;
- Δ_A è il massimo valore delta tra il primo pixel di O_1 e il pixel in esame.

I valori Δ_A e Δ_R sono determinati in funzione dell'immagine di ingresso. In appendice A viene spiegato in dettaglio il codice per il calcolo di O_1 . Dopo il riconoscimento di $O_1 \subseteq O$, PRIAR inizia il riconoscimento di una nuova porzione di oggetto O_2 . Per il riconoscimento di O_2 sono utilizzati dei descrittori d_i specifici per la classe del problema. I descrittori sono memorizzati nella knowledge-base presente in PRIAR. I descrittori relativi ai problemi relativi al grating e ai nanotubuli [42, 34] sono stati ricavati dall'analisi dei documenti [39, 37, 44]. In particolare, nel caso dei nanotubuli, per il riconoscimento di O_2 viene costruito l'ellisse minimo che contiene O_1 . In figura 4.2 vediamo una rappresentazione esemplificativa dell'ellisse E utilizzata. L'ellisse calcolata è la più piccola ellisse che racchiude la regione interessata. Una volta calcolato E PRIAR calcola l'ellissoide \hat{E} . \hat{E} viene dimensionato per essere una densità di probabilità.

Poiché siamo interessati a una figura avente dei descrittori come quelli trattati in [39, 37, 44] il calcolo dell'ellissoide avviene nel modo seguente:

- si calcolano degli opportuni fattori k_{min_axis} e k_{max_axis} relativi agli assi maggiore e minore dell'ellisse E in considerazione delle proprietà descritte [39, 37, 44] (viene così calcolato un'ellisse E');

- La costruzione di E' avviene in uno spazio S' ortonormale a tre dimensioni, l'immagine analizzata giace su un piano di questo spazio.
- $\iint \hat{E} dx dy = K$ con $K \leq 1$

Poiché la struttura che si vuole riconoscere può estendersi anche oltre l'ellissoide la densità di probabilità deve essere calcolata tenendo conto anche di questa ipotesi. Calcoliamo la funzione di probabilità $P(W)$. Consideriamo una funzione $g(x, y)$ e sia $(x, y) \in I$.

$$g(x, y) = \begin{cases} \alpha \cdot \hat{e}(x, y) & (x, y) \in E' \quad \alpha \in \mathbb{R}^+ \\ \epsilon & (x, y) \notin E' \end{cases} \quad (4.7)$$

Chiamiamo il complemento di E' con $\overline{E'}$. Quindi la distribuzione di probabilità è data da:

$$\iint g(x, y) dx dy = \alpha \cdot \iint_{E'} \hat{e}(x, y) dx dy + \epsilon \cdot \iint_{\overline{E'}} dx dy =$$

$$P(W) = \alpha \cdot P(E') + \epsilon \cdot P(\overline{E'}) = 1 \quad (4.8)$$

Calcolata la distribuzione di probabilità (eq. 4.8) andiamo a estendere ulteriormente O_1 sfruttando ancora una volta il crawler. Il crawler in questo caso naviga sulla superficie seguendo una variante delle regole precedentemente descritte:

$$p_{i+1} \in O_2 \iff p_i \in O_2, p_{i+1} \in ad(p_i),$$

$$\|p_i - p_{i+1}\| < g_R(x, y) \cdot \Delta'_R, \|p_0 - p_{i+1}\| < g_A(x, y) \cdot \Delta'_A \quad (4.9)$$

Dove $g_R(x, y)$ e $g_A(x, y)$ sono funzione dell'andamento delle differenze relativa e assoluta, nel caso più semplice abbiamo $g(x, y) = g_R(x, y) = g_A(x, y) \quad \forall x, y \in I$.

In figura 4.3 una esemplificazione della mappa di probabilità relativa all'implementazione in esame.

4.3.2 Analisi multi-point

Durante l'analisi multi-point l'utente traccia alcuni (due o più) punti connessi. I punti scelti, a coppie consecutive, sono uniti da una retta e per ipotesi sia i punti scelti sia i punti della



Figura 4.2: costruzione dell'ellissoide minimo

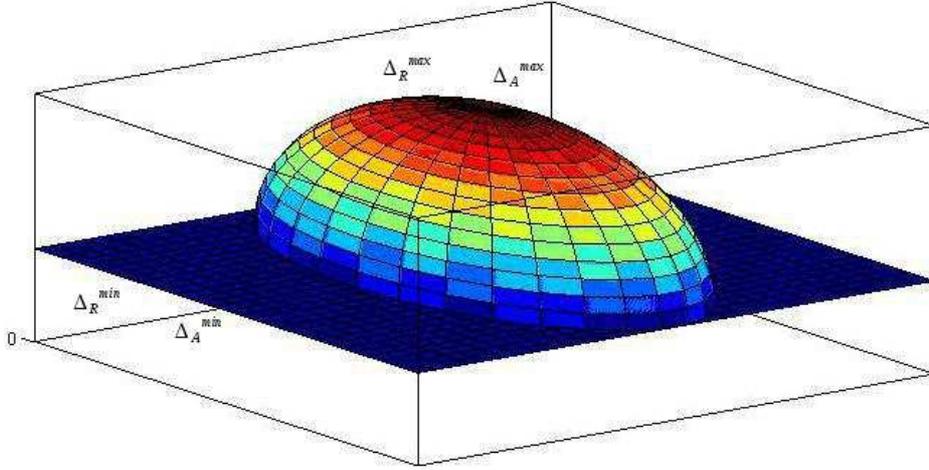


Figura 4.3: mappa di probabilità

retta giacciono all'interno di O . Tutti i punti identificati e i segmenti che congiungono le coppie di punti successivi sono considerati come parte dell'insieme O'_1 .

O'_1 viene esteso in O'_2 ($O'_1 \subseteq O'_2$) utilizzando i parametri Δ_A e Δ_R che, come per l'analisi single-point, sono funzione dell'immagine analizzata.

$$p_{i+1} \in O'_2 \iff p_i \in O'_2, p_{i+1} \in ad(p_i), \|p_0 - p_{i+1}\| < \Delta_R, \|p_i - p_{i+1}\| < \Delta_A \quad (4.10)$$

4.4 Estensione dell'area riconosciuta

L'algoritmo prevede una terza approssimazione che estende il volume delle aree identificate. In funzione della KB è stato individuato un set di parametri f_i tali che O_2 (o O'_2 nel caso dell'analisi multi-traccia) sia esteso calcolando O_3 . Questo calcolo estende effettivamente O_2 quando nell'immagine vi è un elevato rumore oppure un singolo oggetto passa sotto altri oggetti presenti nell'immagine (ad esempio il nanotubulo che stiamo tracciando passa sotto un altro nanotubulo).

O_3 viene calcolato:

1. segmentando l'immagine, poiché in questo tipo di analisi siamo interessati ad avere piccole aree omogenee la segmentazione è calcolata come prodotto dei segmenti generati con l'algoritmo di Sobel [81, 80], il metodo di Prewitt [81, 80], il metodo di Roberts [81, 80], la laplaciana del metodo di Gauss [81, 80], il metodo Canny [81, 80] e il metodo zero-cross [81, 80].
2. per ogni area segmentata A_s e su O_2 vengono calcolate particolari feature f_i . La tipologia delle feature calcolate dipende dal tipo di problema trattato. A tal fine è stato

creato un modulo di supporto alle decisioni. Le basi decisionali del decision-support system dipendono fortemente dal tipo di problema trattato. Nel caso di un nanotubulo sono l'altezza media (f_1) e l'angolo (f_2), ovvero angolo dell'asse maggiore dell'ellissoide calcolato su A_s . In generale, l'insieme di queste feature per un volume viene calcolato dalla funzione F così definita:

$$F : A \rightarrow \{f_i\} \quad (4.11)$$

3. ogni segmento A_s viene confrontato con O_2 . Dato che $O_2 \subseteq O_3$ abbiamo che A_s estende O_2 nel seguente caso:

$$A_s \subseteq O_3 \iff \|F(O_2) - F(A_s)\| < g(x, y) \cdot K \quad (4.12)$$

Ogni volta che l'eq. 4.12 è vera, $O_2' = O_2 \cup A_s$ (e sono aggiornate le features dell'area identificata con $F(O_2')$).

Al termine di questa funzione viene identificato il volume O_3 che è il volume dell'oggetto del quale vogliamo migliorare la risoluzione.

4.5 Calcolo del modello

Il volume individuato viene sostituito con un modello descrittivo che ne incrementa la risoluzione spaziale e che permette di aggiungere altre informazioni quali il volume, le relazioni funzionali nei confronti degli oggetti che ha vicino, ecc. Il modello utilizzato dipende dalla descrizione presente nella KB. PRIAR attualmente ha informazioni solo per classificare e riconoscere immagini contenenti grating e tubuli.

In particolare, all'oggetto riconosciuto O_3 (che approssima l'oggetto reale $o \subseteq i$) viene sostituito il suo modello spaziale in alta risoluzione $M \subseteq I$ che è calcolato per mezzo delle informazioni ottenute durante il processo di riconoscimento del pattern-recognition, dal classificatore e per mezzo delle informazioni memorizzate nella KB. In particolare, l'oggetto O_3 e in generale l'immagine I , è associata a una classe \mathcal{C}_j che fa riferimento alla KB (contenente la descrizione geometrica parametrica della classe di oggetti simili a o).

Il calcolo di M avviene minimizzando le differenze delle i features (eq. 4.13):

$$\min \sum_i \left\| \mathcal{C}_j \left(M^{(i)} \right) - \mathcal{C}_j \left(o^{(i)} \right) \right\| \quad (4.13)$$

L'equazione 4.13, se il modello è ottimale, fornisce 0 come risultato (in quanto il modello ricalca perfettamente il modello reale).

Dato che noi disponiamo solo di un calcolo che rappresenta o lavoriamo con la sua rappresentazione nel nostro spazio

$$\min \sum_i \left\| \mathcal{C}_j \left(M^{(i)} \right) - \mathcal{C}_j \left(O_3^{(i)} \right) \right\| \quad (4.14)$$

4.6 Tool PRIAR Graphic User Interface

L'interfaccia grafica del Tool PRIAR è illustrata in figura 4.4.

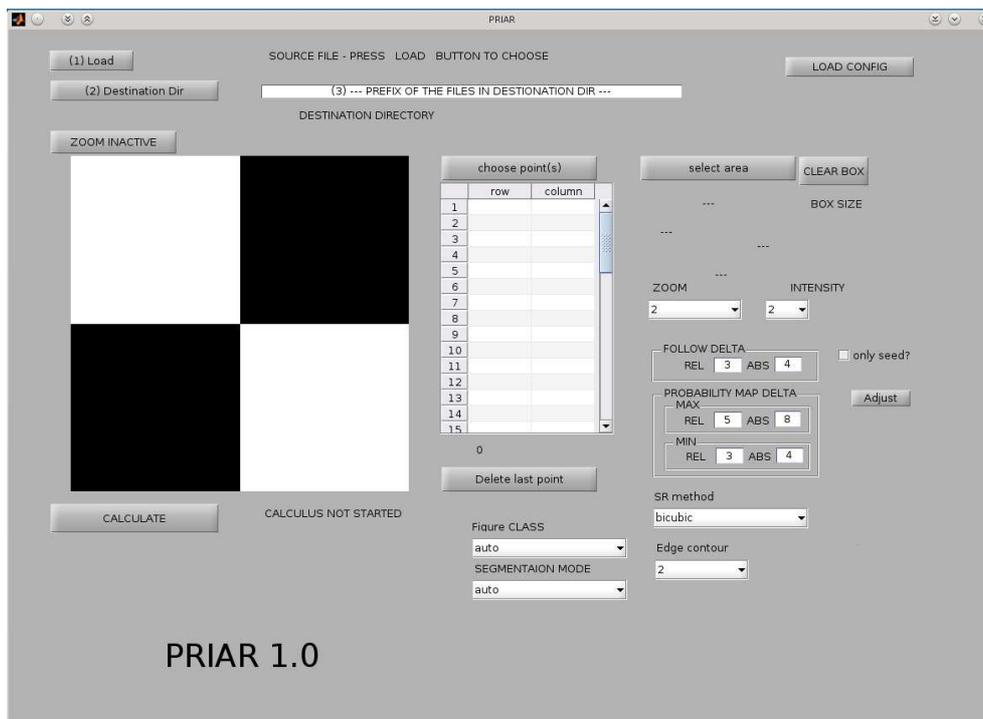


Figura 4.4: PRIAR GUI

Per mezzo dell'interfaccia viene caricata l'immagine da elaborare, viene scelto il punto di interesse che definisce l'oggetto da tracciare e vengono stabiliti i gradienti per tracciare l'oggetto.

In particolare, in figura, sono messe in evidenza le parti dell'interfaccia grafica:

1. pulsante per il caricamento dell'immagine da analizzare;
2. prefisso della directory in cui scrivere i risultati e i file di configurazione utilizzati per il calcolo;
3. suffisso della directory 2 e prefisso dei file risultato e configurazione contenuti nella directory indicata;
4. pulsante per caricare la configurazione precedentemente salvata
5. pulsante per attivare lo zoom sulla figura caricata
6. area in cui viene visualizzata la figura caricata
7. pulsante per attivare la scelta del punto afferente all'oggetto di interesse o per tracciare una prima area inerente l'oggetto di interesse presente nella figura. Selezio-

CAPITOLO 4. PRIAR: PATTERN RECOGNITION IMAGE AUGMENTED RESOLUTION

nato questo pulsante le coordinate si scelgono posizionando il puntatore sul pixel di interesse e premendo il tasto sinistro del mouse.

8. elenco delle coordinate selezionate
9. pulsante per cancellare l'ultimo punto inserito nell'elenco delle coordinate selezionate
10. pulsante per identificare l'area nella quale deve essere effettuata l'elaborazione. Una volta premuto questo pulsante (a) si posiziona il mouse nell'angolo in alto a sinistra dell'area di interesse e si preme il pulsante sinistro del mouse, (b) si posiziona il mouse nell'angolo in basso a destra dell'area di interesse e si preme il pulsante destro del mouse
11. premendo questo pulsante viene annullata la selezione dell'area di interesse
12. in quest'area sono visualizzate le coordinate precedente selezionate durante la scelta dell'area di interesse
13. tramite questa lista viene scelto il fattore di zoom
14. tramite questa lista viene indicato a PRIAR quanto marcare i segni impressi dall'utente sulla figura
15. in questo pannello possono essere forzati i valori Δ_A e Δ_R
16. nel caso dell'analisi single-point, se questa check-box è marcata l'insieme O_2 è calcolato come un'estensione di O_1 , se la check-box non è marcata i dati calcolati per O_1 sono utilizzati nella mappa di probabilità come descritto nel paragrafo 4.3.1 ma non fanno automaticamente parte dell'insieme O_2 .
17. questi campi sono utilizzati per imporre i valori Δ'_A e Δ'_R nel calcolo della mappa di probabilità. In particolare con questi valori è possibile imporre i valori come indicato in figura 4.3.
18. il pulsante calcola i valori del pannello 17 in funzione dei valori del pannello 15.
19. questa check-box permette di forzare il metodo di super-risoluzione blind da utilizzare durante la pre-processing al pattern-recognition.
20. questa lista forza la classe \mathcal{C}_j dell'immagine da tracciare
21. questo parametro forza il modo di segmentare l'immagine
22. questo parametro indica quanto devono essere spessi i bordi dei segmenti calcolati
23. questo pulsante richiama le routine di calcolo sulle immagini
24. in questa area sono riportati i messaggi che indicano l'inizio e la fine del calcolo

Al termine del calcolo PRIAR fornisce nella directory di destinazione vari file:

- le immagini in alta risoluzione dell'area analizzata con il modello elaborato
- le immagini nelle quali viene messo in evidenza l'area tracciata
- una matrice con i dati elaborati
- i file di configurazione con i dati di input usati per l'esecuzione del calcolo

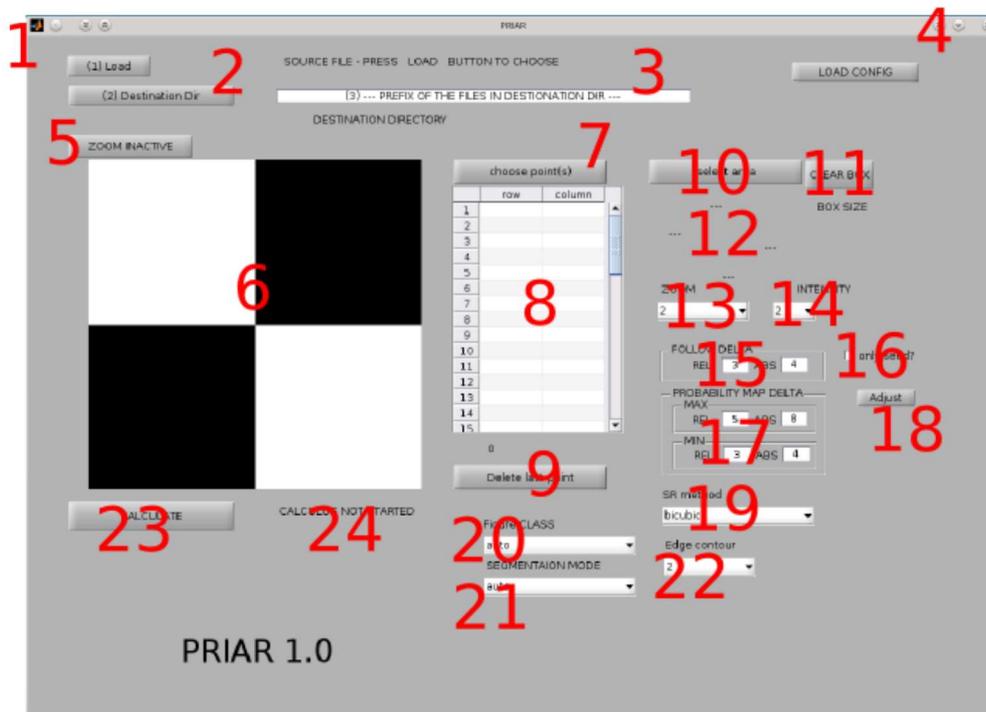


Figura 4.5: PRIAR GUI

4.7 Considerazioni

PRIAR nasce per affrontare il problema di migliorare l'informazione in un'immagine: utilizza algoritmi di super-risoluzione noti che migliorano le performance dell'algoritmo di pattern-recognition.

Le attività di sviluppo del PRIAR prevedono:

- la creazione di un linguaggio generale per la descrizione delle informazioni nella knowledge-base;
- un'estensione del classificatore di immagini e oggetti presenti nelle immagini così da poter riconoscere altri tipi di oggetti;
- estendere il sistema di tag degli oggetti riconosciuti in concomitanza di un maggior numero di oggetti riconosciuti, così da poter creare per ogni immagine una struttura completa delle interconnessioni tra gli oggetti riconosciuti descrivendone al contempo le singole funzionalità.

Durante lo sviluppo di PRIAR e del Tool PRIAR è stata prestata particolare attenzione alla complessità computazionale che è stata mantenuta lineare i risultati in tempi rapidi. Nell'appendice A è riportata una parte del codice sviluppato.

Studio di alcuni casi: applicazioni a immagini AFM e subacquee

5.1 Introduzione

In questo capitolo vengono presentati alcuni risultati sperimentali ottenuti con PRIAR in base ai risultati teorici raggiunti presentando l'analisi di diversi casi testati. La sperimentazione è stata eseguita su immagini ottenute da microscopi a forza atomica (Atomic Force Microscopy, AFM) e immagini da riprese subacquee. Entrambi i tipi di immagine hanno bisogno di essere elaborate per un miglioramento della loro definizione.

L'AFM è un microscopio a scansione di sonda (Scanning Force Microscopy, SFM) che ha come inconveniente che la qualità delle immagini è limitata dal raggio di curvatura della sonda (*bow problem*), che può portare ad un'immagine molto distorta. Inizialmente abbiamo esaminato immagini di superfici di un polimero, che essendo note, ci hanno consentito di mettere a punto il nostro metodo.

Successivamente abbiamo analizzato immagini AFM relative alle cellule staminali mesenchimali (Mesenchymal Stem Cells, MSCs) umane, in particolare quelle relative al citoscheletro. Recentemente, lo studio di queste cellule ha attirato molta attenzione nell'ingegneria dei tessuti e nella medicina rigenerativa. Identificare le componenti del citoscheletro è utile infatti per correlare aspetti morfologici, funzionali e meccanici di MSC, in modo da ottenere una comprensione maggiore delle funzioni cellulari. In particolare, nei nostri test, abbiamo analizzato immagini del citoscheletro per individuare i microtuboli che sono strutture intracellulari. Questa identificazione non è per niente banale a causa delle variazioni fisiologiche nella superficie della cellula e della modalità dell'acquisizione AFM.

Abbiamo successivamente analizzato immagini subacquee che sono state collezionate durante le attività descritte in [5, 58, 152, 129, 6]. Le immagini sono state riprese in tempo reale da veicoli sottomarini autonomi (Autonomous Underwater Vehicle, AUV), ovvero sottomarini robotici in grado di muoversi e compiere azioni autonomamente nell'ambiente marino, utilizzando telecamere subacquee. Con immagini di questo tipo, per loro natura poco nitide, si hanno maggiori difficoltà a individuare gli oggetti. In particolare abbiamo esaminato il caso di oggetti che si trovano sul fondale marino. A scopo

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

archeologico è infatti importante saper identificare reperti, anche se ricoperti da sabbia e incrostazioni, senza muoverli e senza modificare nemmeno l'ambiente circostante.

Più in dettaglio, i casi di studio che presentiamo di seguito sono così suddivisi:

- i primi sono relativi a immagini AFM di due diverse superfici di un polimero che ci hanno permesso di lavorare con pattern a bassa complessità;
- i successivi casi sono relativi all'analisi delle immagini AFM di uno stesso citoscheletro dove tracciamo un unico nanotubo con il metodo single-point e successivamente con il metodo multi-point, il citoscheletro ha un pattern a elevata complessità in quanto i diversi nanotubuli sono intrecciati;
- infine sono presentati dei casi d'uso con immagini di oggetti di interesse archeologico, reperiti sui fondali marini.

5.2 Immagine AFM "A"

L'immagine in esame è ottenuta acquisendo al superficie di un polimero. In figura 5.1 sono visualizzati i parametri impostati per l'analisi dell'immagine. I valori numerici usati per il calcolo sono:

- area analizzata (6, 4) (441, 439). Queste coordinate sono gli estremi della figura analizzata. La parte analizzata è racchiusa in una regione quadrata le cui coordinate sono identificate dalla riga 6 - colonna 4 e riga 441 - colonna 439.
- il punto appartenente alla struttura dal quale inizia l'analisi ha come coordinate (209, 178)
- il fattore di zoom è 2
- il metodo di super risoluzione blind è lo spline
- il valore di Δ_A è 40 e il valore di Δ_R è 25
- il valore di Δ_A^{max} è 48, il valore di Δ_R^{max} è 30, Δ_A^{min} è 40 e il valore di Δ_R^{min} è 25.
- per individuare la classe della figura viene utilizzato il classificatore (modalità auto) e la modalità di segmentazione è dipendente dal modello e dal classificatore (modalità auto).

5.2. IMMAGINE AFM "A"

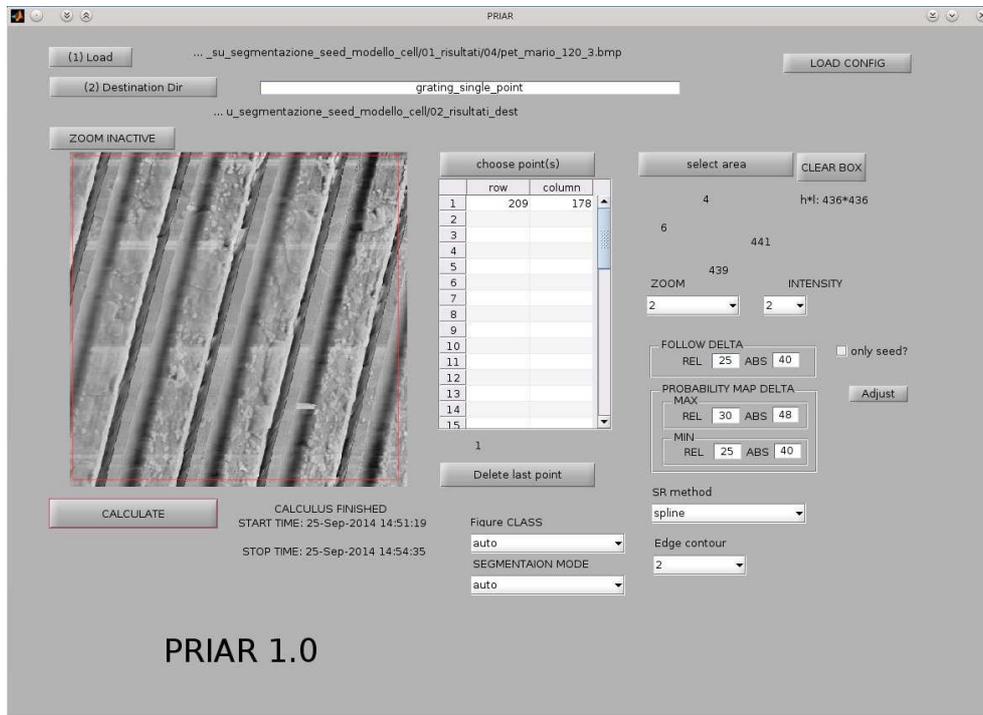


Figura 5.1: immagine AFM "A", in evidenza i parametri utilizzati per l'analisi della superficie a gradino

In figura 5.2 vediamo la prima risposta che PRIAR fornisce all'utente, ovvero, la classe di appartenenza dell'immagine analizzata. La versione 1.0 del Tool PRIAR è in grado di riconoscere due tipi di immagini: i *grating* e le cellule.

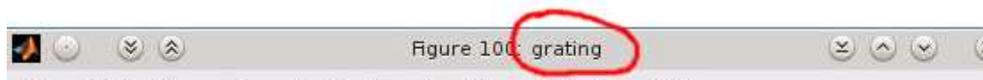


Figura 5.2: immagine AFM "A", cerchiata di rosso la risposta *grating* del classificatore

In figura 5.3 vediamo il primo risultato che presenta PRIAR a video. Questo è il risultato del primo algoritmo che, dopo aver super-risolto in modo blind l'immagine, traccia un sottoinsieme della superficie (O_1).

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

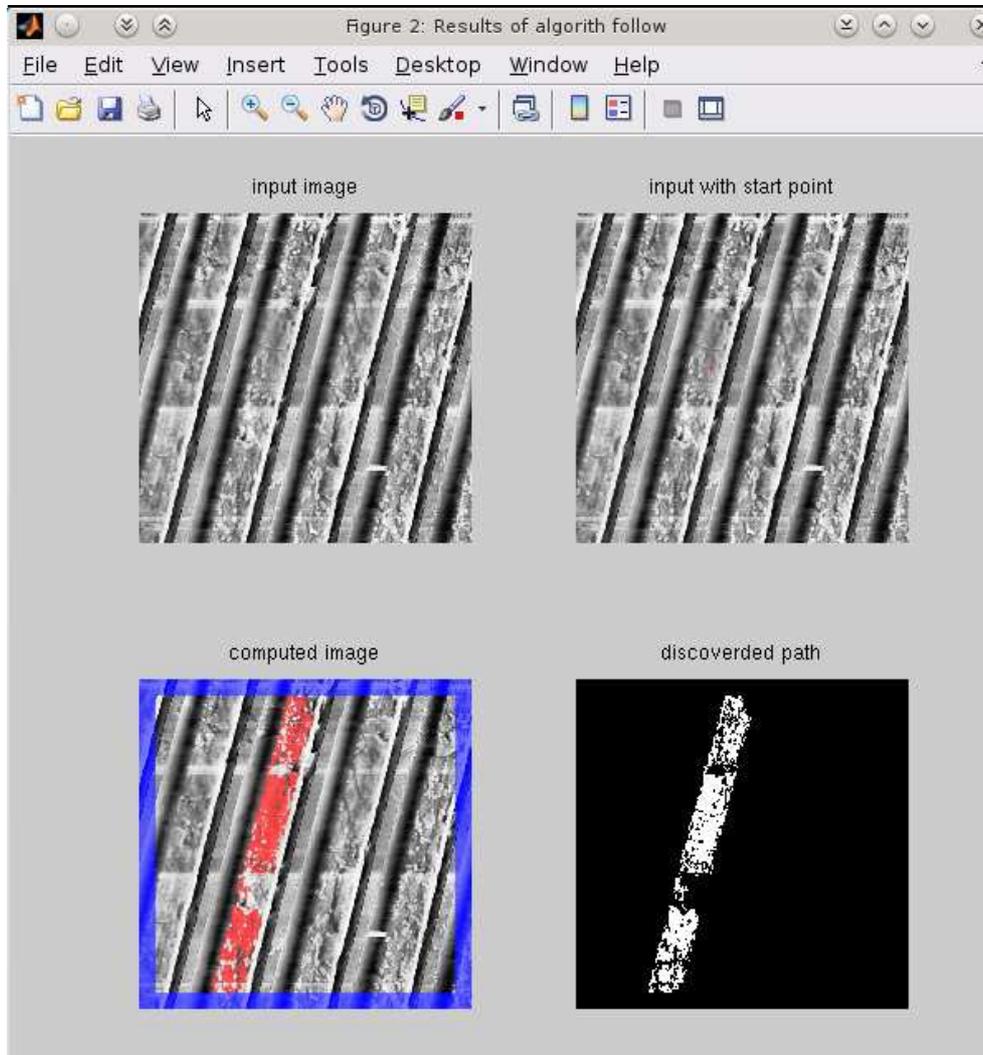


Figura 5.3: immagine AFM "A", durante il calcolo di O_1 sono visualizzati l'immagine di input, l'immagine di input con evidenziato il punto da cui è iniziato il calcolo, il pattern O_1 calcolato sull'immagine e e pixel che costituiscono il pattern O_1

Il calcolo di O_1 avviene applicando una super-risoluzione blind all'immagine di input. O_1 viene esteso approssimando dal basso la figura che O che vogliamo riconoscere. L'approssimazione calcolata, detta O_2 , fa uso della mappa di probabilità come descritto nel capitolo 4. In figura 5.4 vediamo come O_1 viene esteso in O_2 .

5.2. IMMAGINE AFM "A"

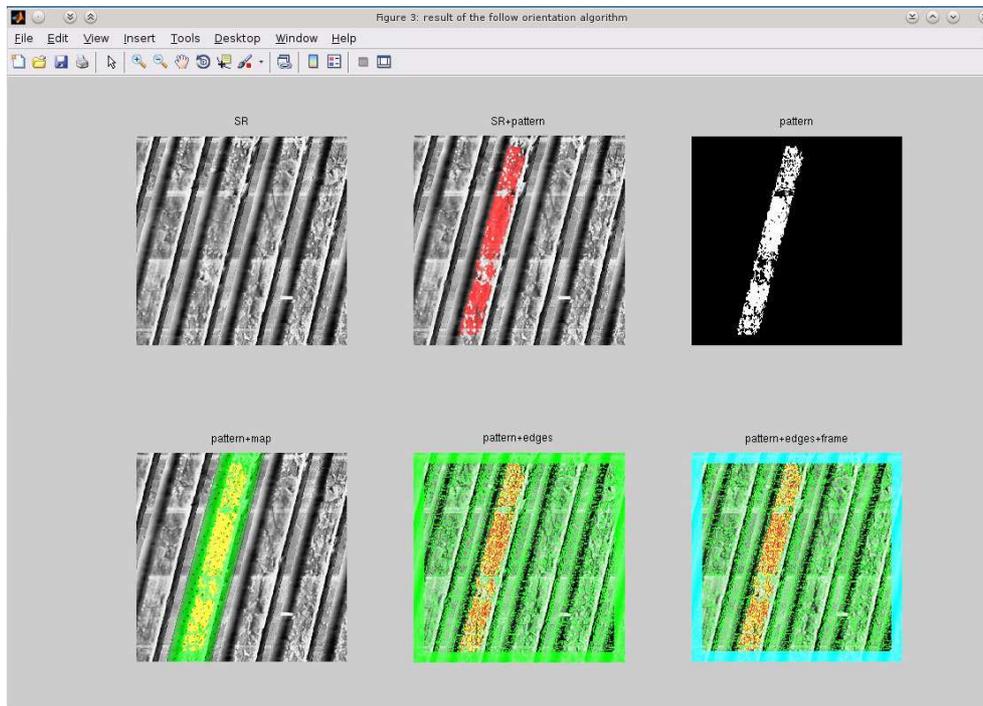


Figura 5.4: immagine AFM "A", durante il calcolo di O_2 sono visualizzati l'immagine in super-risoluzione blind, l'immagine in super-risoluzione blind con il pattern O_2 sovrapposto, il pattern O_2 calcolato, l'immagine in super-risoluzione blind con sovrapposti il pattern O_2 calcolato e la mappa di probabilità, l'immagine in super-risoluzione blind con sovrapposti il pattern O_2 calcolato e i segmenti calcolati, l'immagine in super-risoluzione blind con sovrapposti il pattern O_2 calcolato, i segmenti calcolati e la cornice nella quale non viene effettuato computo

La figura O_2 , facendo nuovamente uso del modello presente nella KB, viene esteso e la nuova multi-area riconosciuta prende il nome di O_3 . Durante questo calcolo l'intera immagine viene segmenta utilizzando i metodi visti in 4 e ogni segmento viene confrontato con O_2 . La KB fornisce la conoscenza per capire quando un segmento può essere una estensione di O_2 , quando questo avviene, O_2 viene esteso. In figura 5.5 vediamo come O_2 viene esteso in O_3 .

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

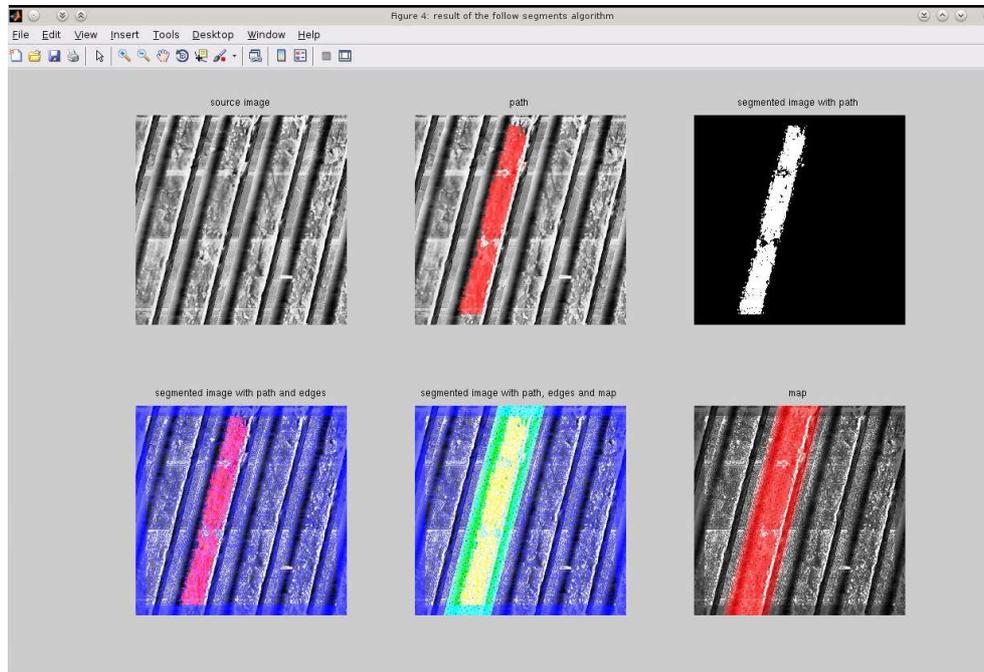


Figura 5.5: immagine AFM "A", durante il calcolo di O_3 sono visualizzati l'immagine in super-risoluzione blind, l'immagine super-risoluzione blind con il pattern O_3 sovrapposto, il pattern O_3 calcolato, l'immagine in super-risoluzione blind con sovrapposti il pattern O_3 calcolato e i segmenti calcolati, l'immagine super-risoluzione blind con sovrapposti il pattern O_3 calcolato, la mappa di probabilità, e i segmenti, l'immagine super-risoluzione blind con sovrapposto il pattern O_3 calcolato

Al termine del calcolo di O_3 l'area riconosciuta, che corrisponde a O_3 viene ulteriormente super-risolta facendo uso delle informazioni contenute nella KB. Nel caso specifico del grating l'area riconosciuta viene sostituita con una superficie piana avente come altezza l'altezza media della superficie. In figura 5.6 vediamo il solo modello calcolato e il modello immerso nell'immagine originale.

5.3. IMMAGINE AFM "B"

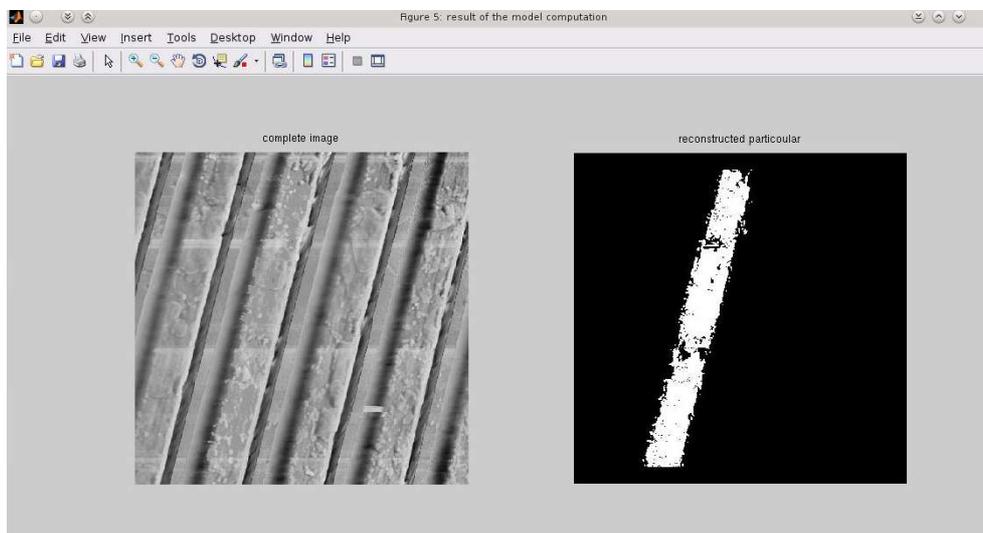


Figura 5.6: immagine AFM "A", a sinistra il modello calcolato e destra il path calcolato

5.3 Immagine AFM "B"

Anche in questo studio viene illustrato come viene elaborata l'immagine di una superficie di un polimero. La figura 5.7 visualizza i parametri impostati per l'analisi dell'immagine. I valori numerici usati per il calcolo sono:

- area analizzata (338, 332) (481, 475)
- il punto appartenente alla struttura dal quale inizia l'analisi ha coordinate (446, 395)
- il fattore di zoom è 2
- il metodo di super risoluzione blind è lo spline
- il valore di Δ_A è 10 e il valore di Δ_R è 15
- il valore di Δ_A^{max} è 18, il valore di Δ_R^{max} è 12, Δ_A^{min} è 15 e il valore di Δ_R^{min} è 10.
- per individuare la classe della figura viene utilizzato il classificatore (modalità auto) e la modalità di segmentazione è dipendente dal modello e dal classificatore (modalità auto).

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

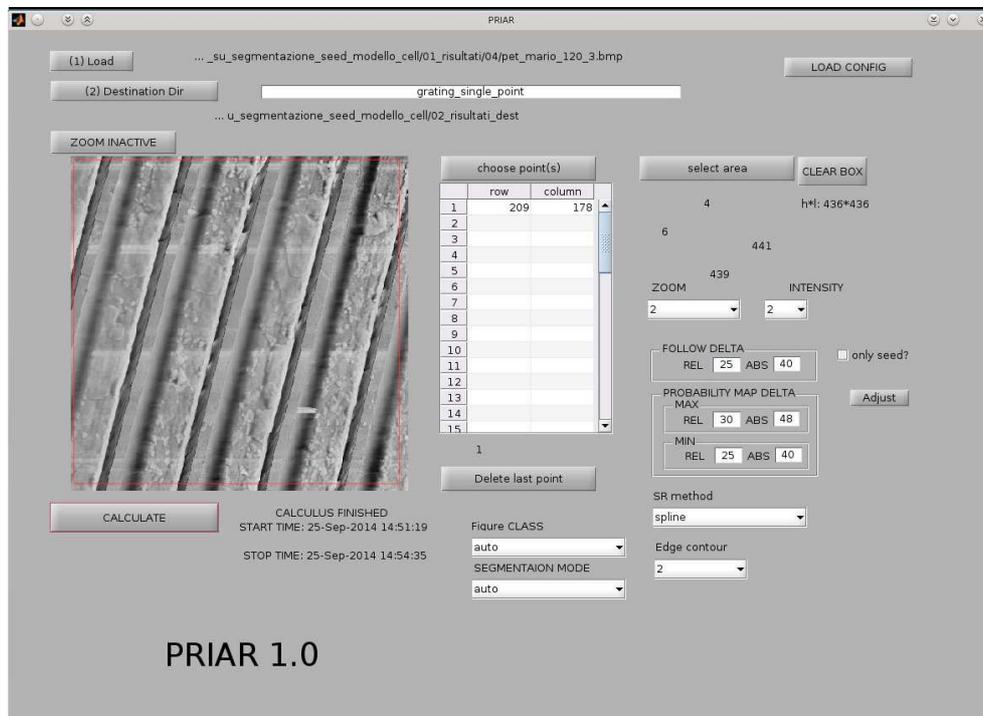


Figura 5.7: immagine AFM "B", in evidenza i parametri utilizzati per l'analisi della superficie a gradino

In figura 5.8 vediamo la prima risposta che PRIAR fornisce all'utente, ovvero, la classe di appartenenza dell'immagine analizzata.

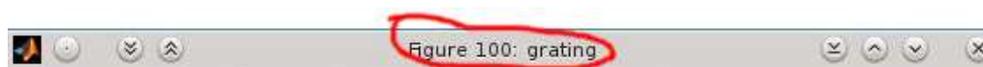


Figura 5.8: immagine AFM "B": risposta "grating" del classificatore

In figura 5.9 vediamo il primo risultato che presenta PRIAR a video. Questo è il risultato del primo algoritmo che, dopo aver super-risolto in modo blind l'immagine, traccia un sottoinsieme della superficie (O_1).

5.3. IMMAGINE AFM "B"

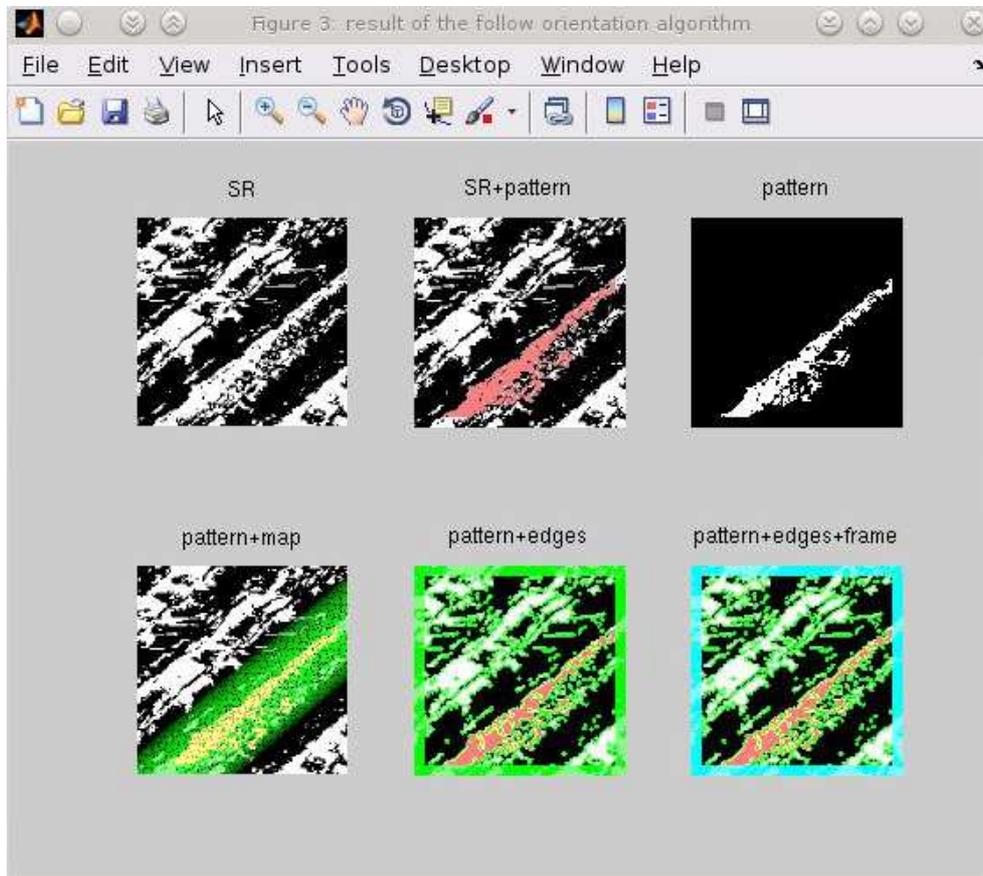


Figura 5.9: immagine AFM "B": durante il calcolo di O_1 sono visualizzati l'immagine di input, l'immagine di input con evidenziato il punto da cui è iniziato il calcolo, il pattern O_1 calcolato sull'immagine e i pixel che costituiscono il pattern O_1

In figura 5.10 vediamo come O_1 viene esteso in O_2 .

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

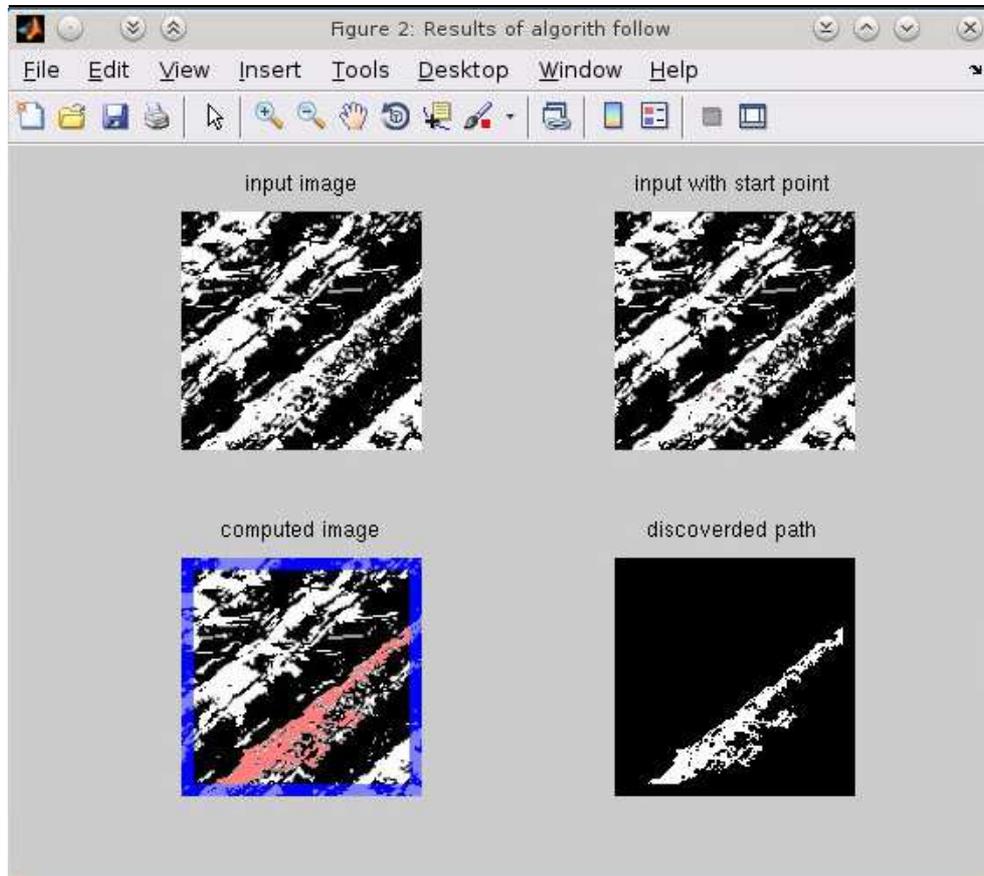


Figura 5.10: immagine AFM "B", durante il calcolo di O_2 sono visualizzati l'immagine in super-risoluzione blind, l'immagine in super-risoluzione blind con il pattern O_2 sovrapposto, il pattern O_2 calcolato, l'immagine in super-risoluzione blind con sovrapposti il pattern O_2 calcolato e la mappa di probabilità, l'immagine in super-risoluzione blind con sovrapposti il pattern O_2 calcolato e i segmenti calcolati, l'immagine in super-risoluzione blind con sovrapposti il pattern O_2 calcolato, i segmenti calcolati e la cornice nella quale non viene effettuato computo

In figura 5.11 vediamo come O_2 viene esteso in O_3 .

5.3. IMMAGINE AFM "B"

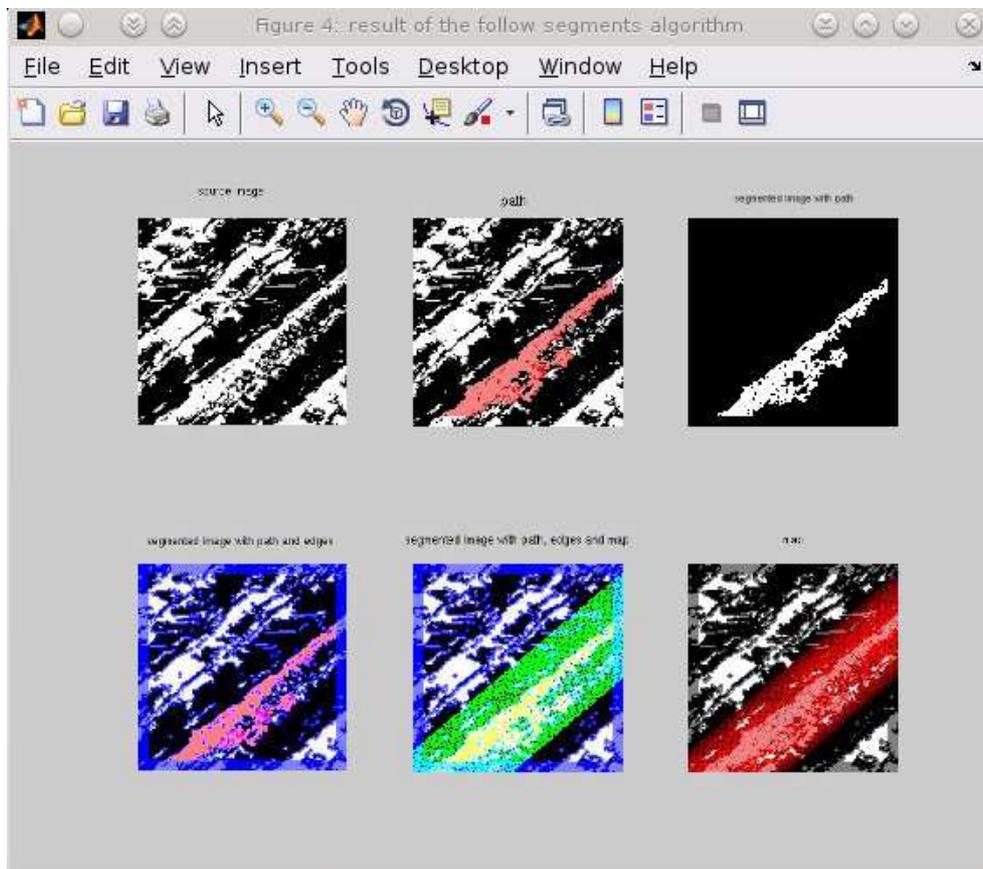


Figura 5.11: immagine AFM "B", durante il calcolo di O_3 sono visualizzati l'immagine in super-risoluzione blind, l'immagine super-risoluzione blind con il pattern O_3 sovrapposto, il pattern O_3 calcolato, l'immagine in super-risoluzione blind con sovrapposti il pattern O_3 calcolato e i segmenti calcolati, l'immagine super-risoluzione blind con sovrapposti il pattern O_3 calcolato, la mappa di probabilità, e i segmenti, l'immagine super-risoluzione blind con sovrapposto il pattern O_3 calcolato

In figura 5.12 vediamo il solo modello calcolato e il modello immerso nell'immagine originale.

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

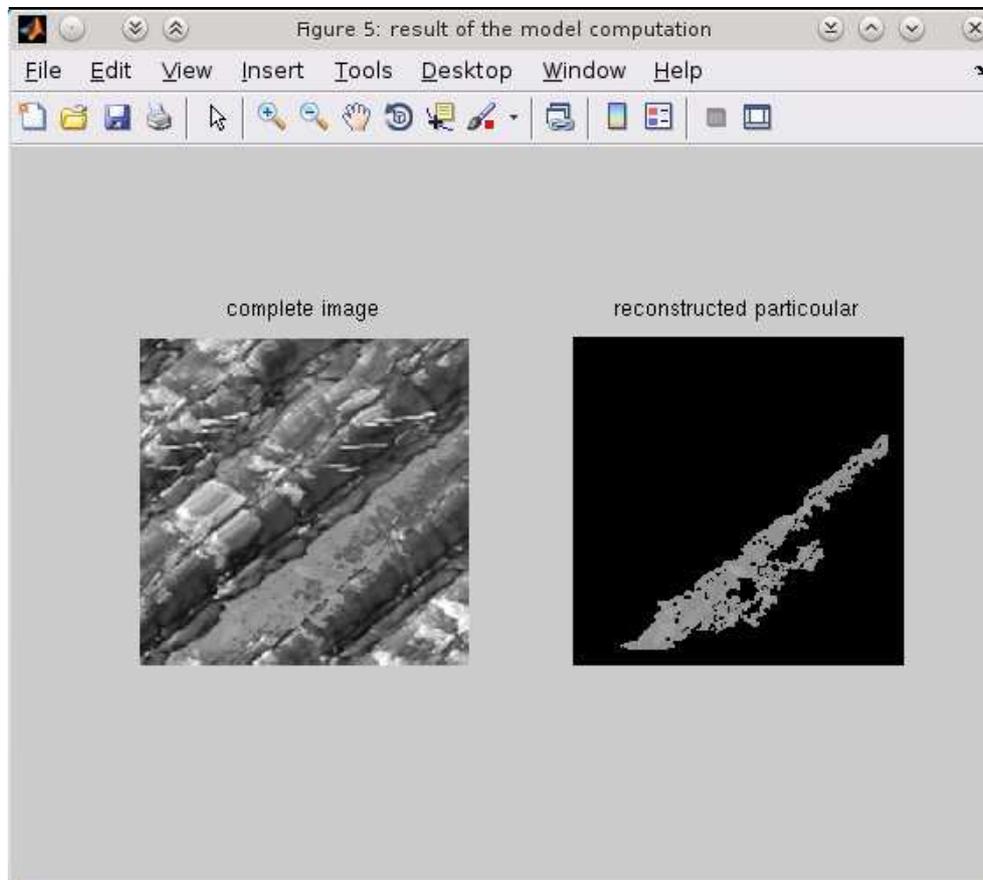


Figura 5.12: immagine AFM "B", a sinistra il modello calcolato e destra il path calcolato

5.4 Immagine AFM "C"

In questo caso viene illustrato come viene elaborata l'immagine di una superficie di una cellula. La figura 5.13 visualizza i parametri impostati per l'analisi dell'immagine. I valori numerici usati per il calcolo sono i seguenti:

- area analizzata (23, 78) (185, 240).
- il punto appartenente alla struttura dal quale inizia l'analisi ha coordinate (95, 78)
- il fattore di zoom è 2
- il metodo di super risoluzione blind è lo spline
- il valore di Δ_A è 8 e il valore di Δ_R è 10
- il valore di Δ_A^{max} è 12, il valore di Δ_R^{max} è 10, Δ_A^{min} è 10 e il valore di Δ_R^{min} è 8.
- per individuare la classe della figura viene utilizzato il classificatore (modalità auto) e la modalità di segmentazione è dipendente dal modello e dal classificatore (modalità auto).

5.4. IMMAGINE AFM "C"

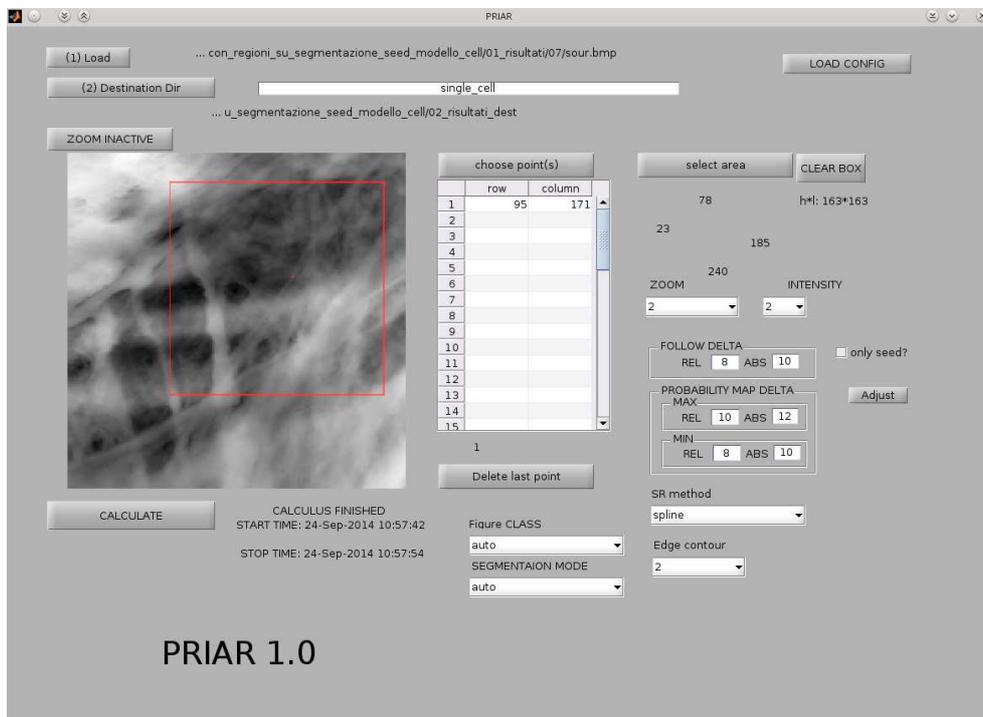


Figura 5.13: immagine AFM "C", in evidenza i parametri utilizzati per l'analisi di una cellula

In figura 5.14 vediamo la prima risposta che PRIAR fornisce all'utente, ovvero, la classe di appartenenza dell'immagine analizzata.

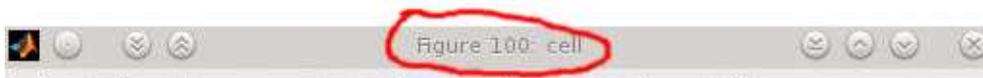


Figura 5.14: immagine AFM "C", risposta "cell" del classificatore

In figura 5.15 vediamo il primo risultato che presenta PRIAR a video. Questo è il risultato del primo algoritmo che, dopo aver super-risolto in modo blind l'immagine, traccia un sottoinsieme della superficie (O_1).

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

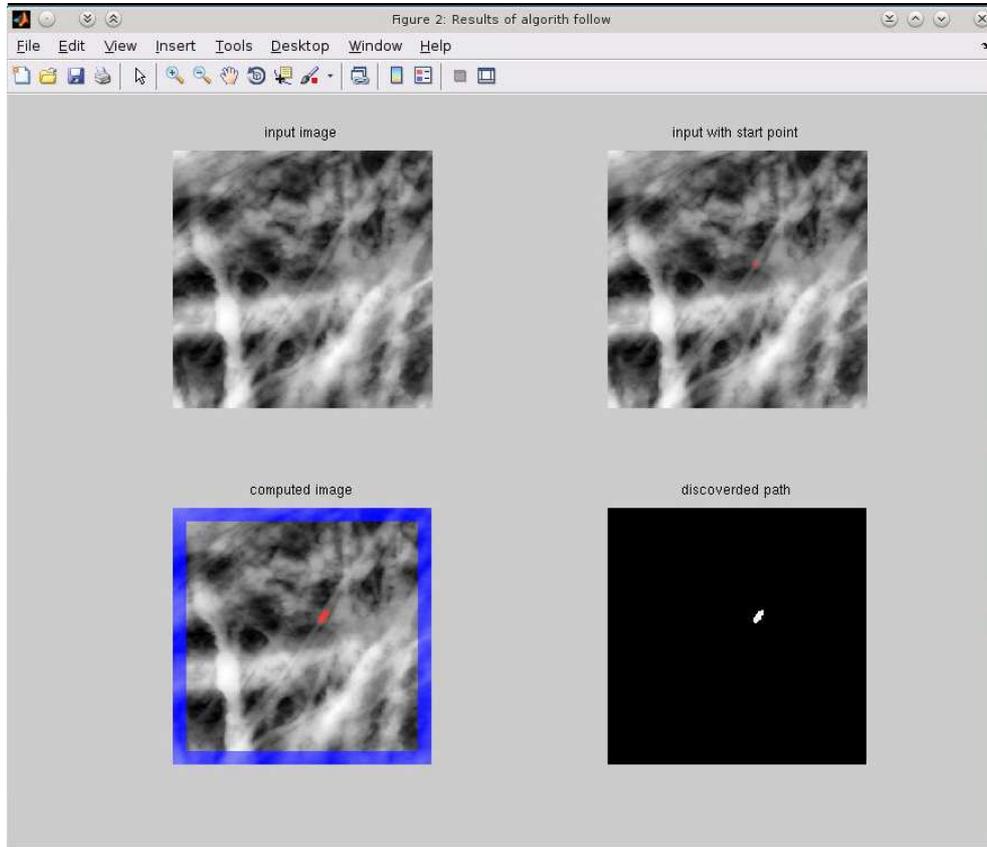


Figura 5.15: Immagine AFM "C", durante il calcolo di O_1 sono visualizzati l'immagine di input, l'immagine di input con evidenziato il punto da cui è iniziato il calcolo, il pattern O_1 calcolato sull'immagine e e pixel che costituiscono il pattern O_1

In figura 5.16 vediamo come O_1 viene esteso in O_2 .

5.4. IMMAGINE AFM "C"

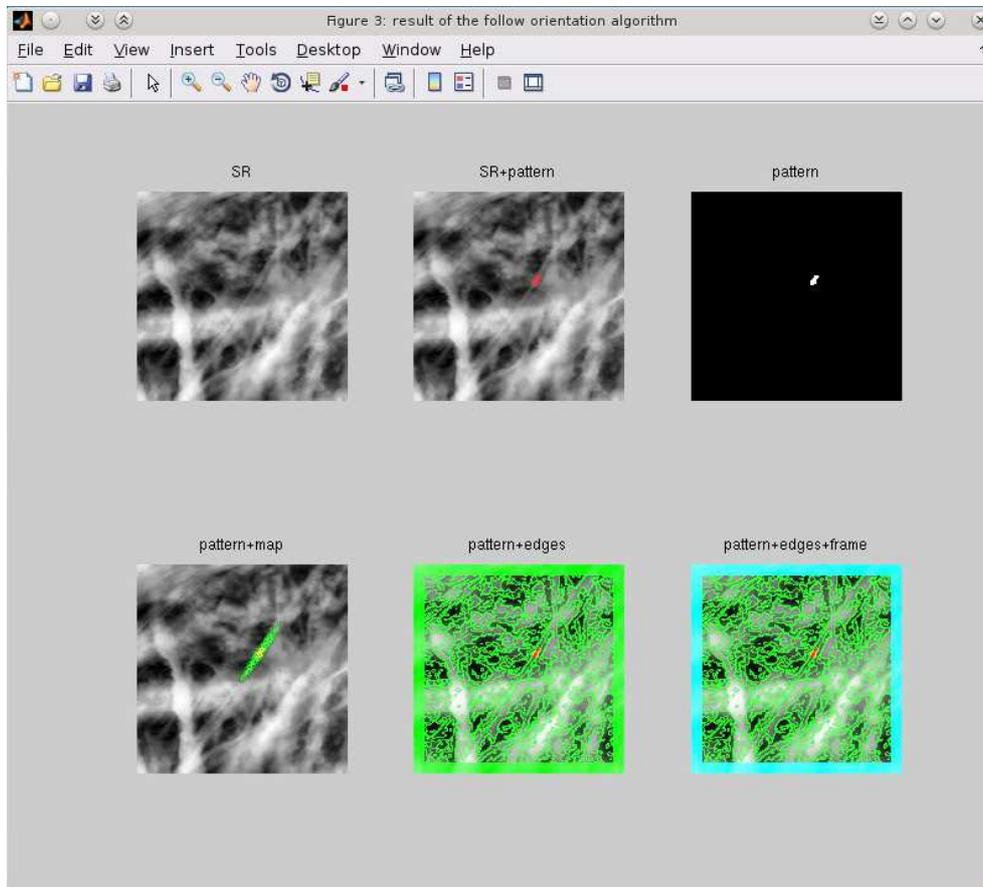


Figura 5.16: Immagine AFM "C", durante il calcolo di O_2 sono visualizzati l'immagine in super-risoluzione blind, l'immagine in super-risoluzione blind con il pattern O_2 sovrapposto, il pattern O_2 calcolato, l'immagine in super-risoluzione blind con sovrapposti il pattern O_2 calcolato e la mappa di probabilità, l'immagine in super-risoluzione blind con sovrapposti il pattern O_2 calcolato e i segmenti calcolati, l'immagine in super-risoluzione blind con sovrapposti il pattern O_2 calcolato, i segmenti calcolati e la cornice nella quale non viene effettuato computo

In figura 5.17 vediamo come O_2 viene esteso in O_3 .

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

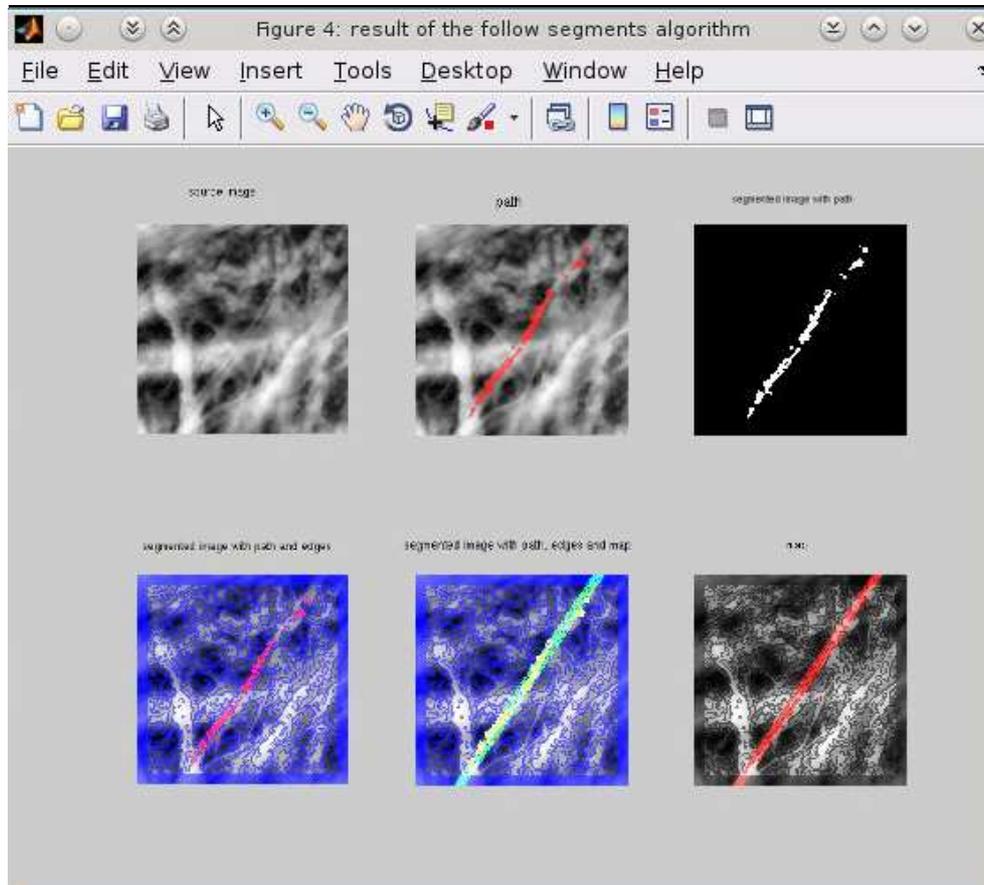


Figura 5.17: immagine AFM "C", durante il calcolo di O_3 sono visualizzati l'immagine in super-risoluzione blind, l'immagine super-risoluzione blind con il pattern O_3 sovrapposto, il pattern O_3 calcolato, l'immagine in super-risoluzione blind con sovrapposti il pattern O_3 calcolato e i segmenti calcolati, l'immagine super-risoluzione blind con sovrapposti il pattern O_3 calcolato, la mappa di probabilità, e i segmenti, l'immagine super-risoluzione blind con sovrapposto il pattern O_3 calcolato

In figura 5.18 vediamo il solo modello calcolato e il modello immerso nell'immagine originale.

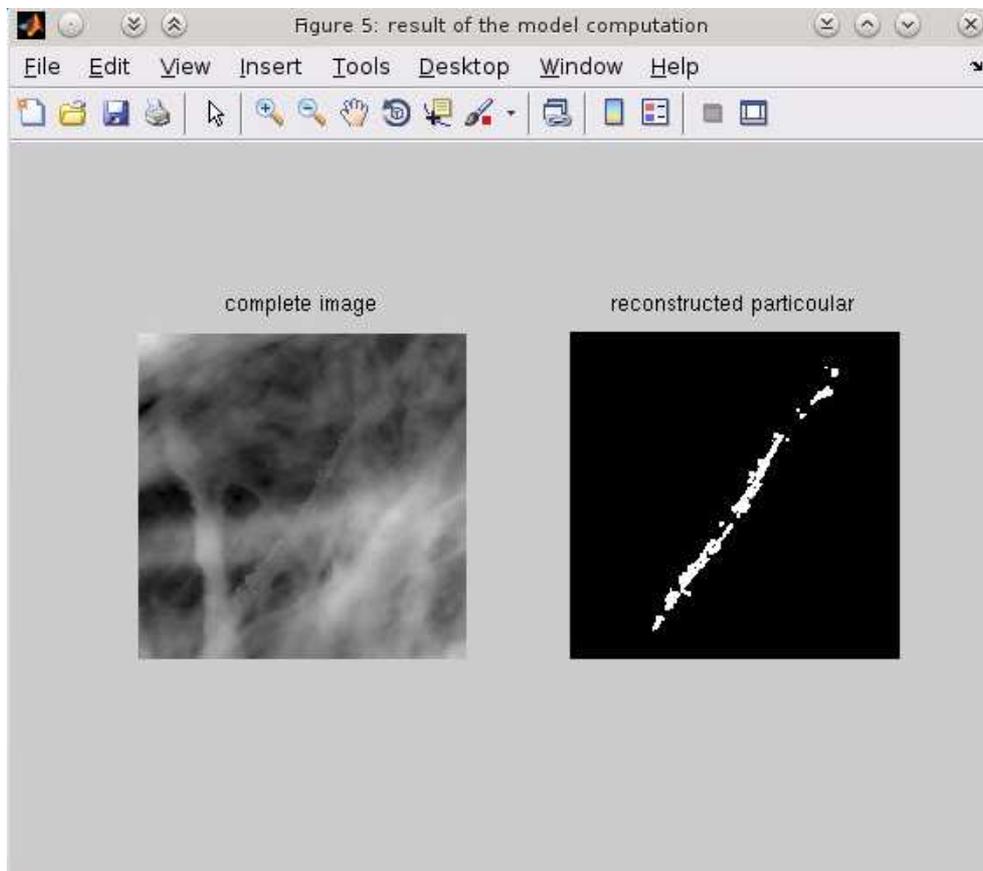


Figura 5.18: immagine AFM "C", a sinistra il modello calcolato e destra il path calcolato

5.5 Immagine AFM "D"

In questo caso viene illustrato come viene elaborata una superficie di una cellula. La figura 5.19 visualizza i parametri impostati per l'analisi dell'immagine. Diversamente dai casi precedenti, per migliorare il riconoscimento dell'oggetto, è stata utilizzata l'analisi multi-point. I valori numerici usati per il calcolo sono:

- area analizzata (6, 4) (441, 439).
- il punto appartenente alla struttura dal quale inizia l'analisi ha coordinate (209, 178)
- il fattore di zoom è 2
- il metodo di super risoluzione blind è lo spline
- il valore di Δ_A è 40 e il valore di Δ_R è 25
- il valore di Δ_A^{max} è 48, il valore di Δ_R^{max} è 30, Δ_A^{min} è 40 e il valore di Δ_R^{min} è 25.

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

- per individuare la classe della figura viene utilizzato il classificatore (modalità auto) e la modalità di segmentazione è dipendente dal modello e dal classificatore (modalità auto).

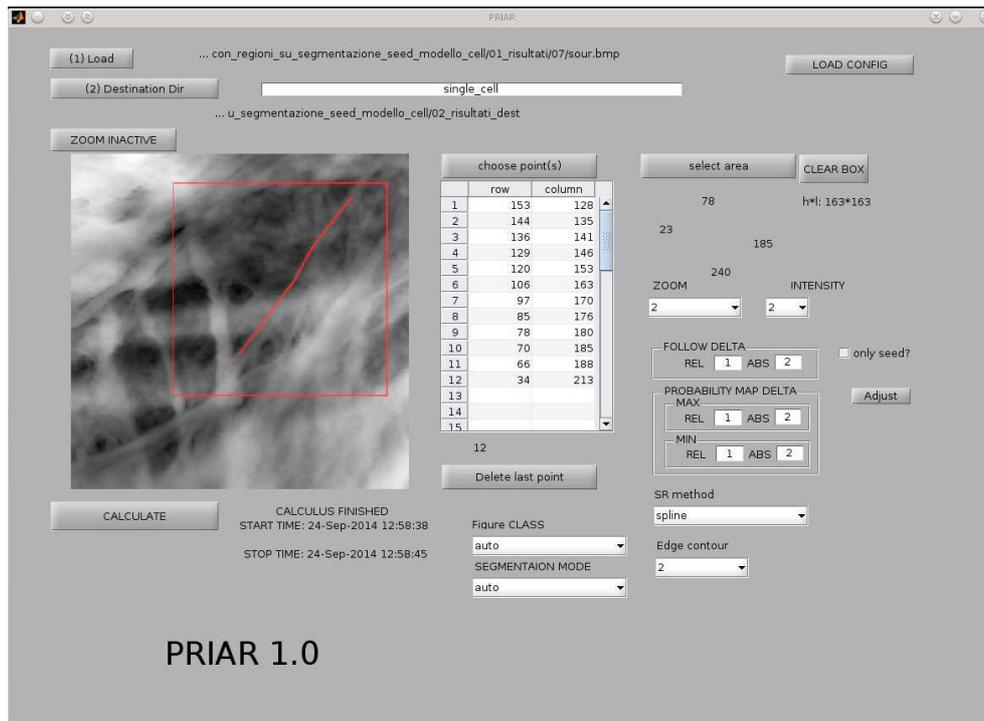


Figura 5.19: immagine AFM "D", in evidenza i parametri utilizzati per l'analisi di una cellula

In figura 5.20 vediamo la prima risposta che PRIAR fornisce all'utente, ovvero, la classe di appartenenza dell'immagine analizzata.

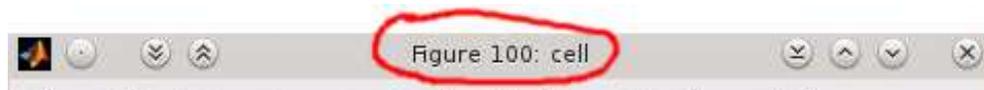


Figura 5.20: Immagine AFM "D": risposta "cell" del classificatore

In figura 5.21 vediamo il primo risultato che presenta PRIAR a video. Questo è il risultato del primo algoritmo che, dopo aver super-risolto in modo blind l'immagine, traccia un sottoinsieme della superficie O_2 a partire da O_1 .

5.5. IMMAGINE AFM "D"

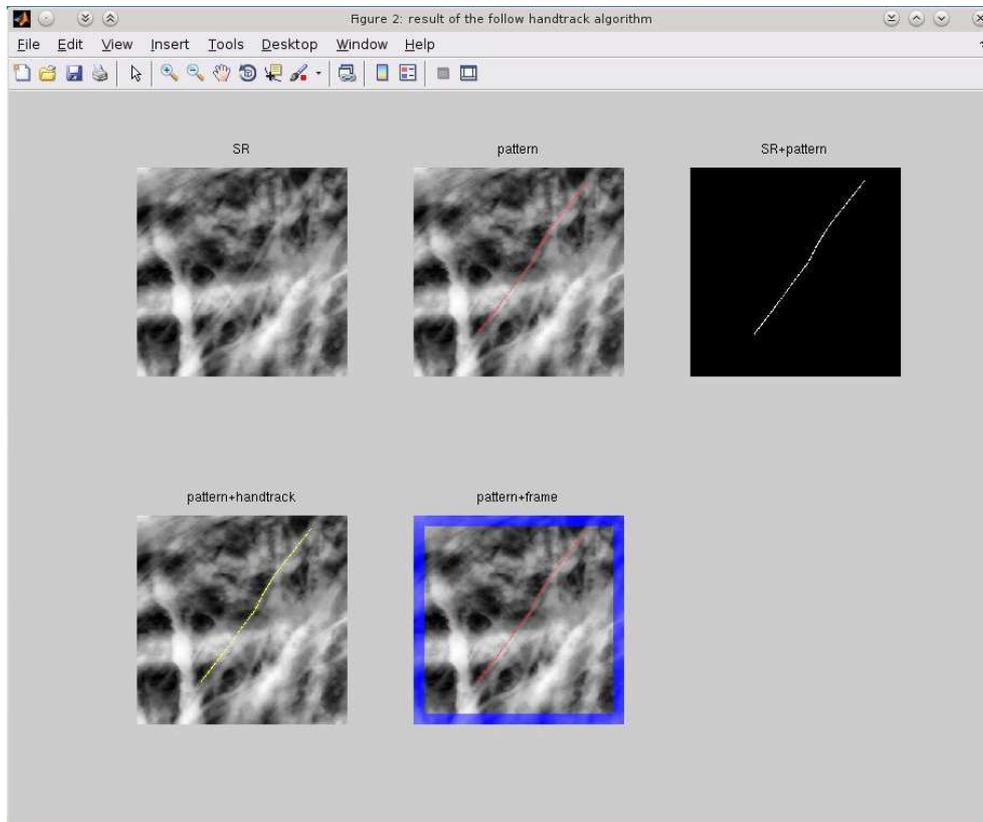


Figura 5.21: immagine AFM "D", durante il calcolo di O_2' sono visualizzati l'immagine in super-risoluzione blind, l'immagine in super-risoluzione blind con il pattern O_2' , il pattern O_2' , il pattern O_2' e la traccia dell'utente, il pattern e la cornice dove non può essere effettuata alcuna ricerca

In figura 5.22 vediamo come O_2' viene esteso in O_3 .

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

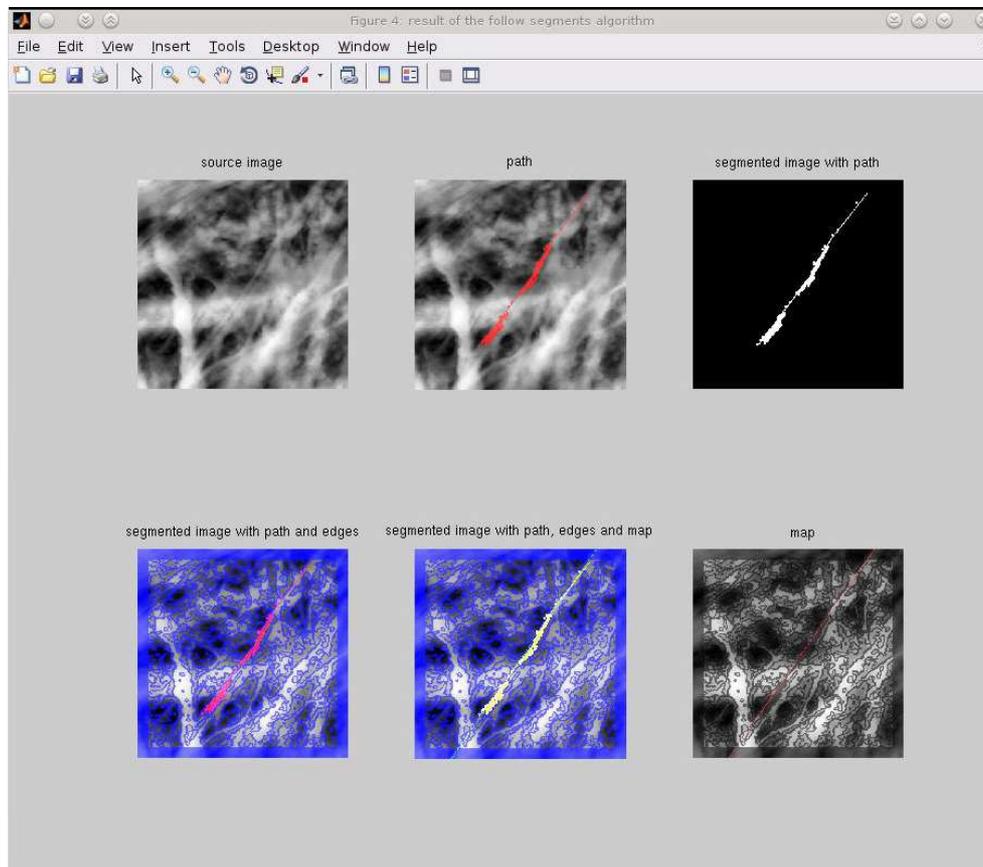


Figura 5.22: immagine AFM "D", durante il calcolo di O_3 sono visualizzati l'immagine in super-risoluzione blind, l'immagine super-risoluzione blind con il pattern O_3 sovrapposto, il pattern O_3 calcolato, l'immagine in super-risoluzione blind con sovrapposti il pattern O_3 calcolato e i segmenti calcolati, l'immagine super-risoluzione blind con sovrapposti il pattern O_3 calcolato, la mappa di probabilità, e i segmenti, l'immagine super-risoluzione blind con sovrapposto il pattern O_3 calcolato

In figura 5.23 vediamo il solo modello calcolato e il modello immerso nell'immagine originale.

5.5. IMMAGINE AFM "D"

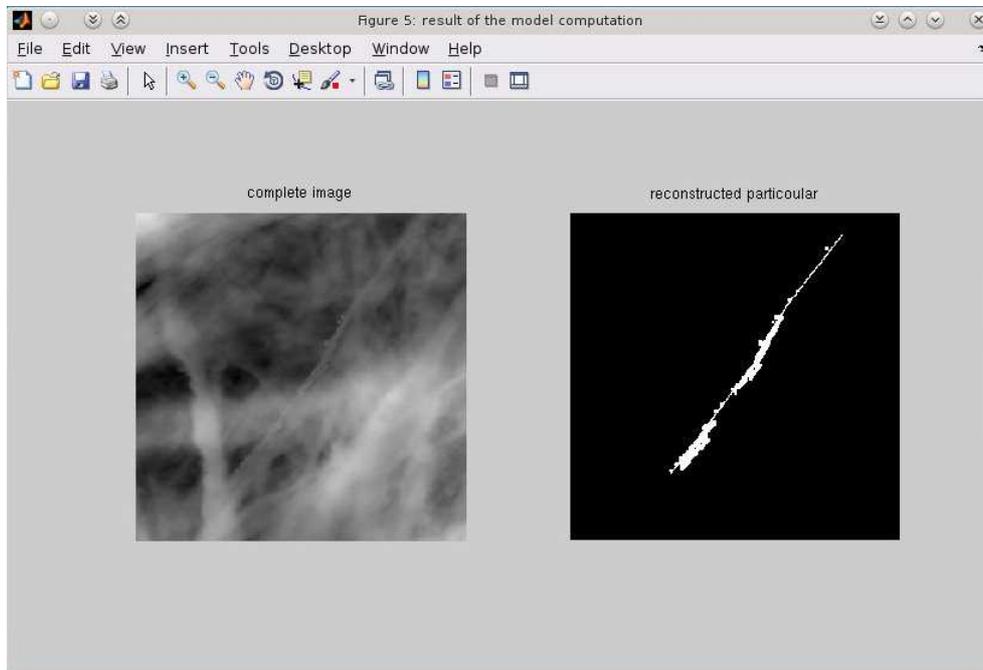


Figura 5.23: immagine AFM "D", a sinistra il modello calcolato e destra il path calcolato

In quest'ultimo esempio, calcolato il modello, mettiamo a confronto l'immagine 3D da elaborare (figura 5.24) e l'immagine 3D dopo l'elaborazione (figura 5.25).

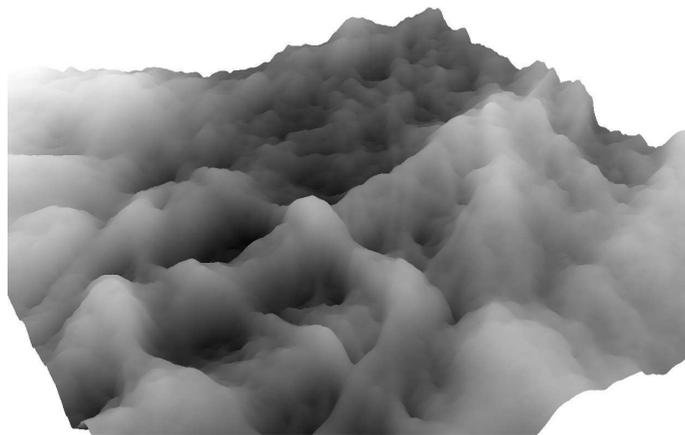


Figura 5.24: immagine AFM "D", immagine 3D prima dell'elaborazione

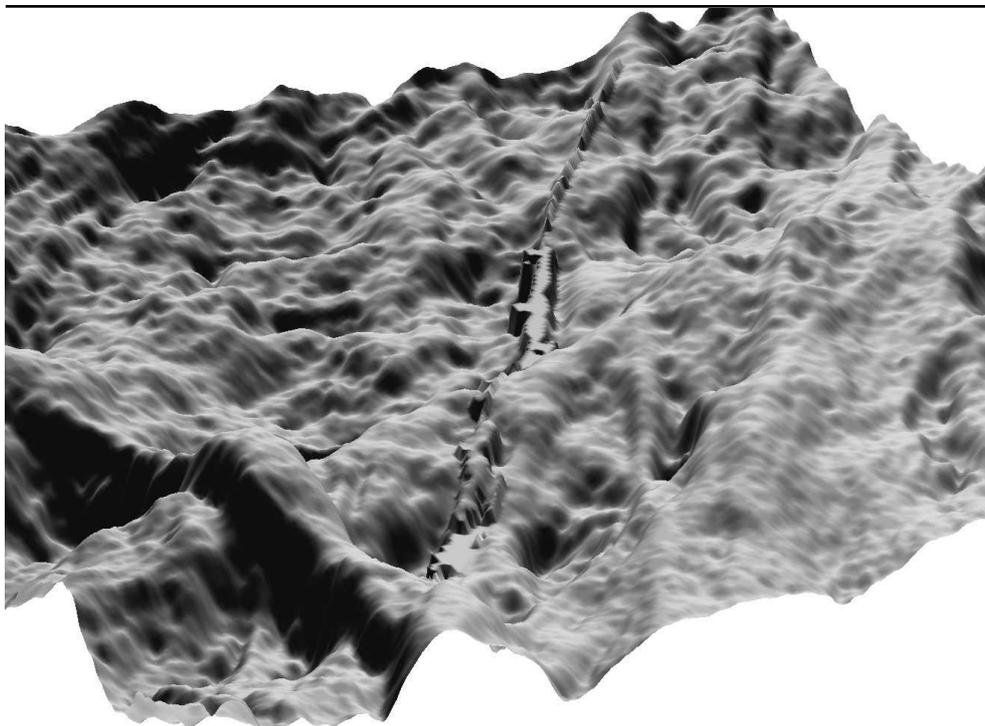


Figura 5.25: immagine AFM "D", immagine 3D dopo l'elaborazione

5.6 Immagini di oggetti subacquei

Il metodo PRIAR è stato implementato con metodologie generiche e capaci di adattarsi all'analisi di un elevato numero di oggetti. PRIAR è stato sperimentato in occasione di alcune campagne di acquisizione effettuate tramite veicoli sottomarini autonomi (*Autonomous Underwater Vehicle, AUV*) [164, 5, 58, 6, 155]. Le immagini raccolte da telecamere durante le missioni sono state messe a confronto con le immagini di alcuni reperti marittimi fotografati in laboratorio. In questo paragrafo sono illustrati i risultati ottenuti dalle immagini acquisite in laboratorio e dalle immagini acquisite durante le missioni.

Le immagini acquisite in laboratorio sono immagini 3D in scala di grigio acquisite per mezzo di una Kinect ¹ [171].

In figura 5.26a l'immagine di un'anfora acquisita in laboratorio. In figura 5.26b il risultato dell'elaborazione: l'anfora viene completamente riconosciuta da PRIAR.

¹ si ringrazia l'ing. Massimo Magrini aver fornito le immagini studiate

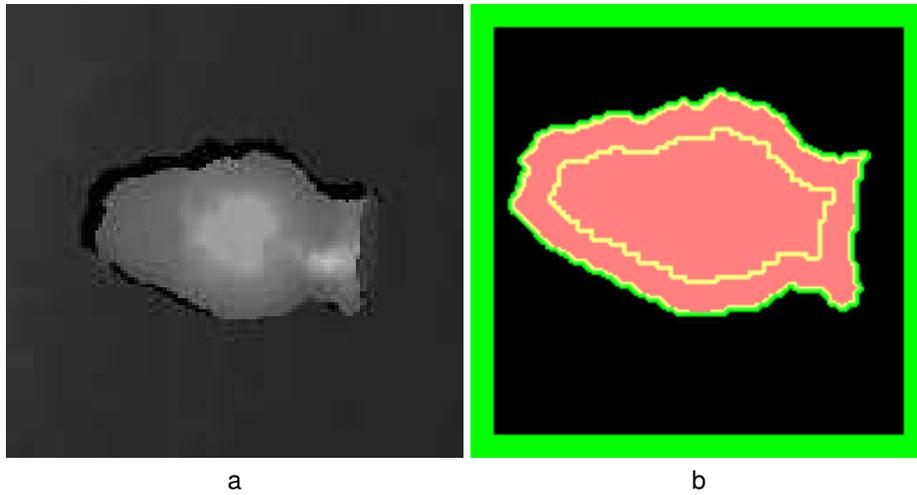


Figura 5.26: in (a) l'immagine acquisita con la kinect, in (b) il vaso completamente riconosciuto

In figura 5.27a l'immagine della medesima anfora acquisita in laboratorio in peggiori condizioni di illuminazione. Le aree meno definite non sono state individuate da PRIAR, viene comunque riconosciuto correttamente oltre il 90% dell'anfora.

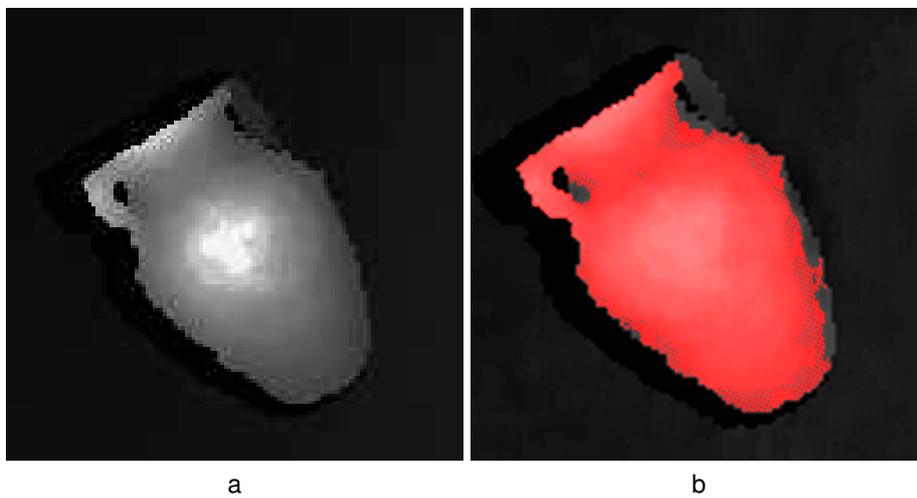


Figura 5.27: in (a) l'immagine acquisita con la kinect, in (b) il vaso completamente riconosciuto

Durante le missioni in mare sono stati fotografati dagli AUV alcuni siti marittimi, tra cui

CAPITOLO 5. STUDIO DI ALCUNI CASI: APPLICAZIONI A IMMAGINI AFM E SUBACQUEE

quelli nella zona delle secche della Meloria presso Livorno².

In figura 5.28 una rappresentazione in livelli di grigio (l'intensità del grigio indica la profondità in 3D dell'oggetto) calcolata dal modello 3D che riproduce un'area del sito archeologico della Meloria. In evidenza la parte dell'immagine che è stata elaborata.

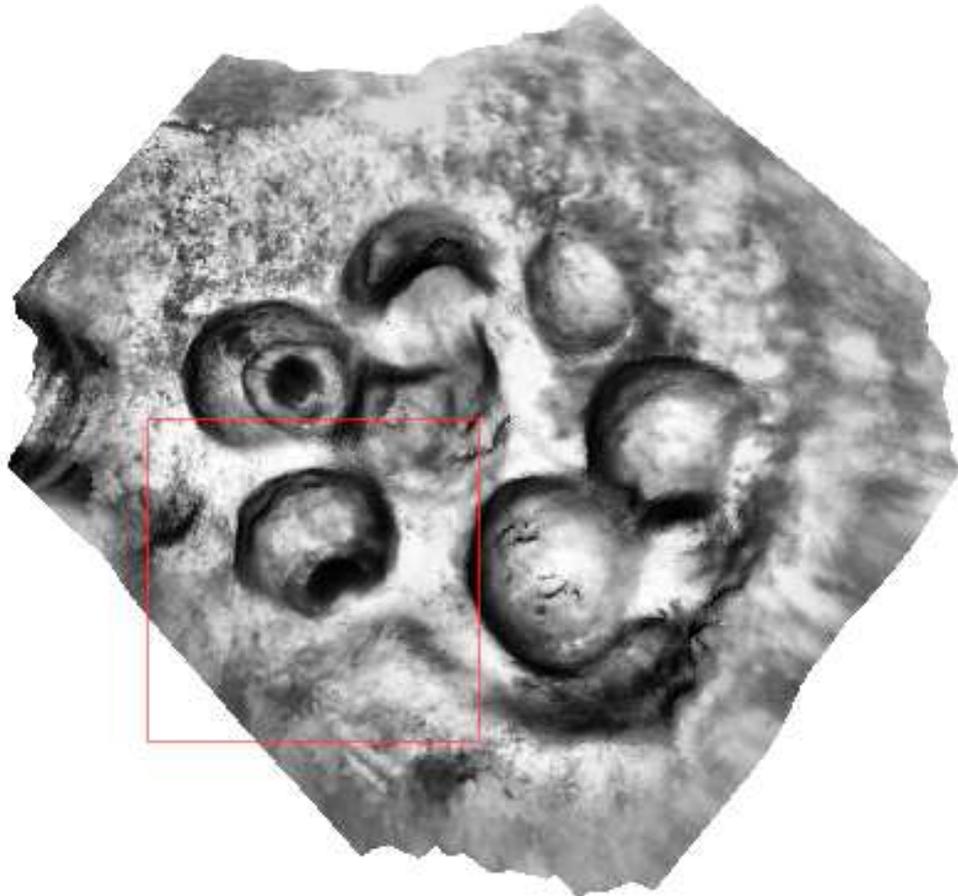


Figura 5.28: rappresentazione a livelli ricavata dal modello 3D che riproduce un'area il sito archeologico della Meloria

In figura 5.29 una delle anfore individuate da PRIAR.

² http://it.wikipedia.org/wiki/Secche_della_Meloria

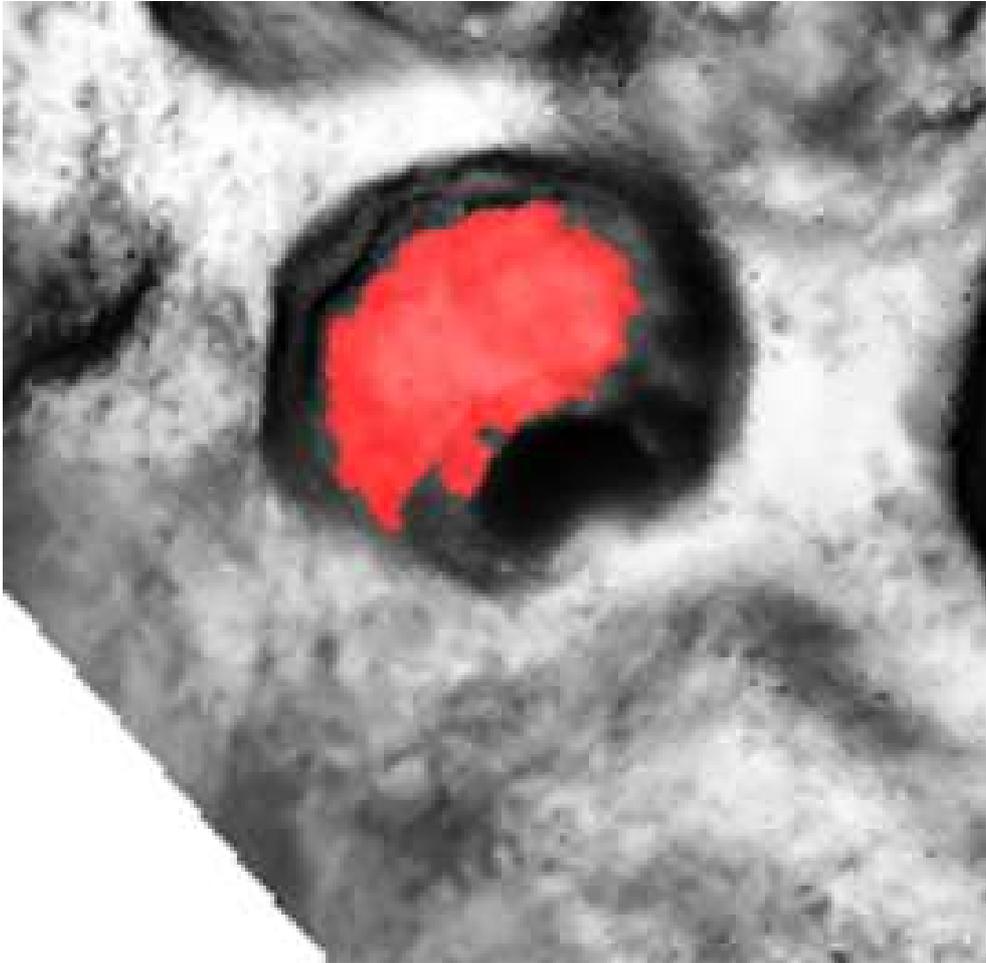


Figura 5.29: rappresentazione 2D a livelli ricavata dal modello 3D che riproduce un'area il sito archeologico della Meloria

5.7 Considerazioni

PRIAR durante il funzionamento ha dimostrato di ottenere ottimi risultati. Infatti è stato in grado di delimitare non solo le immagini per le quali la sua knowledge-base è stata arricchita, ma è anche riuscito a delimitare correttamente immagini di reperti marittimi. La robustezza dell'algoritmo è stata convalidata sia dai risultati che possono essere verificati con un controllo visivo che dalla convalida dello staff dei biologi del Biolab di Pisa [34], soddisfatti del trattamento delle immagini del citoscheletro.

Conclusioni

Il calcolo di super-risoluzione per le immagini sta divenendo sempre più importante con l'introduzione di nuovi dispositivi per l'acquisizione delle immagini a basso costo. In questo lavoro ci siamo voluti confrontare proprio con il problema della super-risoluzione di tipo single-frame (SISO). Per elaborare questo tipo di immagini, si ha a disposizione poca informazione. Gli algoritmi che servono devono quindi cercare dapprima di massimizzare l'informazione disponibile senza inquinarla con ipotesi a contorno. Successivamente devono utilizzare un modello che permetta di capire che cosa rappresenta l'immagine analizzata ed elaborarla ricorrendo a un'opportuna conoscenza.

Il problema è stato affrontato in quest'ottica valutando prima lo stato dell'arte degli algoritmi di super-risoluzione per le immagini single-frame, e quello dei metodi di pattern-recognition basati su knowledge-base. I due metodi sono stati poi combinati in un unico metodo generale in cui la super-risoluzione aiuta il pattern-recognition e successivamente, sfruttando un metodo nuovo per affrontare il problema, il pattern-recognition aiuta a super-risolvere l'immagine andando a riconoscere l'immagine e a utilizzare il modello che più la rispecchia.

A questo scopo sono stati sviluppati nuovi algoritmi non presenti in letteratura in cui vengono sfruttate le tecniche di crawling su grafici unite a metodi statistici e di calcolo delle probabilità. I metodi studiati sono stati implementati sfruttando MATLAB® ed è stato possibile realizzare il *Tool PRIAR* con il quale si può verificare l'efficacia raggiunta nell'analisi delle immagini. Durante lo sviluppo è stata prestata particolare attenzione alla complessità computazionale che si è cercato di contenere. Gli algoritmi sviluppati necessitano di ipotesi per ottimizzare l'analisi delle immagini, ipotesi che al momento sono impostate manualmente a seguito di una pre-analisi dell'immagine. In futuro si prevede l'implementazione dei metodi di pre-analisi per ottenere un tool in grado di analizzare automaticamente le immagini che è in grado di classificare.

Indice analitico

- O_1 , calcolo di, 77
- O_2 , calcolo di, 77
- O_3 , calcolo di, 79

- aberrazione sferica, 9
- AFM, 18
- AFM, artefatti, 19
- AFM, bow, 21
- AFM, convoluzione, 24
- AFM, creep, 21
- AFM, crosstalk, 21
- AFM, inarcatura, 21
- AFM, isteresi, 21
- AFM, overshoot, 24
- AFM, rumore, 25
- AFM, scaling, 20
- AFM, scansione, 18
- aliasing, 8
- Allebach, metodo, 30
- astigmatismo, 12

- Bartlane, metodo, 5
- Bhattacharyya, distanza, 47
- bilineare, metodo di SR, 27
- bivubico, metodo di SR, 27
- blurring, 8
- bow, 21

- Carey, metodo, 32
- Chebyshev, distanza, 47
- coma, 10
- contorni attivi, metodo, 56
- creep, 21
- crosstalk, 21

- distanza, misura, 46

- distorsione, 9

- euclidea, distanza, 47

- f-divergente, funzione, 48
- Felzenszwalb, 62
- Freeman, 37

- Gestalt, 62
- grafi, metodo, 58

- Hamming, distanza, 47
- Hausdorff, 53
- Hausdorff, dimensione, 52
- Hellinger, distanza, 48

- immagine, acquisizione, 8
- immagine, artefatti, 8
- immagine, definizione, 6
- immagine, scalettatura, 8
- inarcatura, 21
- isteresi, 21

- Jenesen, metodo, 28

- Kim, metodo, 39
- Kolmogorov, integrale, 48
- Kullback–Leibler, distanza, 48
- Kullback–Leibler, divergenza, 48

- Lipschitziana, funzione, 34

- Mahalanobis, distanza, 47
- Manhattan, distanza, 47
- Minkowski, distanza, 46
- Moiré, effetto, 8
- Muresan, metodo, 34

Indice analitico

n-connesione, pixel, 45

NEDI, metodo, 35

omogeneità, 46

overshoot, 24

Petzval, campo di curvatura, 14

piramidale, metodo, 55

PRIAR, analisi multi-point, 78

PRIAR, analisi single-point, 77

PRIAR, calcolo del modello, 80

PRIAR, GUI, 81

PRIAR, mappa di probabilità, 77

PRIAR, modello, 74

PRIAR, PR, 77

Renyi, dimensione, 52

rumore elettrico, 16

scaling, 20

segmentazione, 44, 45

sfocatura, 15

Shannon, dimensione, 54

similarità, misura, 46

snake, metodo, 56

spartiacque, metodo, 55

Spectral Embedding MinCut, metodo, 62

spline, metodo di SR, 27

super-visionati, metodi, 44

Voss, algoritmo, 55

Xu, algoritmo, 62

Zahn, 62

Riferimenti bibliografici

1. A. Papoulis. *Probability, random variables, and stochastic processes*. McGraw-Hill, 1991.
2. H. Abdi. A neural network primer. *Journal of Biological Systems*, 2(3):247–281, 1994.
3. N. Adluru. An experimental evaluation of diffusion tensor image segmentation using graph-cuts. In *Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5653–5656, 2009.
4. J. P. Allebach and P. W. Wong. Edge-directed interpolation. In *ICIP (3)*, pages 707–710. IEEE, 1996.
5. B. Allotta et al. Progetto Thesaurus. *L'Archeologo Subacqueo*, XVIII(1), 2012.
6. B. Allotta et al. Thesaurus Project: Design of New Autonomous Underwater Vehicles for Documentation and Protection of Underwater Archaeological Sites. *Lecture Notes in Computer Science*, 7616, 2012.
7. H. J. Antonisse. Image segmentation in pyramids. *Computer Graphics and Image Processing*, 19(4):367–383, 1982.
8. E. Ardizzone, R. Gallea, O. Gambino, and R. Pirrone. Fuzzy-based kernel regression approaches for free form deformation and elastic registration of medical images. *Biomedical Engineering*, 2009.
9. Australasian Conference on Robotics and Automation. *Visual Object Recognition using Template Matching*, 2004.
10. X. Bai and G. Sapiro. Distance cut: interactive segmentation and matting of images and videos. In *IEEE International Conference on Image Processing*, pages 249–252, 2007.
11. P. Baldi. *Calcolo delle probabilità*. McGraw-Hill, 2011.
12. J. J. Berman. *Principles of Big Data: Preparing, Sharing, and Analyzing Complex Information*. Morgan Kaufmann, Amsterdam, 2013.
13. S. Beucher. Watersheds of functions and picture segmentation. Acoustics, Speech, and Signal Processing. In *IEEE International Conference on ICASSP*, 1982.
14. S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In *Mathematical Morphology And Its Applications To Image Processing*, pages 69–76, 1994.
15. A. Bismpiagiannis. Measurements and correction of geometric distortion, 2007. http://scien.stanford.edu/pages/labsite/2007/psych221/projects/07/geometric_distortion/project.htm. Ultimo accesso 10/06/2014.
16. S. Borman and R. L. Stevenson. Super-resolution from image sequences-a review. In *Circuits and Systems, 1998. Proceedings. 1998 Midwest Symposium on*, pages 374–378, Aug 1998.
17. Y. Boykov and G. Funka-Lea. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.
18. Y. Boykov and M. P. Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *ICCV*, pages 105–112, 2001.

19. Y. Boykov and V. Kolmogorov. Computing Geodesics and Minimal Surfaces via Graph Cuts. In *ICCV*, pages 26–33. IEEE Computer Society, 2003.
20. Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.
21. J. P. Braquelaire and L. Brun. Image segmentation with topological maps and inter-pixel representation. *Journal of Visual Communication and Image Representation*, 9(1):62–79, 1998.
22. R. Brunelli and T. Poggio. Template matching: matched spatial filters and beyond. *Pattern Recognition*, 30(5):751–768, 1997.
23. A. A. Bukharaev, N. V. Berdunov, D. V. Ovchinnikov, and K. M. Salikhov. Three dimensional probe and surface reconstruction for atomic force microscopy using a deconvolution algorithm. *Scanning Microscopy International*, 12:225–234, 1998.
24. B. Bushan. *Springer handbook of nanotechnology*. Springer, Berlin, 2010.
25. K. S. Camilus, Govindan V. K., and Sathidevi P. S. Computer-Aided Identification of the Pectoral Muscle in Digitized Mammograms. *J. Digital Imaging*, 23(5):562–580, 2010.
26. J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
27. W. K. Carey, D. B. Chuang, and S. S. Hemami. Regularity-preserving image interpolation. *Image Processing, IEEE Transactions on*, 8(9):1293–1297, Sep 1999.
28. K. R. Castleman and Q. Wu. *Image segmentation*. Elsevier, 2008.
29. C. Corsori. Aberrazione sferica, 2008. http://www.treccani.it/enciclopedia/aberrazione-sferica_%28Enciclopedia_della_Scienza_e_della_Tecnica%29. Ultimo accesso 11/06/2014.
30. O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.
31. B. B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(1):72–77, Jan 1995.
32. C. Chen, H. Li, and X. Zhou. Automated Segmentation of Drosophila RNAi Fluorescence Cellular Images Using Graph Cuts. In *Proceedings of the 13th International Conference on Multimedia Modeling - Volume Part I, MMM'07*, pages 116–125, Berlin, Heidelberg, 2006. Springer-Verlag.
33. L. D. Cohen. On Active Contour Models and Balloons. *CVGIP: Image Underst.*, 53(2):211–218, March 1991.
34. S. Colantonio, M. D'Acunto, M. Righi, and O. Salvetti. Super Resolution-aided recognition of cytoskeletons in scanning probe microscopy images. In *International Conference on Pattern Recognition Applications and Methods, ICPRAM2014*, pages 703–709. SciTePress Digital Library, March 2014.
35. M. Couprie and G. Bertrand. Topological Grayscale Watershed Transformation. In *IN SPIE VISION GEOMETRY V PROCEEDINGS, 3168*, pages 136–146, 1997.
36. J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1362–1374, 2009.
37. M. D'Acunto. *Nanotribology and Biomaterials: New Challenges in Atomic Force Microscopy, in Nanophysics, Nanoclusters and Nanodevices*, chapter 1, pages 1–38. Nova Science publisher, k.s. gehar edition, 2006.
38. M. D'Acunto. Nanotribology and Biomaterials: New Challenges in Atomic Force Microscopy. *Current Research on Nanotechnology*, 1(2):191–228, 2007.
39. M. D'Acunto et al. Inferential Mining for Reconstruction of 3D Cell Structures in Atomic Force Microscopy Imaging. In *KDIR*, pages 348–353. SciTePress, 2011.
40. M. D'Acunto et al. Wize sniffer. A new portable device designed for selective olfaction. In *BIOSTEC 2014 - 7th International Joint Conference on Biomedical Engineering Systems and Technologies. Special Session SUPERHEAL 2014*, pages 577–582. SciTePress Digital Library, 2014.

41. M. D'Acunto, G. Pieri, M. Righi, and O. Salvetti. Enhanced Resolution Methods for Improving Image Analysis and Pattern Recognition in Scanning Probe Microscopy. In *4th International Workshop on Image Mining. Theory and Applications, IMTA2013*, pages 22–28. SciTePress Digital Library, 2013.
42. M. D'Acunto, G. Pieri, M. Righi, and O. Salvetti. A methodological approach for combining super-resolution and pattern-recognition to image identification. *Pattern Recognition and Image Analysis*, 24(2):209–217, 2014.
43. M. D'Acunto, M. Righi, and O. Salvetti. A new method combining Enhanced Resolution and Pattern Identification (submitted). *The Open Medical Informatics Journal*, 2015.
44. S. Danti, M. D'Acunto, L. Trombi, S. Berrettini, and A. Pietrabissa. A Micro/Nanoscale Surface Mechanical Study on Morpho-Functional Changes in Multilineage-Differentiated Human Mesenchymal Stem Cells. *Macromolecular Bioscience*, 7(5):589–598, 2007.
45. H. U. Danzebrink, L. Koenders, G. Wilkening, A. Yacoot, and H. Kunzmann. Advances in Scanning Force Microscopy for Dimensional Metrology. *Annals of the CIRP*, 55(2), 2006.
46. I. Davi. *Characterization of deep, nanometer scale diffraction grating with Atomic Force Microscope*. PhD thesis, DIMEG and IPL, 2005.
47. E. R. Davies. *Computer and machine vision theory, algorithms, practicalities*, 2012.
48. P. A. V. de Miranda. Synergistic arc-weight estimation for interactive image segmentation using graphs. *Computer Vision and Image Understanding*, 114(1):85–99, 2010.
49. Y. Deng, B. S. Manjunath, and H. Shin. Color Image Segmentation. In *CVPR*, pages 2446–2451. IEEE Computer Society, 1999.
50. C. H. Q. Ding, X. He, H. Zha, M. Gu, and H.D. Simon. A Min-max Cut Algorithm for Graph Partitioning and Data Clustering. In *ICDM*, pages 107–114. IEEE Computer Society, 2001.
51. E. Dinic. Algorithm for solution of a problem of maximum flow in networks with power estimation. *Soviet Mathematics*, pages 1277–1280, 1980.
52. J. Dombre. *Multi-scale representation systems for indexing and restoring color medieval archives*. PhD thesis, University of Poitiers, 2004.
53. X.L. Dong and D. Srivastava. Big data integration. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 1245–1248, 2013.
54. L. S. Dongmo et al. Experimental test of blind tip reconstruction for scanning probe microscopy. *Ultramicroscopy*, 85(3):141–153, 2000.
55. A. P. Eriksson, C. Olsson, and F. Kahl. Normalized Cuts Revisited: A Reformulation for Segmentation with Linear Grouping Constraints. In *ICCV*, pages 1–8. IEEE, 2007.
56. F. Estrada, A. D. Jepson, and C. Chennubhotla. Spectral embedding and min-cut for image segmentation. In *In British Machine Vision Conference*, 2004.
57. F. J. Estrada and A. D. Jepson. Benchmarking Image Segmentation Algorithms. *Int. J. Comput. Vision*, 85(2):167–181, November 2009.
58. European Association of Archaeologists. *Thesaurus Project. Planning and Development of new AUVs for Underwater Archaeological Research*, 2012.
59. Ahmed Fahad and Tim Morris. A Faster Graph-Based Segmentation Algorithm with Statistical Region Merge. In George Bebis et al., editors, *ISVC (2)*, volume 4292 of *Lecture Notes in Computer Science*, pages 286–293. Springer, 2006.
60. A. X. Falcão. An Ultra-Fast User-Steered Image Segmentation Paradigm: Live-Wire-On-The-Fly. *IEEE Trans. Med. Imaging*, 19(1):55–62, 2000.
61. A. X. Falcao. The image foresting transform: theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.
62. A. X. Falcão, J. K. Udupa, and S. Samarasekera. User-Steered Image Segmentation Paradigms: Live Wire and Live Lane. *Graphical Models and Image Processing*, 60(4):233–260, 1998.
63. K. Falconer. *Fractal geometry - mathematical foundations and applications*. Wiley, 3rd edition edition, 2014.

64. S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Advances and challenges in super-resolution. *Int. J. Imaging Systems and Technology*, 14(2):47–57, 2004.
65. R. A. Feagin. Heterogeneity versus homogeneity: A conceptual and mathematical theory in terms of scale-invariant and scale-covariant distributions. *Ecological Complexity*, 2(4):339–356, December 2005.
66. P. F. Felzenszwalb and D. P. Huttenlocher. Image Segmentation Using Local Variation. In *CVPR*, pages 98–104. IEEE Computer Society, 1998.
67. P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *Int. J. Comput. Vision*, 59(2):167–181, 2004.
68. C. Fernandez-Maloigne. *Advanced color image processing and analysis*. Springer, New York, NY, USA, 2013.
69. D. Freedman and T. Zhang. Interactive Graph Cut Based Segmentation with Shape Priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1)*, pages 755–762. IEEE Computer Society, 2005.
70. W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-Based Super-Resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.
71. W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning Low-Level Vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.
72. J. E. F. Friedl. *Mastering Regular Expressions*. O'Reilly, 3rd edition, 2006.
73. K. S. Fu and J. K. Mui. A survey on image segmentation. *Pattern Recognition*, 13(1):3–16, 1981.
74. B. V. Funt and G. D. Finlayson. Color Constant Color Indexing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(5):522–529, May 1995.
75. Satish G. and R. F. Sproull. Filtering Edges for Gray-Scale Devices. In *ACM SIGGRAPH*, volume 81, pages 1–5, 1981.
76. D. Geiger and F. Girosi. Parallel and Deterministic Algorithms from MRFs: Surface Reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(5):401–412, 1991.
77. P. Getreuer. Linear Methods for Image Interpolation. *Image Processing On Line*, 1, 2011.
78. A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988.
79. F. Golek, P. Mazur, Z. Ryszka, and S. Zuber. AFM image artifacts. *Applied Surface Science*, 304(0):11–19, 2014.
80. R. C. Gonzalez and R. E. Wood. *Digital image processing*. Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition edition, 2008.
81. R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing Using MATLAB*. Prentice-Hall, 2nd edition edition, 2010.
82. J. W. Goodman. *Introduction to Fourier Optics*. Ben Roberts, 3rd edition edition, 2005.
83. L. Grady. Random Walks for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006.
84. L. Grady and E. L. Schwartz. Isoperimetric Graph Partitioning for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):469–475, 2006.
85. D. M. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, 51(2):271–279, 1989.
86. J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. In *Princeton University Bulletin*, pages 49–52. Princeton University, 1902.
87. D. Halliday, R. Resnick, and J. Walker. *Fundamentals of Physics*. Halliday & Resnick Fundamentals of Physics. John Wiley & Sons Canada, Limited, 2010.
88. R. M. Haralick and L. G. Shapiro. Image Segmentation Techniques. In *Applications of Artificial Intelligence II*, volume 0548, pages 2–9, 1985.
89. R. M. Haralick, L. G. Shapiro, et al. Image Segmentation Techniques. *Computer Vision, Graphics and Image Processing*, 29:100–132, 1985.

90. C. Harris and M. Stephens. A Combined Corner and Edge Detection. In *Proceedings of The Fourth Alvey Vision Conf.*, pages 147–151, 1988.
91. E. Hecht. *Optics*. Addison Wesley, San Francisco, 2002.
92. E. Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die Reine und Angewandte Mathematik*, 1(136):210–271, 1909.
93. D. S. Hochbaum. Polynomial Time Algorithms for Ratio Regions and a Variant of Normalized Cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):889–898, 2010.
94. B. K. P. Horn. *Robot vision*. MIT electrical engineering and computer science series. MIT Press, 1986.
95. R. Howland and L. Benatar. *A practical guide to scanning probe microscopy*. ThermoMicroscopes, 2000.
96. Z. Hu. Clothing segmentation using foreground and background estimation based on the constrained delaunay triangulation. *Pattern Recognition*, 41(5):1581–1592, 2008.
97. J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image Indexing Using Color Correlograms. In *CVPR*, pages 762–768. IEEE Computer Society, 1997.
98. IEEE. *The Birth of Digital Phototelegraphy*, 2003.
99. L. Ikonen and P. J. Toivanen. Distance and Nearest Neighbor Transforms of Gray-Level Surfaces Using Priority Pixel Queue Algorithm. In J. Blanc-Talon et al., editors, *ACIVS*, volume 3708 of *Lecture Notes in Computer Science*, pages 308–315. Springer, 2005.
100. A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, 1989.
101. A. K. Jain, R. P. W. Duin, and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
102. K. Jensen and D. Anastassiou. Subpixel edge localization and the interpolation of still images. *IEEE Transactions on Image Processing*, 4(3):285–295, 1995.
103. I. Jermyn and H. Ishikawa. Globally Optimal Regions and Boundaries as Minimum Ratio Weight Cycles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(10):1075–1088, 2001.
104. A. J. Jerri. The Shannon sampling theorem —Its various extensions and applications: A tutorial review. *Proceedings of the IEEE*, 65(11):1565–1596, Nov 1977.
105. O. Juan and Y. Boykov. Capacity Scaling for Graph Cuts in Vision. In *IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.
106. T. Kapur. Segmentation of Brain Tissue from MR Images. In *Proceedings of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine*, CVRMed '95, pages 429–433, London, UK, UK, 1995. Springer-Verlag.
107. E. Karimi, K. Kangarloo, and S. Javadi. A Survey on Super-Resolution Methods for Image Reconstruction. *International Journal of Computer Applications*, 90(3):32–39, March 2014.
108. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.
109. J. M. Keller, S. Chen, and R. M. Crownover. Texture Description and Segmentation Through Fractal Geometry. *Comput. Vision Graph. Image Process.*, 45(2):150–166, February 1989.
110. T. Kenneth. Anti-Aliasing Through the Use of Coordinate Transformations. *ACM Trans. Graph.*, 1(3):215–234, 1982.
111. K. I. Kim and Y. Kwon. Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(6):1127–1133, 2010.
112. W. Kinsner. A Unified Approach To Fractal Dimensions. *IJCINI*, 1(4):26–46, 2007.
113. W. Kiser, C. Musial, and P. Sen. Accelerating active contour algorithms with the Gradient Diffusion Field. In *ICPR*, pages 1–4. IEEE, 2008.
114. Pushmeet Kohli and Philip H. S. Torr. Dynamic Graph Cuts for Efficient Inference in Markov Random Fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2079–2088, 2007.
115. K. Kpalma and J. Ronsin. Multiscale contour description for pattern recognition. *Pattern Recognition Letters*, 27(13):1545–1559, 2006.
116. E. Kraaikamp. Auto Stakkert. <http://www.autostakkert.com/wp/enhance>. Ultimo accesso 03/06/2014.

117. M. Kumar and Y. Gupta. Methods for solving singular boundary value problems using splines: a review. *Journal of Applied Mathematics and Computing*, pages 265–278, 2010.
118. M. Pawan Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1)*, pages 18–25. IEEE Computer Society, 2005.
119. Bai-Lian L. and C. Loehle. Wavelet analysis of multiscale permeabilities in the subsurface. *Geophysical Research Letters*, 22(23):3123–3126, 1995.
120. V. S. Lempitsky, A. Blake, and C. Rother. Image Segmentation by Branch-and-Mincut. In *ECCV (4)*, volume 5305 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2008.
121. X. Li and M. T. Orchard. New edge-directed interpolation. *IEEE Trans Image Process*, 10(10):1521–7, 2001.
122. X. Li and Z. Tian. Optimum cut-based clustering. *Signal Processing*, 87(11):2491–2502, November 2007.
123. J. Liu. Pattern Recognition: An overview. *International Journal of Computer Science and Network Security*, 6(6), 2006.
124. H. Lombaert, Y. Sun, L. Grady, and C. Xu. A Multilevel Banded Graph Cuts Method for Fast Image Segmentation. In *Tenth IEEE International Conference on Computer Vision*, pages 259–265. IEEE Computer Society, 2005.
125. A. Maalouf and M. Larabi. Image Super-Resolution, a State-of-the-Art Review and Evaluation. In *Advanced Color Image Processing and Analysis*, pages 181–218. Springer New York, 2013.
126. B. Mandelbrot. *The Fractal Geometry of Nature*. Freeman and Co., San Francisco, 1982.
127. R. Marfil, L. Molina-Tanco, A. Bandera, J. A. Rodríguez, and F. Sandoval. Pyramid segmentation algorithms revisited. *Pattern Recognition*, 39(8):1430–1451, 2006.
128. F. Marinello. *Atomic force microscopy in nanometrology: modeling and enhancement of the instrument*. PhD thesis, Università degli studi di Padova, 2006.
129. M. Martinelli et al. THESAURUS - Requisiti e specifiche funzionali del sistema per l'acquisizione e integrazione dati. THESAURUS - Tecniche per l'Esplorazione Sottomarina Archeologica mediante l'Utilizzo di Robot Autonomi in Sciame. Technical report, National Research Council - Institute of Information Science and Technologies, 2012.
130. R. Mc Namara, J. Mc Cormack, and N. P. Jouppi. Prefiltered Antialiased Lines Using Half-Plane Distance Functions. In *Proceedings of the ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware*, 2000.
131. Editors O. McGraw-Hill, editor. *The McGraw-Hill Dictionary of Scientific and Technical Terms, Seventh Edition (McGraw Hill Dictionary of Scientific and Technical Terms)*. McGraw-Hill Professional, 7 edition, October 2009.
132. Facundo Mévoli. Some Properties of Gromov-Hausdorff Distances. *Discrete & Computational Geometry*, 48(2):416–440, 2012.
133. Merigó José M. and Casanovas Montserrat. A New Minkowski Distance Based on Induced Aggregation Operators. *International Journal of Computational Intelligence Systems*, 4(2):123–133, 2011. doi: 10.1080/18756891.2011.9727769.
134. P. A. V. Miranda and A. X. Falcão. Links Between Image Segmentation Based on Optimum-Path Forest and Minimum Cut in Graph. *Journal of Mathematical Imaging and Vision*, 35(2):128–142, 2009.
135. K. Moore. What is a Point Spread Function? Technical report, ZEMAX, 2005.
136. H. Moravec. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, 1980.
137. E. N. Mortensen and W. A. Barrett. Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing*, 60(5):349–384, 1998.
138. T. N. Mundhenk. *Computational modeling and utilization of attention, surprise and attention gating*. PhD thesis, University of Southern California, 2009.
139. D. D. Muresan and T. W. Parks. Prediction of Image Detail. In *ICIP*, pages 323–326, 2000.

140. M. N. Murty and V. S. Devi. *Pattern Recognition - An Algorithmic Approach*. Undergraduate Topics in Computer Science. Springer, 2011.
141. M. S. Nikulin. Hellinger distance, 2001.
142. A. Noma, A. B. V. Graciano, L. A. Consularo, R. M. Cesar-Jr, and I. Bloch. A New Algorithm for Interactive Structural Image Segmentation. *CoRR*, abs/0805.1854, 2008.
143. N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.
144. H. K. Park and M. J. Chung. External force of snake: virtual electric field. *Electronics Letters*, 38(24):1500–1502, 2002.
145. S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, 20(3):21–36, May 2003.
146. V. S. Pedro, H. Hoppe, J. Snyder, and S. J. Gortler. Discontinuity edge overdraw. In *SI3D*, pages 167–174. ACM, 2001.
147. S. Pelletier. *Acceleration methods for image super-resolution*. PhD thesis, Department of Electrical & Computer Engineering, 2009.
148. W. K. Pratt. *Digital image processing: PIKS Inside*. Wiley, 4th edition edition, 2007.
149. A. Protiere and G. Sapiro. Interactive Image Segmentation via Adaptive Weighted Distances. *IEEE Transactions on Image Processing*, 16(4):1046–1057, 2007.
150. S. S. Qureshi, X. M. Li, and T. Ahmad. Investigating image super resolution techniques: What to choose? In *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, pages 642–647, Feb 2012.
151. M. Raposo. A Guide for Atomic Force Microscopy Analysis of Soft- Condensed Matter . In *Modern Research and Educational Topics in Microscopy* , 2007.
152. M. Reggiannini et al. ARROWS - Development of an interactive mixed reality environment for virtual exploration of archaeological sites. Technical report, ISTI - CNR Pisa, 2014.
153. A. Renyi. On a new axiomatic theory of probability. *Acta Mathematica Hungarica*, 6(3):285–335, September 1955.
154. M. R. Rezaee, P. M. J. van der Zwet, B. P. F. Lelieveldt, R. J. van der Geest, and J. H. C. Reiber. A multiresolution image segmentation technique based on pyramidal segmentation and fuzzy clustering. *IEEE Transactions on Image Processing*, 9(7):1238–1248, 2000.
155. M. Righi. PRIAR - Pattern Recognition Image Augmented Resolution a tool for image analysis. Technical report, National Research Council - Institute of Information Science and Technologies, 2014.
156. M. Righi, D'Acunto M., and Salvetti O. PRIAR (Pattern Recognition Image Augmented Resolution) – a tool to combine pattern-recognition with super-resolution. In *9th International Conference on Open German-Russian Workshop on PATTERN RECOGNITION and IMAGE UNDERSTANDING, OGRW2014*, 2014.
157. S. Roberts and R. Everson, editors. *Independent Component Analysis: Principles and Practice*, volume 19, pages 459–462. Cambridge University Press, 2001.
158. L. M. Rocha. Data clustering as an optimum-path forest problem with applications in image analysis. *Int. J. Imaging Systems and Technology*, 19(2):50–68, 2009.
159. J. B. Roerdink and A. Meijster. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae*, 31(1):187–228, 2000.
160. Sudeep Sarkar and Padmanabhan Soundararajan. Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(5):504–525, 2000.
161. S. C. Saunders, J. Chenb, T. D. Drummerc, E. J. Gustafsond, and K. D. Brosofskea. Identifying scales of pattern in ecological data: a comparison of lacunarity, spectral and wavelet analyses. *Ecological Complexity*, 2(1):87–105, 2005.
162. F. R. Schmidt, E. Toppe, and D. Cremers. Efficient planar graph cuts with applications in Computer Vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 351–356. IEEE, 2009.

163. M. Ó Searcóid. *Metric spaces*. Springer, 2006.
164. S. Settis et al. Thesaurus: un database per il patrimonio culturale sommerso (in press). *Archeologia e Calcolatori*, 25, 2015.
165. C. E. Shannon. The mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.
166. Sharon Eitan, Galun Meirav, Sharon Dahlia, Basri Ronen, and Brandt Achi. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810–813, aug 2006. 10.1038/nature04977.
167. J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.
168. H. Simson. Causeway Security Solutions. <http://www.causewaysecuritysolutions.com/resolution-calculated-tv-lines.html>. Ultimo accesso 03/06/2014.
169. A. Singh, D. Terzopoulos, and D. B. Goldgof. Deformable models in medical image analysis. *IEEE Computer Society Press*, 1998.
170. W. Siu and K. Hung. Review of image interpolation and super-resolution. In *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–10, Dec 2012.
171. J. Smisek, M. Jancosek, and T. Pajdla. 3D with Kinect. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011.
172. W. J. Smith. *Modern Optical Engineering*. McGraw-Hill, 4th edition edition, 2007.
173. M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis and and machine vision (3. ed.)*. Thomson, 2008.
174. H. Stark and P. Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. *J Opt Soc Am A*, 6(11):1715–26, November 1989.
175. K. W. Sum and P. Y. S. Cheung. Boundary vector field for parametric active contours. *Pattern Recognition*, 40(6):1635–1645, 2007.
176. M. F. Tappen and W. T. Freeman. Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters. In *ICCV*, pages 900–907. IEEE Computer Society, 2003.
177. D. Terzopoulos. Deformable Models: Classic, Topology-Adaptive and Generalized Formulations. In S. Osher and N. Paragios, editors, *Geometric Level Set Methods in Imaging, Vision, and Graphics*, chapter 2, pages 21–40. Springer-Verla, New Yor, 2003.
178. S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Fourth Edition*. Academic Press, 2009.
179. P. Thévenaz, T. Blu, and M. Unser. Image Interpolation and Resampling. In I.N. Bankman, editor, *Handbook of Medical Imaging, Processing and Analysis*, , chapter 25, pages 393–420. Academic Press, San Diego CA, USA, 2000.
180. A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed problems*. Winston, 1977.
181. M. Trimeche. *Super-Resolution Image Reconstruction Using Non-Linear Filtering Techniques*. PhD thesis, Tampere University of Technology, Finland, 2006.
182. M. Turner et al. *Landscape Ecology in Theory and Practice: Pattern and Process*. Springer, April 2001.
183. M. G. Turner, R. H. Gardner, and R. V. O'Neill. Signal matching through scale space. *International Journal of Computer Vision*, 1(2):133–144, 1987.
184. F. van der Heijden, R. Duin, D. de Ridder, and D. M. J. Tax. *Classification, parameter estimation and state estimation - an engineering approach using Matlab*. John Wiley & Sons, Chichester, 2004.
185. S. J. van der Walt. *Super-resolution Imaging*. PhD thesis, Engineering, 2010.
186. M. Venguerov and P. Cunningham. *Generalised Syntactic Pattern Recognition as a Unifying Approach in Image Analysis*, volume 1451, pages 913–920. Springer Verlag, 1998.
187. L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

188. C. Wang, Q. Yang, M. Chen, X. Tang, and Z. Ye. Progressive cut. In *ACM Multimedia*, pages 251–260. ACM, 2006.
189. Q. Wang and R. Ward. A new edge-directed image expansion scheme. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 899–902 vol.3, 2001.
190. S. Wang and J. M. Siskind. Image Segmentation with Minimum Mean Cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):675–690, 2001.
191. T. Wang, I. Cheng, and A. Basu. Fluid Vector Flow and Applications in Brain Tumor Segmentation. *IEEE Trans. Biomed. Engineering*, 56(3):781–789, 2009.
192. M. Wertheimer. *Laws of organization in perceptual forms*. Harcourt, Brace & Jovanovitch, London, 1938.
193. P. West and N. Starostina. A guide to AFM Image Artifacts. Technical report, Pacific Nanotechnology, 2010.
194. A. P. Witkin. Scale-Space Filtering. In A. Bundy, editor, *IJCAI*, pages 1019–1022. William Kaufmann, 1983.
195. Z. Wu and R. Leahy. An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 2002.
196. Y. Xia, D. D. Feng, and R. Zhao. Morphology-based multifractal estimation for texture segmentation. *Image Processing, IEEE Transactions on*, 15(3):614–623, March 2006.
197. C. Xu and J. L. Prince. Gradient Vector Flow: A New External Force for Snakes. In *CVPR*, pages 66–71. IEEE Computer Society, 1997.
198. C. Xu and J. L. Prince. Generalized gradient vector flow external forces for active contours. *Signal Processing*, 71(2):131–139, 1998.
199. C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.
200. L. Xu, W. Li, and D. Shuurmans. Fast normalized cut with linear constraints. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2866–2873, 2009.
201. N. Xu, N. Ahujab, and R. Bansalc. Object Segmentation Using Graph Cuts Based Active Contours. *Computer Vision and Image Understanding*, 107(3):210–224, 2007.
202. N. Xu, R. Bansal, and N. Ahuja. Object segmentation using graph cuts based active contours. In *Computer Vision and Pattern Recognition Conference*, pages 46–53, 2003.
203. Y. Xu, V. Olman, and E. C. Uberbacher. A segmentation algorithm for noisy images. In *IEEE International Joint Symposia on Intelligence and Systems*, pages 220–226, 1996.
204. Ying Xu and Edward C. Uberbacher. 2D image segmentation using minimum spanning trees. *Image and Vision Computing*, 15(1):47–57, 1997.
205. H. Yang and Y. Choe. Cell Tracking and Segmentation in Electron Microscopy Images Using Graph Cuts. In *ISBI*, pages 306–309. IEEE, 2009.
206. Q. Yang, X. Tang, C. Wang, Z. Ye, and M. Chen. Progressive Cut: An Image Cutout Algorithm that Models User Intentions. *IEEE MultiMedia*, 14(3):56–66, 2007.
207. S. X. Yu and J. Shi. Segmentation Given Partial Grouping Constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):173–183, 2004.
208. C. T. Zahn. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, C-20(1):68–86, 1971.
209. Y. Zeng, D. Samaras, W. Chen, and Q. Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in N-D images. *Computer Vision and Image Understanding*, 112(1):81–90, 2008.
210. M. Zhang and R. Alhajj. Improving the Graph-Based Image Segmentation Method. In *ICTAI*, pages 617–624. IEEE Computer Society, 2006.
211. Y. J. Zhang. *Advances in image and video segmentation*. IRM Press, 2006.
212. H. Zhilan, G. Wang, X. Lin, and H. Yan. Skin segmentation based on graph cuts. *Tsinghua Science and Technology*, 14(4):478–486, 2009.
213. X. Zhu. Semi-Supervised Learning Literature Survey, 2008.

214. W. W. W. Zou and P. C. Yuen. Very Low Resolution Face Recognition Problem. *IEEE Transactions on Image Processing*, 21(1):327–340, 2012.

Ringraziamenti

Le prime parole di questi ringraziamenti sono per le persone che in questi anni mi hanno sostenuto, nella vita, nello studio e nel lavoro, consigliandomi e incoraggiandomi.

In questi anni ho avuto la fortuna di poter lavorare in ambiente di ricerca molto stimolante dove ho potuto approfondire gli argomenti del mio dottorato. Il merito di tutto questo va al mio gruppo di lavoro e un ringraziamento particolare va ai miei tutor Ovidio Salvetti, Mario D'Acunto e Francesco Marcelloni.

A

Implementazione

Questa appendice mette in evidenza alcuni aspetti implementativi del metodo illustrato nel capitolo 4. La descrizione completa dell'implementazione del *Tool PRIAR* richiederebbe la trascrizione di oltre 8000 righe di codice e la relativa descrizione. Per questo motivo abbiamo scelto di mettere in evidenza solamente una parte dello stesso, in particolare qui viene descritta la funzione di esplorazione che è stata centrale per capire le reali capacità del metodo di pattern-reconition sviluppato e che ha permesso, in seguito, di realizzare immagini a risoluzione aumentata.

A.1 La funzione di esplorazione

La funzione di esplorazione crea una griglia M delle dimensioni dell'immagine e utilizza questa griglia come mappa. Su M i valori delle celle sono tutti inizializzati a `untouched_cell`. La cella dalla quale si inizia l'esplorazione (definita dalle coordinate `ex` e `ey`) prende il valore `pixel_of_the_set`, ovvero, il valore di tutte le celle che fanno parte dell'insieme.

L'algoritmo per ogni cella afferente all'insieme ricorda:

- quali sono le celle adiacenti sono state visitate e quali non lo sono;
- l'elenco delle celle afferenti la cui visita degli adiacenti non è ancora stata completata.

Una funzione di valutazione, che è variabile in base al problema da affrontare viene passata all'algoritmo in forma di matrici (`delta_map_definition`, `abs_map_definition`) e viene utilizzata per confrontare la distanza tra due pixel vicini. La distanza viene misurata con la norma 1. Essenzialmente le matrici rappresentano nello spazio una mappa di probabilità che è funzione dell'oggetto che stiamo di trovare. Se la distanza è inferiore a una certa soglia, il pixel vicino fa parte dell'insieme dei pixel afferenti all'insieme cercato e viene marcato con un `pixel_of_the_set`, diversamente viene marcato con un `untouched_cell`. Questo algoritmo di ricerca ha complessità algoritmica $O(n)$, dove n è il numero delle celle dell'immagine da analizzare e ha una occupazione di memoria $K \cdot n$

(la funzione è lineare e dipende dal numero delle celle occupate e dalla dimensione dei tipi usati, K è l'occupazione in byte di una variabile memorizzante il valore della cella).

La funzione permette inoltre di forzare un certo numero di pixel come afferenti all'insieme. Questo valore viene passato con la variabile `accepted_area_map`. La funzione accetta inoltre un'area vietata nella quale tutti i pixel sono marcati come non afferenti all'insieme cercato. Quest'area è risultata particolarmente utile su immagini da AFM che presentavano errori particolarmente rilevanti sulla cornice.

```
1 function image_pattern=explore(ey,ex,map_to_explore,forbidden_zone,  
   delta_map_definition, abs_map_definition,accepted_area_map)  
3  
4 if isequal(accepted_area_map,[])  
5     accepted_area_map= map_to_explore .* 0 ;  
6 end  
7  
8 if isequal(forbidden_zone,[])  
9     forbidden_zone= map_to_explore .* 0 ;  
10 end  
11  
12 white=255;  
13  
14 untouched_cell=255;  
15 pixel_of_the_set=128;  
16 pixel_out_of_the_set=0;  
17  
18 pmin.x=1;  
19 pmin.y=1;  
20 input_image_size=size(map_to_explore);  
21 pmax.x=input_image_size(2);  
22 pmax.y=input_image_size(1);  
23  
24 map_to_explore=double(map_to_explore);  
25 image_pattern=double(ones(size(map_to_explore)))*untouched_cell;  
26 next_near_to_check=uint8(ones(size(map_to_explore)));  
27 from_node=init_from_node(ey,ex,pmin,pmax);  
28  
29 near_point(8).x=0;  
30 near_point(8).y=0;  
31 insert_counter=1;  
32  
33 for dx=-1:1:1  
34     for dy=-1:1:1  
35         if dx~=0 || dy~=0  
36             near_point(insert_counter).x=dx;  
37             near_point(insert_counter).y=dy;  
38             insert_counter=insert_counter+1;
```

```

39     end
40     end
41 end

43 continue_exploration=true;
44 current.x=ex;
45 current.y=ey;

47 while continue_exploration
48     while (next_near_to_check(current.y,current.x)<=8)
49
50         if image_pattern(current.y,current.x)==untouched_cell
51             image_pattern(current.y,current.x)=pixel_of_the_set;
52         end

53
54         next.x=next_near_to_check(current.y,current.x).x
55             +current.x;
56         next.y=next_near_to_check(current.y,current.x).y
57             +current.y;

58
59         if next.y>=pmin.y && next.y<=pmax.y && next.x>=pmin.x &&
60             next.x<=pmax.x
61             if image_pattern(next.y,next.x)==untouched_cell
62                 if (
63                     abs(double(map_to_explore(current.y,current.x))
64                         -double(map_to_explore(next.y,next.x)))
65                     <=delta_map_definition(next.y,next.x)
66                         && abs(double(map_to_explore(ey,ex))
67                             -double(map_to_explore(next.y,next.x)))<=
68                             abs_map_definition(next.y,next.x)
69                         && forbidden_zone(next.y,next.x)~=1
70                     ) || (accepted_area_map(next.y,next.x) == 1)
71                     if image_pattern(next.y,next.x)==untouched_cell
72                         from_node(next.y,next.x).x=current.x;
73                         from_node(next.y,next.x).y=current.y;
74                         current.x=next.x;
75                         current.y=next.y;
76                     end
77                 else
78                     if image_pattern(next.y,next.x)==untouched_cell
79                         image_pattern(next.y,next.x)=
80                             pixel_out_of_the_set;
81                     end
82                     next_near_to_check(current.y,current.x)=
83                         next_near_to_check(current.y,current.x)+1;
84                 end
85             else

```

```

87         next_near_to_check(current.y,current.x)=
            next_near_to_check(current.y,current.x)+1;
            end
89     else
            next_near_to_check(current.y,current.x)=
91         next_near_to_check(current.y,current.x)+1;
            end
93
94     end
95     if next_near_to_check(current.y,current.x)>8
96         && from_node(current.y,current.x).x==0
97         && from_node(current.y,current.x).y==0
98         continue_exploration=false;
99     end
100    if continue_exploration
101    && next_near_to_check(current.y,current.x)>8
102
103        tmp.x=current.x;
104        tmp.y=current.y;
105        current.x=from_node(tmp.y,tmp.x).x;
106        current.y=from_node(tmp.y,tmp.x).y;
107    end
108 end
109 image_pattern=uint8(image_pattern);
110 end
111
112
113 function initialized_from_node=init_from_node(y,x,pmin,pmax)
114
115 for vx=pmax.x:-1:pmin.x
116     for vy=pmax.y:-1:pmin.y
117         initialized_from_node(vy,vx).x=-1;
118         initialized_from_node(vy,vx).y=-1;
119     end
120 end
121 initialized_from_node(y,x).x=0;
122 initialized_from_node(y,x).y=0;
123 end

```

A.2 Generazione della mappa di probabilità

La mappa di probabilità è generata in funzione della forma prevista dell'oggetto da tracciare. Per oggetti simili a filamenti, abbiamo sperimentalmente verificato sulle immagini campioni che l'ellissoide in figura A.1 fornisce buoni risultati. Dove la funzione ellissoide presenta un valore maggiore, è maggiore la probabilità di avere pixel adiacenti che appar-

tengono allo stesso insieme e quindi la funzione che decide se due pixel appartengono allo stesso insieme è maggiormente rilassata rispetto alle aree esterne.

```

1 function [hyperboloid_in_image_space]=
  generate_hyperboloid_in_image_space(props,image_space_dimensions)
3
  centroid_local.x=props.Centroid(1);
5  centroid_local.y=props.Centroid(2);
  x_axis=props.max_axis;
7  y_axis=props.min_axis;
  search_area=max(y_axis,x_axis);
9  estensione_area_calcolo=search_area/4;
  x_axis_search_area=round(search_area+estensione_area_calcolo);
11 y_axis_search_area=round(search_area+estensione_area_calcolo);
  if x_axis_search_area>image_space_dimensions.x
13     x_axis_search_area=image_space_dimensions.x;
  end
15
  if y_axis_search_area>image_space_dimensions.y
17     y_axis_search_area=image_space_dimensions.y;
  end
19  hyperboloid_in_image_space=zeros(image_space_dimensions.y,
    image_space_dimensions.x);
21  hyperboloid_rotated_size=hyperboloid_in_image_space;
  if isfield (props,'hyperboloid_height')
23     hyperboloid_height=props.hyperboloid_height;
  else
25     hyperboloid_height=1;
  end
27  for y_point=-y_axis_search_area:1:y_axis_search_area
    for x_point=-x_axis_search_area:1:x_axis_search_area
29     tmp_x=((x_point^2)/(x_axis^2));
    tmp_y=((y_point^2)/(y_axis^2));
31     v=hyperboloid_height*sqrt(tmp_x+tmp_y-1);
    if isreal (v)
33     punto.x=x_point;
    punto.y=y_point;
35     [new_punto]=
    round_coordinate(ruota_punti(-props.Orientation,punto));
37     new_punto.x=centroid_local.x+new_punto.x;
    new_punto.y=centroid_local.y+new_punto.y;
39     new_x=round(new_punto.x);
    new_y=round(new_punto.y);
41     if (new_x>0) && (new_y>0) && (new_x<=image_space_dimensions.x)
    && (new_y<=image_space_dimensions.y)
43     hyperboloid_in_image_space(new_y,new_x)=v;
    end
45  end

```

```

    end
47 end

49 image=hyperboloid_in_image_space;
clear hyperboloid_in_image_space;
51 hyperboloid_in_image_space.data_image=image;
white=255;
53 hyperboloid_in_image_space.saturated_image=uint8((image>0).*white);
hyperboloid_in_image_space.logical_image=
55 logical(hyperboloid_in_image_space.saturated_image);
end

57 function coordinate_nuove=ruota_punti(angolo,coordinate)
59 tmp=[cosd(angolo),-sind(angolo);
sind(angolo),cosd(angolo)]*[coordinate.x;coordinate.y];
61 coordinate_nuove.x=tmp(1);
coordinate_nuove.y=tmp(2);
63 end

65 function coordinate_nuove=round_coordinate(coordinate)
coordinate_nuove.x=round(coordinate.x);
67 coordinate_nuove.y=round(coordinate.y);
end

```

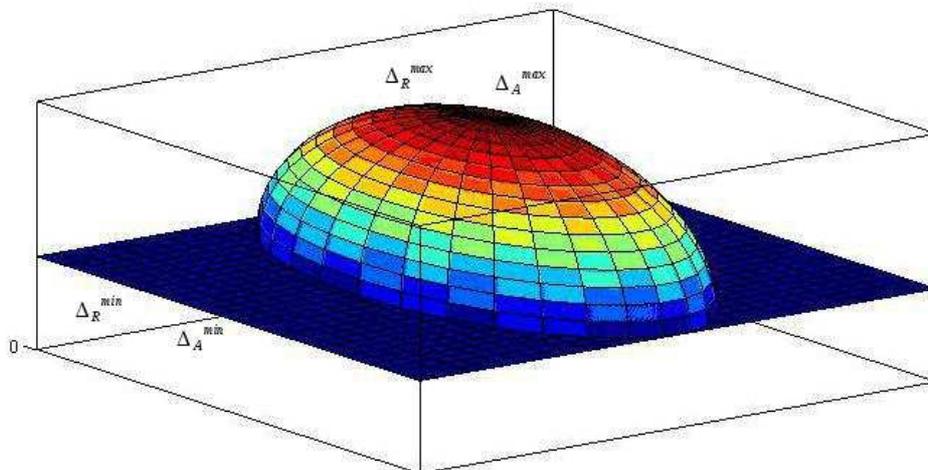


Figura A.1: mappa di probabilità