



SCUOLA DOTTORALE DI INGEGNERIA
INGEGNERIA DELL'ELETTRONICA BIOMEDICA,
DELL'ELETTROMAGNETISMO E DELLE TELECOMUNICAZIONI

XXIII CICLO

Motori di ricerca di immagini basati sulle funzioni armoniche circolari

Ing. Luca Costantini

A.A. 2010/2011

Docente Guida - Relatore: Prof. Alessandro Neri

Coordinatore: Prof. Lucio Vegni

Correlatore: Ing. Licia Capodiferro

Abstract

Il presente lavoro di tesi affronta il problema della gestione automatica di archivi di immagini. A tal fine sono stati sviluppati due sistemi, che pur avendo degli aspetti comuni, affrontano il problema da due punti di vista differenti. Gli aspetti comuni riguardano l'architettura generale in cui un utente sottopone la richiesta utilizzando un'immagine di esempio, successivamente il sistema classifica le immagini nell'archivio in base alla similitudine con questa di esempio. Un altro aspetto comune riguarda l'utilizzo delle funzioni armoniche circolari come strumento per l'analisi e la caratterizzazione delle immagini. Di tale classe di funzioni vengono utilizzate i polinomi di Zernike e le funzioni di Laguerre Gauss. La differenza principale degli approcci studiati e sviluppati risiede nel fatto che mentre il primo descrive l'immagine utilizzando un numero finito di caratterizzazioni locali, il secondo la rappresenta utilizzando delle caratteristiche globali. Infatti, nel primo sistema le funzioni armoniche circolari vengono utilizzate come descrittori di *pattern* 2D, sfruttandone le loro proprietà come, ad esempio, l'invarianza alla rotazione. Nel secondo sistema, invece, le funzioni armoniche circolari vengono utilizzate per analizzare le immagini al fine di descriverle utilizzando delle caratteristiche di basso livello, quali ad esempio le tessiture ed i bordi. In questo caso, è stato sviluppato un algoritmo che raggruppa in sottoinsiemi le immagini che potrebbero essere rilevanti. Si fa ciò per facilitare le operazioni di analisi dei risultati da parte dell'utente e per riassumere i contenuti presenti nelle immagini considerate rilevanti. I risultati ottenuti mostrano sia l'efficacia dei sistemi sviluppati, sia il fatto che le funzioni armoniche circolari sono degli ottimi strumenti per la descrizione di *pattern* 2D e per l'analisi delle immagini.

Alla mia famiglia ed ai miei amici.

Ringraziamenti

Il presente lavoro è stato svolto nell'ambito di una collaborazione tra Università Roma Tre e Fondazione Ugo Bordoni; desidero quindi ringraziare il Prof. Alessandro Neri e l'Ing. Licia Capodiferro per aver reso possibile tutto questo e per i preziosi insegnamenti.

Un ringraziamento speciale va all'area di Elaborazione dei Segnali Audio Video della Fondazione Ugo Bordoni per avermi accolto con gentilezza ed affetto fin dal primo giorno.

Desidero inoltre ringraziare il gruppo di Telecomunicazioni di Roma Tre ed in particolare l'Ing. Marco Carli per essermi stato vicino durante questi anni.

Infine, ringrazio tutti gli amici "bordoniani" con cui ho condiviso piacevoli momenti di pausa; gli amici di sempre e la mia famiglia per avermi sostenuto e sopportato.

Indice

Elenco delle figure	v
Elenco delle tabelle	x
Glossario	xi
1 Introduzione	1
2 Funzioni armoniche circolari	6
2.1 Introduzione	6
2.2 Polinomi di Zernike	7
2.2.1 Relazione con i polinomi di Taylor	12
2.3 Funzioni di Laguerre Gauss	15
3 Recupero di immagini basato su <i>pattern</i> 2D	21
3.1 Introduzione	21
3.2 Selezioni di punti salienti	23
3.3 Algoritmo del <i>QuadTree</i>	30
3.4 Mappa di verosimiglianza tra due immagini	32
3.5 Risultati sperimentali	38
3.5.1 Selezione dei punti salienti	38
3.5.2 Confronto di immagini basato sulla <i>ZMLM</i>	41

3.5.2.1	Complessità computazionale	46
4	Classificazione di immagini	50
4.1	Introduzione	50
4.2	Estrazione e confronto delle caratteristiche di basso livello	52
4.2.1	Le tessiture	52
4.2.2	Il colore	61
4.2.3	I bordi	62
4.3	Algoritmo per il raggruppamento delle immagini rilevanti	69
4.4	Valutazione dell'informazione di Fisher per l'estrazione delle caratteristiche di basso livello	74
4.5	Risultati sperimentali	79
4.5.1	Analisi e confronto di tessiture	79
4.5.2	Raggruppamento delle immagini rilevanti	85
5	Conclusioni	92
	Bibliografia	95

Elenco delle figure

1.1	Schema a blocchi di un sistema che utilizza una immagine di esempio per le operazioni di recupero.	3
2.1	Esempi dei polinomi di Zernike per $n = 2$	9
2.2	Andamento delle parte radiale, equazione 2.2, per i primi tre ordini.	10
2.3	Campionamento con una griglia quadrata di un cerchio, [18]. . . .	11
2.4	Immagine originale Barbara.	12
2.5	Immagine di figura 2.4 filtrata con i polinomi di Zernike per $n = 2$.	13
2.6	Andamento delle parte radiale delle funzioni di Laguerre Gauss fino al secondo ordine.	16
2.7	Esempi delle funzioni di Laguerre Gauss fino al secondo ordine. . .	17
2.8	Immagine di figura 2.4 filtrata con le funzioni di Laguerre Gauss fino al secondo ordine.	20
3.1	Versione originale dell'immagine Einstein.	28
3.2	Mappa associata a J_b , J_φ e J_a dell'immagine mostrata nella figura 3.1.	28
3.3	Primi 10 punti salienti selezionati sull'immagine Einstein.	29
3.4	Struttura del <i>QuadTree</i>	31
3.5	Esempio del funzionamento dell'algoritmo del <i>QuadTree</i>	31

3.6	Immagine originale 3.6(a), <i>pattern</i> posizionati in corrispondenza dei punti salienti 3.6(b), <i>pattern</i> ricostruiti 3.6(c). La grandezza dei <i>pattern</i> è scelta in accordo con il <i>QuadTree</i>	33
3.7	Immagini utilizzati per le sperimentazioni.	39
3.8	Andamento medio del <i>Repeatability Rate</i> , rispetto alla rotazione 3.8(a) e rispetto al fattore di scala 3.8(b), calcolato sui tre livelli di sfocatura con filtro gaussiano.	39
3.9	Andamento medio del <i>Repeatability Rate</i> , rispetto alla rotazione 3.9(a) e rispetto al fattore di scala 3.9(b), calcolato utilizzando la compressione JPEG con tre fattori di compressione diversi.	39
3.10	Andamento medio del <i>Repeatability Rate</i> , rispetto alla rotazione 3.10(a) e rispetto al fattore di scala 3.10(b), calcolato cambiando uniformemente l'illuminazione a tre diversi livelli.	40
3.11	Andamento medio del <i>Repeatability Rate</i> , rispetto alla rotazione 3.11(a) e rispetto al fattore di scala 3.11(b), calcolato degradando l'immagine con rumore gaussiano additivo a media nulla e con tre diverse varianze.	40
3.12	Andamento medio del <i>Repeatability Rate</i> , rispetto alla rotazione 3.12(a) e rispetto al fattore di scala 3.12(b), calcolato utilizzando tutte le degradazioni a tutti i livelli.	40
3.13	Esempi delle classi presenti nell'archivio <i>COREL-1000-A</i>	42
3.14	Esempi degli oggetti rappresentati nel database <i>COIL-100</i>	43
3.15	Grafico <i>Precision-Recall</i> ottenuto utilizzando il database <i>COIL</i>	45
4.1	Casi particolari di GGD che coincidono con la distribuzione di Gauss, $\beta = 2$, e con la distribuzione di Laplace, $\beta = 1$	54
4.2	Esempi di gaussiane generalizzate per diversi valori di α e β	55

4.3	Immagine <i>Bark.0000</i>	57
4.4	Parte reale, 4.4(a), e della parte immaginaria, 4.4(b), del piano <i>wavelet</i> di Laguerre Gauss $n = 1$ e $k = 0$ ad alta risoluzione, calcolato sull'immagine mostrata in figure 4.3.	58
4.5	Parte reale, 4.5(a), e della parte immaginaria, 4.5(b), del piano <i>wavelet</i> di Laguerre Gauss $n = 3$ e $k = 0$ ad alta risoluzione, calcolato sull'immagine mostrata in figure 4.3.	58
4.6	Densità marginali della parte reale, 4.6(a), e parte immaginaria, 4.6(b), del piano <i>wavelet</i> di Laguerre Gauss $n = 1$ e $k = 0$ ad alta risoluzione, calcolato sull'immagine mostrata in figure 4.3.	59
4.7	Densità marginali della parte reale, 4.7(a), e parte immaginaria, 4.7(b), del piano <i>wavelet</i> di Laguerre Gauss $n = 3$ e $k = 0$ ad alta risoluzione, calcolato sull'immagine mostrata in figure 4.3.	59
4.8	Versione originale dell'immagine Lena.	62
4.9	Componenti RGB, 4.9(a), 4.9(b), 4.9(c), dell'immagine mostrata nella figura 4.8.	63
4.10	Componenti YUV, 4.10(a), 4.10(b), 4.10(c), dell'immagine mostrata nella figura 4.8.	64
4.11	Versione originale dell'immagine Cameraman.	65
4.12	Mappa ottenuta applicando l'operatore di Laplace all'immagine mostrata in figura 4.11.	66
4.13	Esempio di punto di <i>zero-crossing</i>	66
4.14	Parte reale, 4.14(a), parte immaginaria, 4.14(b), e modulo, 4.14(c), dell'immagine mostrata in figura 4.11 filtrata con la funzione di Laguerre Gauss $n = 1$ e $k = 0$	67

4.15	Mappa in cui sono selezionati i punti utilizzati per caratterizzare l'informazione sui bordi presenti nell'immagine di figura 4.11.	68
4.16	Istogramma delle fasi dei bordi calcolato sull'immagine di figura 4.11.	68
4.17	Esempio della variazione di $\delta_{th}(n)$, equazione 4.31, al variare di λ e n^*	73
4.18	Immagine che appartiene alla classe dinosauri del database <i>COREL-1000-A</i>	75
4.19	Immagine che appartiene alla classe palazzi del database <i>COREL-1000-A</i>	75
4.20	Mappe $\varepsilon_{f_{x_1}}$ 4.20(a), $\varepsilon_{f_{x_2}}$ 4.20(b), $\varepsilon_{f_{x_1}f_{x_2}}$ 4.20(c) e J_b 4.20(d), calcolate sull'immagine 4.18.	76
4.21	Mappe $\varepsilon_{f_{x_1}}$ 4.21(a), $\varepsilon_{f_{x_2}}$ 4.21(b), $\varepsilon_{f_{x_1}f_{x_2}}$ 4.21(c) e J_b 4.21(d), calcolate sull'immagine 4.19.	77
4.22	Esempio della zona non considerata per l'estrazione delle caratteristiche, calcolata sull'immagine 4.18.	78
4.23	Esempio della zona non considerata per l'estrazione delle caratteristiche, calcolata sull'immagine 4.19.	78
4.24	Esempi delle classi di tessiture utilizzate nelle sperimentazioni.	79
4.25	Immagine originale di 512×512 pixel, 4.25(a), divisione in 16 immagini non sovrapposte di 128×128 pixel, 4.25(b).	80
4.26	Risultati ottenuti utilizzando la funzione di Laguerre Gauss $n = 1$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza di Kullback-Leibler.	80
4.27	Risultati ottenuti utilizzando la funzione di Laguerre Gauss $n = 1$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza euclidea.	81

4.28 Risultati ottenuti utilizzando le funzioni di Laguerre Gauss $n = 1$ e $k = 0$, $n = 3$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza di Kullback-Leibler.	82
4.29 Risultati ottenuti utilizzando le funzioni di Laguerre Gauss $n = 1$ e $k = 0$, $n = 3$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza euclidea.	82
4.30 Risultati ottenuti utilizzando l'informazione sul colore, le funzioni di Laguerre Gauss $n = 1$ e $k = 0$, $n = 3$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza di Kullback-Leibler.	84
4.31 Distribuzione dei valori di S , equazione 4.35, tra immagini che appartengono alla stessa classe e immagini che appartengono a classi diverse.	84
4.32 Percentuali di recupero di ogni classe.	89
4.33 Miglioramento delle percentuali di recupero di ogni classe.	89
4.34 Percentuali di recupero totale.	90
4.35 Risultati ottenuti utilizzando una parte del database <i>COREL-1000-A</i>	90

Elenco delle tabelle

2.1	Espressione dei polinomi di Taylor, fino al terzo ordine, espressi dall'equazione 2.10, in coordinate cartesiane.	14
2.2	Espressione dei polinomi di Zernike, equazione 2.11, in coordinate polari ed in coordinate cartesiane, fino al terzo ordine.	14
3.1	Risultati ottenuti utilizzando il database <i>COREL-1000-A</i>	44
3.2	Confronto con alcuni metodi presenti in letteratura.	44
4.1	Risultati ottenuti utilizzando alcune <i>wavelet</i> di Daubechies, [37].	83
4.2	Esempio delle immagini che vengono considerate rilevanti al primo passo dell'algoritmo descritto nel paragrafo 4.3, considerando come <i>query</i> l'immagine <i>Bark.0000.00</i>	86
4.3	Risultato dell'algoritmo descritto nel paragrafo 4.3, considerando come <i>query</i> l'immagine <i>Bark.0000.00</i>	87
4.4	Risultati ottenuti utilizzando la metrica definita nell'equazione 4.35.	87
4.5	Risultati ottenuti quando l'immagini viene caratterizzata utilizzando le tessiture, il colore ed i bordi.	88

Glossario

1D	Una dimensione
2D	Due dimensioni
3D	Tre dimensioni
CBIR	<i>Content Based Image Retrieval</i>
CHF	Funzioni armoniche circolari
CHW	<i>Wavelet</i> armoniche circolari
ED	Distanza euclidea
GGD	Distribuzione gaussiana generalizzata
JPEG	<i>Joint photographic experts group</i>
KLD	Distanza di Kullback-Leibler
RGB	Spazio di colore RGB
SVM	<i>Support Vector Machine</i>
YUV	Spazio di colore YUV
ZMLM	Mappa della verosimiglianza calcolata con i momenti di Zernike

1

Introduzione

In questi ultimi anni c'è stato un notevole incremento della diffusione di gadget e strumenti elettronici in grado di acquisire, visualizzare e diffondere immagini digitali. Questo fenomeno è dovuto, da un lato, alla diffusione dell'elettronica di consumo a basso costo dall'altro, all'accesso al web a larga banda disponibile ad un numero di persone sempre maggiore, sebbene il *digital divide* sia ancora un problema aperto. In questo scenario, in cui ognuno è in grado di creare oggetti come immagini e/o video ed è in grado di condividerli con gli altri in modo estremamente rapido, fa nascere l'esigenza di avere degli strumenti sempre più efficienti e veloci, che siano in grado di gestire questi oggetti. Ormai con il termine gestire non si intende più la creazione e l'archiviazione, ma si intende anche l'indicizzazione, l'aggregazione, la navigazione e il recupero di oggetti simili, anche dal punto di vista semantico. In questo modo gli utenti interessati ad un determinato argomento possono visualizzare tutte le immagini e/o i video inerenti l'argomento di interesse. Il metodo più semplice e che viene utilizzato dalla maggior parte degli strumenti di questo tipo, si basano sull'utilizzo di metadati creati, in parte, manualmente. Questi metadati, oltre a contenere una serie di informazioni quali, data di acquisizione, tipo di camera utilizzata per l'acquisizione, le impostazioni

utilizzate dalla fotocamera, contengono anche delle annotazioni che descrivono il contenuto di ciò che è rappresentato nell'immagine. È evidente che un sistema di questo tipo, pur permettendo l'implementazioni di tutte le operazioni di gestione necessarie, presenta alcuni aspetti negativi. Il limite principale di questo tipo di approccio è il fatto che le annotazioni devono essere scritte da un utente. Questo, inevitabilmente, richiede una enorme quantità di risorse; inoltre, le annotazioni sono soggettive e dipendono quindi dalla persona che le ha scritte. Considerato il numero di immagini che vengono create, non solo da utenti, ma anche da strumenti automatici, basti pensare alle immagini acquisite dai satelliti, o alla immagini acquisite dai sistemi di video sorveglianza, rende questi sistemi non adeguati. Infatti la soggettività delle annotazioni, fa sì che due immagini simili, potrebbero avere due annotazioni diverse, se annotate da due persone distinte. Inoltre la soggettività lascia spazio a comportamenti maliziosi, in cui le annotazioni non rappresentano il contenuto dell'immagine, ma vengono scritte con il fine di rendere il più possibile accessibile una determinata immagine. È quindi evidente la necessità di rendere i sistemi CBIR (*Content Based Image Retrieval*) oggettivi e completamente automatici. In questo modo vengono superati tutti i limiti dei sistemi basati sulle annotazioni. A tal fine vengono utilizzati degli algoritmi che caratterizzano le immagini in modo oggettivo. Vengono quindi utilizzate le caratteristiche di basso livello, quali ad esempio le tessiture, il colore ed i bordi, basandosi sull'assunzione che tali caratteristiche siano legate al contenuto semantico dell'immagine. Quelle menzionate sono caratteristiche globali, estratte cioè considerando l'intera immagine, ma possono essere usate anche caratteristiche locali. In questo modo le immagini vengono rappresentate mediante una serie di vettori che rappresentano le caratteristiche estratte. È importante notare che le caratteristiche estratte devono presentare delle invarianze per permettere

una caratterizzazione efficiente dell'immagine. Una volta calcolati i vettori che rappresentano le caratteristiche di basso livello, è necessario compiere un ulteriore passo per permettere di legarle ai concetti semantici, per colmare il cosiddetto gap semantico. A tal fine, possono essere seguite due diverse strategie. Da un lato il gap semantico può essere colmato con l'utilizzo di strumenti di *machine learning* [15], come ad esempio le SVM (*Support Vector Machine*) o le reti neurali. Questi strumenti, dopo una fase di addestramento, permettono di classificare una qualsiasi immagine. L'altro tipo di approccio, è quello basato su una immagine di esempio [4], in cui il sistema classifica le immagini in archivio in base alla similitudine con una immagine di esempio, detta *query*, fornita dall'utente. In questo caso non è necessaria una fase di addestramento, ma è necessaria la definizione di una metrica per confrontare i vettori associati alle caratteristiche di basso livello. Un sistema di questo tipo può essere schematizzato come mostrato nella figura 1.1. Nella figura 1.1, in rosso sono indicate le operazioni effettuate quando una

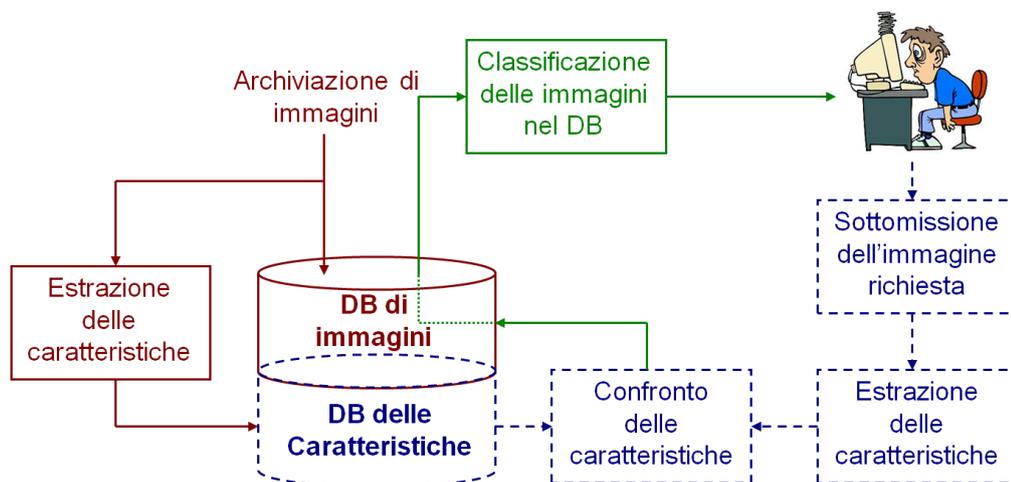


Figura 1.1: Schema a blocchi di un sistema che utilizza una immagine di esempio per le operazioni di recupero.

nuova immagine viene inserita nell'archivio, in particolare, a parte l'operazioni di archiviazione, vengono calcolati i vettori associati alle caratteristiche di bas-

so livello. In blu vengono indicate le operazioni che il sistema compie quando l'utente sottometta una nuova richiesta. In questa fase, l'operazione principale e più delicata, è quella in cui avviene il confronto dei vettori delle caratteristiche di basso livello. Infine, in verde è indicata l'operazione con cui il sistema mostra il risultato all'utente. Le immagini, una volta classificate, vengono mostrate all'utente in base alla similitudine, calcolata utilizzando le caratteristiche di basso livello, tra le immagini nell'archivio e l'immagine di esempio che ha fornito al sistema. Gli aspetti che si sono approfonditi in questo lavoro sono quelli che riguardano l'estrazione e il confronto delle caratteristiche. Si sono utilizzati due approcci diversi, il primo, ha l'obiettivo di ricercare un oggetto all'interno di un archivio, stimando la posizione e l'eventuale rotazione. Il secondo, basato sull'utilizzo di caratteristiche globali, ha l'obiettivo di recuperare tutte le immagini semanticamente rilevanti avendo a disposizione una immagine di esempio. Entrambi gli approcci si basano sull'utilizzo delle funzioni armoniche circolari, quali ad esempio i polinomi di Zernike e le funzioni di Laguerre Gauss. Mentre nel primo caso queste funzioni vengono utilizzate come descrittori, sfruttando le loro proprietà come ad esempio l'invarianza rispetto alla rotazione, nel secondo caso vengono utilizzate per l'estrazione delle caratteristiche di basso livello. La ricerca di un oggetto all'interno di un archivio è tanto più efficiente quanto più sono efficienti sia il metodo con cui esso viene descritto, che il metodo di confronto. La descrizione dell'oggetto avviene attraverso una descrizione locale delle zone che presentano un alto contenuto informativo. Inoltre è possibile scegliere la grandezza di queste zone in modo da ottenere sempre una descrizione accurata. Per effettuare il confronto e la stima della posizione e dell'eventuale rotazione, viene utilizzato l'operatore di verosimiglianza, calcolato utilizzando i coefficienti dello sviluppo fatto con le funzioni armoniche circolari. Invece, per quanto

riguarda le caratteristiche di basso livello, vengono utilizzate le tessiture, il colore ed i bordi. Anche in questo caso, la caratterizzazione delle tessiture e dei bordi avviene utilizzando le funzioni armoniche circolari. Una volta definita il metodo di confronto e di caratterizzazione dell'immagine, viene utilizzato un metodo per il raggruppamento delle immagini rilevanti in sottoinsiemi per permettere all'utente di avere una panoramica più ampia dei risultati. Infatti, mostrando all'utente solo il rappresentante di ogni sottoinsieme, nei primi risultati, quelli a cui mediamente l'utente presta la maggiore attenzione, sono rappresentate più immagini. Questo, da un lato, rende le prestazioni del sistema più robuste rispetto alla scelta dell'immagine di esempio, dall'altro, permette di dare importanza a quelle immagini che sono rilevanti, ma che non vengono classificate nelle primissime posizioni.

Il resto delle tesi è organizzato come segue. Nel capitolo 2 vengono descritte le funzioni armoniche circolari, in particolare i polinomi di Zernike e le funzioni di Laguerre Gauss. Il metodo per il ritrovamento di oggetti all'interno di un archivio basato sull'utilizzo dell'operatore di verosimiglianza è descritto nel capitolo 3. Mentre, nel capitolo 4, viene descritto il metodo basato sull'utilizzo delle caratteristiche di basso livello e l'algoritmo di *clustering* di immagini. Infine, nel capitolo 5 vengono tratte le conclusioni.

2

Funzioni armoniche circolari

2.1 Introduzione

Le funzioni armoniche circolari (CHF) sono una particolare classe di funzioni utilizzate in molte applicazioni riguardanti l'elaborazione delle immagini. In generale queste funzioni, definite in coordinate polari, compongono una base ortogonale. Queste funzioni sono composte da un profilo angolare armonico, da cui ne deriva una importante proprietà come l'invarianza rispetto alla rotazione, e da un profilo radiale. Il profilo radiale è composto da un polinomio, come ad esempio il polinomio di Laguerre o il polinomio di Zernike, moltiplicato per una finestra. Le funzioni armoniche circolari possono essere usate sia per la caratterizzazione di *pattern* bidimensionale che per effettuare l'analisi delle caratteristiche di basso livello di una immagine. Nel primo caso si ottengono dei descrittori di *pattern* bidimensionali che sono invarianti alla rotazione e con i quali è quindi possibile ottenere ottimi risultati in tutte quelle applicazioni che riguardano il recupero di immagini, o di oggetti, all'interno di un archivio. Nel secondo caso, invece, vengono usate per analizzare le immagini ed in particolare le caratteristiche di basso livello. Infatti, ogni funzione è in grado di estrarre una particolare caratteristica

come ad esempio le linee, gli incroci tra due linee, tra tre linee. Verranno utilizzate due particolari famiglie della classe delle funzioni armoniche circolari, che sono i polinomi di Zernike e le funzioni di Laguerre Gauss. Come detto in precedenza, queste due famiglie di funzioni si differenziano solo nella parte radiale e quindi mantengono entrambe le proprietà generali delle funzioni armoniche circolari. La diversa natura della parte radiale fa sì che manifestano comportamenti diversi che rendono una famiglia più adatta ad un certo tipo di applicazioni ed un'altra ad altri tipi di applicazioni. Nel seguito del capitolo verranno descritti prima i polinomi di Zernike e successivamente le funzioni di Laguerre Gauss.

2.2 Polinomi di Zernike

I polinomi di Zernike appartengono alla classe delle funzioni armoniche circolari e furono introdotti nel campo dell'analisi di immagini da Teague [36]. Grazie alle loro proprietà sono molto usati per la caratterizzazione di *pattern* 2D [8], per l'analisi delle immagini, ed in generale in tutti gli aspetti che riguardano l'elaborazione delle immagini [18, 31]. Come detto in precedenza le funzioni armoniche circolari, e quindi anche i polinomi di Zernike, sono composti da un profilo radiale e da un profilo angolare. Indicando con $\mathbf{x} = [x_1, x_2]$ le generiche coordinate cartesiane nel piano reale \mathfrak{R}^2 , il polinomio di Zernike V_{nm} è

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) e^{im\theta} \quad (2.1)$$

dove $\rho = \sqrt{x_1^2 + x_2^2}$, $\theta = \tan^{-1}(x_2/x_1)$ e R_{nm} rappresenta la parte radiale, definita come segue

$$R_{nm}(\rho) = \begin{cases} \sum_{s=0}^{(n-|m|)/2} \frac{(-1)^s (n-s)! \rho^{n-2s}}{s! \left(\frac{n+|m|}{2}-s\right)! \left(\frac{n-|m|}{2}-s\right)!} & \text{per } n - |m| \text{ pari} \\ 0 & \text{per } n - |m| \text{ dispari} \end{cases} \quad (2.2)$$

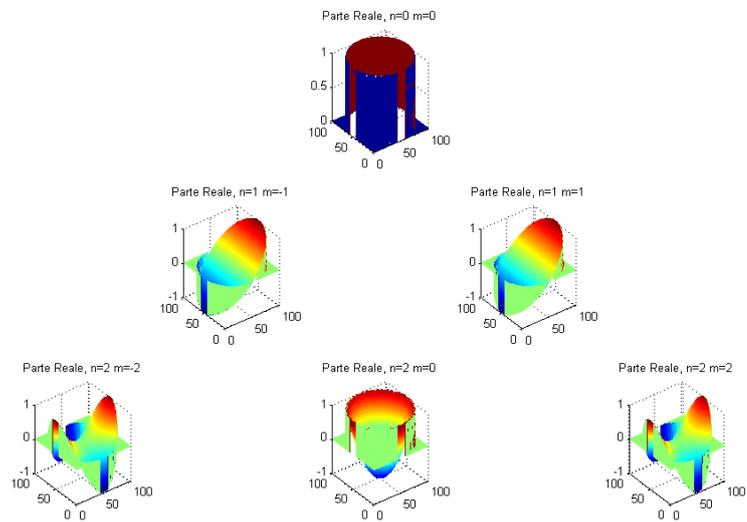
per $\rho \leq 1$, perciò la finestra che pesa il polinomio in questo caso è un disco unitario. Alcuni esempi di R_{nm} sono mostrati nella figura 2.2. Ogni polinomio di Zernike, come evidenziato nelle equazioni 2.1, 2.2 è definito da due indici. Il primo indice, n , è detto ordine ed è un intero non negativo, mentre, il secondo, m , è detto ripetizione ed è definito per $n - |m|$ pari e non negativo. Nella figura 2.1 sono mostrati degli esempi dei polinomi di Zernike fino al terzo ordine. Un esempio dei momenti di Zernike calcolati sull'immagine 2.4, sono mostrati nella figura 2.5, in particolare in 2.5(a) è mostrata la parte reale, mentre, in 2.5(b) è mostrata la parte immaginaria. I polinomi di Zernike compongono una base ortogonale per cui

$$\int_0^{2\pi} \int_0^1 V_{nm}^*(\rho, \theta) V_{n'm'}(\rho, \theta) \rho d\rho d\theta = \frac{\pi}{n+1} \delta_{nn'} \delta_{mm'} \quad (2.3)$$

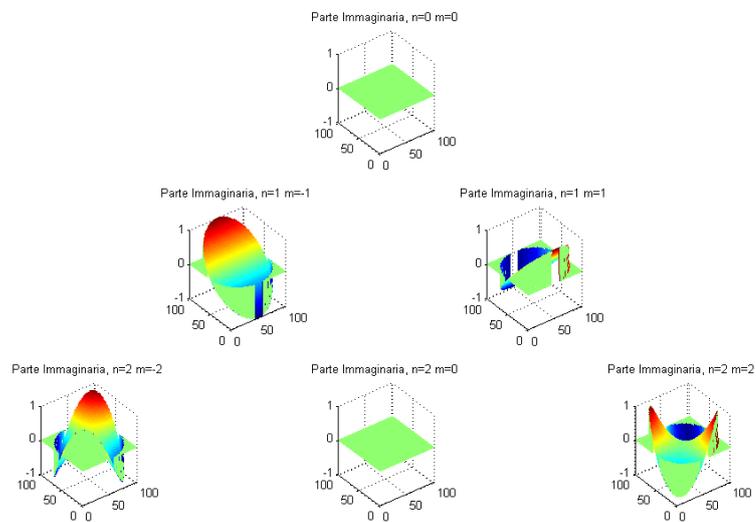
dove δ indica la delta di Kronecker. Anche la parte radiale, equazione 2.2, soddisfa la condizione di ortogonalità, quindi

$$\int_0^1 R_{nm}(\rho) R_{n'm}(\rho) \rho d\rho = \frac{1}{2(n+1)} \delta_{nn'} \quad (2.4)$$

dove, anche in questo caso, δ indica la delta di Kronecker. Se si indica con $f(\mathbf{x}) \in L^2(\mathfrak{R}^2, d^2\mathbf{x})$ una qualsiasi funzione bidimensionale, è possibile calcolare i



(a) Parte reale



(b) Parte immaginaria

Figura 2.1: Esempi dei polinomi di Zernike per $n = 2$.

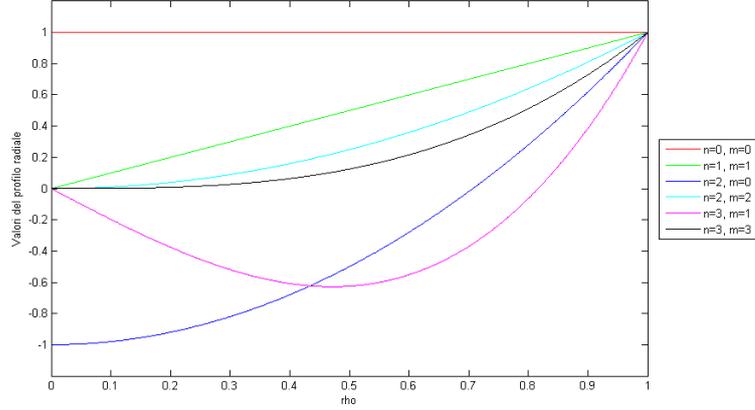


Figura 2.2: Andamento delle parte radiale, equazione 2.2, per i primi tre ordini.

momenti di Zernike come segue

$$A_{nm}(\mathbf{x}_0) = \frac{n+1}{\pi} \int \int_{\|\mathbf{x}-\mathbf{x}_0\| \leq 1} f(\mathbf{x}) V_{nm}^*(\mathbf{x}-\mathbf{x}_0) dx_1 dx_2 \quad (2.5)$$

dove \mathbf{x}_0 è il punto dove è centrato il disco unitario. Il calcolo dei momenti di Zernike in coordinate cartesiane introduce un errore di approssimazione geometrico. Infatti, la griglia di campionamento quadrata, non approssima bene il dominio in cui sono definiti i polinomi di Zernike come è mostrato nella figura 2.3. Dalla figura si può notare che specialmente nella parte periferica del cerchio la griglia di campionamento quadrata non segue bene la sua curvatura. Infatti, in alcuni casi il pixel quadrato copre una regione che è al di fuori del cerchio, mentre, in altri, delle regioni all'interno del cerchio non sono coperte da nessun pixel. Per minimizzare questo errore è necessario effettuare il calcolo in coordinate polari, piuttosto che in coordinate cartesiane, [18]. La funzione $f(\mathbf{x})$ può essere ricostruita a partire dai momenti di Zernike nel seguente modo

$$f(\mathbf{x}) = \sum_{n=0}^{+\infty} \sum_{m=-\infty}^{+\infty} A_{nm}(\mathbf{x}_0) V_{nm}(\mathbf{x}-\mathbf{x}_0); \quad (2.6)$$

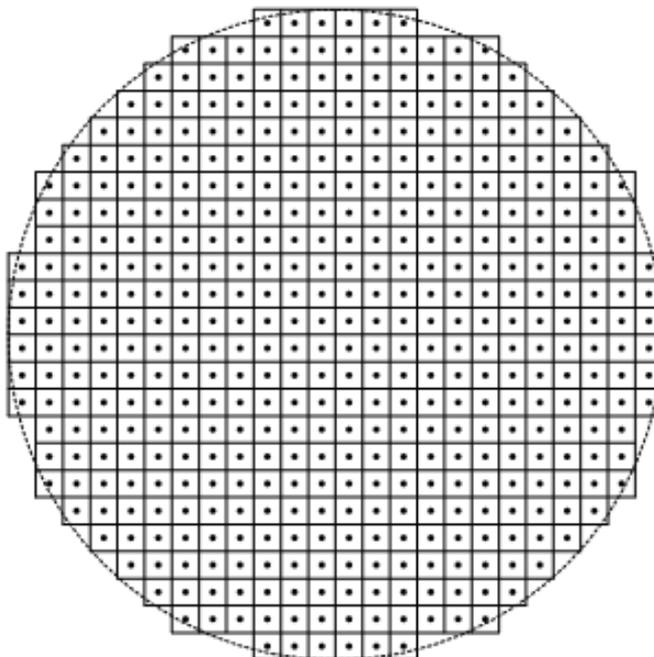


Figura 2.3: Campionamento con una griglia quadrata di un cerchio, [18].

tuttavia, nei casi pratici, non si hanno a disposizione infiniti momenti, ma solo un numero finito, per cui è possibile solo una ricostruzione approssimata dalla funzione $f(\mathbf{x})$

$$\hat{f}(\mathbf{x}) = \sum_{n=0}^N \sum_{m=-M}^{+M} A_{nm}(\mathbf{x}_0) V_{nm}(\mathbf{x} - \mathbf{x}_0). \quad (2.7)$$

In questo caso, l'errore di ricostruzione dipende sia dalla grandezza dell'immagine che dal numero di momenti utilizzati. Questo perchè se la zona da rappresentare è grande, questa conterrà anche molti dettagli che saranno ricostruiti soltanto se si utilizzano tanti momenti di Zernike, il che corrisponde ad utilizzare filtri di ordine elevato. D'altra parte però, non è possibile utilizzare filtri di ordine elevato con un supporto piccolo, perchè, in questo caso, questo potrebbe non essere in grado di seguire tutte le oscillazioni del filtro. Una delle importanti proprietà delle funzioni armoniche circolari è che, grazie al profilo radiale armonico, sono invarianti rispetto alla rotazione. Infatti, se si ruota di un angolo α la funzione

$f(\mathbf{x})$ si ottiene che

$$A_{nm}^{(\alpha)}(\mathbf{x}_0) = A_{nm}(\mathbf{x}_0) e^{-im\alpha}, \quad (2.8)$$

per cui

$$|A_{nm}^{(\alpha)}(\mathbf{x}_0)| = |A_{nm}(\mathbf{x}_0)|. \quad (2.9)$$

Proprio grazie a questa proprietà i polinomi di Zernike vengono applicati come descrittori di *pattern* 2D.



Figura 2.4: Immagine originale Barbara.

2.2.1 Relazione con i polinomi di Taylor

I polinomi di Zernike sono strettamente legati ai polinomi di Taylor, [25], che possono essere generalizzati come segue

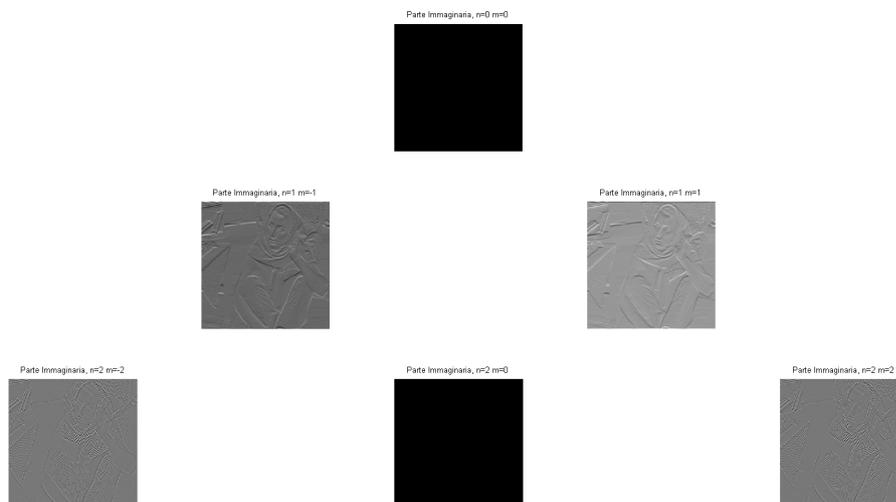
$$T_{pq}(\mathbf{x}) = x_1^{(p-q)} x_2^q. \quad (2.10)$$

Esprimendo l'equazione 2.1 come segue

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) e^{im\theta} = \begin{cases} R_{nm}(\rho) \cos(m\theta), & \text{per } m \geq 0 \\ R_{nm}(\rho) \sin(m\theta), & \text{per } m < 0 \end{cases} \quad (2.11)$$



(a) Parte reale



(b) Parte immaginaria

Figura 2.5: Immagine di figura 2.4 filtrata con i polinomi di Zernike per $n = 2$.

p	q	Coordinate cartesiane
0	0	1
1	0	x_1
1	1	x_2
2	0	x_1^2
2	1	x_1x_2
2	2	x_2^2
3	0	x_1^3
3	1	$x_1^2x_2$
3	2	$x_1x_2^2$
3	3	x_2^3

Tabella 2.1: Espressione dei polinomi di Taylor, fino al terzo ordine, espressi dall'equazione 2.10, in coordinate cartesiane.

si ottengono dei polinomi di Zernike che sono reali. Confrontando la terza colonna

n	m	Coordinate polari	Coordinate cartesiane
0	0	1	1
1	-1	$\rho \sin \theta$	x_1
1	1	$\rho \cos \theta$	x_2
2	-2	$\rho^2 \sin 2\theta$	$2x_1x_2$
2	0	$2\rho^2 - 1$	$-1 + 2x_1^2 + 2x_2^2$
2	2	$\rho^2 \cos 2\theta$	$-x_1^2 + x_2^2$
3	-3	$\rho^3 \sin 3\theta$	$-x_1^3 + 3x_1x_2^2$
3	-1	$(3\rho^3 - 2\rho) \sin \theta$	$-2x_1 + 3x_1^3 + 3x_1x_2^2$
3	1	$(3\rho^3 - 2\rho) \cos \theta$	$-2x_2 + 3x_2^3 + 3x_1^2x_2$
3	3	$\rho^3 \cos 3\theta$	$x_2^3 - 3x_1^2x_2$

Tabella 2.2: Espressione dei polinomi di Zernike, equazione 2.11, in coordinate polari ed in coordinate cartesiane, fino al terzo ordine.

della tabella 2.1 con la quarta colonna della tabella 2.2 si nota che i polinomi di Zernike espressi in forma reale, equazione 2.11, sono una combinazione lineare dei polinomi di Taylor, equazione 2.10. Perciò il passaggio dai i polinomi di Taylor ai polinomi di Zernike, avviene semplicemente attraverso l'utilizzo della seguente

matrice di trasformazione

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 \end{bmatrix}. \quad (2.12)$$

È da notare che la matrice mostrata nell'equazione 2.12 è valida solo per polinomi di ordine pari a tre.

2.3 Funzioni di Laguerre Gauss

Le funzioni di Laguerre Gauss, [17, 24], sono un'altra famiglia di funzioni che appartengono alla classe delle funzioni armoniche circolari, così come i polinomi di Zernike. Perciò anche queste funzioni sono composte da una parte radiale e da una parte armonica angolare. Mentre la parte angolare è la stessa sia nel caso dei polinomi di Zernike che nel caso della funzioni di Laguerre Gauss, la parte radiale è profondamente diversa. Infatti, come si vedrà in seguito, non solo è diverso il polinomio ma è diversa anche la finestra che lo pesa ed il relativo dominio in cui sono definite. L'espressione generale delle funzioni di Laguerre Gauss è la

seguinte

$$\mathcal{L}_{nk}(\rho, \theta) = (-1)^k 2^{(n+1)/2} \pi^{|n|/2} \left[\frac{k!}{(|n|+k)!} \right]^{1/2} \rho^{|n|} \ell_{nk}(2\pi\rho^2) e^{-\pi\rho^2} e^{in\theta} \quad (2.13)$$

dove ℓ_{nk} rappresenta il polinomio di Laguerre generalizzato espresso mediante la formula di Rodriguez

$$\ell_{nk}(t) = \frac{t^{-n} e^t}{k!} \frac{d^k}{dt^k} [t^{k+n} e^{-t}] = \sum_{h=0}^k (-1)^h \binom{n+k}{n-h} \frac{t^h}{h!}. \quad (2.14)$$

Dall'equazione 2.13 si può notare che, in questo caso, la parte radiale è composta

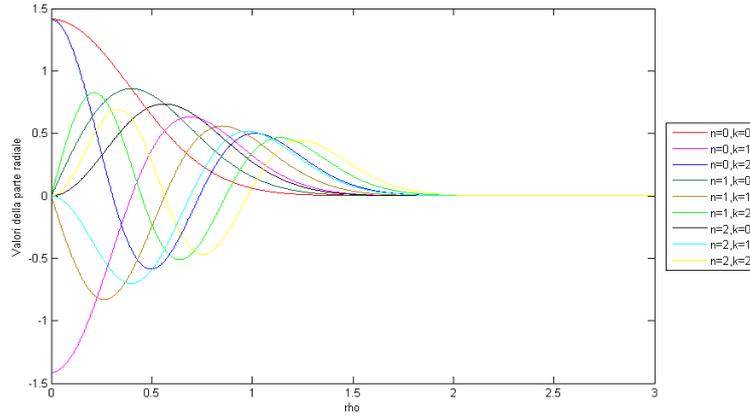
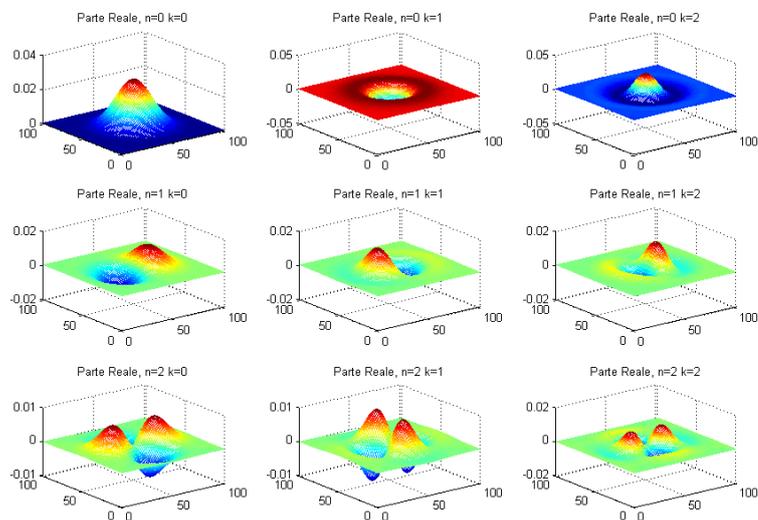
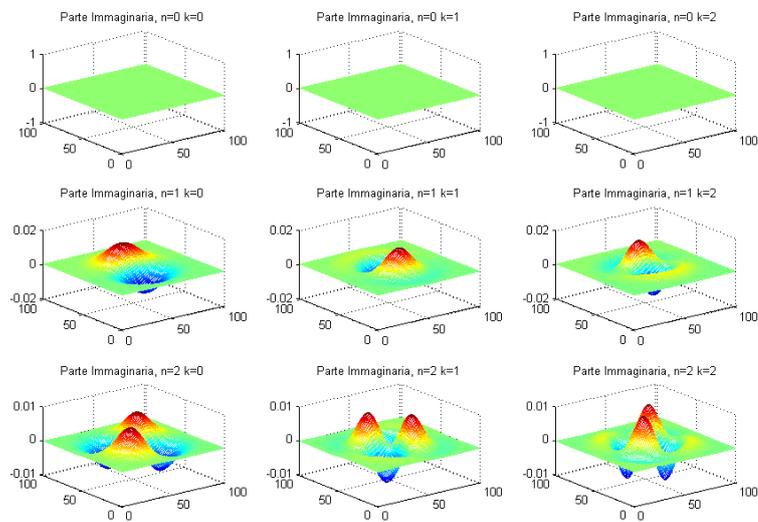


Figura 2.6: Andamento delle parte radiale delle funzioni di Laguerre Gauss fino al secondo ordine.

dal polinomio di Laguerre pesata con una finestra gaussiana. Gli andamenti della parte radiale, in funzione di ρ sono mostrati nella figura 2.6. Come mostrato dall'equazione 2.13, ogni funzione di Laguerre Gauss è identificata da due indici, n e k . In questo caso, a differenza degli indici che identificano i polinomi di Zernike, entrambi gli indici sono interi non negativi e non c'è nessun legame tra loro. Nella figura 2.7 sono mostrati alcuni esempi delle funzioni di Laguerre Gauss fino al secondo ordine. Un'immagine $f(\mathbf{x})$, pesata con una finestra gaussiana w_g ,



(a) Parte reale



(b) Parte immaginaria

Figura 2.7: Esempi delle funzioni di Laguerre Gauss fino al secondo ordine.

può essere rappresentata mediante i coefficienti dell'espansione di Laguerre Gauss come segue

$$f(\mathbf{x}) w_g \left(\frac{\mathbf{x} - \mathbf{x}_0}{\sigma} \right) = \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} D_{nk}(\mathbf{x}_0) \frac{1}{s} \mathcal{L}_{nk} \left(\frac{|\mathbf{x} - \mathbf{x}_0|}{\sigma}, \theta(\mathbf{x} - \mathbf{x}_0) \right) \quad (2.15)$$

dove i coefficienti dell'espansione sono

$$D_{nk}(\mathbf{x}_0) = \left\langle f(\mathbf{x}) w_g \left(\frac{\mathbf{x} - \mathbf{x}_0}{\sigma} \right), \frac{1}{s} \mathcal{L}_{nk} \left(\frac{|\mathbf{x} - \mathbf{x}_0|}{\sigma}, \theta(\mathbf{x} - \mathbf{x}_0) \right) \right\rangle. \quad (2.16)$$

Un esempio dei coefficienti di Laguerre Gauss, calcolati sull'immagine 2.4 è mostrato nella figura 2.8, dove la parte reale è mostrato nella figura 2.8(a), e la parte immaginaria nella figura 2.8(b). Se si ruota di un angolo α la funzione $f(\mathbf{x})$ si ha

$$D_{nk}^{(\alpha)}(\mathbf{x}_0) = D_{nk}(\mathbf{x}_0) e^{-in\alpha}, \quad (2.17)$$

quindi

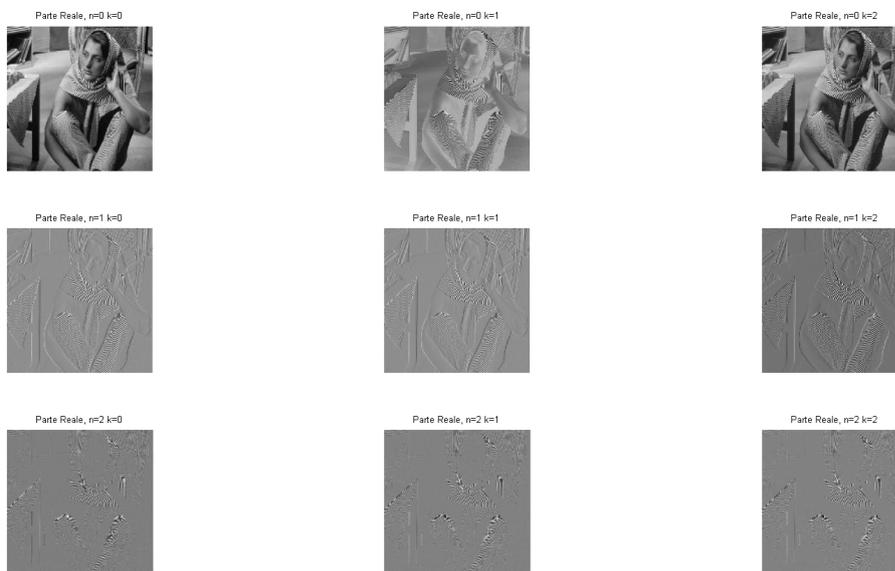
$$\left| D_{nk}^{(\alpha)}(\mathbf{x}_0) \right| = |D_{nk}(\mathbf{x}_0)|. \quad (2.18)$$

Anche in questo caso, queste funzioni, godono della proprietà di invarianza rispetto alla rotazione. Inoltre, ogni funzione di Laguerre Gauss genera una *wavelet* armonica circolare (CHW) diadica, [24]. Infatti una qualsiasi funzione di Laguerre Gauss soddisfa alla condizione di ammissibilità. Quindi una qualsiasi immagine $f(\mathbf{x})$ può essere rappresentata mediante la sua trasformata *wavelet* continua

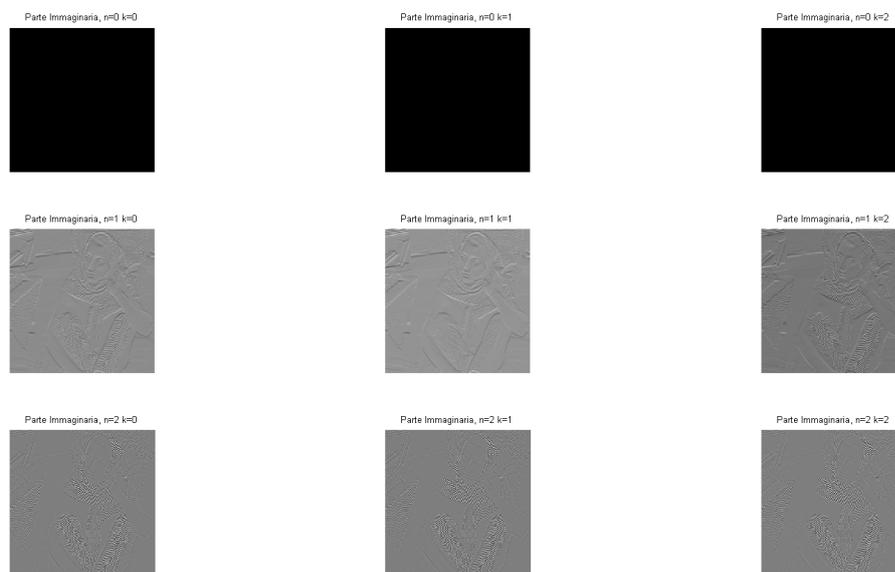
$$W_{\mathcal{L}_{nk}}[f](\mathbf{b}, \alpha, \sigma)$$

dove \mathbf{b} , α e σ rappresentano, rispettivamente, i parametri di traslazione, di rotazione e di scala della *wavelet* madre \mathcal{L}_{nk} . È possibile quindi effettuare un'analisi a multirisoluzione dell'immagine f ottenendo, in questo caso, un piano *wavelet*

complesso.



(a) Parte reale



(b) Parte immaginaria

Figura 2.8: Immagine di figura 2.4 filtrata con le funzioni di Laguerre Gauss fino al secondo ordine.

3

Recupero di immagini basato su *pattern 2D*

3.1 Introduzione

La nuova generazione di sistemi di indicizzazione, recupero e distribuzione di documenti multimediali, tra cui le immagini, richiedono l'utilizzo di algoritmi efficienti e il più possibile automatici. In questo modo si ottengono sistemi più efficaci e veloci rispetto a quelli basati sull'utilizzo di annotazioni scritte dagli utenti. Nasce quindi la necessità di avere degli strumenti in grado di riconoscere e verificare, se è presente un determinato oggetto in un archivio. Ovviamente, l'oggetto può essere ruotato o scalato, e questo rende ancora più difficile questo compito. I metodi classici che calcolano degli indici di similarità come la mappa di verosimiglianza, oppure la cross-correlazione normalizzata, tra un insieme finito di versioni, ruotate e scalate, dell'oggetto da riconoscere, non presentano buone prestazioni in termini di costo computazionale. È quindi evidente la necessità di utilizzare dei descrittori che presentano delle invarianze rispetto alla rotazione o alla scala, riducendo di molto la complessità computazionale, [21, 36]. Infat-

ti, utilizzando ad esempio dei descrittori invarianti alla rotazione, è sufficiente confrontare i descrittori per sapere, non solo se l'oggetto è presente o meno, ma anche se una sua versione ruotata è presente nell'immagine in esame. È evidente che se da un lato è necessario il calcolo dei descrittori, il numero dei confronti si riduce drasticamente portando un notevole miglioramento delle prestazioni. Dei semplici descrittori di forme bidimensionali possono essere, ad esempio, creati calcolando i momenti geometrici, [16, 34]. Una estensione di questa metodologie è il calcolo dei momenti di Zernike che presentano ottime prestazioni in termini di caratterizzazione delle forme, [8, 9, 23, 35]. Nel seguito del capitolo è descritto un metodo che permette il ritrovamento e la localizzazione di un oggetto all'interno di un archivio. Il metodo, si basa sull'utilizzo delle funzioni armoniche circolare [8, 9, 17, 30], e in particolare sull'utilizzo dei polinomi di Zernike, utilizzati per il calcolo dei momenti di Zernike. Nel precedente capitolo sono stati descritti gli aspetti matematici che riguardano questa particolare famiglia di funzioni. Il metodo descritto in questo capitolo può essere suddiviso in due passi principali. Nel primo passo l'oggetto, o l'immagine, da ricercare è rappresentata mediante l'utilizzo dei momenti di Zernike. Nel secondo passo si utilizza l'operatore di verosimiglianza come indice di similarità tra l'immagine da ricercare e le immagini presenti nell'archivio. I momenti di Zernike permettono di rappresentare l'area che cade all'interno del cerchio unitario, dominio in cui essi sono definiti. Per limitare il numero dei momenti di Zernike utilizzati per descrivere l'oggetto di interesse ad avere comunque una descrizione accurata, necessaria per ottenere buone prestazioni, l'oggetto viene diviso in zone la cui grandezza è scelta in base alla bontà della rappresentazione. Le zone da rappresentare sono scelte dopo una valutazione dell'informazione di Fisher rispetto al parametro di traslazione, di rotazione e di scale, [12, 25, 33]. In questo modo vengono utilizzate solo zone con

le quali è possibile caratterizzare in modo efficiente l'immagine. La grandezza dei polinomi di Zernike, e quindi delle zone descritte dai momenti di Zernike viene scelta con l'algoritmo del *QuadTree*. Questo algoritmo permette di selezionare la zona più grande che permette di avere un errore di ricostruzione inferiore ad una certa soglia. In questo modo, fissato il numero di momenti di Zernike che si vogliono utilizzare, si è certi che la zona di interesse è ben rappresentata. Una volta rappresentato l'oggetto o l'immagine da ricercare, è necessario calcolare la similarità tra questo e le immagini in archivio. A tal fine è stata calcolata la verosimiglianza, generalizzando il metodo descritto in [17] al caso dei polinomi di Zernike, [9]. In questo modo si ottiene un metodo veloce ed efficiente per ottenere una misura di similarità, ed anche la stima della posizione e della rotazione dell'oggetto. Le zone utilizzate per la rappresentazione dell'oggetto di interesse vengono analizzate uno dopo l'altra fino a che non interviene il criterio di arresto basato sull'energia del gradiente, [9]. Inoltre, al fine di limitare ulteriormente il costo computazionale, una volta che si ha a disposizione la prima stima della rotazione dell'immagine in archivio, si seleziona una zona di interesse per calcolare la verosimiglianza. Infatti se si conosce la posizione relativa tra due zone che rappresentano l'oggetto, la posizione di una prima zona sull'immagine dell'archivio che si sta analizzando e la rotazione di quest'ultima, è possibile stimare la regione dove cercare la seconda zona, [9].

3.2 Selezioni di punti salienti

Le selezioni di punti salienti che siano robusti rispetto alle possibili degradazioni e alle possibili alterazioni del segnale è molto importante in varie applicazioni che riguardano la *computer vision* e l'elaborazione delle immagini. Come si vedrà nei seguenti paragrafi, questi algoritmi saranno usati per selezionare i punti sui

quali centrare le finestre delle funzioni armoniche circolari. In questo modo, sfruttando la loro proprietà di caratterizzare *pattern* 2D, per ogni punto si avrà un vettore di coefficienti che rappresenta la zona dove è centrata la finestra della funzione circolare armonica. Così facendo, si otterranno un numero di vettori pari al numero di punti selezionati, che saranno utilizzati come rappresentazione dell'immagine. Per la selezione dei punti salienti, viene calcolata l'informazione di Fisher [12, 17] associata al vettore $\vartheta = [\mathbf{b}, \varphi, a]$ dove \mathbf{b} , φ e a rappresentano rispettivamente, il parametro di traslazione, di rotazione e di scala del *pattern*. L'informazione di Fisher, definita come

$$\mathbf{J}(\vartheta) = E \left\{ \nabla_{\vartheta} \ln p(f; \vartheta) [\nabla_{\vartheta} \ln p(f; \vartheta)]^T \right\}, \quad (3.1)$$

ed in forma matriciale è

$$\mathbf{J}(\vartheta) = \begin{bmatrix} J_b & J_{b,\varphi} & J_{b,a} \\ J_{b,\varphi}^T & J_{\varphi} & J_{\varphi,a} \\ J_{b,a}^T & J_{\varphi,a}^T & J_a \end{bmatrix}. \quad (3.2)$$

Per la selezione dei punti vengono utilizzate solo le componenti dell'informazione di Fisher, equazione 3.2, che sono sulla componente principale. La componente J_b è una matrice 2×2 , proporzionale alle energie delle derivate parziali, definita come segue

$$J_b = \begin{bmatrix} \varepsilon_{f_{x_1}} & \varepsilon_{f_{x_1}f_{x_2}} \\ \varepsilon_{f_{x_1}f_{x_2}} & \varepsilon_{f_{x_2}} \end{bmatrix} \quad (3.3)$$

dove

$$\varepsilon_{f_{x_1}} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left| \frac{\partial f(\mathbf{x})}{\partial x_1} \right|^2 |w(a\mathbf{x})|^2 dx_1 dx_2, \quad (3.4)$$

$$\varepsilon_{f_{x_2}} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left| \frac{\partial f(\mathbf{x})}{\partial x_2} \right|^2 |w(a\mathbf{x})|^2 dx_1 dx_2, \quad (3.5)$$

$$\varepsilon_{f_{x_1} f_{x_2}} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\partial f(\mathbf{x})}{\partial x_1} \frac{\partial f(\mathbf{x})}{\partial x_2} |w(a\mathbf{x})|^2 dx_1 dx_2. \quad (3.6)$$

Mentre, le altre due componenti della diagonale principale della matrice 3.3 sono

$$J_\varphi = a^2 [\mathcal{M}_{x_1, x_1}^{(0,2)} - 2\mathcal{M}_{x_1, x_2}^{(1,1)} + \mathcal{M}_{x_2, x_2}^{(2,0)}], \quad (3.7)$$

$$J_a = [\mathcal{M}_{x_1, x_1}^{(2,0)} + 2\mathcal{M}_{x_1, x_2}^{(1,1)} + \mathcal{M}_{x_2, x_2}^{(0,2)}], \quad (3.8)$$

dove i momenti delle derivate parziali sono

$$\mathcal{M}_{x_n, x_m}^{(h,k)} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_1^h x_2^k \frac{\partial f(\mathbf{x})}{\partial x_n} \frac{\partial f(\mathbf{x})}{\partial x_m} w(a\mathbf{x}) dx_1 dx_2. \quad (3.9)$$

Se si considera il disco unitario come finestra w e si sviluppa la funzione $f(\mathbf{x})$ in serie di Taylor, e quindi anche le sue derivate parziali, è possibile calcolare le componenti della diagonale principale dell'informazione di Fisher, come combinazione lineare dei momenti di Zernike, paragrafo 2.2.1, facilmente calcolabili con un banco di filtri. A tal fine lo sviluppo in serie di Taylor della funzione $f(\mathbf{x})$ è

$$\begin{aligned} f(\mathbf{x}) \cong & f(\mathbf{0}) + \frac{\partial}{\partial x_1} f(\mathbf{0}) x_1 + \frac{\partial}{\partial x_2} f(\mathbf{0}) x_2 + \frac{1}{2} \frac{\partial^2}{\partial x_1^2} f(\mathbf{0}) x_1^2 + \frac{1}{2} \frac{\partial^2}{\partial x_2^2} f(\mathbf{0}) x_2^2 + \\ & + \frac{\partial^2}{\partial x_1 \partial x_2} f(\mathbf{0}) x_1 x_2 + \frac{1}{6} \frac{\partial^3}{\partial x_1^3} f(\mathbf{0}) x_1^3 + \frac{1}{6} \frac{\partial^3}{\partial x_2^3} f(\mathbf{0}) x_2^3 + \frac{1}{2} \frac{\partial^3}{\partial x_1^2 \partial x_2} f(\mathbf{0}) x_1^2 x_2 + \\ & + \frac{1}{2} \frac{\partial^3}{\partial x_1 \partial x_2^2} f(\mathbf{0}) x_1 x_2^2 = B_{00} + B_{10} x_1 + B_{01} x_2 + B_{20} x_1^2 + B_{02} x_2^2 + \\ & + B_{11} x_1 x_2 + B_{30} x_1^3 + B_{03} x_2^3 + B_{21} x_1^2 x_2 + B_{12} x_1 x_2^2 \end{aligned} \quad (3.10)$$

dove con B si indicano i coefficienti dello sviluppo di Taylor. A questo punto le derivate parziali $\frac{\partial f(\mathbf{x})}{\partial x_1}$ e $\frac{\partial f(\mathbf{x})}{\partial x_2}$ possono essere calcolate utilizzando lo sviluppo

3.10. Sostituendo le derivate parziali di $f(\mathbf{x})$ così calcolate nelle equazioni 3.4, 3.5, 3.6, 3.9, si ottiene

$$\begin{aligned}
 \varepsilon_{f_{x_1}} &= \pi B_{10}B_{10} + \frac{\pi}{8}B_{10}B_{30} + \frac{\pi}{8}B_{10}B_{12} + \\
 &+ \frac{\pi}{4}B_{20}B_{20} + \frac{\pi}{4}B_{11}B_{11} + \frac{\pi}{8}B_{30}B_{10} + \\
 &+ \frac{\pi}{32}B_{30}B_{30} + \frac{\pi}{96}B_{30}B_{12} + \frac{\pi}{24}B_{21}B_{21} + \\
 &+ \frac{\pi}{8}B_{12}B_{12} + \frac{\pi}{96}B_{12}B_{30} + \frac{\pi}{32}B_{12}B_{12};
 \end{aligned} \tag{3.11}$$

$$\begin{aligned}
 \varepsilon_{f_{x_2}} &= \pi B_{01}B_{01} + \frac{\pi}{8}B_{10}B_{30} + \frac{\pi}{8}B_{01}B_{21} + \\
 &+ \frac{\pi}{4}B_{02}B_{02} + \frac{\pi}{4}B_{11}B_{11} + \frac{\pi}{8}B_{03}B_{01} + \\
 &+ \frac{\pi}{32}B_{03}B_{03} + \frac{\pi}{96}B_{03}B_{21} + \frac{\pi}{8}B_{01}B_{21} + \\
 &+ \frac{\pi}{96}B_{21}B_{03} + \frac{\pi}{32}B_{21}B_{21} + \frac{\pi}{24}B_{12}B_{12};
 \end{aligned} \tag{3.12}$$

$$\begin{aligned}
 \varepsilon_{f_{x_1}f_{x_2}} &= \pi B_{10}B_{01} + \frac{\pi}{8}B_{10}B_{03} + \frac{\pi}{8}B_{10}B_{21} + \\
 &+ \frac{\pi}{4}B_{20}B_{11} + \frac{\pi}{4}B_{11}B_{02} + \frac{\pi}{8}B_{03}B_{01} + \\
 &+ \frac{\pi}{96}B_{30}B_{03} + \frac{\pi}{32}B_{30}B_{21} + \frac{\pi}{24}B_{21}B_{12} + \\
 &+ \frac{\pi}{8}B_{12}B_{01} + \frac{\pi}{32}B_{12}B_{03} + \frac{\pi}{96}B_{12}B_{21};
 \end{aligned} \tag{3.13}$$

$$\begin{aligned}
 \mathcal{M}_{x_1, x_1}^{(0,2)} &= \frac{\pi}{4}B_{10}B_{10} + \frac{\pi}{48}B_{10}B_{30} + \frac{\pi}{24}B_{20}B_{20} + \\
 &+ \frac{\pi}{8}B_{11}B_{11} + \frac{\pi}{48}B_{30}B_{10} + \frac{\pi}{256}B_{30}B_{30} + \\
 &+ \frac{\pi}{256}B_{30}B_{12} + \frac{\pi}{64}B_{21}B_{21} + \frac{\pi}{16}B_{12}B_{10} + \\
 &+ \frac{\pi}{256}B_{12}B_{30} + \frac{5\pi}{256}B_{12}B_{12} + \frac{\pi}{16}B_{10}B_{12};
 \end{aligned} \tag{3.14}$$

$$\begin{aligned}
 \mathcal{M}_{x_1, x_2}^{(1,1)} &= \frac{\pi}{24}B_{10}B_{12} + \frac{\pi}{24}B_{20}B_{02} + \frac{\pi}{24}B_{11}B_{11} + \\
 &+ \frac{\pi}{128}B_{30}B_{12} + \frac{\pi}{24}B_{21}B_{01} + \frac{\pi}{128}B_{21}B_{03} + \\
 &+ \frac{\pi}{128}B_{21}B_{21} + \frac{\pi}{128}B_{12}B_{12};
 \end{aligned} \tag{3.15}$$

$$\begin{aligned}
\mathcal{M}_{x_2, x_2}^{(2,0)} &= \frac{\pi}{4} B_{01} B_{01} + \frac{\pi}{48} B_{01} B_{03} + \frac{\pi}{16} B_{01} B_{21} + \\
&+ \frac{\pi}{24} B_{02} B_{02} + \frac{\pi}{8} B_{11} B_{11} + \frac{\pi}{48} B_{03} B_{01} + \\
&+ \frac{\pi}{256} B_{03} B_{03} + \frac{\pi}{256} B_{03} B_{21} + \frac{\pi}{16} B_{21} B_{01} + \\
&+ \frac{\pi}{256} B_{21} B_{03} + \frac{5\pi}{256} B_{21} B_{21} + \frac{\pi}{64} B_{12} B_{12};
\end{aligned} \tag{3.16}$$

$$\begin{aligned}
\mathcal{M}_{x_1, x_1}^{(2,0)} &= \frac{\pi}{4} B_{10} B_{10} + \frac{\pi}{16} B_{10} B_{30} + \frac{\pi}{48} B_{10} B_{12} + \\
&+ \frac{\pi}{8} B_{20} B_{20} + \frac{\pi}{24} B_{11} B_{11} + \frac{\pi}{16} B_{30} B_{10} + \\
&+ \frac{5\pi}{256} B_{30} B_{30} + \frac{\pi}{256} B_{30} B_{12} + \frac{\pi}{64} B_{21} B_{21} + \\
&+ \frac{\pi}{48} B_{12} B_{10} + \frac{\pi}{256} B_{12} B_{30} + \frac{\pi}{256} B_{12} B_{12};
\end{aligned} \tag{3.17}$$

$$\begin{aligned}
\mathcal{M}_{x_2, x_2}^{(0,2)} &= \frac{\pi}{4} B_{01} B_{01} + \frac{\pi}{16} B_{01} B_{03} + \frac{\pi}{48} B_{01} B_{21} + \\
&+ \frac{\pi}{8} B_{02} B_{02} + \frac{\pi}{24} B_{11} B_{11} + \frac{\pi}{16} B_{03} B_{01} + \\
&+ \frac{5\pi}{256} B_{03} B_{03} + \frac{\pi}{256} B_{03} B_{21} + \frac{\pi}{48} B_{21} B_{01} + \\
&+ \frac{\pi}{256} B_{21} B_{03} + \frac{\pi}{256} B_{21} B_{21} + \frac{\pi}{64} B_{12} B_{12}.
\end{aligned} \tag{3.18}$$

In questo modo le componenti principali della matrice di Fisher sono espresse come combinazione lineare dei coefficienti di Taylor. All'interno del cerchio unitario, i polinomi di Taylor possono essere espressi come una combinazione lineare dei polinomi di Zernike, paragrafo 2.2.1. Il legame che esiste tra queste due famiglie di polinomi vale anche per i rispettivi coefficienti B_{pq} e A_{nm} . Tramite la matrice di conversione 2.12 è possibile esprimere i coefficienti dello sviluppo di Taylor delle equazioni 3.10, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18 in funzione dei momenti di Zernike, esprimendo così l'informazione di Fisher in funzione dei momenti di Zernike. Poichè J_b è una matrice, la mappa associata a questa componente, sarà costituita dagli autovalori della matrice. Mentre, per le altre due componenti, J_φ e J_a le mappe sono costruite proprio con il valore di queste componenti. Nella figura 3.2 sono mostrate le tre mappe calcolate per l'immagine di figura 3.1. I punti salienti vengono selezionati come i massimi locali della

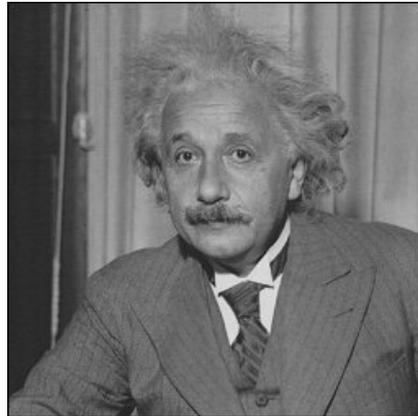


Figura 3.1: Versione originale dell'immagine Einstein.



Figura 3.2: Mappa associata a J_b , J_φ e J_a dell'immagine mostrata nella figura 3.1.

mappa associata a J_b che soddisfano le seguenti condizioni

$$J_\varphi(\mathbf{x}) > th_\varphi \max_{\mathbf{x}} [J_\varphi(\mathbf{x})], \quad (3.19)$$

$$J_a(\mathbf{x}) > th_a \max_{\mathbf{x}} [J_a(\mathbf{x})] \quad (3.20)$$

dove th_φ e th_a sono due soglie. In questo modo si selezionano i punti a cui è associata la massima informazione rispetto al parametro \mathbf{b} , che tiene conto della traslazione, al parametro φ , che tiene conto della rotazione, e, infine, al parametro a , che tiene conto della scala. Nella figura 3.3 è mostrato un esempio dei primi 10 punti selezionati sull'immagine mostrata nella figura 3.1. Come si può notare

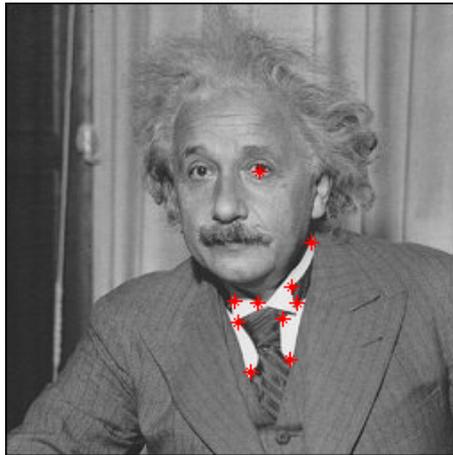


Figura 3.3: Primi 10 punti salienti selezionati sull'immagine Einstein.

l'algoritmo non seleziona i punti in quelle zone che sono difficili da riconoscere come le zone omogenee. L'algoritmo seleziona i punti nelle zone di vertice che sono quelle che contengono informazione rispetto alla traslazione, la rotazione e la scala. Infatti, se si considera una zona omogenea, che può essere ruotata, traslata o anche scalata, e si confronta con una sua versione originale, non è possibile stabilire che tipo di alterazione ha subito. Se invece, la zona è un vertice, è possibile individuare l'alterazione a cui è stata sottoposta. A titolo di esempio

si consideri una zona che rappresenta un vertice di un triangolo. Confrontandola con la versione originale, si nota immediatamente se è stato ruotato oppure no. Quindi calcolare l'informazione di Fisher per selezionare i punti salienti permette di selezionare le zone dell'immagine che contengono informazione rispetto al parametro di traslazione, di rotazione e di scala.

3.3 Algoritmo del *QuadTree*

Le immagini naturali sono generalmente composte da una serie di oggetti, più o meno complessi, che compongono la scena. Rappresentare ogni oggetto nella sua completezza attraverso l'uso delle funzioni armoniche circolari, richiederebbe di arrestare lo sviluppo ad un ordine molto elevato. Ciò significa un costo computazionale molto elevato. Per ridurre il costo computazionale, l'oggetto non viene rappresentato nella sua completezza, ma viene suddiviso in varie parti. Così facendo, si diminuisce la complessità di ogni regione che si vuole rappresentare, si può quindi arrestare lo sviluppo ad un ordine più basso riducendo il costo computazionale. A tal fine viene utilizzato l'algoritmo del *QuadTree*, [8, 9, 30]. Data una regione di interesse quadrata R , questo algoritmo la suddivide in quattro blocchi. Questa operazione viene ripetuta per ogni blocco, come mostrato dalla figura 3.4, fino a che non si verifica la seguente condizione

$$P(R_i) = \left\{ \left\| w_i \left(\frac{\mathbf{x} - \mathbf{x}_0}{\sigma} \right) [f(\mathbf{x}) - \hat{f}(\mathbf{x})] \right\|^2 < \gamma \right\}. \quad (3.21)$$

Come mostrato dall'equazione 3.21 l'algoritmo del *QuadTree* è pilotato dalla norma L^2 dell'errore di ricostruzione. Così facendo, per un dato ordine a cui si tronca lo sviluppo, si ha la certezza che ogni zona in cui è suddivisa la regione di interesse è ben rappresentata. Nei casi pratici come regione di interesse viene

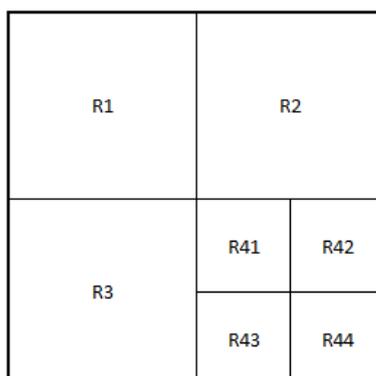


Figura 3.4: Struttura del *QuadTree*.

considerata l'intorno dei punti salienti e si applica una versione modificata dell'algoritmo del *QuadTree*, [8, 9, 12]. Infatti, se non è verificata la condizione 3.4, si prende una regione di interesse più piccola ma centrata sempre nel punto \mathbf{x}_0 . Se si utilizzano i polinomi di Zernike, poichè essi sono definiti all'interno del cerchio unitario, la zona che effettivamente viene analizzata è quella all'interno del cerchio circoscritto nel quadrato, la cui dimensione è selezionata dall'algoritmo del *QuadTree*, che seleziona la regione di interesse. Come mostrato nella figura 3.5,

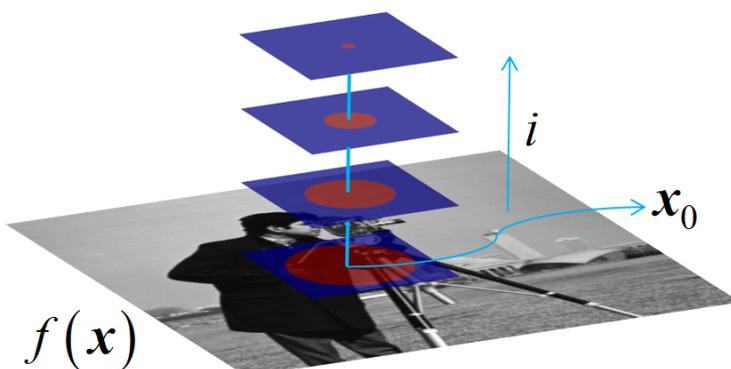


Figura 3.5: Esempio del funzionamento dell'algoritmo del *QuadTree*.

una volta selezionato il punto \mathbf{x}_0 viene verificata la condizione 3.21. Se l'errore di ricostruzione non è inferiore al valore γ , si seleziona una regione più piccola e si verifica nuovamente la condizione 3.21. Queste operazioni vengono ripetute fino a che, la condizione 3.21, non è verificata.

3.4 Mappa di verosimiglianza tra due immagini

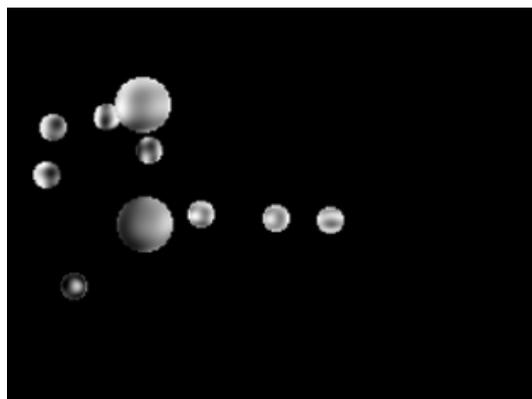
In questo paragrafo verrà descritto il metodo utilizzato per la ricerca di una immagine all'interno di un archivio, [9]. Il metodo è basato sull'utilizzo dei polinomi di Zernike per il calcolo della verosimiglianza tra l'immagine *query* e le immagini in archivio. I momenti di Zernike sono calcolati utilizzando l'algoritmo del *QuadTree*, descritto nel paragrafo 3.3, ottenendo così una descrizione dettagliata delle zone dell'immagine di interesse. Una volta calcolata la mappa della verosimiglianza questa viene usata sia per le operazioni di confronto che per effettuare una stima dell'eventuale rotazione relativa tra l'immagine *query* e l'immagine dell'archivio. La rappresentazione di una immagine è composta essenzialmente da due passi, nel primo passo vengono selezionati i punti salienti, cioè quei punti a cui corrispondono un intorno che ha una larghezza di banda spaziale ed angolare grande, [12, 25, 33]. Nella pratica la selezione dei punti salienti avviene utilizzando il metodo di Harris [33]. Nel secondo passo, gli intorni dei punti selezionati, vengono rappresentati utilizzando i polinomi di Zernike, in accordo con l'algoritmo del *QuadTree*. Nella figura 3.6 è mostrato un esempio in cui vengono selezionati, sull'immagine originale di figure 3.6(a), i punti salienti. La grandezza degli intorni dei punti salienti viene scelta in accordo con l'algoritmo del *QuadTree*. Come si può notare nella figura 3.6(b) la grandezze degli intorni è variabile. Le zone dell'immagine così selezionate rappresentano i *pattern* che vengono utilizzati per le procedure di ricerca. Invece nella figura 3.6(c) sono mostrate come vengono ricostruiti i *pattern* utilizzando i momenti di Zernike. Poiché la rotazione di un *pattern* corrisponde ad uno sfasamento lineare dei momenti di Zernike, proporzionale all'ordine del momento armonico angolare, come mostrato dall'equazione 2.8, il riconoscimento del *pattern* può essere fatto utilizzando una procedura iterativa quasi-newtoniana. Ottenendo così una stima a



(a)



(b)



(c)

Figura 3.6: Immagine originale 3.6(a), *pattern* posizionati in corrispondenza dei punti salienti 3.6(b), *pattern* ricostruiti 3.6(c). La grandezza dei *pattern* è scelta in accordo con il *QuadTree*.

3.4 Mappa di verosimiglianza tra due immagini

massima verosimiglianza della traslazione e della rotazione. Questa metodologia può essere utilizzata anche nel caso in cui si vogliono utilizzare le funzioni di Laguerre Gauss, [17, 30]. Se si indica con $f(\mathbf{x})$ l'immagine candidata a contenere un dettaglio di $g(\mathbf{x})$, in posizione \mathbf{b} e ruotato di un angolo φ , si ha

$$w[R_\varphi(\mathbf{x} - \mathbf{b})]f(\mathbf{x}) = w[R_\varphi(\mathbf{x} - \mathbf{b})]g[R_\varphi(\mathbf{x} - \mathbf{b})] + v(\mathbf{x}) \quad (3.22)$$

dove $w(\mathbf{x})$ è una generica finestra, $v(\mathbf{x})$ è il rumore di osservazione modellato come rumore gaussiano a media nulla e spettro di densità di potenza pari a $(N_0/4)$ e R_φ è la matrice di rotazione definita come segue

$$R_\varphi = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix}. \quad (3.23)$$

Massimizzando il logaritmo del funzionale $\Lambda[f(\mathbf{x}); \mathbf{b}, \varphi]$ si può effettuare la stima della posizione \mathbf{b} e della rotazione φ

$$\ln \Lambda[f(\mathbf{x}); \mathbf{b}, \varphi] = -\frac{2}{N_0} \int \int |w[R_\varphi(\mathbf{x} - \mathbf{b})]|^2 |f(\mathbf{x}) - g[R_\varphi(\mathbf{x} - \mathbf{b})]|^2 d\mathbf{x}. \quad (3.24)$$

La stima a massima verosimiglianza è una procedura complessa, perchè massimizzare la verosimiglianza rispetto alla posizione \mathbf{b} e alla rotazione φ implica una ricerca in tre dimensioni. Scegliendo come finestra $w(\mathbf{x})$ un disco di raggio σ , si possono utilizzare i momenti di Zernike per rappresentare $f(\mathbf{x})$ e $g(\mathbf{x})$. Quindi si ottiene:

$$w[R_\varphi(\mathbf{x} - \mathbf{b})]f(\mathbf{x}) = \sum_{n=0}^{\infty} \sum_{m=-\infty}^{+\infty} A_{nm}^f(\mathbf{x}_0) \frac{1}{\sigma} V_{nm} \left(\frac{\mathbf{x} - \mathbf{x}_0}{\sigma} \right) \quad (3.25)$$

3.4 Mappa di verosimiglianza tra due immagini

e

$$w(\mathbf{x})g(\mathbf{x}) = \sum_{n=0}^{\infty} \sum_{m=-\infty}^{+\infty} A_{nm}^g(\mathbf{x}_0) \frac{1}{\sigma} V_{nm} \left(\frac{\mathbf{x}}{\sigma} \right) \quad (3.26)$$

dove

$$A_{nm}^f(\mathbf{b}) = \frac{n+1}{\pi} \int_{\|\mathbf{x}-\mathbf{b}\| \leq \sigma} \int f(\mathbf{x}) \frac{1}{\sigma} V_{nm}^* \left(\frac{\mathbf{x}-\mathbf{b}}{\sigma} \right) dx_1 dx_2, \quad (3.27)$$

$$A_{nm}^g = \frac{n+1}{\pi} \int_{\|\mathbf{x}-\mathbf{b}\| \leq \sigma} \int g(\mathbf{x}) \frac{1}{\sigma} V_{nm}^* \left(\frac{\mathbf{x}}{\sigma} \right) dx_1 dx_2. \quad (3.28)$$

Sostituendo le equazioni 3.27 e 3.28 nell'equazione 3.24 si ottiene

$$\begin{aligned} \ln \Lambda[f(\mathbf{x}); \mathbf{b}, \varphi] &= -\frac{2}{N_0} \int_{\|\mathbf{x}-\mathbf{b}\| \leq \sigma} |f(\mathbf{x}) - g[R_\varphi(\mathbf{x}-\mathbf{b})]|^2 dx_1 dx_2 = \\ &= -\frac{2}{N_0} \sum_{n=0}^{\infty} \sum_{m=-\infty}^{+\infty} \frac{\pi}{(n+1)\sigma^2} |A_{nm}^f(\mathbf{b}) - A_{nm}^g e^{im\varphi}|^2 = \\ &= -\frac{2}{N_0} \sum_{n=0}^{\infty} \sum_{m=-\infty}^{\infty} \frac{\pi}{(n+1)\sigma^2} \left[|A_{nm}^f(\mathbf{b})|^2 + |A_{nm}^g|^2 \right] + \\ &+ \frac{4}{N_0} \sum_{n=0}^{\infty} \sum_{m=-\infty}^{\infty} \frac{\pi}{(n+1)\sigma^2} \operatorname{Re} \left\{ \sum_{n=0}^{\infty} [A_{nm}^f(\mathbf{b})]^* A_{nm}^g e^{im\varphi} \right\}. \end{aligned} \quad (3.29)$$

Nella pratica si utilizzano un numero finito di momenti di Zernike, troncando così lo sviluppo, quindi si ottiene che la mappa di verosimiglianza (*ZMLM*) è

$$ZMLM(\mathbf{b}) = \max_{\varphi} \left\{ -\frac{2}{N_0} \sum_{n=0}^N \sum_{m=-M}^{+M} \frac{\pi}{(n+1)\sigma^2} |A_{nm}^f(\mathbf{b}) - A_{nm}^g e^{im\varphi}|^2 \right\}. \quad (3.30)$$

Per ottenere un efficiente stimatore di $\varphi(\mathbf{b})$, si può scrivere

$$\begin{aligned} ZMLM(\mathbf{b}) &= -\frac{2}{N_0} \sum_{n=0}^{\infty} \sum_{m=-\infty}^{\infty} \frac{\pi}{(n+1)\sigma^2} \left[|A_{nm}^f(\mathbf{b})|^2 + |A_{nm}^g|^2 \right] + \\ &+ \max_{\varphi} \left\{ \frac{4\pi}{N_0\sigma^2} \sum_{m=-M}^M \mu_m(\mathbf{b}) \cos[m\varphi + \eta_m(\mathbf{b})] \right\} \end{aligned} \quad (3.31)$$

dove

$$\mu_m(\mathbf{b}) e^{i\eta_m(\mathbf{b})} = \frac{1}{(n+1)} \sum_{n=0}^N [A_{nm}^f(\mathbf{b})]^* A_{nm}^g. \quad (3.32)$$

3.4 Mappa di verosimiglianza tra due immagini

Per ogni posizione, il calcolo della rotazione $\hat{\varphi}(\mathbf{b})$ può essere fatto utilizzando la procedura iterativa di Newton-Raphson che, nel presente caso, si esplicita come segue

$$\hat{\phi}_{ML}^{(h)}(\mathbf{b}) = \hat{\phi}_{ML}^{(h-1)}(\mathbf{b}) - \frac{\sum_{m=-M}^M m \mu_m(\mathbf{b}) \sin \left[m \hat{\phi}_{ML}^{(h-1)}(\mathbf{b}) + \eta_m(\mathbf{b}) \right]}{\sum_{m=-M}^M m^2 \mu_m(\mathbf{b}) \cos \left[m \hat{\phi}_{ML}^{(h-1)}(\mathbf{b}) + \eta_m(\mathbf{b}) \right]}. \quad (3.33)$$

La stima della posizione $\hat{\mathbf{b}}$ è quindi data dal valore di \mathbf{b} che massimizza la $ZMLM(\mathbf{b})$

$$\hat{\mathbf{b}} = \arg \left\{ \max_{\mathbf{b}} [ZMLM(\mathbf{b})] \right\}. \quad (3.34)$$

Nel caso in cui si voglia ricercare una immagine, all'interno di un archivio, il *pattern* dell'immagine *query* viene confrontato con tutte le immagini in archivio mediante $ZMLM^{max} = \max[ZMLM(\mathbf{b})]$ valutata per ogni immagine. A questo punto le immagini in archivio sono classificate in accordo con la $ZMLM^{max}$ calcolata. L'immagine che presenta il valore massimo viene considerata come la migliore candidata per il primo *pattern* ottenendo la stima della posizione $\hat{\mathbf{b}}_1$ e della rotazione $\hat{\varphi}_1 = \hat{\varphi}(\hat{\mathbf{b}}_1)$. Queste stime vengono ulteriormente raffinate considerando i successivi *pattern*, associati agli altri punti salienti selezionati. Quindi per il k -esimo *pattern* si hanno $\hat{\mathbf{b}}_k$ e $\hat{\varphi}_k$, rispettivamente per la posizione e per la rotazione. In generale, avendo a disposizione le stime $[\hat{\mathbf{b}}_{k-1}, \hat{\varphi}_{k-1}]$, è possibile stimare il punto $\tilde{\mathbf{c}}_k$ dov'è centrato il k -esimo *pattern* come segue

$$\tilde{\mathbf{c}}_k = \hat{\mathbf{b}}_{k-1} + R_{\hat{\varphi}_{k-1}} \delta \mathbf{x}_k \quad (3.35)$$

dove $R_{\hat{\varphi}_{k-1}}$ è la matrice di rotazione, equazione 3.23, e $\delta \mathbf{x}_k$ è la differenza tra la posizione del centro del k -esimo *pattern* rispetto alla posizione del centro

3.4 Mappa di verosimiglianza tra due immagini

del primo *pattern*. In pratica quello che si vuole sfruttare è l'informazione sulla posizione relativa tra i punti selezionati sull'immagine *query*. Infatti, conoscendo l'orientazione e la lunghezza del segmento che unisce due punti sull'immagine *query*, la rotazione dell'immagine che si sta analizzando e la posizione di un estremo del segmento, è possibile stimare la posizione dell'altro estremo. Quindi la *ZMLM* è calcolata solo in un intorno del punto $\tilde{\mathbf{c}}_k$ per tutti i *pattern* successivi al primo. Inoltre, la stima $\hat{\varphi}_{k-1}$ viene utilizzata per inizializzare la procedura quasi-newtoniana utilizzate per il calcolo della mappa di verosimiglianza *ZMLM*. La deviazione standard dell'errore di predizione del punto $\tilde{\mathbf{c}}_k$ è

$$\sigma_{\tilde{\mathbf{c}}_k}^2 = \sigma_{\hat{\mathbf{b}}_{k-1}}^2 + \sigma_{\hat{\varphi}_{k-1}}^2 |\delta \mathbf{x}_k|^2 \quad (3.36)$$

dove $\sigma_{\hat{\mathbf{b}}_{k-1}}^2$ è la varianza di $\hat{\mathbf{b}}_{k-1}$, e $\sigma_{\hat{\varphi}_{k-1}}^2$ è la varianza di $\hat{\varphi}_{k-1}$. Se la differenza tra l'energia del gradiente dei primi k *pattern* e dell'immagine che si sta considerando è superiore ad una certa soglia viene aumentata l'accuratezza della stima e viene analizzato il successivo *pattern*. In particolare, se A_{nm}^{gh} sono i momenti di Zernike dell' h -esimo *pattern* di g , al k -esimo stadio si ha che il funzionale 3.22 è

$$\begin{aligned} \ln \Lambda^{(k)}[f(\mathbf{x}); \mathbf{b}, \varphi] &= \\ &= -\frac{2}{N_0} \sum_{n=0}^N \sum_{m=-M}^{+M} \sum_{h=1}^K \frac{\pi}{(n+1)\sigma^2} \left| A_{nm}^f(\mathbf{b} + R_\varphi \delta \mathbf{x}_h) - A_{nm}^{gh} e^{im\varphi} \right|^2. \end{aligned} \quad (3.37)$$

Nella pratica, in accordo con il criterio di arresto, solo pochi *pattern* sono utilizzati per migliorare la stima iniziale. Se l'energia del gradiente dei k *pattern* valutati è maggiore dell'energia del gradiente dell'intera immagine moltiplicata per un fattore strettamente compreso tra 0.5 e 1, nessun altro blocco viene analizzato. Inoltre, quando il modulo di $\delta \mathbf{x}_k$ è sufficientemente alto, l'angolo di rotazione può essere stimato sulla base dell'orientazione della retta che collega il centro

del primo *pattern* al centro del k -esimo *pattern*. In ogni caso, la valutazione dell'equazione 3.37 è necessaria per verificare se il *pattern* è presente o meno nell'immagine.

3.5 Risultati sperimentali

3.5.1 Selezione dei punti salienti

Le prestazioni dell'algoritmo che seleziona i punti salienti descritto nel paragrafo 3.2 sono state confrontate con il metodo presente in letteratura proposto da Harris in [33]. Le prestazioni sono state valutate in termini di *Repeatability Rate*, che calcola la percentuale dei punti selezionati sull'immagine originale che vengono selezionati anche sull'immagine artefatta. Il *Repeatability Rate* è calcolato rispetto all'angolo di rotazione e rispetto al fattore di scala considerando quattro tipi di degradazioni a 3 diversi livelli. Le degradazioni sono:

1. sfocatura con filtro gaussiano,
2. compression JPEG,
3. cambiamento uniforme dell'illuminazione,
4. rumore additivo gaussiano a media nulla e varianza σ .

Le immagini utilizzate nelle sperimentazioni sono mostrate in figura 3.7. Come si può notare dalle figure 3.8(a), 3.9(a), 3.10(a), 3.11(a), 3.12(a), utilizzando i polinomi di Zernike si ottiene un netto miglioramento del *Repeatability Rate* rispetto alla rotazione. Questo è dovuto proprio al fatto che i polinomi di Zernike godono, per loro natura, della proprietà di invarianza rispetto alla rotazione. Anche nel caso delle variazioni di scala, utilizzando i polinomi di Zernike, si ha



Figura 3.7: Immagini utilizzati per le sperimentazioni.

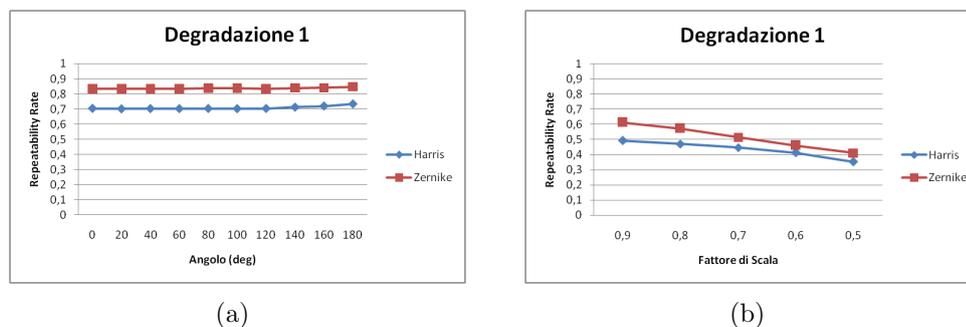


Figura 3.8: Andamento medio del *Repeatability Rate*, rispetto alla rotazione 3.8(a) e rispetto al fattore di scala 3.8(b), calcolato sui tre livelli di sfocatura con filtro gaussiano.

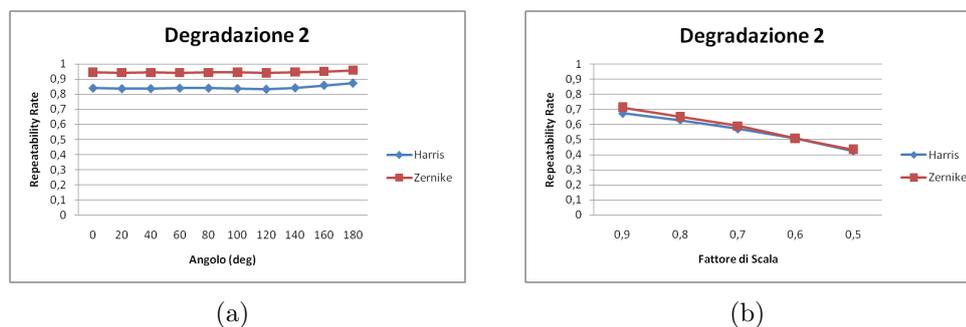


Figura 3.9: Andamento medio del *Repeatability Rate*, rispetto alla rotazione 3.9(a) e rispetto al fattore di scala 3.9(b), calcolato utilizzando la compressione JPEG con tre fattori di compressione diversi.

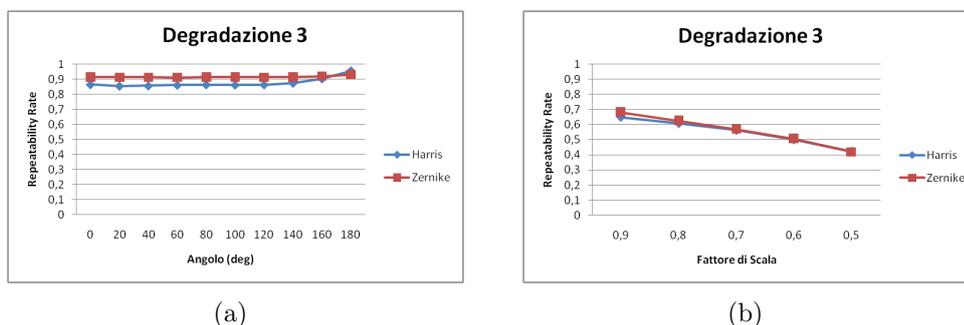


Figura 3.10: Andamento medio del *Repeatability Rate*, rispetto alla rotazione 3.10(a) e rispetto al fattore di scala 3.10(b), calcolato cambiando uniformemente l'illuminazione a tre diversi livelli.

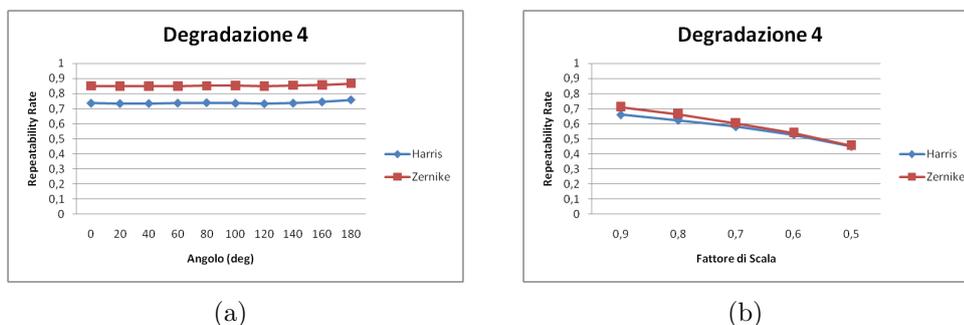


Figura 3.11: Andamento medio del *Repeatability Rate*, rispetto alla rotazione 3.11(a) e rispetto al fattore di scala 3.11(b), calcolato degradando l'immagine con rumore gaussiano additivo a media nulla e con tre diverse varianze.

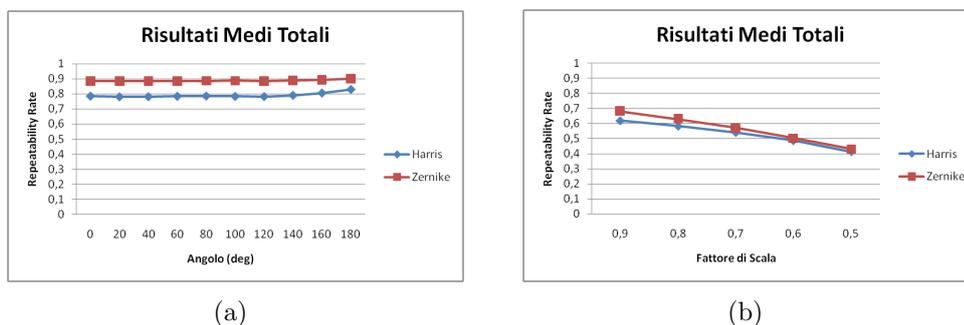


Figura 3.12: Andamento medio del *Repeatability Rate*, rispetto alla rotazione 3.12(a) e rispetto al fattore di scala 3.12(b), calcolato utilizzando tutte le degradazioni a tutti i livelli.

un miglioramento del *Repeatability Rate*, come mostrano le figure 3.8(b), 3.9(b), 3.10(b), 3.11(b), 3.12(b).

3.5.2 Confronto di immagini basato sulla *ZMLM*

Le prestazioni sono valutate utilizzando due diversi archivi di immagini, il *COREL-1000-A* [2] e il *Columbia Object Image Library* (COIL-100) [1]. Il database *COREL-1000-A* è composto da 1000 immagini divise in dieci classi, quali africa, mare, palazzi, autobus, elefanti, dinosauri, fiori, cavalli, montagna ed infine cibo. Un esempio per ogni classe è mostrato nella figura 3.13. Nell'archivio utilizzato nella sperimentazioni sono state aggiunte le versioni ruotate di un angolo *random*, ottenendo così un archivio di 2000 immagini. Il database COIL invece è composto dalle immagini che rappresentano cento oggetti diversi ruotati da 0 a 360 gradi con intervalli di 5 gradi. Nella sperimentazione sono stati utilizzato soltanto le immagini relative alle seguenti rotazioni degli oggetti: $-30, -15, 0, 15, 30$ gradi, ottenendo così un archivio di 500 immagini. Alcuni esempi dell'archivio utilizzato sono mostrati nella figura 3.14. Nelle simulazioni sono stati utilizzati i seguenti parametri:

- due livelli per il *QuadTree*, rispettivamente di 27 e 13 pixel,
- il valore di γ , eq. 3.21, è stato impostato pari a 0.007,
- lo sviluppo di Zernike è stato troncato al quinto ordine, quindi $N = 5$, ottenendo un descrittore di 15 elementi complessi, in questo modo per tutti i livelli del *QuadTree* il supporto è in grado di seguire le oscillazioni dei filtri,
- le immagini sono convertite nello spazio di colore YUV ed è stata utilizzata soltanto la componente di luminanza.



(a)



(b)



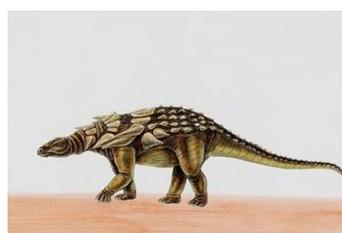
(c)



(d)



(e)



(f)



(g)



(h)



(i)



(j)

Figura 3.13: Esempi delle classi presenti nell'archivio *COREL-1000-A*.

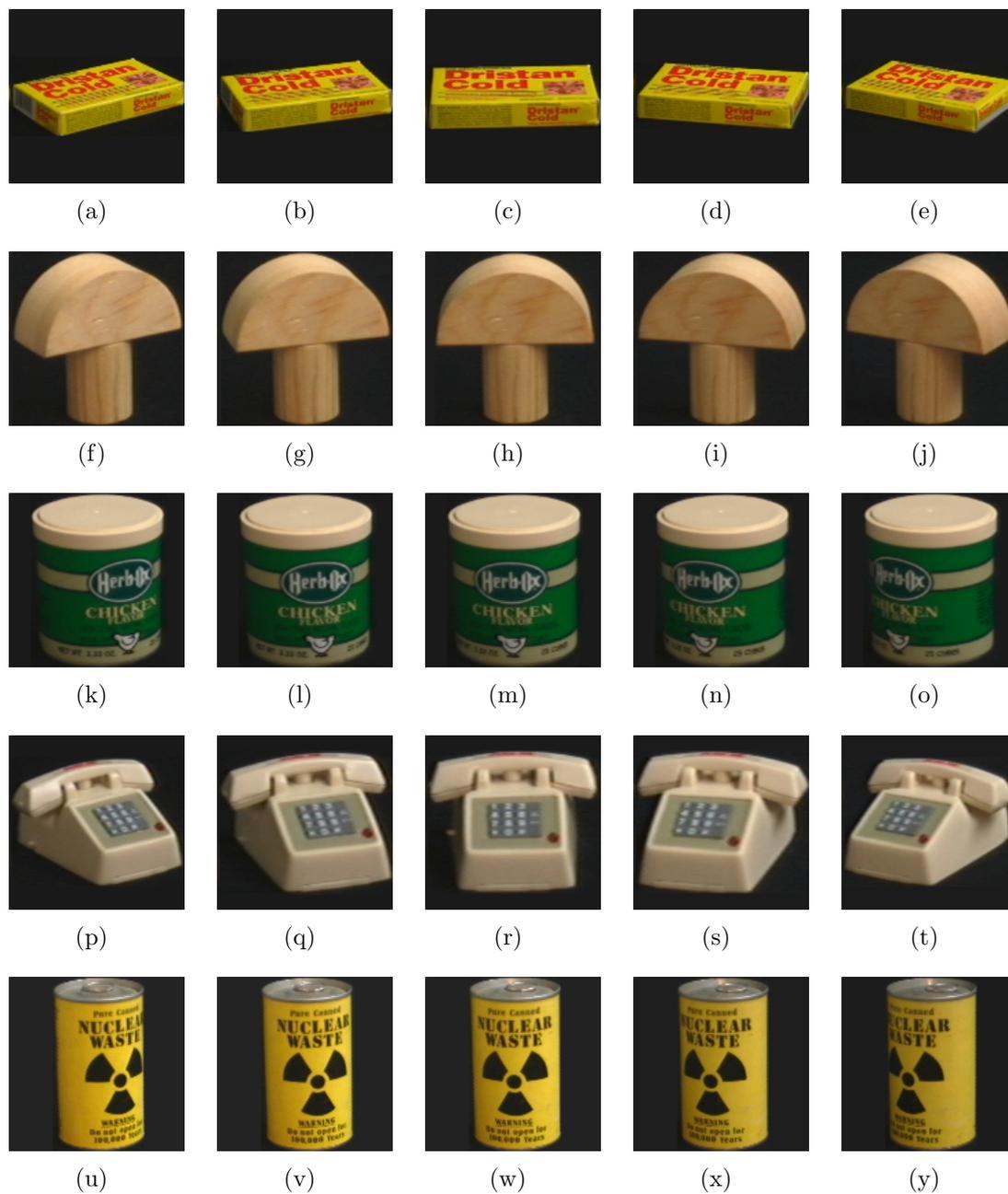


Figura 3.14: Esempi degli oggetti rappresentati nel database COIL-100.

Percentuale media di recupero di immagini rilevanti	95%
Percentuale media di stima corretta dell'angolo di rotazione	91.95%
Errore medio di stima dell'angolo	0.35 deg

Tabella 3.1: Risultati ottenuti utilizzando il database *COREL-1000-A*.

I risultati ottenuti sul database *COREL-1000-A* sono mostrati nella tabella 3.1. Ogni immagine è stata utilizzata come *query* e le immagini rilevanti sono l'originale e la sua versione ruotata. La stima dell'angolo di rotazione è considerata corretta se l'errore è inferiore a 5 gradi, e l'errore medio è considerato nei soli casi in cui la stima è ritenuta corretta. In accordo con il criterio di arresto, il numero di *pattern* considerati è circa cinque. Le prestazioni ottenute sul database *COREL-1000-A* sono state confrontate con quelle ottenute con metodi presenti in letteratura, come quelli basati sull'utilizzo di caratteristiche come il colore e le tessiture (*Color-Texture*) [14], metodi che utilizzano ricerche globali (*Global Search*)¹, o locali (*Regional Search*) [39] o il metodo utilizzato in *SIMPLIcity* [38]. Nelle ricerche globali le caratteristiche vengono estratte considerando tutta l'immagine *query*, mentre in quelle locali, l'utente seleziona una regione di interesse. Il metodo utilizzato in *SIMPLIcity*, invece, estrae le caratteristiche nel dominio wavelet. I metodi utilizzati per il confronto comprendono anche quelli basati su algoritmi genetici (GA), [14]. Nella tabella 3.2 è riportato il confronto, in termini di percentuale media di recupero, con i metodi appena citati. Il database

	Percentuale media di recupero (%)
ZM	95
Global Search	87.2
Regional Search	84.8
Color-Texture	43
SIMPLIcity	46
GA	53

Tabella 3.2: Confronto con alcuni metodi presenti in letteratura.

COIL, invece è stato utilizzato per confrontare i risultati con quelli ottenuti con il

metodo *Invariant Zernike Moments Descriptor* (IZMD) [32]. Le prestazioni sono valutate utilizzando il grafico *Precision-Recall*. In cui la *Precision* è definita come il rapporto tra il numero totale di oggetti rilevanti recuperati e il numero totale di oggetti recuperati, mentre la *Recall* è definita come il rapporto tra il numero di oggetti rilevanti recuperati e il numero di oggetti rilevanti presenti nel database. In particolare,

$$\text{Precision} = \frac{vp}{vp + fp} \quad (3.38)$$

e

$$\text{Recall} = \frac{vp}{vp + fn} \quad (3.39)$$

dove vp sono i veri positivi, fp sono i falsi positivi e fn sono i falsi negativi. Nella figure 3.15 è mostrato il grafico *Precision-Recall* ottenuto utilizzando il

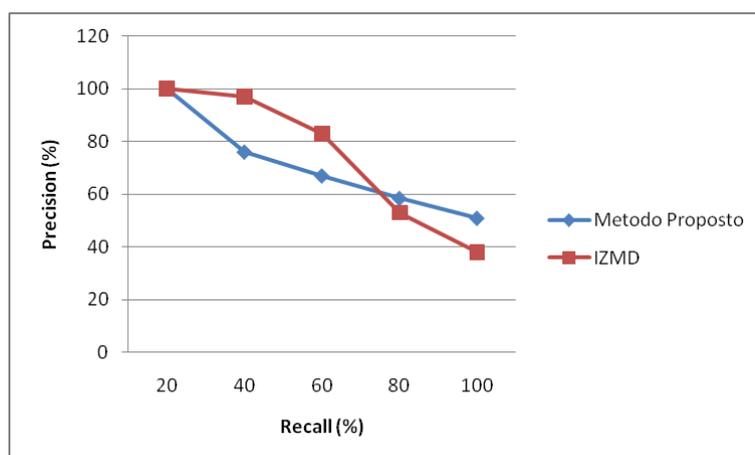


Figura 3.15: Grafico *Precision-Recall* ottenuto utilizzando il database COIL.

database COIL. Come si può notare per bassi valori del *Recall* il metodo IZMD ha prestazioni migliori, tuttavia questo si inverte per valori di *Recall* prossimi al 75%. Poichè un sistema ideale dovrebbe avere *Precision* e *Recall* uguali e pari al 100%, si evince che il metodo proposto si adatta meglio alle esigenze delle applicazioni pratiche che richiedono una elevata percentuale di *Recall*.

3.5.2.1 Complessità computazionale

Per analizzare la complessità computazionale del metodo descritto nel paragrafo 3.4 è stato valutato il numero di operazioni aritmetiche elementari necessarie nella fase di analisi dell'immagine e nella fase di confronto tra l'immagine *query* e l'immagine in archivio. Nell'analisi effettuata sono stati utilizzati i seguenti parametri:

- R_{V_i}, C_{V_i} sono le dimensioni, in pixel, del filtro,
- R, C sono le dimensioni, in pixel, dell'immagine,
- i è il numero di livelli del *QuadTree*,
- L è il numero di filtri utilizzati che dipende dall'ordine a cui si tronca lo sviluppo,
- k è il numero di punti salienti, e quindi di *pattern* utilizzati.

Il numero di addizioni e di moltiplicazioni necessarie nella fase di analisi dell'immagini sono:

1. calcolo dei momenti di Zernike

- addizioni: $L \cdot R_{V_i} \cdot C_{V_i} \cdot R \cdot C$,
- moltiplicazioni: $L \cdot R_{V_i} \cdot C_{V_i} \cdot R \cdot C$,

2. ricostruzione dell'immagini a partire dai momenti di Zernike (ripetuto i volte)

- addizioni: $L \cdot R_{V_i} \cdot C_{V_i} \cdot R \cdot C + L \cdot R \cdot C$,
- moltiplicazioni: $L \cdot R_{V_i} \cdot C_{V_i} \cdot R \cdot C$,

3. calcolo dell'errore di ricostruzione (ripetuto i volte)

- addizioni: $2 \cdot R_{V_i} \cdot C_{V_i}$.

Quindi, approssimativamente, per completare la fase di analisi sono necessarie

$$2 \cdot i \cdot (L \cdot R \cdot C \cdot R_{V_i} \cdot C_{V_i}), \quad (3.40)$$

addizioni e

$$2 \cdot i \cdot (L \cdot R \cdot C \cdot R_{V_i} \cdot C_{V_i}) \quad (3.41)$$

moltiplicazioni. Mentre, per quanto riguarda la fase di confronto tra l'immagine *query* e l'immagine in archivio il numero di operazioni necessarie sono:

1. calcolo dell'energia dell'immagine

- addizioni: $2 \cdot R \cdot C$,
- moltiplicazioni: $2 \cdot R \cdot C$,

2. stima della massima verosimiglianza

- addizioni: $L \cdot (R \cdot C + 1)$,
- moltiplicazioni: $L \cdot (R \cdot C + 4) + 1$,

3. stima dell'angolo di rotazione

- addizioni: $200 \cdot L$,
- moltiplicazioni: $300 \cdot L$,

4. calcolo del residuo dell'energia dell'immagine

- addizioni: 2,
- moltiplicazioni: 2,

5. stima della regione di interesse (ripetuto $k - 1$ volte)

- addizioni: 7,
 - moltiplicazioni: 5,
6. stima della massima verosimiglianza della regione di interesse (ripetuto $k-1$ volte)
- addizioni: $L \cdot \left(\frac{R \cdot C + 1}{8}\right)$,
 - moltiplicazioni: $L \cdot \left(\frac{R \cdot C + 4}{8}\right) + 1$,
7. stima dell'angolo di rotazione (ripetuto $k-1$ volte)
- addizioni: $200 \cdot L$,
 - moltiplicazioni: $300 \cdot L$,
8. calcolo del residuo dell'energia dell'immagine (ripetuto $k-1$ volte)
- addizioni: 2,
 - moltiplicazioni: 2.

In questa fase, sono quindi necessarie

$$\begin{aligned}
 & 2 \cdot R \cdot C + L \cdot (R \cdot C + 1) + 200 \cdot L + 2 + \\
 & + (k-1) \cdot \left[7 + L \cdot \left(\frac{R \cdot C + 1}{8}\right) + 200 \cdot L + 2\right] = \\
 & = R \cdot C \cdot (2 + L) + 201 \cdot L + 2 + \\
 & + (k-1) \cdot \left\{9 + L \cdot \left[\left(\frac{R \cdot C + 1}{8}\right) + 200\right]\right\} \cong \\
 & \cong R \cdot C \cdot L + (k-1) \cdot \left(L \cdot \frac{R \cdot C}{8}\right)
 \end{aligned} \tag{3.42}$$

addizioni e

$$\begin{aligned}
 & 2 \cdot R \cdot C + L \cdot (R \cdot C + 4) + 1 + 300 \cdot L + 2 + \\
 & + (k - 1) \cdot [5 + L \cdot (\frac{R \cdot C + 4}{8}) + 1 + 300 \cdot L + 2] = \\
 & = R \cdot C \cdot (2 + L) + 304 \cdot L + 3 + \tag{3.43} \\
 & + (k - 1) \cdot \{8 + L \cdot [(\frac{R \cdot C + 4}{8}) + 300]\} \cong \\
 & \cong R \cdot C \cdot L + (k - 1) \cdot (L \cdot \frac{R \cdot C}{8})
 \end{aligned}$$

moltiplicazioni.

4

Classificazione di immagini

4.1 Introduzione

Come detto in precedenza, dagli attuali sistemi di recupero di immagini ci si aspetta che siano completamente automatici e che, in qualche modo, siano in grado di classificare gli oggetti in base al loro contenuto semantico. Nel precedente capitolo è stato descritto un metodo che permette il recupero di oggetti complessi all'interno di un archivio multimediale. Questo metodo utilizza un insieme di descrittori estratti mediante le funzioni armoniche circolari, per rappresentare localmente l'oggetto in questione. Mediante l'insieme delle rappresentazioni locali si ottiene una completa caratterizzazione dell'oggetto da ricercare. Un altro tipo di approccio è quello di caratterizzare globalmente l'immagine estraendo una serie di caratteristiche di basso livello. Tra le caratteristiche di basso livello più utilizzate sono le tessiture, il colore e i bordi, [10, 13, 20]. Le caratteristiche di basso livello vengono usate per cercare di caratterizzare il contenuto semantico dell'immagine. Le tessiture sono l'insieme delle trame presenti in una immagine e solitamente vengono rappresentate in modo statistico. I metodi classici utilizzati per la rappresentazione delle tessiture si basano sull'utilizzo delle *wavelet* [37],

oppure sull'utilizzo dei filtri di Gabor [7, 15]. A questo punto, i piani *wavelet* o le risposte al filtro di Gabor, vengono caratterizzate statisticamente utilizzando la media e la varianza. Un altro metodo, proposto in [37], è quello di caratterizzare le densità marginali dei piani *wavelet* mediante delle gaussiane generalizzate. È stato scelto questo tipo di approccio applicato però alle densità marginali dei piani ottenuti con le *wavelet* armoniche circolari, ed in particolare alla *wavelet* di Laguerre Gauss [28]. Il colore, invece è stato caratterizzato mediante i momenti del primo, secondo e terzo ordine delle sole componenti cromatiche, [7]. Infine, per quanto riguarda i bordi, questi vengono solitamente caratterizzati mediante l'istogramma delle fasi. Anche in questo caso, la selezione e la stima della fase di ogni punto di bordo viene effettuata utilizzando le funzioni di Laguerre Gauss, [27]. Una volta estratte le caratteristiche di basso livello queste possono essere utilizzate in sistemi che utilizzano tecniche di *machine learning* [15], oppure in sistemi che effettuano le operazioni di ricerca basandosi su una immagine di esempio [4]. Il metodo descritto in questo capitolo segue questa seconda strada, cioè organizza le immagini in archivio in base alla similitudine con una immagine di esempio fornita dall'utente. Ovviamente la similitudine viene calcolata utilizzando i vettori che rappresentano le caratteristiche di basso livello. Poiché in immagini naturali, potrebbero esserci delle zone che non sono utili al fine dell'estrazione delle caratteristiche di basso livello, queste vengono estratte solo nelle zone importanti, dal punto di vista dell'informazione di Fisher, [29]. La classificazione basata su un'immagine di esempio presenta alcuni limiti, come ad esempio il fatto che le prestazioni sono strettamente legate all'immagine fornita dall'utente. Per superare questi limiti viene utilizzato un algoritmo che raggruppa in sottoinsiemi le immagini che vengono considerate rilevanti [26]. Questo metodo è totalmente generale, infatti le immagini rilevanti possono essere selezionate in qualsiasi modo.

4.2 Estrazione e confronto delle caratteristiche di basso livello

Per esempio possono essere selezionate considerando la similarità con l'immagine di esempio fornita dall'utente utilizzando le caratteristiche di basso livello o può essere applicato a sistemi che classificano le immagini sull'utilizzo di parole chiave, largamente diffusi sul web. Raggruppando i risultati in sottoinsiemi è possibile mostrare all'utente solo il rappresentante di ogni sottoinsieme, riuscendo così a mostrargli una panoramica più ampia di risultati. In questo modo, se l'immagine di esempio non è scelta correttamente, l'utente ha più possibilità di trovare ciò a cui è interessato. Infatti, nella classificazione classica le prime posizioni sarebbero occupate da immagini molto simili all'immagine di esempio, mentre in questo modo tutte queste immagini vengono raggruppate e rappresentate da una sola immagine lasciando spazio ad immagini che, pur essendo meno simili, possono risultare essere rilevanti per l'utente.

4.2 Estrazione e confronto delle caratteristiche di basso livello

4.2.1 Le tessiture

La rappresentazione statistica dell'informazione contenuta in una immagine $f(\mathbf{x})$ è tipicamente preceduto da una trasformazione in un dominio in cui l'informazione può essere rappresentata con pochi parametri. Per l'analisi delle tessiture le immagini vengono analizzate mediante la trasformazione *wavelet*, che per loro natura, suddividono lo spettro in sottobande calcolate a diverse scale e a diverse orientazioni. Se si assume che l'energia delle sottobande può essere utilizzate come descrittori delle tessiture si ha che, il vettore che rappresenta questa caratteristica

è composto da

$$\frac{1}{L} \sum_{j=1}^L W_{\psi}^i [f] (j) \quad (4.1)$$

e da

$$\left(\frac{1}{L} \sum_{j=1}^L |W_{\psi}^i [f] (j)|^2 \right)^{\frac{1}{2}} \quad (4.2)$$

dove $W_{\psi}^i [f] (j)$ è il j –esimo coefficiente della sottobanda i –esima associato all'immagine f . Per completezza ψ indica la generica *wavelet* madre utilizzata per l'analisi [37]. Questo tipo di approccio può essere applicato anche alle *wavelet* di Laguerre Gauss [22], ed in questo caso si ha

$$\frac{1}{L} \sum_{j=1}^L W_{\mathcal{L}_{nk}}^i [f] (j), \quad (4.3)$$

$$\left(\frac{1}{L} \sum_{j=1}^L |W_{\mathcal{L}_{nk}}^i [f] (j)|^2 \right)^{\frac{1}{2}}, \quad (4.4)$$

dove \mathcal{L}_{nk} rappresenta *wavelet* di Laguerre Gauss, [22]. L'estensione di questo metodo passa attraverso la caratterizzazione delle tessiture mediante le densità marginale dei coefficienti *wavelet*. Questo è giustificato dal fatto che due tessiture sono difficili da distinguere se la densità marginali delle risposte ad un banco di filtri sono simili. Le densità marginali dei coefficienti *wavelet* di una generica sottobanda può essere approssimata da una gaussiana generalizzata, [11]

$$p (W_{\mathcal{L}_{nk}}^i [f] ; \alpha, \beta) = \frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} e^{-\left(\frac{|W_{\mathcal{L}_{nk}}^i [f]|}{\alpha}\right)^{\beta}} \quad (4.5)$$

4.2 Estrazione e confronto delle caratteristiche di basso livello

dove $\Gamma(\cdot)$ è la funzione Gamma definita come

$$\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} dt. \quad (4.6)$$

Come si può osservare dall'equazione 4.5, la gaussiana generalizzata è definita mediante due parametri, α e β . Il primo è proporzionale alla larghezza della distribuzione, quindi alla deviazione standard, mentre, il secondo, è proporzionale al tasso di decrescita della distribuzione. Quindi α può essere chiamato parametro di scala e β parametro di forma. Le distribuzioni gaussiane generalizzate includono due casi particolari in cui coincidono con due distribuzioni note, in particolare per $\beta = 2$ coincidono con la gaussiana, e per $\beta = 1$ coincidono con la distribuzione di Laplace, come si vede nella figura 4.1. Nella figura 4.2 sono mostrati alcuni

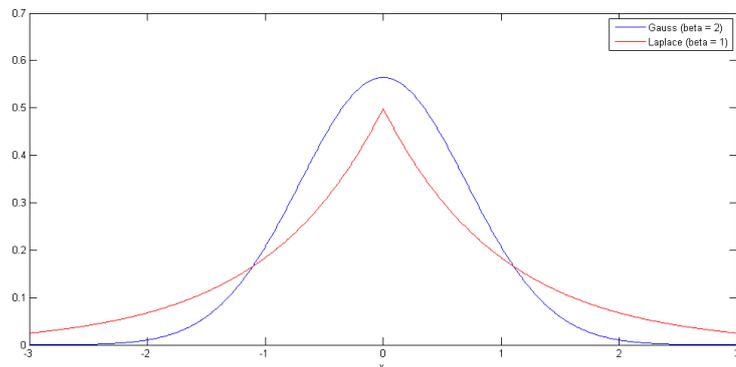


Figura 4.1: Casi particolari di GGD che coincidono con la distribuzione di Gauss, $\beta = 2$, e con la distribuzione di Laplace, $\beta = 1$.

esempi di gaussiane generalizzate. Nel grafico, le curve che hanno lo stesso colore hanno lo stesso valore di β , mentre le curve che hanno lo stesso tipo di linea, hanno lo stesso valore di α . Si può quindi osservare come varia la forma della distribuzione al variare dei due parametri α e β . In [5] è stato dimostrato che utilizzando lo stimatore a massima verosimiglianza nei casi di distribuzioni con

4.2 Estrazione e confronto delle caratteristiche di basso livello

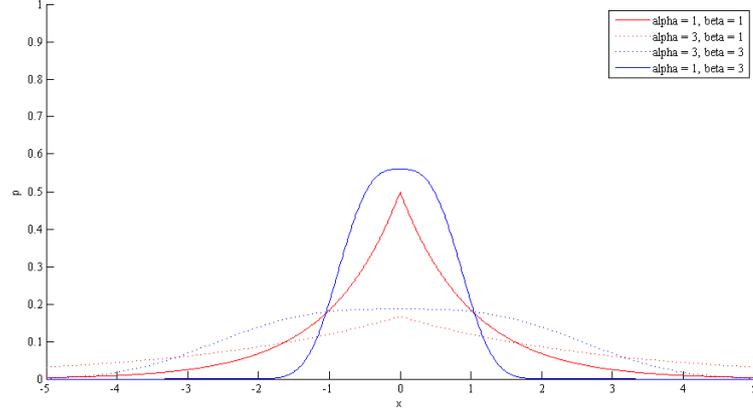


Figura 4.2: Esempi di gaussiane generalizzate per diversi valori di α e β .

code lunghe si ottengono i risultati migliori. Perciò si ottiene

$$L(\mathbf{W}_{\mathcal{L}_{nk}}; \alpha, \beta) = \log \prod_{j=1}^L p(W_{\mathcal{L}_{nk}}(j); \alpha, \beta) \quad (4.7)$$

dove con $\mathbf{W}_{\mathcal{L}_{nk}}$ si indica la generica sottobanda il cui elemento è $W_{\mathcal{L}_{nk}}$, e α e β sono i parametri da stimare. Per semplicità di notazione si omette l'immagine f a cui corrispondono i coefficienti *wavelet*. Dall'equazione 4.7 segue che

$$\frac{\partial L(\mathbf{W}_{\mathcal{L}_{nk}}; \alpha, \beta)}{\partial \alpha} = -\frac{L}{\alpha} + \sum_{j=1}^L \frac{\beta |W_{\mathcal{L}_{nk}}(j)| \alpha^{-\beta}}{\alpha} \quad (4.8)$$

$$\frac{\partial L(\mathbf{W}_{\mathcal{L}_{nk}}; \alpha, \beta)}{\partial \beta} = \frac{L}{\beta} + \frac{L\Psi\left(\frac{1}{\beta}\right)}{\beta^2} - \sum_{j=1}^L \left(\frac{|W_{\mathcal{L}_{nk}}(j)|}{\alpha}\right)^{\beta} \log\left(\frac{|W_{\mathcal{L}_{nk}}(j)|}{\alpha}\right) \quad (4.9)$$

dove Ψ è la funzione digamma definita come

$$\Psi(z) = \frac{\Gamma'(z)}{\Gamma(z)}. \quad (4.10)$$

4.2 Estrazione e confronto delle caratteristiche di basso livello

L'equazione 4.8 ha un'unica soluzione reale, per $\beta > 0$, in

$$\hat{\alpha} = \left(\frac{\beta}{L} \sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta \right)^{\frac{1}{\beta}}. \quad (4.11)$$

Sostituendo l'equazione 4.11 nell'equazione 4.9 si ottiene la seguente equazione trascendente che può essere risolta numericamente

$$1 + \frac{\Psi\left(\frac{1}{\hat{\beta}}\right)}{\hat{\beta}} - \frac{\sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^{\hat{\beta}} \log |W_{\mathcal{L}_{nk}}(j)|}{\sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^{\hat{\beta}}} + \frac{\log\left(\frac{\hat{\beta}}{L} \sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^{\hat{\beta}}\right)}{\hat{\beta}} = 0. \quad (4.12)$$

Un possibile algoritmo per trovare $\hat{\beta}$ è quello di Newton-Raphson, in cui

$$\beta_{k+1} = \beta_k - \frac{h(\beta_k)}{h'(\beta_k)} \quad (4.13)$$

dove $h(\beta)$ è

$$h(\beta) = 1 + \frac{\Psi\left(\frac{1}{\beta}\right)}{\beta} - \frac{\sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta \log |W_{\mathcal{L}_{nk}}(j)|}{\sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta} + \frac{\log\left(\frac{\beta}{L} \sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta\right)}{\beta} \quad (4.14)$$

e $h'(\beta)$ è la sua derivata

$$\begin{aligned} h'(\beta) = & -\frac{\Psi\left(\frac{1}{\beta}\right)}{\beta^2} - \frac{\Psi'\left(\frac{1}{\beta}\right)}{\beta^3} - \frac{1}{\beta^2} - \frac{\sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta \log |W_{\mathcal{L}_{nk}}(j)|}{\sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta} + \\ & + \frac{\left(\sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta \log |W_{\mathcal{L}_{nk}}(j)|\right)^2}{\left(\sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta\right)^2} + \frac{\log\left(\frac{\beta}{L} \sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta\right)}{\beta \sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta} + \\ & - \frac{\log\left(\frac{\beta}{L} \sum_{j=1}^L |W_{\mathcal{L}_{nk}}(j)|^\beta\right)}{\beta^2} \end{aligned} \quad (4.15)$$

4.2 Estrazione e confronto delle caratteristiche di basso livello

in cui Ψ' indica la funzione trigamma definita come segue

$$\Psi'(z) = \frac{d}{dz}\Psi(z). \quad (4.16)$$

Questo tipo di approccio è stato utilizzato per la caratterizzazione delle tessiture utilizzando le *wavelet* di Laguerre Gauss, [28]. Ovviamente, poichè in questo caso i coefficienti *wavelet* sono complessi la parte reale e la parte immaginaria sono state trattate in modo indipendente. Quindi le densità marginali della parte reale di un piano *wavelet* è stato approssimato con una GGD e la parte immaginaria con un'altra GGD. In questo modo, a differenza delle *wavelet* reali, servono quattro parametri per caratterizzare completamente un piano *wavelet*. Le funzioni di Laguerre Gauss che meglio si adattano alla caratterizzazione delle tessiture sono quelle identificate dagli indici $n = 1$ e $k = 0$, e $n = 3$ e $k = 0$, utilizzando tre risoluzioni. L'analisi delle tessiture avviene utilizzando la sola componente di luminanza, che contiene tutte le informazioni riguardanti la struttura delle tessiture. Come si vede nelle figure 4.6, 4.7 le densità marginali dei coefficienti



Figura 4.3: Immagine *Bark.0000*.

wavelet sono ben approssimate dalle gaussiane generalizzate, anche nel caso in cui vengono utilizzate le *wavelet* di Laguerre Gauss. Una volta caratterizzate completamente le tessiture presenti in una immagine, utilizzando i parametri che

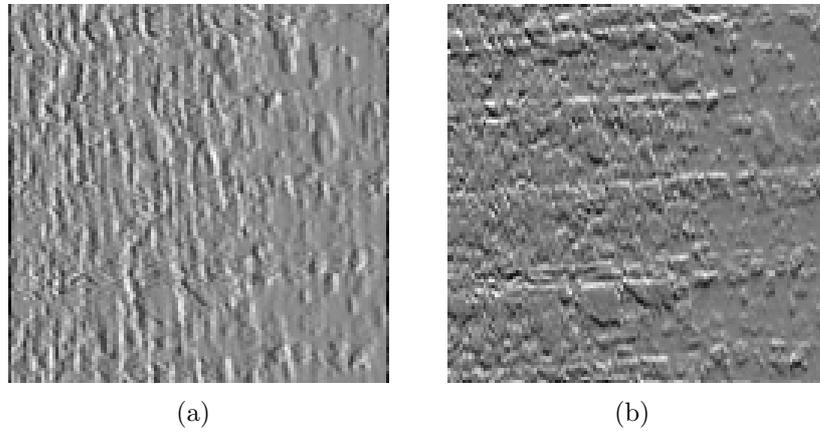


Figura 4.4: Parte reale, 4.4(a), e della parte immaginaria, 4.4(b), del piano *wavelet* di Laguerre Gauss $n = 1$ e $k = 0$ ad alta risoluzione, calcolato sull'immagine mostrata in figure 4.3.

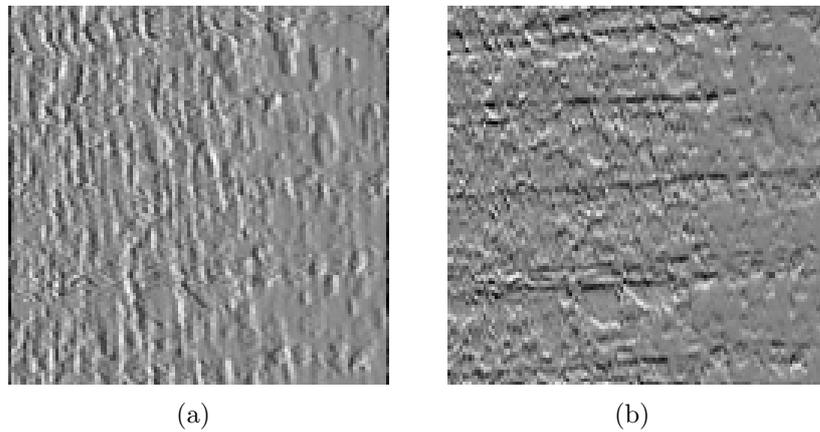


Figura 4.5: Parte reale, 4.5(a), e della parte immaginaria, 4.5(b), del piano *wavelet* di Laguerre Gauss $n = 3$ e $k = 0$ ad alta risoluzione, calcolato sull'immagine mostrata in figure 4.3.

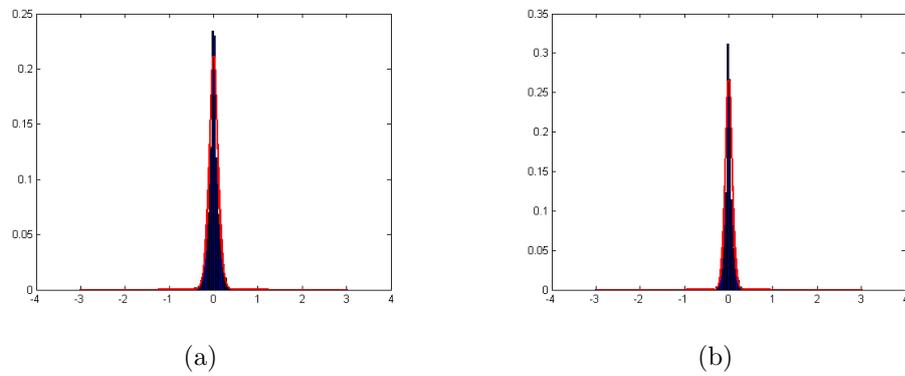


Figura 4.6: Densità marginali della parte reale, 4.6(a), e parte immaginaria, 4.6(b), del piano *wavelet* di Laguerre Gauss $n = 1$ e $k = 0$ ad alta risoluzione, calcolato sull'immagine mostrata in figure 4.3.

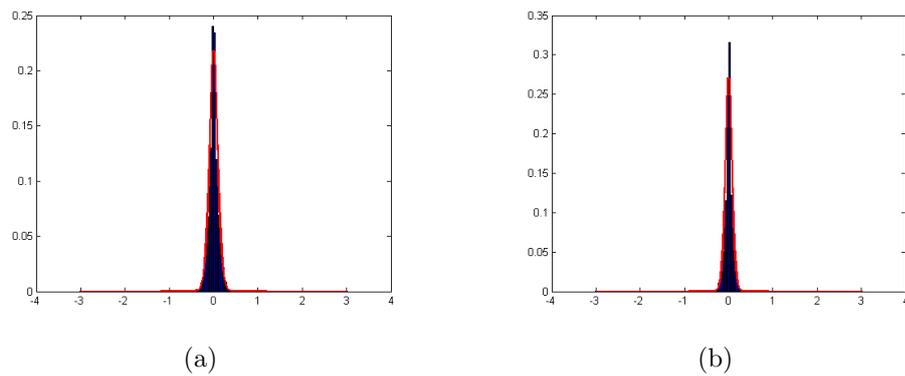


Figura 4.7: Densità marginali della parte reale, 4.7(a), e parte immaginaria, 4.7(b), del piano *wavelet* di Laguerre Gauss $n = 3$ e $k = 0$ ad alta risoluzione, calcolato sull'immagine mostrata in figure 4.3.

4.2 Estrazione e confronto delle caratteristiche di basso livello

definiscono una gaussiana generalizzata, è necessario definire una metrica che permetta di effettuare il confronto tra due tessiture. L'approccio più classico è quello di calcolare la distanze Euclidea tra i vettori associati alle due tessiture. Tuttavia, sfruttando il fatto che gli elementi di questi vettori rappresentano due distribuzioni è possibile definire una metrica più precisa. In particolare, si utilizza la distanza di Kullback-Leibler, [28, 37]. Minimizzare la distanza di Kullback-Leibler tra due distribuzioni equivale a massimizzare la loro verosimiglianza e a minimizzare l'entropia relativa tra le due distribuzioni. La distanza di Kullback-Leibler tra due gaussiane generalizzate definite da α_f, β_f e da α_g, β_g è

$$\begin{aligned} KLD(p(\cdot; \alpha_f, \beta_f) \| p(\cdot; \alpha_g, \beta_g)) &= \\ &= \log \left(\frac{\beta_f \alpha_g \Gamma\left(\frac{1}{\beta_g}\right)}{\beta_g \alpha_f \Gamma\left(\frac{1}{\beta_f}\right)} \right) + \left(\frac{\alpha_f}{\alpha_g} \right)^{\beta_g} \frac{\Gamma\left(\frac{\beta_g+1}{\beta_f}\right)}{\Gamma\left(\frac{1}{\beta_f}\right)} - \frac{1}{\beta_f} \end{aligned} \quad (4.17)$$

dove f e g rappresentano due generiche immagini da confrontare. La distanza di Kullback-Leibler viene anche detta divergenza, questo perchè non rispetta la disuguaglianza triangolare. Infatti, la distanza di Kullback-Leibler tra A ed B , è generalmente diversa da quella calcolata tra B ed A . Come detto in precedenza le tessiture presenti nell'immagine f vengono caratterizzate utilizzando due funzioni di Laguerre Gauss a tre risoluzioni diverse. Inoltre, poichè i coefficienti sono complessi, ad ogni piano *wavelet* corrispondono due gaussiane generalizzate, una per la parte reale, ed una per la parte immaginaria. Quindi, utilizzando B piani *wavelet* complessi si ottiene che la distanza tra due immagini f e g è

$$KLD(f, g) = \sum_{j=1}^{2B} KLD(p(\cdot; \alpha_f^j, \beta_f^j) \| p(\cdot; \alpha_g^j, \beta_g^j)). \quad (4.18)$$

L'equazione 4.18 è valida ipotizzando che i piani *wavelet* siano tra loro indipendenti.

4.2.2 Il colore

Un'altra caratteristica di basso livello utilizzata per descrivere il contenuto di una immagine è il colore. Infatti, se per esempio si considerano alcuni tipo di tessiture, queste da un punto di vista della struttura sono indistinguibili, mentre presentano un colore completamente diverso. È quindi evidente che la completa caratterizzazione di una immagine non può prescindere dall'informazione sul colore. A tal fine, viene utilizzato lo spazio di colore YUV e vengono utilizzate le sole componenti cromatiche. Il passaggio dallo spazio di colore RGB allo spazio di colore YUV avviene come segue

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (4.19)$$

Le distribuzioni delle sole componenti cromatiche vengono caratterizzate utilizzando la media, la varianza ed il momento centrato del terzo ordine [7]

$$\mu = \frac{1}{RC} \sum_{r=1}^R \sum_{c=1}^C f(r, c), \quad (4.20)$$

$$\sigma = \left[\frac{1}{RC} \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} (f(r, c) - \mu)^2 \right]^{\frac{1}{2}}, \quad (4.21)$$

$$t = \left[\frac{1}{RC} \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} (f(r, c) - \mu)^3 \right]^{\frac{1}{3}}, \quad (4.22)$$

dove con $f(r, c)$ rappresenta il pixel in posizione r, c della componente cromatica che si vuole caratterizzare. Nel caso di immagini naturali, è evidente che una rappresentazione di questo tipo potrebbe non essere sufficiente. In questi casi per avere una rappresentazione più completa le immagini vengono suddivise in



Figura 4.8: Versione originale dell'immagine Lena.

blocchi non sovrapposti. Perciò, ogni blocco viene caratterizzato come descritto in precedenza, come se fosse una singola immagine. In questo modo si ottiene un vettore composto da più elementi ma anche una rappresentazione più completa. La metrica utilizzata per analizzare la similarità tra due vettori che rappresentano l'informazione sul colore è la distanza euclidea, quindi

$$ED_c(f, g) = \left[\sum_j (CV_f(j) - CV_g(j))^2 \right]^{\frac{1}{2}} \quad (4.23)$$

dove con CV_f indica il vettore che rappresenta il colore dell'immagine f , e, analogamente, CV_g indica il vettore che rappresenta il colore dell'immagine g .

4.2.3 I bordi

Per completare la rappresentazione del contenuto informativo presente in una immagine è necessario caratterizzare i bordi presenti in una immagine. Per poter caratterizzare in modo efficiente i bordi presente in una immagine sono necessarie principalmente due operazioni. La prima consiste nella selezioni dei bordi principali di una immagine. Con bordi principali si intendono i bordi che descrivono



(a)



(b)



(c)

Figura 4.9: Componenti RGB, 4.9(a), 4.9(b), 4.9(c), dell'immagine mostrata nella figura 4.8.



(a)

(b)



(c)

Figura 4.10: Componenti YUV, 4.10(a), 4.10(b), 4.10(c), dell'immagine mostrata nella figura 4.8.

le forme degli oggetti. Si cerca quindi di non utilizzare i bordi che definiscono le trame e le tessiture. Considerando l'immagine mostrata nella figura 4.11, i



Figura 4.11: Versione originale dell'immagine Cameraman.

bordi che, in teoria, non si dovrebbero usare sono quelli che identificano l'erba nella parte bassa dell'immagine. D'altra parte, i bordi che identificano la sagoma dell'uomo, della videocamera e anche quelle che delimitano i palazzi nello sfondo, sono i soli, almeno teoricamente, che si dovrebbero usare. Una volta selezionati i bordi principali, questi vengono caratterizzati calcolando la loro direzione. È quindi necessario stimare la direzione dei bordi e, successivamente, calcolarne l'istogramma, [27]. Per selezionare i bordi principali di una immagine viene usati l'operatore di Laplace definito come segue:

$$\nabla^2 f(\mathbf{x}) = \frac{\partial^2}{\partial x_1^2} f(\mathbf{x}) + \frac{\partial^2}{\partial x_2^2} f(\mathbf{x}). \quad (4.24)$$

Nella figura 4.12 è mostrata la mappa che si ottiene applicando l'equazione 4.24 all'immagine di figura 4.11. Una volta calcolata questa mappa, vengono selezionati i punti di *zero-crossing*. Questi sono i punti in cui il segnale interseca l'asse delle ascisse. Per esempio, nel caso monodimensionale, se si considera una sinusoide, nel periodo c'è un punto di *zero-crossing*, figura 4.13. Per ottenere una



Figura 4.12: Mappa ottenuta applicando l'operatore di Laplace all'immagine mostrata in figura 4.11.

buona stima della direzione dei bordi vengono utilizzate le funzioni di Laguerre Gauss. In particolare viene utilizzata quella identificata dagli indici $n = 1$ e $k = 0$, perchè intrinsecamente estrae le linee presenti in una immagine. La map-

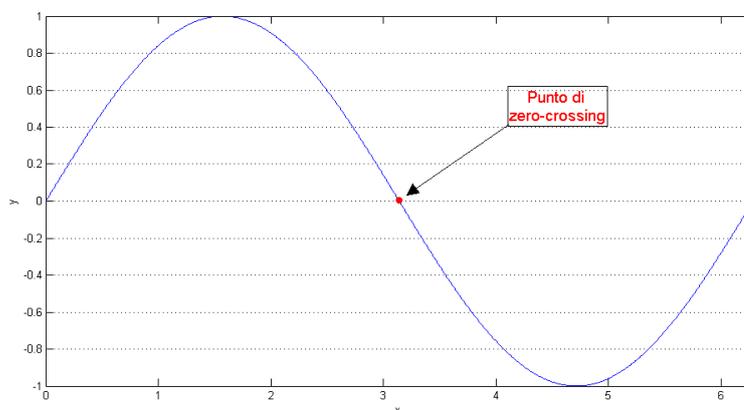


Figura 4.13: Esempio di punto di *zero-crossing*.

pa ottenuta dopo aver selezionato i punti di *zero-crossing* e la mappa ottenuta filtrando l'immagine con la funzione di Laguerre Gauss $n = 1$ e $k = 0$, vengono unite mediante l'operatore di AND. In questo modo si ottiene una mappa che è diversa da zero solo nei punti corrispondenti ai bordi principali dell'immagine. In questi punti la mappa assume i valori pari al coefficiente di Laguerre Gauss.

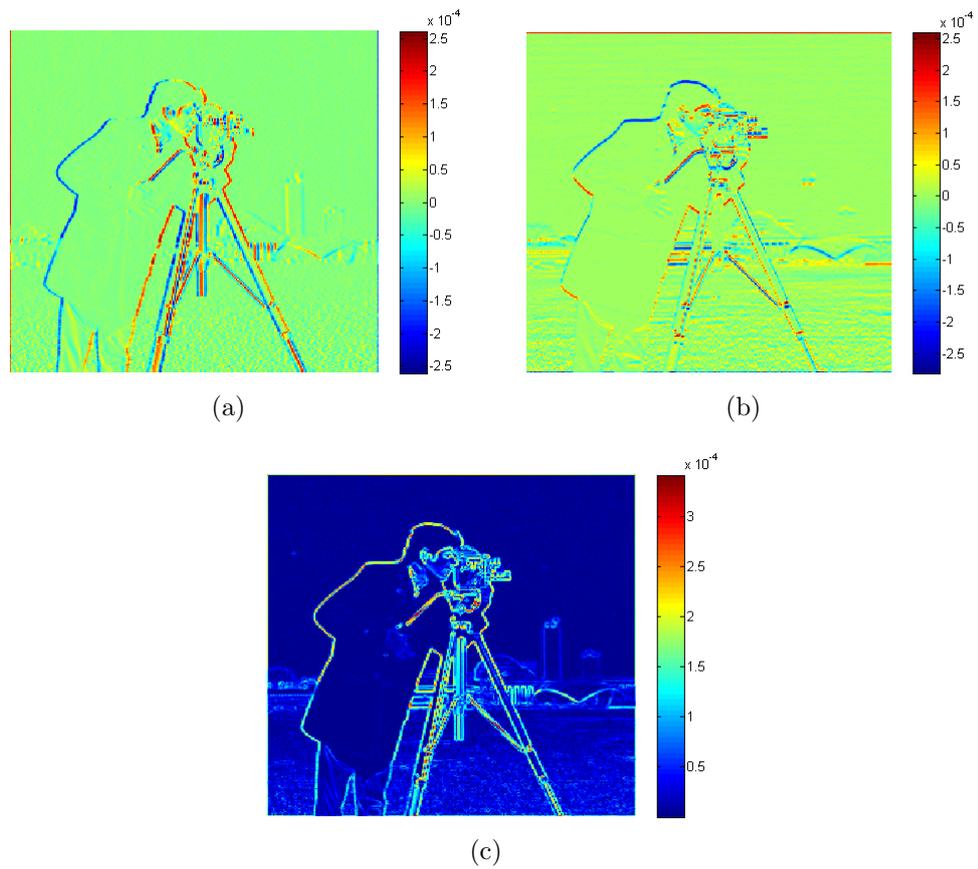


Figura 4.14: Parte reale, 4.14(a), parte immaginaria, 4.14(b), e modulo, 4.14(c), dell'immagine mostrata in figura 4.11 filtrata con la funzione di Laguerre Gauss $n = 1$ e $k = 0$.

4.2 Estrazione e confronto delle caratteristiche di basso livello

Nell'immagine mostrata in figura 4.15 sono mostrati i punti che vengono utilizzati



Figura 4.15: Mappa in cui sono selezionati i punti utilizzati per caratterizzare l'informazione sui bordi presenti nell'immagine di figura 4.11.

per caratterizzare i bordi. Sfruttando le proprietà delle funzioni armoniche circolari, ed in particolare delle funzioni di Laguerre Gauss, una stima della direzione del punto di bordo è la fase del coefficiente di Laguerre Gauss. L'istogramma delle fasi dei punti di bordo è calcolato considerando il dominio $[0, \pi]$ perchè non è importante il verso del bordo, bensì la direzione. Inoltre, questo dominio viene quantizzato in 36 intervalli di 5 gradi ognuno. Nella figura 4.16 è mostrato

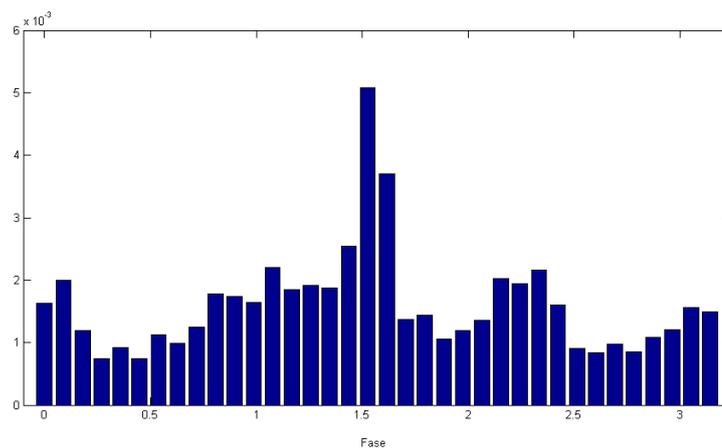


Figura 4.16: Istogramma delle fasi dei bordi calcolato sull'immagine di figura 4.11.

4.3 Algoritmo per il raggruppamento delle immagini rilevanti

l'istogramma delle fasi calcolato con il metodo descritto sull'immagine mostrata nella figura 4.11. In questo modo si ottiene un vettore composto da 36 elementi che rappresenta i bordi presenti in una immagine. Per effettuare il confronto tra due generici vettori si utilizza la distanza euclidea

$$ED_e(f, g) = \left[\sum_{j=1}^{36} (EV_f(j) - EV_g(j))^2 \right]^{\frac{1}{2}} \quad (4.25)$$

dove con EV_f e EV_g indicano l'istogramma delle fasi calcolato, rispettivamente, sull'immagine f e sull'immagine g .

4.3 Algoritmo per il raggruppamento delle immagini rilevanti

Generalmente i motori di ricerca, data una richiesta da parte dell'utente, misurano la distanza tra questa e gli oggetti presenti nell'archivio, utilizzando una qualsiasi metrica, e poi mostra all'utente gli oggetti in archivio ordinandoli dal più simile al meno simile. Ovviamente, questo tipo di approccio è utilizzato per qualsiasi oggetto digitale, come ad esempio testi, immagini, video. Nel caso delle immagini, la richiesta molto spesso è una immagine che in qualche modo rappresenta ciò che l'utente sta cercando. Questo tipo di approccio presenta però alcuni limiti. Infatti quando si hanno a disposizione grandi archivi, cosa che attualmente, accade molto spesso, classificare le immagini dalla più simile alla meno simile fa sì che le prime posizioni, quelle a cui generalmente l'utente presta la maggiore attenzione, sono occupate da immagini che sono molto simili all'immagine *query*. Trascurando così quelle immagini che sono rilevanti ma che non presentano una grandissima similarità con l'immagine che l'utente ha fornito al

4.3 Algoritmo per il raggruppamento delle immagini rilevanti

sistema per effettuare la sua richiesta. Un altro limite dei sistemi appena descritti è che i risultati sono fortemente influenzati dall'immagine che l'utente fornisce al sistema nel momento in cui sottomette la sua richiesta. Infatti, l'utente potrebbe scegliere una immagine che non rappresenta bene ciò a cui è interessato, sia a causa di un proprio errore che per la mancanza oggettiva di una immagine più appropriata, oppure potrebbe non avere le idee chiare su ciò che sta cercando. In queste situazioni, un sistema che mostra all'utente le immagini organizzandole dalla più simile alla meno simile, in generale, difficilmente può soddisfare l'utente. Per cercare di superare questi tipi di problemi è stato pensato ad un metodo che permetta al sistema di mostrare all'utente una panoramica più ampia dei risultati. A tal fine è stato sviluppato un algoritmo di *clustering*, che ha lo scopo di raggruppare le immagini che potrebbero essere rilevanti rispetto alla *query* fornita dall'utente, [26]. Prima di descrivere l'algoritmo è necessario definire la metrica con cui vengono confrontate due immagini. Nei paragrafi precedenti sono stati descritti dei metodi con cui caratterizzare e confrontare delle caratteristiche di basso livello come le tessiture, il colore ed i bordi. Le equazioni 4.18, 4.23 e 4.25 mostrano le metriche per confrontare, rispettivamente, le tessiture, il colore ed i bordi. La metrica utilizzata per confrontare due immagini f e g è

$$S(f, g) = \delta_t \cdot KLD(f, g) + \delta_c \cdot ED_c(f, g) + \delta_e \cdot ED_e(f, g) \quad (4.26)$$

dove $\delta_t, \delta_c, \delta_e$ sono tre pesi con cui modificare l'importanza relativa data alle tessiture, al colore ed ai bordi. L'algoritmo di *clustering* può essere riassunto in quattro passi principali:

1. Le immagini in archivio vengono classificate, in accordo con l'equazione 4.26, in base alla similitudine con l'immagine *query* fornita dall'utente. Considerando le prima n immagini rilevanti queste vengono raggruppate in

4.3 Algoritmo per il raggruppamento delle immagini rilevanti

un sottoinsieme U che sarà utilizzato nei successivi passi. Le immagini che non appartengono a questo sottoinsieme si considerano irrilevanti, mentre gli elementi $u \in U$ sono le immagini che vengono considerate rilevanti.

2. L'ultima immagine del sottoinsieme U^1 viene utilizzata come *query*, indicata con u_{query}^1 , e le immagini che appartengono al sottoinsieme U^1 vengono riordinate in base alla similitudine con questa nuova *query*. A questo punto le immagini che presentano un valore di $S(u_i, u_{query}^1)$ inferiore ad una certa soglia vengono raggruppate in un sottoinsieme v^1 , ottenendo quindi che

$$v^1 = \{u_i \in U^1 \mid S(u_i, u_{query}^1) < th_{STEP1}\} \quad (4.27)$$

Al sottoinsieme v^1 vengono associati tre vettori che lo rappresentano, uno per le tessiture, uno per il colore ed, infine, uno per i bordi. Questi vettori sono ottenuti mediando i vettori che rappresentano tutte le immagini $u \in v^1$. Le immagini che compongono il sottoinsieme v^1 vengono tolte dall'insieme U^1 , ottenendo per la k -esima iterazione

$$U^k = U^{k-1} \setminus v^{k-1}. \quad (4.28)$$

e quindi

$$v^k = \{u_i \in U^k \mid S(u_i, u_{query}^k) < th_{STEP1}\} \quad (4.29)$$

Questo passo finisce quando $U^{k+1} = \emptyset$. Tutti gli elementi v^k creati vengono uniti nell'insieme V utilizzato nel seguente passo.

3. Il risultato del passo precedente è quindi un insieme V composto da un numero finito di sottoinsiemi v che sono rappresentanti esattamente come le immagini $u \in U$. È quindi possibile ripetere le operazioni del passo

4.3 Algoritmo per il raggruppamento delle immagini rilevanti

precedente a questi oggetti. In particolare, per la prima iterazione, si ha che

$$q_j^1 = \left\{ v_i \in V_j \mid S(v_i, v_{query_j}^1) < \delta_{th} \cdot th_{STEP2} \right\} \quad (4.30)$$

dove δ_{th} è un fattore con cui si pesa la soglia th_{STEP2} . δ_{th} è proporzionale numero di elementi che appartengono ai sottoinsiemi che si stanno confrontando v_i e v_{query} ed è definito come segue

$$\delta_{th}(n) = \frac{1}{2} \left(1 + \left(\frac{2}{\pi} \tan^{-1}(-\lambda(n - n^*)) \right) \right) \quad (4.31)$$

dove n^* è il numero di oggetti medio che si vorrebbero in ogni sottoinsieme, λ definisce quanto questo vincolo deve essere stringente e $n = \text{Max} [|v_i|, |v_{query}|]$. Nella figura 4.17 è mostrato come varia il valore di $\delta_{th}(n)$ in funzione dei parametri λ e n^* . Il colore uguale indica che le curve corrispondono allo stesso valore di n^* , mentre lo stesso tipo di linea delle curve indica che queste corrispondono allo stesso valore di λ . In questo modo se un sottoinsieme è composto da un numero di oggetti maggiore di n^* la soglia viene pesata con un fattore minore di uno e quindi si inserisce un nuovo oggetto solo se questo risulta essere molto simile a quello con cui si è confrontato. Viceversa, se il sottoinsieme è composto da pochi oggetti la soglia è pesata con un fattore prossimo ad uno. Ad ogni iterazione j si ha che

$$V_j = V_{j-1} \setminus q_{j-1}^1, \quad (4.32)$$

quando $V_j = \emptyset$ la prima iterazione di questo passo finisce. Dall'equazione 4.30 è evidente che se la soglia è troppo bassa, l'insieme Q^1 , composto da tutti i sottoinsiemi q_j^1 , corrisponde all'insieme V e questo vuol dire che nessun elemento $v \in V$ è stato unito in un unico sottoinsieme q . In questo

4.3 Algoritmo per il raggruppamento delle immagini rilevanti

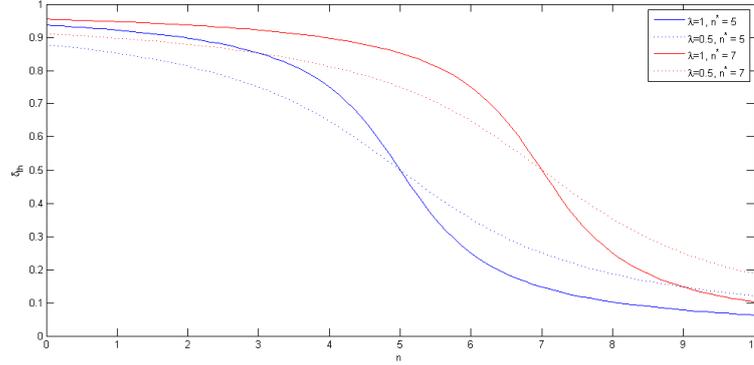


Figura 4.17: Esempio della variazione di $\delta_{th}(n)$, equazione 4.31, al variare di λ e n^* .

caso la soglia th_{STEP2} viene incrementata, in generale, questo accade ogni volta che $Q^k = Q^{k-1}$. Se $Q^k \neq Q^{k-1}$ vuol dire che qualche sottoinsieme è stato unito, e il passo viene ripetuto con lo stesso valore di th_{STEP2} . Alla k -esima iterazione si ha

$$q_j^k = \left\{ q_i^k \in Q_j^k \mid S(q_i^k, q_{query_j}^k) < \delta_{th} \cdot th_{STEP2} \right\}, \quad (4.33)$$

e

$$Q_j^k = Q_{j-1}^k \setminus q_{j-1}^k \quad (4.34)$$

dove la soglia δ_{th} in questo caso è proporzionale a $n = \text{Max} \left[|q_i^k|, |q_{query_j}^k| \right]$. Quando $Q_j^k = \emptyset$ l'iterazione k -esima finisce e gli insiemi creati q_j^k compongono l'insieme Q^{k+1} che viene utilizzato nella $(k+1)$ -esima iterazione. Questo passo viene interrotto se il numero di $q^k \in Q^k$ è inferiore al numero scelto dall'utente oppure la soglia th_{STEP2} diventa maggiore di un valore massimo.

4. In questo ultimo passo i sottoinsiemi $q^k \in Q^k$, creati al passo precedenti, vengono classificati in base al numero di immagini di cui sono composti e al-

4.4 Valutazione dell'informazione di Fisher per l'estrazione delle caratteristiche di basso livello

l'utente viene mostrata solo una immagine per ogni sottoinsieme. L'immagine scelta può essere la prima inserita nel sottoinsieme oppure l'immagine che ha i vettori che la rappresentano più simili ai vettori che rappresentano il sottoinsieme di appartenenza.

Così facendo è possibile mostrare all'utente poche immagini che però rappresentano un numero di immagini molto maggiore, dando importanza a quelle immagini che potrebbero essere rilevanti ma a cui non viene data la giusta importanza perchè non classificate nelle primissime posizioni. Questo algoritmo è del tutto generale e si può applicare con qualsiasi metrica. Infatti non c'è nessun legame tra l'algoritmo e il modo scelto per rappresentare e confrontare due immagini, che però inevitabilmente ne influenza i risultati. Inoltre, poichè non c'è nessun vincolo sulla selezione delle immagini rilevanti che compongono l'insieme U , creato al primo passo, queste possono essere scelte in qualunque maniera utilizzando una qualsiasi motore di ricerca.

4.4 Valutazione dell'informazione di Fisher per l'estrazione delle caratteristiche di basso livello

Un metodo per rendere più robusti i vettori che rappresentano le caratteristiche di basso livello, quali tessiture, colore e bordi, può essere quello di calcolarli nelle zone delle immagini di interesse, [29]. Le zone di interesse possono essere definite come quelle zone ad alto contenuto informativo. Utilizzando la relazione trovata tra informazione di Fisher, rispetto al parametro di traslazione, equazione 3.3, ed i polinomi di Zernike, equazioni 3.11, 3.12, 3.13, è facile, e non costoso dal punto di vista computazionale, la realizzazione di una mappa che sia proporzionale al

4.4 Valutazione dell'informazione di Fisher per l'estrazione delle caratteristiche di basso livello

contenuto informativo del generico punto \mathbf{x}_0 e di un suo intorno. Questo tipo di approccio ha senso in immagini naturali, che rappresentano più oggetti, come ad esempio quelle mostrate nella figura 3.13, piuttosto che su immagini più semplici mostrate nella figura 4.24. Nelle figure 4.20 e 4.21 sono mostrati degli esempi di



Figura 4.18: Immagine che appartiene alla classe dinosauri del database *COREL-1000-A*.



Figura 4.19: Immagine che appartiene alla classe palazzi del database *COREL-1000-A*.

mappe calcolate utilizzando le immagini mostrate, rispettivamente, nelle figure 4.18 e 4.21. Per rimuovere le regioni a basso contenuto informativo, quelle a cui corrisponde un basso valore sulla mappa J_b viene utilizzato l'algoritmo detto *region growing* [40]. Questo algoritmo permette di selezionare tutte le regioni connesse che presentano una determinata caratteristica. Infatti, nel caso specifico,

4.4 Valutazione dell'informazione di Fisher per l'estrazione delle caratteristiche di basso livello

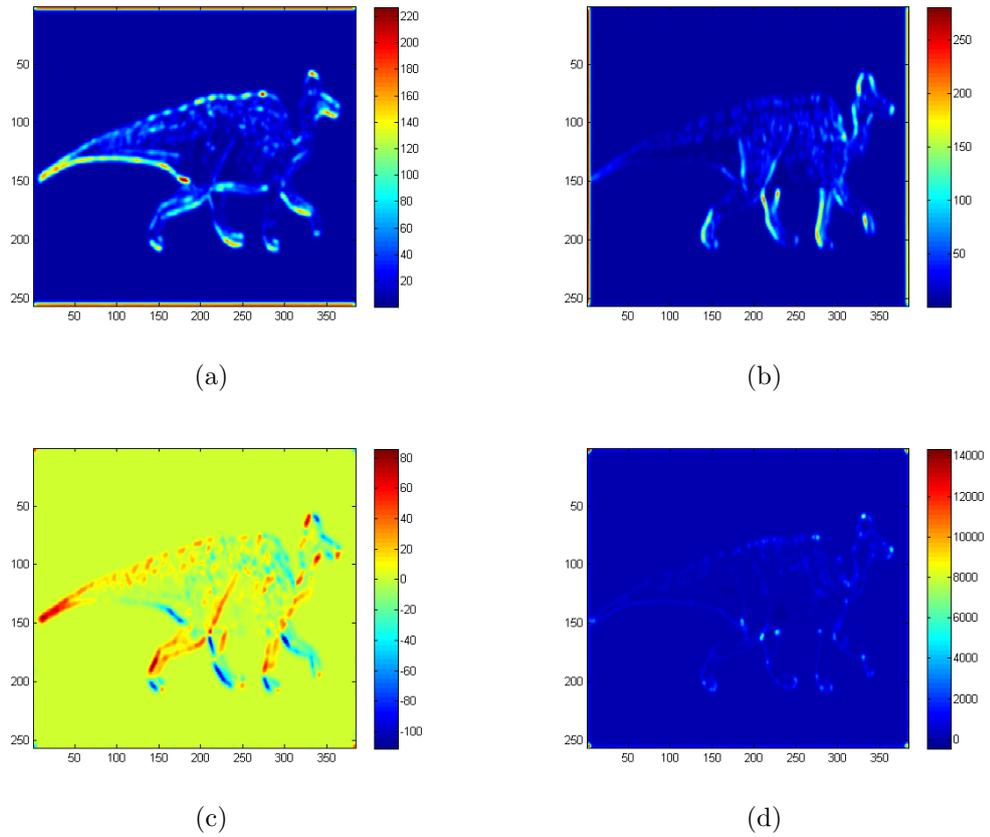


Figura 4.20: Mappe $\varepsilon_{f_{x_1}}$ 4.20(a), $\varepsilon_{f_{x_2}}$ 4.20(b), $\varepsilon_{f_{x_1}f_{x_2}}$ 4.20(c) e J_b 4.20(d), calcolate sull'immagine 4.18.

4.4 Valutazione dell'informazione di Fisher per l'estrazione delle caratteristiche di basso livello

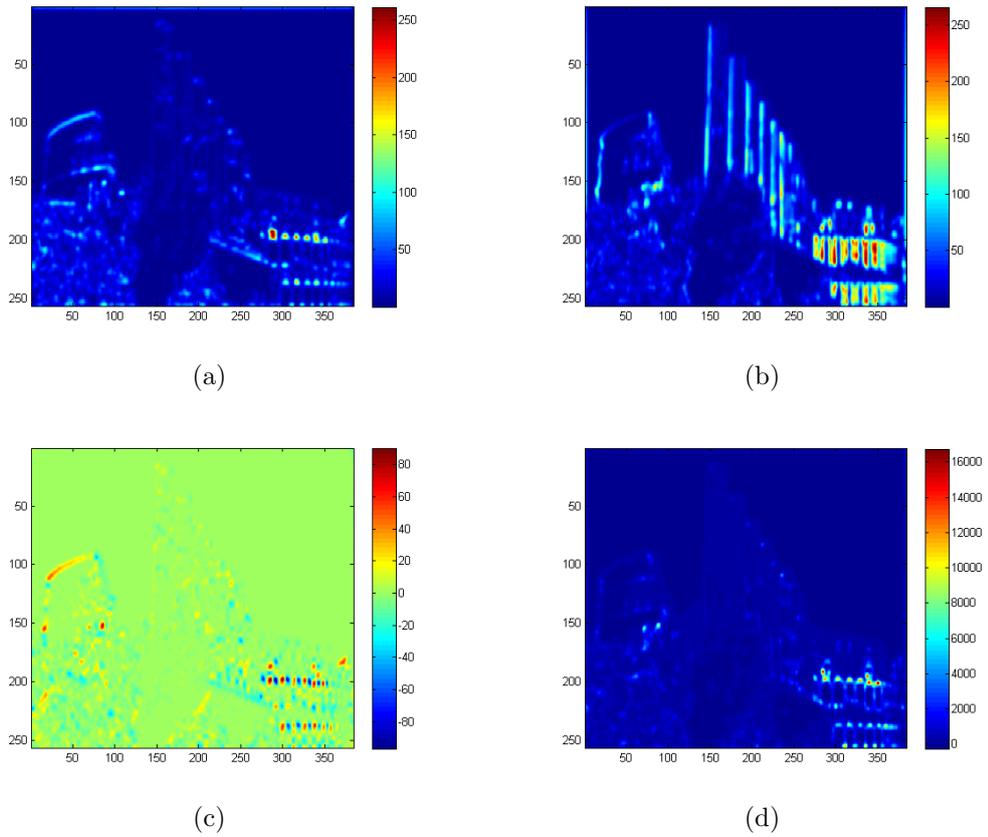


Figura 4.21: Mappe $\varepsilon_{f_{x_1}}$ 4.21(a), $\varepsilon_{f_{x_2}}$ 4.21(b), $\varepsilon_{f_{x_1}f_{x_2}}$ 4.21(c) e J_b 4.21(d), calcolate sull'immagine 4.19.

4.4 Valutazione dell'informazione di Fisher per l'estrazione delle caratteristiche di basso livello

partendo da alcuni punti fissi, detti *semi*, questo algoritmo seleziona tutti i punti connessi al *seme* che hanno un valore della mappa J_b prossimo allo zero. I *semi* vengono scelti nella zone periferica dell'immagini, in questo modo il soggetto principale, che solitamente è nel centro dell'immagine, rimane inalterato. Infatti, una volta selezionata una zona se questa confina con dei punti a cui corrisponde un valore di J_b maggiore di zero l'algoritmo del *region growing* si interrompe. Nelle

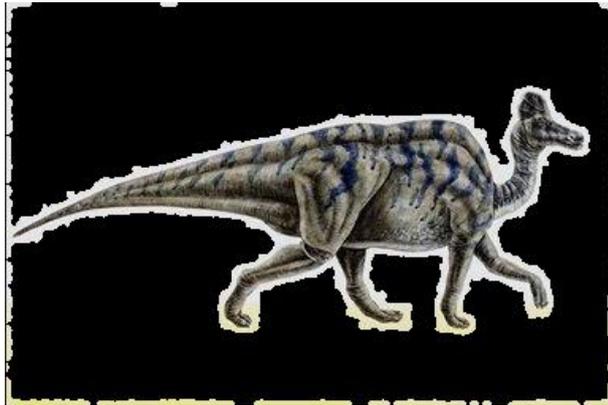


Figura 4.22: Esempio della zona non considerata per l'estrazione delle caratteristiche, calcolata sull'immagine 4.18.



Figura 4.23: Esempio della zona non considerata per l'estrazione delle caratteristiche, calcolata sull'immagine 4.19.

figure 4.22 e 4.23 vengono mostrate le zone dell'immagine che vengono selezionate e che quindi non vengono utilizzate per l'estrazione delle caratteristiche di basso livello.

4.5 Risultati sperimentali

4.5.1 Analisi e confronto di tessiture

Il metodo per la caratterizzazione ed il confronto delle tessiture, paragrafo 4.2.1, è stato valutato utilizzando il database *Vision Texture*, [3]. Le sperimentazioni sono state condotte utilizzando 40 classi diverse di tessiture, rappresentate da 16 elementi ognuna, mostrate nella figura 4.24. Originariamente ogni classe è

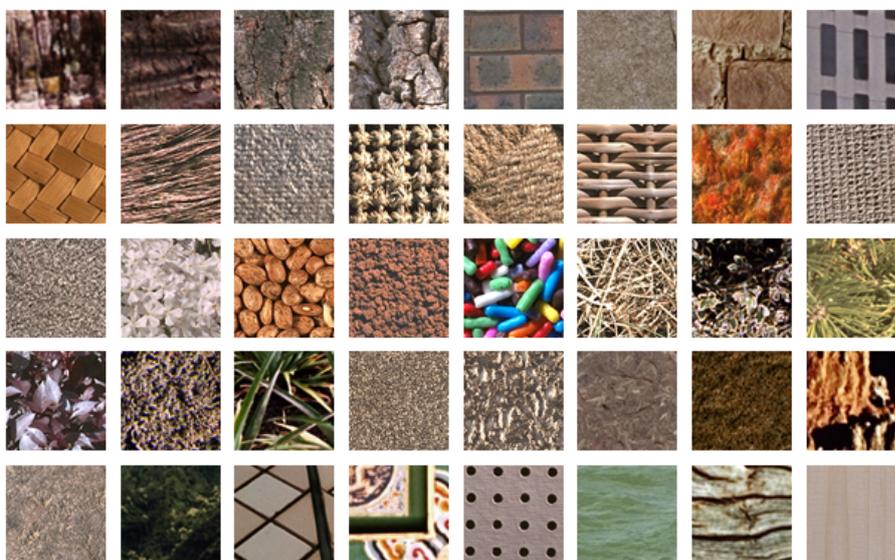


Figura 4.24: Esempi delle classi di tessiture utilizzate nelle sperimentazioni.

una immagine di 512×512 pixel che rappresenta una particolare tessitura. Per ottenere i 16 rappresentanti di questa classe, l'immagine viene divisa in 16 parti non sovrapposte di 128×128 pixel. In questo modo si ottiene un *dataset* composta da 640 immagini di dimensione 128×128 pixel. Le prestazioni del metodo descritto nel paragrafo 4.2.1 sono state valutate calcolando la percentuali di recupero media. In particolare, ogni immagine del *dataset*, è stata considerata come *query*. Una volta effettuata la classificazione di tutte le immagini nell'archivio in base alla similitudine, equazione 4.18, sono state valutate quante immagini della stessa classe dell'immagine *query*, erano classificate nelle prime sedici posizioni.

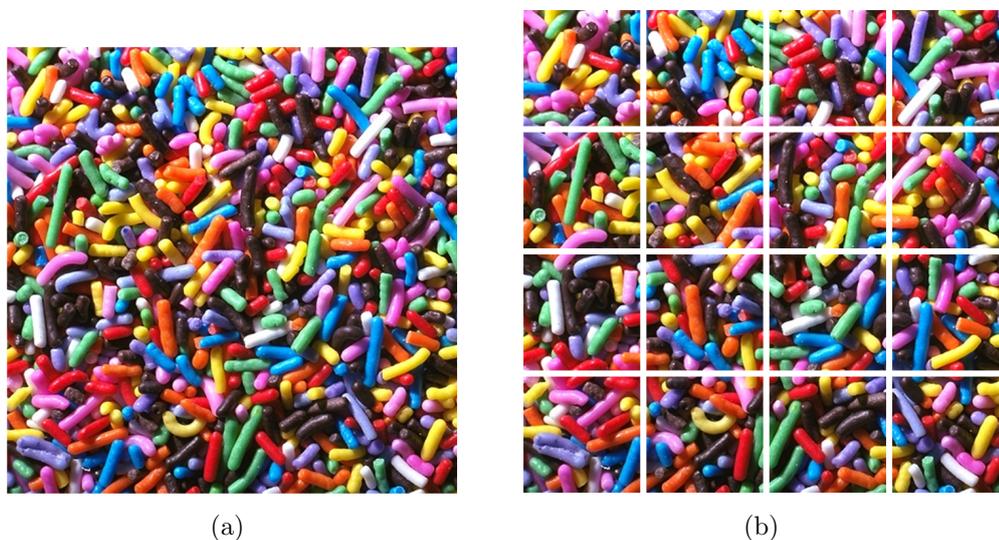


Figura 4.25: Immagine originale di 512×512 pixel, 4.25(a), divisione in 16 immagini non sovrapposte di 128×128 pixel, 4.25(b).

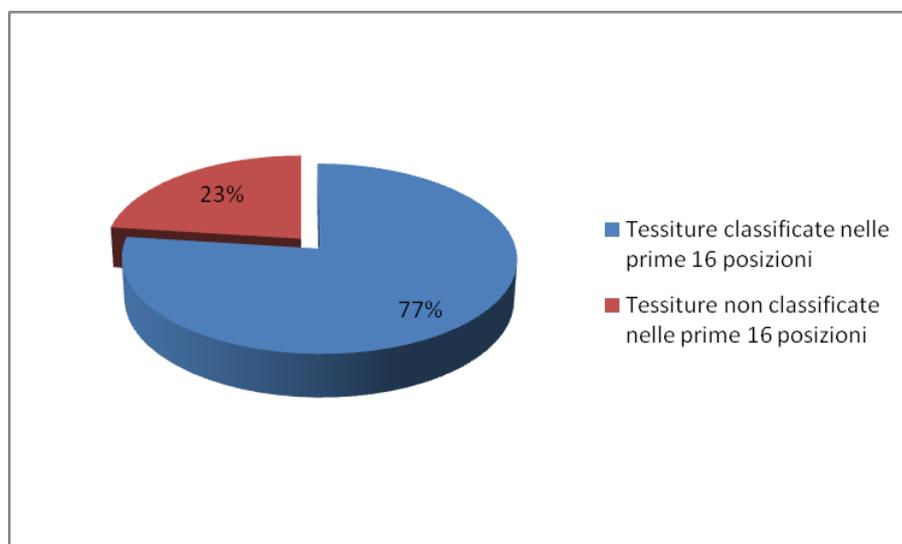


Figura 4.26: Risultati ottenuti utilizzando la funzione di Laguerre Gauss $n = 1$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza di Kullback-Leibler.

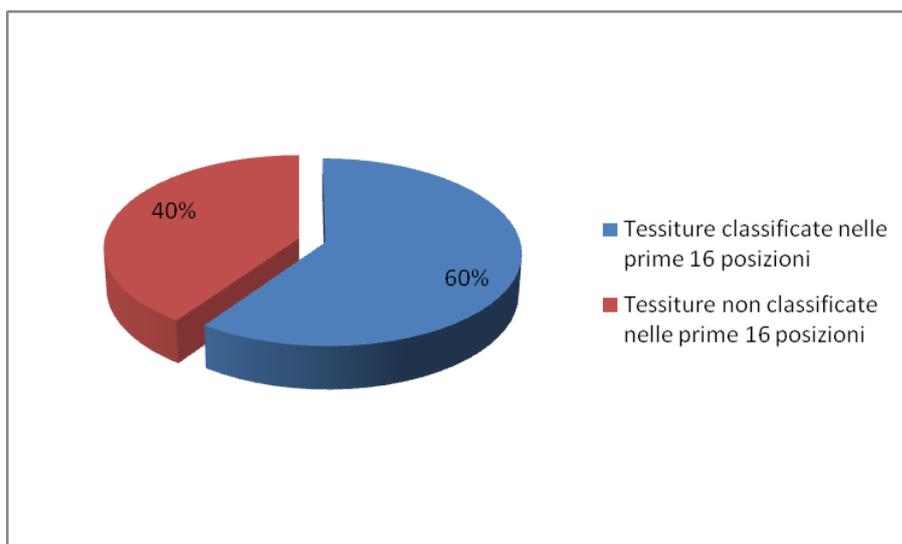


Figura 4.27: Risultati ottenuti utilizzando la funzione di Laguerre Gauss $n = 1$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza euclidea.

Nella figura 4.26 sono mostrati i risultati che si sono ottenuti utilizzando una sola funzione di Laguerre Gauss, quella identificata dagli indici $n = 1$ e $k = 0$, a tre diverse risoluzioni. In questo modo le tessiture vengono rappresentate con 6 gaussiane generalizzate, che corrisponde ad avere un vettore composto da 12 elementi. Perché, come descritto in dettaglio nel paragrafo 4.2.1, ogni gaussiana generalizzata è completamente caratterizzata da due parametri. Nella figura 4.26 sono mostrati i risultati ottenuti utilizzando come metrica la distanza di Kullback-Leibler, mentre, nella figura 4.27, come metrica è stata utilizzata la distanza euclidea. Come si può notare le prestazioni ottenute utilizzando la distanza di Kullback-Leibler sono nettamente migliori rispetto a quelle ottenute utilizzando la distanza euclidea. Un comportamento analogo si ha quando vengono utilizzate due funzioni di Laguerre Gauss, quelle definite dalla coppia di indici $n = 1$ e $k = 0$, e $n = 3$ e $k = 0$, ognuna a tre diverse risoluzioni, come mostrato nelle figure 4.28 e 4.29. Anche in questo caso si ha un miglioramento utilizzando la distanza di Kullback-Leibler. Inoltre, confrontando la figura 4.26 con la figura 4.28, e la figura 4.27 con la figura 4.29, si nota come utilizzando anche la funzione

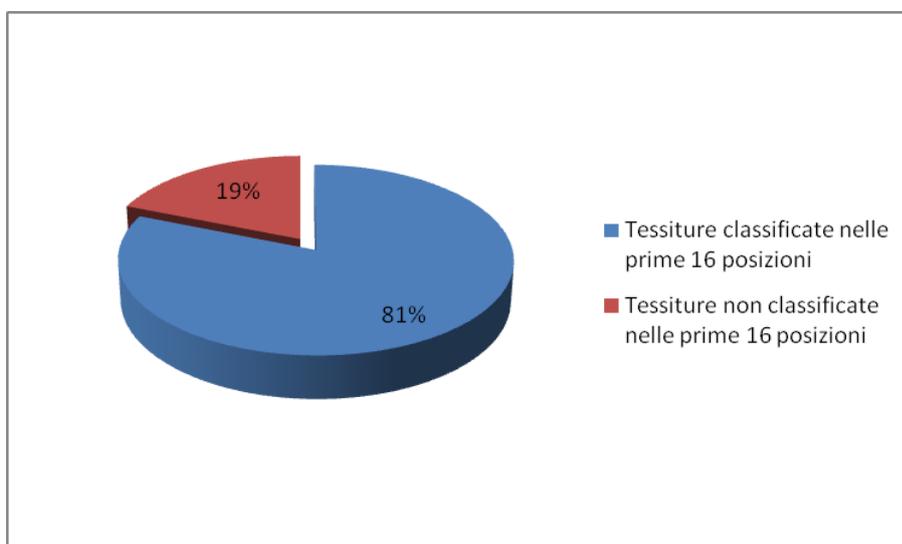


Figura 4.28: Risultati ottenuti utilizzando le funzioni di Laguerre Gauss $n = 1$ e $k = 0$, $n = 3$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza di Kullback-Leibler.

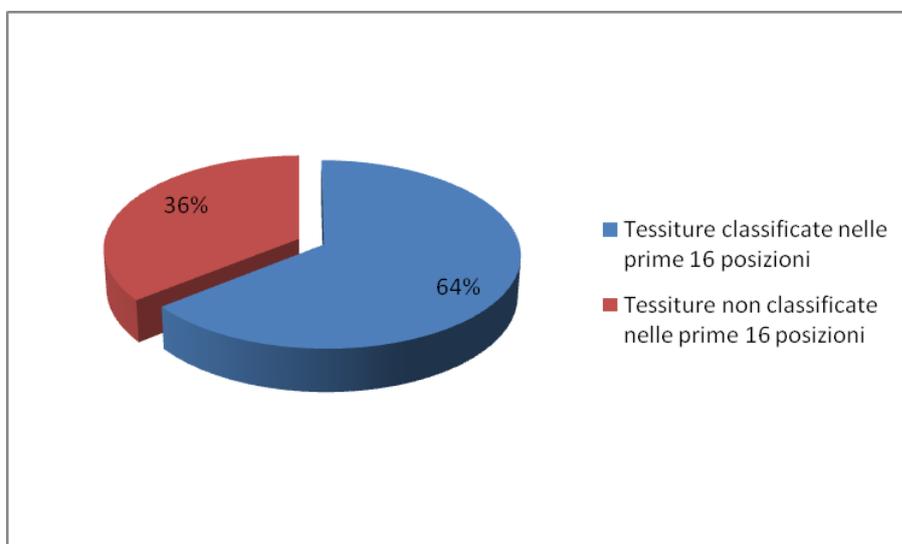


Figura 4.29: Risultati ottenuti utilizzando le funzioni di Laguerre Gauss $n = 1$ e $k = 0$, $n = 3$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza euclidea.

di Laguerre Gauss $n = 3$ e $k = 0$ si ha un miglioramento delle percentuali di recupero. Ciò, ovviamente, implica l'utilizzo di 12 gaussiane generalizzate, per rappresentare ogni tessitura, e quindi un vettore composto da 24 elementi. È stata scelta proprio questa funzione di Laguerre Gauss perchè con le altre non si aveva un miglioramento così netto delle prestazioni. I risultati mostrano che l'uti-

Wavelet	Kullback-Leibler	Distanza euclidea
D_2	76.92%	62.22%
D_4	76.57%	61.18%
D_6	75.51%	60.35%
D_8	75.63%	61.73%
D_{10}	74.66%	59.73%

Tabella 4.1: Risultati ottenuti utilizzando alcune *wavelet* di Daubechies, [37].

lizzo della distanza di Kullback-Leibler, che di fatto calcola la verosimiglianza tra due gaussiane generalizzate, corrisponde ad un miglioramento delle prestazioni, come mostrato non solo nei risultati ottenuti utilizzando le *wavelet* di Laguerre Gauss, ma anche utilizzando alcune *wavelet* di Daubechies, [37]. Inoltre, si può notare che le metodologie descritte per l'analisi delle tessiture in [37] possono essere applicate con ottimi risultati, alle *wavelet* armoniche circolari, in particolare alle *wavelet* di Laguerre Gauss. Un ulteriore aumento delle prestazioni può essere ottenuto tenendo in considerazione anche l'informazione sul colore. Infatti, caratterizzando anche questa caratteristica di basso livello, come descritto nel paragrafo 4.2.2, si ottengono i risultati riportati nella figura 4.30. In questo scenario ogni immagine viene caratterizzata con un vettore di 24 elementi per caratterizzare le tessiture e con un vettore di 6 elementi che caratterizza il colore. Al solito, per confrontare due vettori che rappresentano le tessiture viene utilizzata la distanza di Kullback-Leibler che, come si è visto in precedenza, è più performante della distanza euclidea, mentre per confrontare due vettori che rappresentano il colore si utilizza la distanza euclidea. Così facendo la metrica

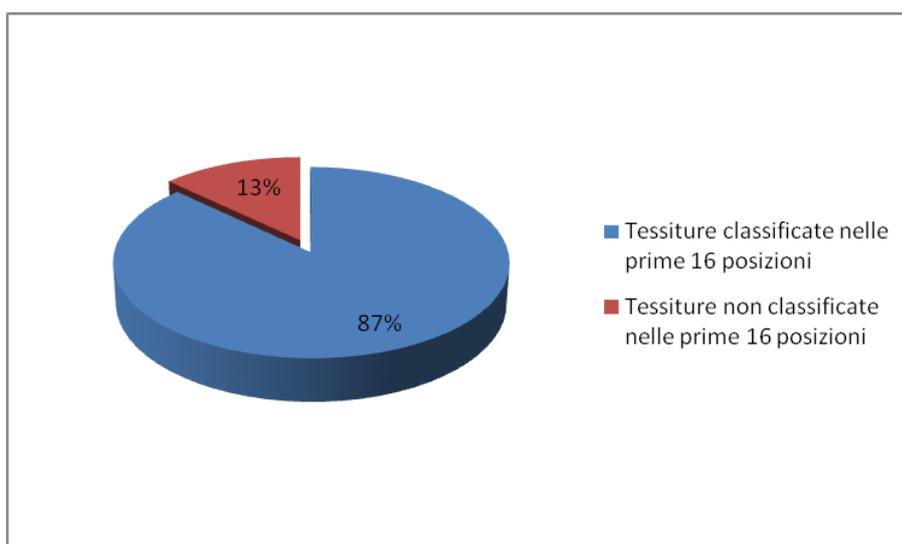


Figura 4.30: Risultati ottenuti utilizzando l'informazione sul colore, le funzioni di Laguerre Gauss $n = 1$ e $k = 0$, $n = 3$ e $k = 0$, a tre risoluzioni, e utilizzando la distanza di Kullback-Leibler.

utilizzata per il confronto di due immagini è la seguente

$$S(f, g) = \delta_t \cdot KLD(f, g) + \delta_c \cdot ED_c(f, g) \quad (4.35)$$

dove empiricamente è stato trovato che i risultati migliori, quelli mostrati nella figura 4.30, si ottengono con $\delta_t = 1$ e $\delta_c = 12$. Nella figura 4.31 sono mostrate

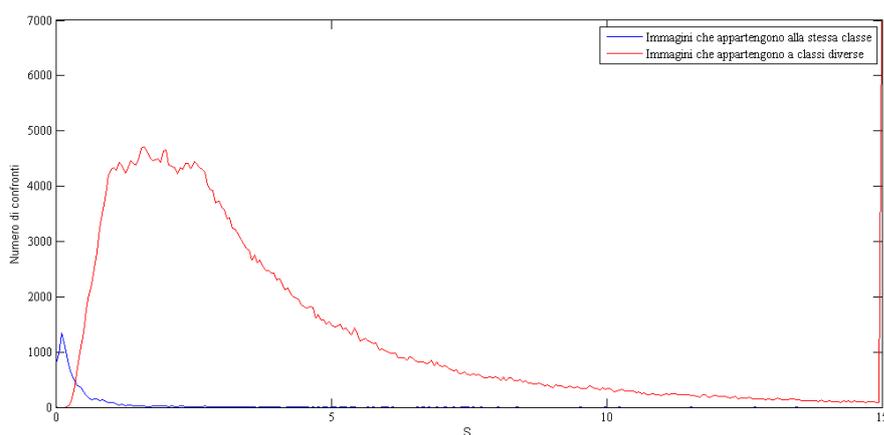


Figura 4.31: Distribuzione dei valori di S , equazione 4.35, tra immagini che appartengono alla stessa classe e immagini che appartengono a classi diverse.

le distribuzioni delle distanze S , equazione 4.35, calcolate tra immagini che appartengono alla stessa classe e immagini che appartengono a classi diverse. Come si può notare le due distribuzioni sono piuttosto separate e l'area di sovrapposizione tra le due distribuzioni è decisamente piccola. Idealmente, si dovrebbero avere due distribuzioni separate, il che significherebbe che i valori di S che si ottengono quando f e g appartengono alla stessa classe non si possono ottenere quando f e g appartengono a classi diverse, che corrisponderebbe ad avere una percentuale di recupero del 100%.

4.5.2 Raggruppamento delle immagini rilevanti

Le prestazioni del metodo descritto nel paragrafo 4.3 sono state valutate utilizzando il *dataset* descritto nel precedente paragrafo. Una prima parte di simulazioni sono state effettuate utilizzando soltanto l'informazione sulle tessiture e l'informazione sul colore. In questo caso si utilizza come metrica quella mostrata nell'equazione 4.35. Le prestazioni sono state valutate considerando:

- numero totale di sottoinsiemi creati: numero totale di sottoinsiemi creati dall'algoritmo considerando tutte le immagini come *query*;
- numero totale di sottoinsiemi “sporchi” creati: numero totale di sottoinsiemi “sporchi” considerando tutte le immagini come *query*;
- numero totale di “intrusi”: numero totale di elementi inseriti nel sottoinsieme sbagliato considerando tutte le immagini come *query*;
- percentuale media di sottoinsiemi “sporchi”: percentuale di sottoinsiemi “sporchi” creati rispetto al numero totale di sottoinsiemi creati;

- numero medio di elementi “intrusi” per ogni cluster “sporco”: numero di elementi inseriti in un sottoinsieme sbagliato rispetto al numero totale di sottoinsiemi “sporchi”;
- numero medio di sottoinsiemi creati: numero di sottoinsiemi creati rispetto al numero totale di *query* utilizzate;
- numero medio di cluster reali: numero di sottoinsiemi ideale in cui dovrebbero essere suddivise le immagini considerate rilevanti.

È importante definire bene cosa si intendo con “intruso” e con sottoinsieme “sporco”. Un “intruso” è una immagine che viene inserita dall’algoritmo in un sottoinsieme sbagliato. Mentre un sottoinsieme “sporco” è un sottoinsieme che contiene almeno un elemento intruso. Nelle tabelle 4.2 e 4.3 è riportato un

	Immagine		Immagine
1	Bark.0000.00	17	Bark.0000.02
2	Bark.0000.08	18	Brick.0005.14
3	Bark.0000.09	19	Bark.0000.11
4	Bark.0000.05	20	Wood.0001.01
5	Bark.0000.12	21	Wood.0001.05
6	Bark.0000.01	22	Flowers.0005.13
7	Bark.0000.04	23	Brick.0005.02
8	Bark.0000.13	24	Leaves.00016.15
9	Bark.0000.06	25	Flowers.0005.12
10	Wood.0001.15	25	Wood.0001.09
11	Bark.0000.10	27	Leaves.0016.09
12	Wood.0001.13	28	Brick.0005.01
13	Wood.0001.02	29	Flowers.0005.09
14	Bark.0000.07	30	Leaves.00016.14
15	Wood.0001.12	31	Leaves.00016.13
16	Wood.0001.08	32	Brick.0005.00

Tabella 4.2: Esempio delle immagini che vengono considerate rilevanti al primo passo dell’algoritmo descritto nel paragrafo 4.3, considerando come *query* l’immagine *Bark.0000.00*.

esempio del funzionamento dell’algoritmo utilizzato per raggruppare le immag-

	Rappresentante del sottoinsieme
1	Bark.0000.05
2	Wood.0001.01
3	Bark.0000.11
4	Brick.0005.14
5	Leaves.0016.13
6	Flowers.0005.09
7	Wood.0001.13
8	Wood.0001.12

Tabella 4.3: Risultato dell’algoritmo descritto nel paragrafo 4.3, considerando come *query* l’immagine *Bark.0000.00*.

ini che vengono considerate rilevanti, mostrate nella tabella 4.2. Invece, nella tabella 4.3 sono mostrati le immagini che sono i rappresentanti dei sottoinsiemi creati. Nella tabella 4.4 sono mostrati i risultati che si ottengono utilizzando la

Numero totale di sottoinsiemi creati	3496
Numero medio di sottoinsiemi “sporchi” creati	319
Numero medio di “intrusi”	974
Percentuale media di sottoinsiemi “sporchi”	9.12%
Numero medio di elementi “intrusi” per ogni cluster “sporco”	3.05
Numero medio di sottoinsiemi creati	5.46
Numero medio di cluster reali	4.67

Tabella 4.4: Risultati ottenuti utilizzando la metrica definita nell’equazione 4.35.

metrica mostrata nell’equazione 4.35, perciò utilizzando soltanto l’informazione sulle tessiture e sul colore. Nella tabella si può notare che il numero di sottoinsiemi in cui è presente almeno un elemento sbagliato è inferiore al 10% sebbene il numero totale di sottoinsiemi creati è molto simile al numero di sottoinsiemi reali. Questo è molto importante perchè se una immagine viene inserita in un sottoinsieme sbagliato, il rappresentante di questo sottoinsieme non la rappresenterà e quindi, se l’utente è interessato proprio a questa immagine, l’algoritmo non lo facilita. Se si utilizza anche l’informazione sui bordi, caratterizzata e confrontata come è descritto nel paragrafo 4.2.3, la percentuale di sottoinsiemi “sporchi” decresce ulteriormente, come è possibile osservare nella tabella 4.5. In

Numero totale di sottoinsiemi creati	3675
Numero medio di sottoinsiemi “sporchi” creati	227
Numero medio di “intrusi”	804
Percentuale media di sottoinsiemi “sporchi”	6.17%
Numero medio di elementi “intrusi” per ogni cluster “sporco”	3.54
Numero medio di sottoinsiemi creati	5.74
Numero medio di cluster reali	4.63

Tabella 4.5: Risultati ottenuti quando l’immagini viene caratterizzata utilizzando le tessiture, il colore ed i bordi.

questo caso, per confrontare due immagini viene utilizzata la metrica mostrata nell’equazione 4.26. I risultati mostrano da un lato che, come ci si aspettava, le prestazioni dell’algoritmo dipendono dalla bontà dei metodi utilizzati per la caratterizzazione e il confronto di due immagini. Dall’altro che il metodo utilizzato per caratterizzare e confrontare i bordi, paragrafo 4.2.3, è efficace. Questo algoritmo è stato applicato anche ad un archivio di immagini composto da 5 classi del database *COREL-1000-A*. Le classi utilizzate sono: autobus, cavalli, dinosauri, fiori e palazzi, gli esempi sono mostrati nella figura 3.13. Prima di analizzare le prestazioni dell’algoritmo si è valutata la bontà della metrica utilizzata per il confronto tra due immagini. In particolare, si sono confrontati i risultati, in termini di percentuale di recupero, calcolando i vettori delle caratteristiche di basso livello considerando l’intera immagine oppure considerando solo le zone ad alto contenuto informativo, selezionate come descritto nel paragrafo 4.4. Come si vede nelle figure 4.32, 4.33, 4.34 calcolando i vettori delle caratteristiche di basso livello nelle zone ad alto contenuto informativo si ottengono percentuali di recupero migliori rispetto all’utilizzo dell’intera immagine. Questo approccio è stato utilizzato nell’algoritmo descritto nel paragrafo 4.3. I risultati sono analizzati con un grafico che mostra la percentuale media di sottoinsiemi “sporchi”, rispetto al numero medio di sottoinsiemi creati. Il grafico, figura 4.35, mostra che si ottengono buone percentuali di sottoinsiemi “sporchi”. È da notare che il

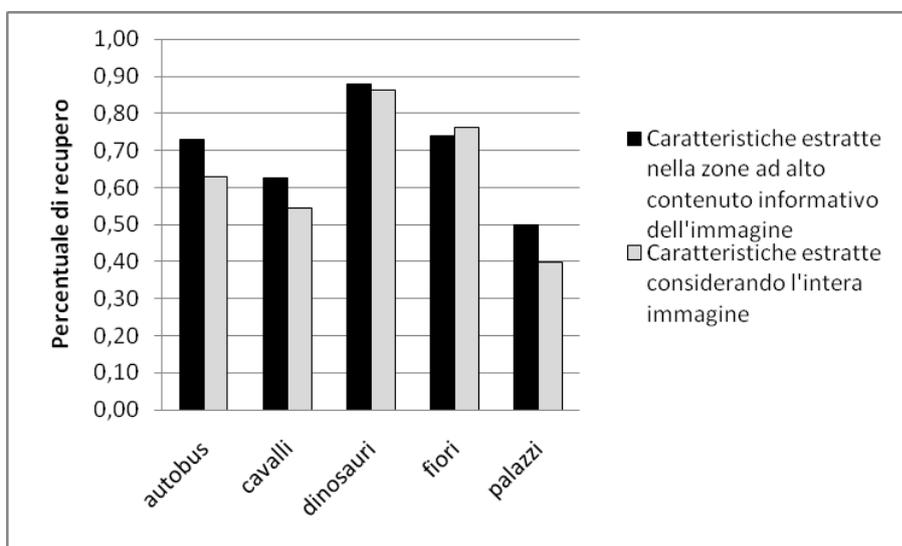


Figura 4.32: Percentuali di recupero di ogni classe.

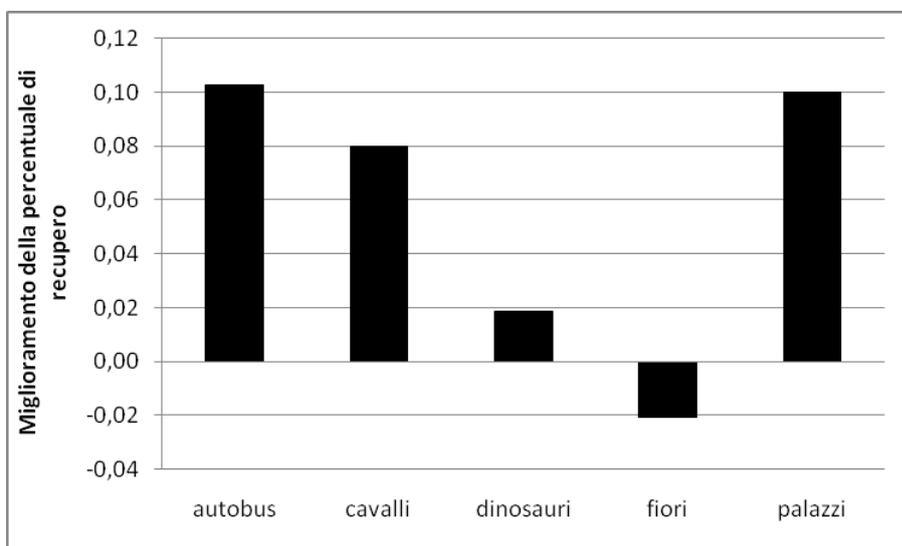


Figura 4.33: Miglioramento delle percentuali di recupero di ogni classe.

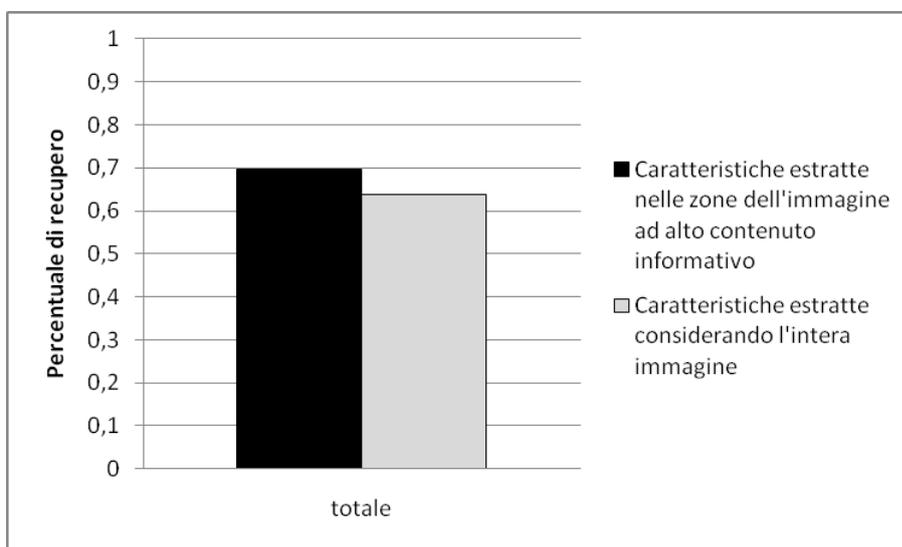


Figura 4.34: Percentuali di recupero totale.

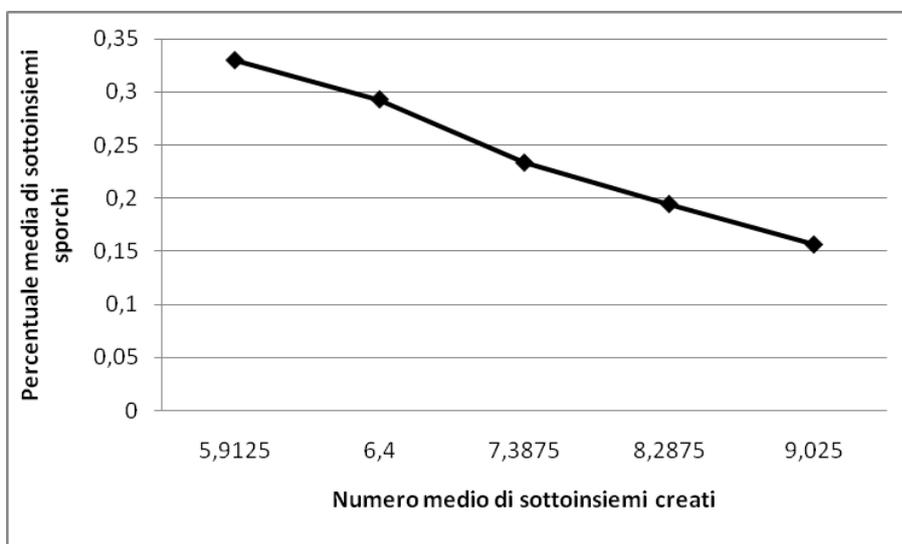


Figura 4.35: Risultati ottenuti utilizzando una parte del database COREL-1000-A.

numero reale di sottoinsiemi è 4.3. Sebbene numericamente i risultati ottenuti siano peggiori di quelli mostrati nelle tabelle 4.4 e 4.5, si deve tener conto che in questo caso le percentuali di recupero è prossima al 70%, rispetto al 90% del caso precedente. Questo è dovuto al fatto che le immagini utilizzate sono più complesse e quindi, a immagini che nominalmente appartengono a classi diverse, potrebbero corrispondere caratteristiche di basso livello simili.

5

Conclusioni

In questo lavoro si è affrontato il problema della ricerca automatica di immagini all'interno di un archivio. Sono stati sviluppati due approcci, che pur nella loro diversità, presentano degli aspetti in comune ed hanno il medesimo obiettivo. Infatti, entrambi hanno come scopo il recuperare tutte le immagini che, sulla base delle loro caratteristiche visive sono rilevanti rispetto ad una immagine fornita dall'utente. Tra gli aspetti che gli approcci hanno in comune il più importante è costituito dal fatto che entrambi si basano sull'utilizzo delle funzioni armoniche circolari. Questa particolare classe di funzioni, a cui appartengono sia le funzioni di Laguerre Gauss, che i polinomi di Zernike, sono state utilizzate sia per la rappresentazione di *pattern* 2D, che per l'analisi delle immagini. Nel capitolo 3 è stato descritto un metodo per il recupero di oggetti all'interno di un archivio. Questo metodo caratterizza l'oggetto, o l'immagine, di interesse con un numero finito di descrittori locali. Vengono quindi selezionate le zone ad alto contenuto informativo, ed in seguito queste vengono rappresentate utilizzando le funzioni armoniche circolari come descrittori di *pattern* 2D. La grandezza delle finestra che pesa la parte radiale dei polinomi di Zernike viene scelta in accordo con l'algoritmo del *QuadTree*, in modo tale da avere sempre una rappresentazione accurata del

pattern. Utilizzando l'operatore di verosimiglianza, calcolato utilizzando i coefficienti dello sviluppo, fatto con le funzioni armoniche circolari, è possibile stimare la posizione e l'eventuale angolo di rotazione del *pattern* all'interno dell'archivio. I risultati ottenuti mostrano che utilizzando i sistemi sviluppati, basati su questa classe di funzioni, si ottengono degli ottimi risultati sia per quanto riguarda la rappresentazione di *pattern* 2D, sia per quanto riguarda la selezione delle zone ad alto contenuto informativo. Infatti, i metodi descritti permettono di ottenere delle prestazioni migliori rispetto a quelle ottenute con metodi presenti in letteratura. Come detto in precedenza, le funzioni armoniche circolari, possono essere utilizzate anche per l'analisi delle immagini. In questo tipo di approccio la rappresentazione di una immagine viene effettuata mediante le caratteristiche globali, come ad esempio le tessiture, il colore ed i bordi. Nel capitolo 4 le funzioni armoniche circolari vengono utilizzate per caratterizzare le tessiture ed i bordi presenti in una immagine. In particolare, per quanto riguarda le tessiture, queste vengono caratterizzate approssimando le densità marginali dei coefficienti *wavelet* con delle gaussiane generalizzate. I bordi vengono invece caratterizzati calcolando l'istogramma della fase dei punti di bordo stimata proprio mediante le funzioni di Laguerre Gauss. Il colore, infine, viene rappresentato utilizzando i momenti del primo, del secondo e del terzo ordine delle componenti cromatiche. Questo approccio per la caratterizzazione ed il confronto delle immagini, viene utilizzato in un algoritmo che, dato un insieme di immagini rilevanti, le raggruppa in sottoinsiemi. In questo modo, mostrando all'utente solo il rappresentante di ogni sottoinsieme, si facilitano le operazioni di analisi dei risultati all'utente ed è possibile mostrargli una panoramica più ampia di immagini a cui potrebbe essere interessato. I risultati sperimentati, descritti nei paragrafi 3.5 e 4.5, mostrano l'efficacia dei sistemi sviluppati e il fatto che le funzioni armoniche circolari sono un

ottimo strumento sia per la descrizione di *pattern* 2D sia per l'analisi delle immagini. Per questo motivo, gli sviluppi futuri riguarderanno la generalizzazione 3D di queste funzioni, che si ottiene sostituendo la parte armonica angolare con le funzioni armoniche sferiche, come descritto in [6, 19]. In questo caso tali funzioni, da un lato potrebbero essere utilizzate come descrittori, dall'altro potrebbero essere sfruttate per analizzare lo spazio-tempo di sequenze video.

Bibliografia

- [1] Columbia objects image library.
<http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php>. 41
- [2] Corel gallery magic (2000). <http://www.corel.com>. 41
- [3] Mit, vision and modeling group, vision texture.
<http://vismod.media.mit.edu/pub/VisTex/VisTex.tar.gz>. 79
- [4] Muvis. <http://muvis.cs.tut.fi/>. 3, 51
- [5] M. K. Varanasi; B. Aazhang. Parametric generalized gaussian density estimation. *The Journal of the Acoustical Society of America*, 86, 1989. 54
- [6] N. Canterakis. 3d zernike moments and zernike affine invariants for 3d image analysis and recognition. *11th Scandinavian Conf. on Image Analysis*, 1999. 94
- [7] A. Yanagawa; W. Hsu; S.-F. Chang. Brief descriptions of visual features for baseline trecvid concept detectors. *ADVENT Technical Report #219-2006-5 Columbia University*, 2006. 51, 61

- [8] A. Neri; M. Carli; V. Palma; L. Costantini. Template matching based on quadtree zernike decomposition. *IS&T SPIE's Electronic Imaging Science and Technology*, 2009. 7, 22, 30, 31
- [9] A. Neri; M. Carli; V. Palma; L. Costantini. Image search based on quadtree zernike decomposition. *Journal of Electronic Imaging*, 2010. 22, 23, 30, 31, 32
- [10] O. A. Bizzetto Penatti; R. da Silva Torres. Color descriptors for web image retrieval: a comparative study. *XXI Brazilian Symp. on Computer Graphics and Image Processing*, 2008. 50
- [11] G. Van de Wouwer; P. Scheunders; D. Van Dyck. Statistical texture characterization from discrete wavelet representations. *IEEE Transaction on Image Processing*, 1999. 53
- [12] L. Capodiferro; E. D. Di Claudio; G. Jacovitti; F. Mangiatordi. Application of local information analysis to salient points extraction. *IASTED International Conference on Signal Processing, Pattern Recognition*, 2008. 22, 24, 31, 32
- [13] Y. Zhu; N. Xiong; J. H. Park; R. He. A web image retrieval re-ranking scheme with cross-modal association rules. *Int. Symp. on Ubiquitous Multimedia Computing*, 2008. 50
- [14] Z. Stejic; Y. Takama; K. Hirota. Genetic algorithm-based relevance feedback for image retrieval using local similarity patterns. *Inform. Process. Manage*, 2003. 44
- [15] A. Yanagawa; S.-F. Chang; L. Kennedy; W. Hsu. Columbia university's baseline detectors for 374 lscm semantic visual concepts. *ADVENT Technical Report #222-2006-8 Columbia University*, 2007. 3, 51

- [16] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transaction on Informatio Theory*, 1962. 22
- [17] A. Neri; G. Jacovitti. Maximum likelihood localization of 2-d patterns in the gauss-laguerre transform domain: Theoretic framework and preliminary results. *IEEE Transaction on Image Processing*, 13(1), 2004. 15, 22, 23, 24, 34
- [18] Y. Xin; M. Pawlak; Simon Liao. Image reconstruction with polar zernike moments. *Pattern Recognition and Image Analysis*, 2005. v, 7, 10, 11
- [19] R. Klein M. Novotni. Shape retrieval using 3d zernike descriptors. *Computer-Aided Design*, 36(11), 2004. 94
- [20] T. Quack; U. Monich; L. Thiele; B. S. Manjunath. Cortina: a system for largescale, content-based web image retrieval. *12th annual ACM Int. Conf. on Multimedia*, 2004. 50
- [21] Z. Zalevsky; I. Ouzieli; D. Mendlovic. Wavelet transform based composite filter for invariant pattern recognition. *Applied Optics*, 35, 1996. 21
- [22] M. Carli; A. Neri; S. Hatipoglu; S. K. Mitra. Texture based classification and segmentation using the laguerre-gauss transform. *Image and Signal Processing and Analysis*, 2001. 53
- [23] C.-W. Chong; P. Raveendran; R. Mukundan. Translation invariants of zernike moments. *Pattern recognition*, 2003. 22
- [24] G. Jacovitti; A. Neri. Multiresolution circular harmonic decomposition. *IEEE Transaction on Signal Processing*, 48(11), 2000. 15, 18

- [25] L. Costantini; F. Mangiatordi; L. Capodiferro; A. Neri. Key points selection by using zernike polynomials. *IS&T SPIE's Electronic Imaging Science and Technology*, 2010. 12, 22, 32
- [26] L. Costantini; L. Capodiferro; M. Carli; A. Neri. A fair ranking method for image database retrieval. *The first International Workshop on Living Web: Making Web Diversity a true asset collocated with the 8th International Semantic Web Conference*, 2009. 51, 70
- [27] L. Costantini; L. Capodiferro; M. Carli; A. Neri. Impact of edges characterization on image clustering. *EUVIP 2nd European Workshop on Visual Information Processing*, 2010. 51, 65
- [28] L. Costantini; P. Sitá; L. Capodiferro; A. Neri. Laguerre gauss analysis for image retrieval based on color texture. *IS&T SPIE's Electronic Imaging Science and Technology*, 2010. 51, 57, 60
- [29] L. Costantini; P. Sitá; M. Carli; A. Neri. Features extraction based on fisher's information. *IS&T SPIE's Electronic Imaging Science and Technology*, 2011. 51, 74
- [30] L. Capodiferro; M. Carli; L. Costantini; A. Neri; V. Palma. Adaptive riesz basis decomposition for image search. *EUSIPCO, 17th European Signal Processing Conference*, 2009. 22, 30, 34
- [31] C. Y. Wee; R. Paramesran. On the computational aspects of zernike moments. *Image and Vision Computing*, 2007. 7
- [32] S. Li; M. C. Lee; C. M. Pun. Complex zernike moments features for shape-based image retrieval. *IEEE Trans. Syst. Man Cybernetics, Part A: Syst. Humans*, 2009. 45

- [33] C. Harris; M. Stephens. A combined corner and edge detector. *4th Alvey Vision Conference*, 1988. 22, 32, 38
- [34] J. Flusser; T. Suk. Rotation moment invariants for recognition of symmetric objects. *IEEE Transaction on Image Processing*, 2006. 22
- [35] Z. Chen; S. K. Sun. A zernike moment phase-based descriptor for local image representation and matching. *IEEE Transaction on Image Processing*, 19, 2010. 22
- [36] M. R. Teague. Image analysis via the general theory of moments. *IEEE Journal Opt. Soc. Am.*, 1980. 7, 21
- [37] M. N. Do; M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kulback-leibler distance. *IEEE Transaction on Image Processing*, 11(02), 2002. x, 50, 51, 53, 60, 83
- [38] J. Z. Wang. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. Patt. Anal. Mach. Intell.*, 2001. 44
- [39] S. Rudinac; M. Rudinac; B. Reljin; M. Uscumlic; G. Zajic. Global image search vs. regional search in cbir systems. *Proc. 8th Intl. Workshop Imag Analy. Multimedia Interactive Services*, 2007. 44
- [40] W. Cui; Z. Guan; Z. Zhang. An improved region growing algorithm for image segmentation. *International Conference on Computer Science and Software Engineering*, 2008. 75