



**UNIVERSITÀ DEGLI STUDI DI PADOVA**

---

**FACOLTÀ DI INGEGNERIA DELL'INFORMAZIONE**

*Corso di Laurea Magistrale in Ingegneria Informatica*

**ANALISI E IMPLEMENTAZIONE DI ALGORITMI PER LA  
REIDENTIFICAZIONE DI TAVOLE IN LEGNO  
ALL'INTERNO DI UNA LINEA DI PRODUZIONE**

*Laureando*

**Andrea Busatto**

*Relatore*

**Ch.mo Prof.**

**Emanuele Menegatti**

*Co-relatore*

**Dott. Ing. Mauro Cossi**

---

ANNO ACCADEMICO 2013/2014

*Alla mia famiglia.*

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Riconoscimento di pattern e features . . . . .	1
1.2	Il legno . . . . .	1
1.3	Riconoscimento di tavole in legno . . . . .	2
1.4	Affinità con il riconoscimento dell'iride . . . . .	2
1.5	Obiettivo . . . . .	3
<b>2</b>	<b>Strumenti</b>	<b>5</b>
2.1	OpenCV . . . . .	5
2.2	OpenMP . . . . .	5
<b>3</b>	<b>Segmentazione</b>	<b>7</b>
3.1	Thresholding . . . . .	7
3.1.1	Thresholding locale . . . . .	9
3.1.2	Thresholding globale . . . . .	10
3.2	Canny edge detection . . . . .	11
3.2.1	Smoothing . . . . .	12
3.2.2	Ricerca dei gradienti . . . . .	12
3.2.3	Soppressione Non-maximum . . . . .	12
3.2.4	Doppio Threshold . . . . .	13
3.2.5	Isteresi . . . . .	13
3.3	Hough transform . . . . .	14
3.4	Active Contour Models . . . . .	15
<b>4</b>	<b>Estrazione del descrittore</b>	<b>17</b>
4.1	Trasformate Wavelet . . . . .	18
4.1.1	Trasformata Wavelet continua . . . . .	18
4.1.2	Trasformata Wavelet discreta . . . . .	18
4.1.3	Haar Wavelet . . . . .	19
4.2	Filtri di Gabor . . . . .	21
4.2.1	Filtro mono-dimensionale . . . . .	22

4.2.2	Filtro bidimensionale . . . . .	22
4.3	Filtri Log-Gabor . . . . .	23
4.4	Gaussian-Hermite Moments . . . . .	24
4.4.1	Momenti Gaussian-Hermite monodimensionali . . . . .	25
4.4.2	Calcolo ricorsivo . . . . .	26
4.4.3	Finestra di smoothing . . . . .	27
4.4.4	Formulazione generica . . . . .	28
4.4.5	Significato dei momenti Gaussian-Hermite . . . . .	28
4.5	Zero-crossing . . . . .	29
4.6	SIFT . . . . .	29
4.6.1	Identificazione degli estremi locali . . . . .	31
4.6.2	Localizzazione accurata dei keypoints . . . . .	31
4.6.3	Assegnamento dell'orientazione . . . . .	32
4.6.4	Descrittore dell'immagine . . . . .	32
<b>5</b>	<b>Implementazione</b>	<b>33</b>
5.1	Introduzione . . . . .	33
5.1.1	Posizionamento delle stazioni . . . . .	33
5.1.2	Riconoscimento con inchiostro . . . . .	35
5.1.3	Invarianza a fattori esterni . . . . .	37
5.2	Estrazione della tavola . . . . .	37
5.2.1	Bordo verticale . . . . .	37
5.2.2	Bordi superiore e inferiore . . . . .	38
5.2.3	Ransac con pesatura Gaussiana . . . . .	39
5.2.4	Invarianza alla rotazione . . . . .	40
5.2.5	Invarianza alla scala . . . . .	41
5.3	Creazione del descrittore . . . . .	41
5.3.1	Haar Wavelet . . . . .	46
5.3.2	VDi . . . . .	48
5.3.3	Gaussian Hermite moments . . . . .	49
5.4	Distanza di matching . . . . .	54
5.4.1	Haar wavelet . . . . .	54
5.4.2	VDi . . . . .	55
5.4.3	Gaussian-Hermite moments . . . . .	56
<b>6</b>	<b>Testing dell'algorithmo</b>	<b>57</b>
6.1	Dataset . . . . .	57
6.2	Definizione dei test . . . . .	58
6.3	Risultati . . . . .	59
6.3.1	Haar Wavelet . . . . .	59
6.3.2	VDi . . . . .	60

6.3.3	Gaussian Hermite moments . . . . .	64
6.4	Tempi computazionali . . . . .	67
6.5	Conclusioni sui risultati . . . . .	67
<b>7</b>	<b>Installazioni</b>	<b>71</b>
7.1	iNKLESS iDSCAN . . . . .	71
7.2	Berneck (Brasile) . . . . .	71
7.3	Hasslacher (Austria) . . . . .	71
<b>8</b>	<b>Conclusioni</b>	<b>75</b>
8.1	Problemi attuali . . . . .	76
8.2	Sviluppi futuri . . . . .	76
<b>9</b>	<b>Appendice</b>	<b>79</b>
9.1	Test di robustezza . . . . .	79
	<b>Bibliography</b>	<b>85</b>

## Sommario

Il matching di oggetti è uno dei campi più studiati nell'ambito della computer vision. Il riconoscimento di un particolare soggetto in immagini acquisite in momenti diversi è essenziale in molteplici situazioni: da applicazioni industriali all'autenticazione tramite informazioni biometriche.

Il lavoro presentato si concentra in particolare nell'identificazione e accoppiamento di tavole in legno utilizzando immagini ottenute in diverse fasi della catena di lavorazione; il matching delle tavole è necessario per poter fondere i dati ricavati dall'analisi della tavola da parte di più macchine. Non sono stati trovati in letteratura sistemi esistenti che eseguono il compito prefissato sulla tipologia di tavole studiata.

Il sistema di riconoscimento consiste nella segmentazione automatica della tavola all'interno dell'immagine, l'applicazione di accorgimenti geometrici al fine di garantire l'invarianza a trasformazioni affini, il calcolo di un descrittore univoco per la tavola e infine il matching con le tavole presenti nella coda del sistema.

Sono stati implementati e confrontati tre diversi metodi per costruire il descrittore delle tavole: Haar Wavelet, Gaussian-Hermite Moments e un metodo basato su valutazioni statistiche denominato come VDi (Variance of Difference).

Il sistema ha ottenuto percentuali di matching incondizionato pari a 99.9869% simulando la situazione reale di funzionamento in un database di 15191 immagini di tavole in pino radiata acquisite all'interno di un impianto equivalente a quelli dove sarà installato.

# Capitolo 1

## Introduzione

### 1.1 Riconoscimento di pattern e features

Per riconoscimento di pattern si intende l'analisi di un oggetto per identificare possibili costituenti di pattern, che permettono di descrivere l'oggetto servendosi solo di tali elementi.

Una feature è un'informazione rilevante all'interno di un set di dati; è possibile descrivere univocamente un oggetto selezionando le features più importanti, creando così un descrittore. Un descrittore può rappresentare l'oggetto a cui è legato in uno spazio dimensionale con un numero limitato di gradi di libertà, permettendo sia di diminuire la complessità computazionale del matching sia di enfatizzare il grado di unicità dell'entità all'interno del set. L'insieme di tutti i possibili descrittori forma lo spazio dei descrittori, all'interno del quale tutti i possibili oggetti sono rappresentati. Ne deriva che più lo spazio dei descrittori è ampio e più difficilmente due entità saranno legate allo stesso descrittore; allo stesso tempo uno spazio dei descrittori ampio comporta una complessità computazionale elevata sia in fase di costruzione sia in fase di matching.

Per identificare le migliori features presenti sulla superficie di una tavola in legno è necessaria un'analisi approfondita del materiale in questione.

### 1.2 Il legno

Il legno è il materiale ricavato dai fusti delle piante, in particolare da alberi e arbusti. Il legno è impiegato utilmente dall'uomo sin dall'antichità. Il legno è classificato in hardwood e softwood: il primo si ricava dalle conifere (come pino e abete) mentre il secondo si ricava dalle angiosperme (come quercia e noce). Il legno ricavato da diverse specie presenta diverse caratteristiche di

densità, venatura, colore e durezza. Osservando un tronco tagliato in sezione trasversale si possono distinguere due zone: la zona centrale di colore scuro è detta durame, la zona esterna di colore chiaro è detta albarno.

I nodi sono il prolungamento di un ramo all'interno del fusto o di un ramo più grande; i nodi sono presenti perchè i rami crescono a partire dal midollo sviluppandosi verso la parte esterna del tronco: quando un ramo si secca lascia dei nodi. Le venature sono segni tipici del legno, dovuti alla particolare struttura del materiale. Quando il legno viene tagliato parallelamente all'asse del tronco, le venature appaiono drittte. Le venature sono frutto dello sviluppo del tronco, che avviene con il deposito di uno strato di legno nuovo tra la cortecchia e il vecchio legno. Al variare della tipologia di legno, le venature possono essere più o meno marcate.

### **1.3 Riconoscimento di tavole in legno**

La superficie di una tavola in legno presenta varie caratteristiche visive che permettono ad un soggetto umano di riconoscere una tavola da un'altra: le più affidabili sono i nodi, le venature e l'intensità del colore. Nelle tipologie di legno prese in considerazione in questo lavoro (pino radiata e abete) i nodi non sono presenti sulla totalità delle tavole, quindi le caratteristiche individuate che possono rappresentare una tavola sono le venature e l'intensità dei tre canali rosso-verde-blu.

È necessaria l'acquisizione di una foto della tavola a risoluzione adeguata, operazione già eseguita all'interno della catena di lavorazione in cui viaggiano le tavole.

Per quanto riguarda il riconoscimento di tavole in legno nel paper [1] è stato proposto un sistema che si basa sulla posizione reciproca dei nodi, considerazione non utilizzabile nel sistema presentato.

### **1.4 Affinità con il riconoscimento dell'iride**

Il problema di riconoscere una tavola in legno è in parte riconducibile al problema del riconoscimento dell'iride: in entrambi i casi l'oggetto di interesse contiene delle informazioni che sono date da particolari pattern e forme contenuti all'interno dell'immagine. L'iride è composta da un numero di strati che formano un pattern casuale ed unico, per questo motivo il riconoscimento bio-



metrico utilizzando l'iride umana ha riscosso un grande successo. Il pattern formato dalle fibre e dalle venature nel legno è altrettanto unico e casuale, nonostante abbia forme diverse rispetto al collaretto dell'occhio.

I sistemi di riconoscimento di individui basati sul riconoscimento dell'iride sono stati ampiamente studiati in letteratura, quindi sono stati proposti molteplici metodi per il matching di iridi umane. Nei primi anni novanta John Daugman [2] ha implementato un sistema di riconoscimento dell'iride che ottiene risultati perfetti basato sulle 2D Gabor Wavelet. Wildes et al. [3] hanno sviluppato un sistema che ottiene ottimi risultati su un database di 520 immagini. Boles e Boashash [4] hanno proposto un algoritmo basato sulla Wavelet transform zero crossing ottenendo buoni risultati nel descrivere univocamente le differenti iridi. Libor Masek [5] nel suo lavoro di tesi ha realizzato un sistema con fattori FAR 0.005% e FRR 0.238% che utilizza i filtri Log-Gabor. Ma et al. [6] hanno proposto un sistema basato sui momenti di Gauss-Hermite che ottiene percentuale di riconoscimento incondizionato pari a 99.60% sul CASIA Iris Database. Nam et al. [7] calcolano i descrittori dell'iride umana utilizzando il passaggio per lo zero dei segnali orizzontali sulla regione estratta. Lim et al. [8] realizzano un sistema basato su wavelet bidimensionali. Ali et al. [9] applicano la wavelet di Haar per costruire un descrittore dell'iride, ottenendo un sistema con fattori FAR 0.001% e FRR 0.55% su un database di 240 immagini. Elgamal et al. [10] applicano la tecnica Discrete Wavelet Transformation affiancata a K-Nearest Neighbor per implementare un sistema di reidentificazione basato sull'iride con percentuali di riconoscimento incondizionato del 99.5% su un campione di 2240 immagini. Panganiban et al. [11] applicano la scomposizione tramite wavelet sull'iride estratta per poi analizzare i risultati ottenibili. Ma et al. [12] applicano la trasformata wavelet sull'iride scomposta in segnali monodimensionali e ottengono una percentuale del 100% testando il sistema su 2255 immagini.

Vista la mancanza di letteratura nel campo del riconoscimento di tavole in legno, l'affinità con un problema importante come il riconoscimento dell'iride ha comportato una forte ricerca in questa direzione con l'ottica di sfruttare tali conoscenze per la realizzazione del proprio sistema.

## 1.5 Obiettivo

L'obiettivo è l'implementazione di un sistema in grado di riconoscere tavole in legno utilizzando immagini acquisite in diversi istanti della catena di lavorazione. Poiché il sistema sarà utilizzato in impianti industriali reali deve essere

preciso, robusto e rispettare determinati tempi di esecuzione. Il linguaggio di programmazione utilizzato è C++. Per l'implementazione iniziale (5) è stata utilizzata la libreria OpenCV, mentre il sistema finale è stato implementato completamente dalla radice utilizzando una libreria proprietaria. Per testare il sistema si ha a disposizione una database di 15191 coppie di immagini contenenti tavole di pino radiata appositamente raccolte in un impianto simile a quelli dove il sistema sarà installato (6). Successivamente (7.3) si testa il sistema su un ulteriore database di 5295 immagini contenenti tavole di abete. Il legno di abete presenta caratteristiche visuali meno marcate rispetto al pino radiata, per questo motivo si desidera mettere alla prova il sistema anche su tale tipologia di input.

# Capitolo 2

## Strumenti

### 2.1 OpenCV

OpenCV (Open Source Computer Vision Library) è una libreria software open source per computer vision e machine learning. La libreria è sotto licenza BSD quindi è facile per gli utilizzatori modificarla o ottimizzarla. Contiene più di 2500 algoritmi ottimizzati, tra cui algoritmi che fanno parte dello stato dell'arte della computer vision. Tali algoritmi possono essere utilizzati per molteplici funzioni, tra cui: il riconoscimento facciale, l'identificazione di oggetti, la classificazione di azioni umane, il tracking video, l'estrazione di modello tridimensionali di oggetti, la produzione di nuvole di punti da camere stereo, e molte altre.

Tra le grandi compagnie che utilizzano OpenCV si possono citare Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota.

La libreria ha interfacce per C++, C, Python, Java e Matlab inoltre è dotata di supporto multiplatforma (Windows, Linux, Android, Mac OS). Attualmente è in corso di sviluppo l'implementazione di interfacce per CUDA e per OpenCL.

### 2.2 OpenMP

OpenMP (Open Multiprocessing) è un insieme di procedure per la creazione di applicazioni parallele su sistemi che presentano memoria condivisa. È supportato da linguaggi di programmazione come C/C++ e Fortran.

Le porzioni di codice marcate con le direttive di OpenMP vengono eseguite concorrentemente da un certo numero di thread, allocati dal run-time system ai processori disponibili. Ogni thread esegue la propria porzione di codice in maniera indipendente, permettendo di diminuire il tempo computazionale necessario ad eseguire le operazioni richieste.

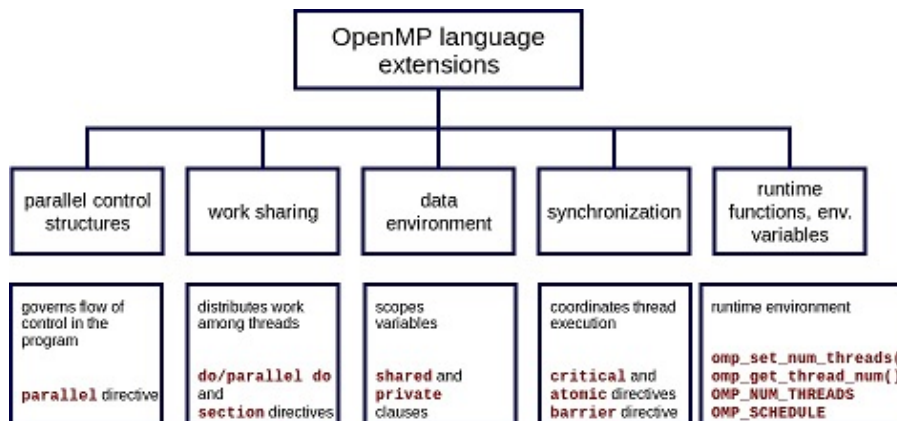


Figura 2.1: Struttura semplificata di OpenMP.

# Capitolo 3

## Segmentazione

Il termine segmentazione si riferisce alla partizione di un'immagine in un set di regioni la cui unione copre l'intera immagine. Lo scopo della segmentazione è la scomposizione dell'immagine in aree che saranno successivamente analizzate o la rappresentazione dell'immagine in una modalità che possa semplificare le operazioni successive sull'input. Il risultato della segmentazione può essere un set di contours o un set di aree.

La segmentazione è utile in tutte le applicazioni che richiedono di suddividere e rappresentare diverse regioni dell'input come la localizzazione di: una struttura anatomica in biomedica, un viso per il riconoscimento facciale, una figura umana nell'identificazione di pedoni, un'iride nel riconoscimento biometrico, una vettura nel controllo del traffico, una tavola di legno nel sistema presentato.

In letteratura la segmentazione è un argomento trattato a fondo, i metodi proposti sono numerosi e adattati per particolari situazioni. In questo capitolo vengono presentati i metodi di segmentazione allo stato dell'arte presi in considerazione per essere utilizzati nel sistema.

### 3.1 Thresholding

La tecnica di thresholding suddivide l'immagine utilizzando come elemento di disgregazione il livello di intensità di ogni pixel. Il risultato di un'operazione di threshold sarà un'immagine suddivisa in aree, binaria nel particolare caso di threshold binario.

Definendo l'immagine di input con  $i(x, y)$ , la versione thresholded con  $o(x, y)$  e la soglia di threshold con  $T$ , si possono calcolare i principali tipi di threshold:

- Threshold binario

$$o(x, y) = \begin{cases} \text{maxValue} & \text{se } i(x, y) \geq T \\ 0 & \text{altrimenti} \end{cases} \quad (3.1)$$

- Threshold binario invertito

$$o(x, y) = \begin{cases} 0 & \text{se } i(x, y) \geq T \\ \text{maxValue} & \text{altrimenti} \end{cases} \quad (3.2)$$

- Troncamento

$$o(x, y) = \begin{cases} T & \text{se } i(x, y) \geq T \\ i(x, y) & \text{altrimenti} \end{cases} \quad (3.3)$$

- Threshold a zero

$$o(x, y) = \begin{cases} i(x, y) & \text{se } i(x, y) \geq T \\ 0 & \text{altrimenti} \end{cases} \quad (3.4)$$

- Threshold a zero invertito

$$o(x, y) = \begin{cases} 0 & \text{se } i(x, y) \geq T \\ i(x, y) & \text{altrimenti} \end{cases} \quad (3.5)$$

Il problema principale del thresholding è considerare solo l'intensità nella scelta del valore di soglia: spesso i pixel non sono legati dal solo valore di energia. Con un threshold fisso è facile inserire pixel estranei alla regione considerata, soprattutto se si sa a priori che le condizioni di luminosità delle immagini sono variabili.

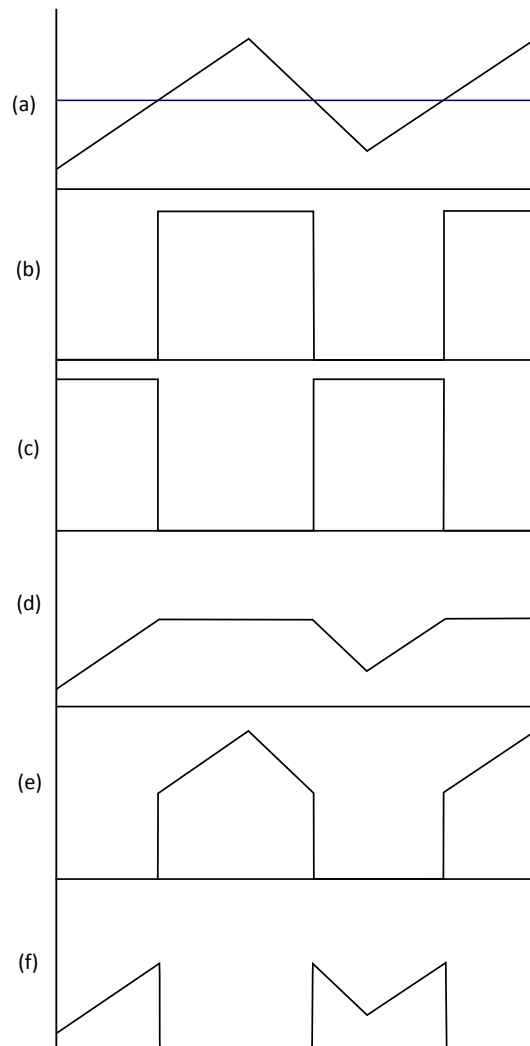


Figura 3.1: Esempio dei diversi tipi di threshold: (a) segnale di input, (b) Threshold binario, (c) Threshold binario invertito, (d) Troncamento, (e) Threshold a zero, (f) Threshold a zero invertito. La linea blu rappresenta la soglia T.

### 3.1.1 Thresholding locale

Per risolvere il problema della luminosità non omogenea o di regioni che presentano valori di intensità non omogenei è possibile applicare un threshold locale, ovvero si permette alla soglia di variare durante lo spostamento all'interno dell'immagine.

La tecnica di adaptive threshold setta soglie differenti per differenti aree dell'input. L'immagine viene suddivisa in aree di dimensione  $R \times C$ . All'interno di ogni area la soglia viene settata pari alla media dell'intensità dei pixel contenuti nella regione (nel caso di Adaptive Threshold Mean) o pari alla media dell'intensità dei pixel pesata su una Gaussiana (nel caso di Adaptive Threshold Gaussian). Al threshold calcolato è possibile sottrarre o sommare una costante  $k$  in modo da adattare la soglia calcolata per immagini molto rumorose. Questa tecnica permette di ottenere ottime segmentazioni nel caso di regioni con forme non troppo frastagliate.

### 3.1.2 Thresholding globale

Il thresholding globale applica lo stesso valore di soglia per tutta l'immagine da segmentare. Per settare una soglia ottimale spesso si analizza l'istogramma dell'immagine per individuare possibili modelli distinti all'interno dell'input (ad esempio sfondo e oggetto).

L'istogramma di un'immagine è la distribuzione di probabilità dell'intensità dei pixel:

$$p(g) = \frac{n_g}{n} \quad (3.6)$$

Dove  $n$  è il numero di pixel dell'immagine e  $n_g$  sono i pixel di intensità  $g$ .

Una tecnica molto semplice per individuare una soglia è trovare i punti di estremo nell'istogramma e successivamente estrarre i minimi tra i valori trovati. Il problema di questo approccio è dato dalla poca affidabilità in caso di rumore e in caso di segmentazione di oggetti con intensità non uniforme.

Il metodo Otsu è usato per applicare un thresholding clustering-based. La tecnica consiste nel trovare il threshold che permette di minimizzare la sovrapposizione tra i due cluster (bi-modal histogram). Si può definire la varianza intra-classe come la somma pesata delle varianze di ogni cluster:

$$\sigma^2_{intra}(T) = n_1(T)\sigma^2_1(T) + n_2(T)\sigma^2_2(T) \quad (3.7)$$

Dove l'immagine ha range di intensità  $[0, N - 1]$  e

$$n_1(T) = \sum_{i=1}^T p(i) = \sum_{i=1}^T \frac{n_i}{n} \quad (3.8)$$



$$n_2(T) = \sum_{i=T+1}^{N-1} p(i) = \sum_{i=T+1}^{N-1} \frac{n_i}{n} \quad (3.9)$$

$$\sigma^2_1(T) = \text{varianza della regione sotto la soglia } T \quad (3.10)$$

$$\sigma^2_2(T) = \text{varianza della regione sopra la soglia } T \quad (3.11)$$

Calcolare la varianza intra-classe in questo modo richiede un alto tempo computazionale. Otsu ha dimostrato che si può equivalentemente lavorare sulla massimizzazione della varianza inter-classe:

$$\sigma^2_{inter}(T) = \sigma^2 - \sigma^2_{intra}(T) = n_1(T)n_2(T)[\mu_1(T) - \mu_2(T)]^2 \quad (3.12)$$

Dove

$$\mu_1(T) = \frac{\sum_1^T ip(i)}{n_1(T)} \quad (3.13)$$

$$\mu_2(T) = \frac{\sum_{T+1}^{N-1} ip(i)}{n_2(T)} \quad (3.14)$$

È possibile calcolare ricorsivamente i termini definiti sopra utilizzando le formule:

$$n_1(T+1) = n_1(T) + p(T+1) \quad (3.15)$$

$$n_2(T+1) = 1 - n_1(T+1) \quad (3.16)$$

$$\mu_1(T+1) = \frac{\mu_1(T)n_1(T) + p(T+1)(T+1)}{n_1(T+1)} \quad (3.17)$$

$$\mu_2(T+1) = \frac{\mu_2(T)n_2(T) - p(T+1)(T+1)}{n_2(T+1)} \quad (3.18)$$

## 3.2 Canny edge detection

Questa tecnica fa parte della segmentazione edge-based, ovvero la suddivisione dell'immagine è ottenuta identificando i contorni delle regioni che la compongono.

L'algoritmo di Canny permette di identificare i contorni presenti in un'immagine seguendo cinque step:

- Smoothing: blurring dell'immagine per rimuovere il rumore
- Ricerca dei gradienti: gli edges si trovano in corrispondenza di magnitudine alta

- Soppressione Non-maximum: solo i massimi locali vengono scelti come edges
- Doppio threshold: threshold dei potenziali edges
- Isteresi: eliminazione dei candidati non connessi ad edges forti

### 3.2.1 Smoothing

L'identificazione dei bordi è molto sensibile al rumore, quindi la prima operazione è un filtro Gaussiano per ridurre questo problema. Il blur è ottenuto convolvendo l'immagine con un nucleo:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.19)$$

Rispetto ad altri filtri, permette di mantenere in maggior evidenza i contorni ottenendo nel contempo l'effetto di smoothing desiderato.

### 3.2.2 Ricerca dei gradienti

L'algoritmo associa a forti sbalzi di intensità un edge; queste zone sono individuate studiando il gradiente dell'immagine. Per calcolare la mappa dei gradienti sono applicati i kernel di Sobel:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Direzione e intensità dei gradienti sono calcolati con le formule

$$G = \sqrt{G_x^2 + G_y^2} \quad \theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (3.20)$$

### 3.2.3 Soppressione Non-maximum

Lo scopo di questo passo è convertire i bordi smoothed della mappa dei gradienti in bordi ben definiti. Per ottenere questo risultato si mantengono solo i pixel che sono massimi locali, sopprimendo tutti gli altri. Per ogni pixel nella mappa dei gradienti le operazioni effettuate sono:



Figura 3.2: Canny edge detection. A sinistra l'immagine di input, a destra il risultato ottenuto applicando Canny.

- approssimare le direzioni dei gradienti al più vicino multiplo di 45 gradi
- confrontare l'intensità del gradiente con i pixel che si trovano nelle celle puntate dalla direzione positiva e negativa del gradiente
- se la forza del gradiente del pixel è la più alta, mantenere il valore del pixel; in caso contrario sopprimerlo

### 3.2.4 Doppio Threshold

In questa fase i pixel con intensità del gradiente maggiore del threshold alto  $T_h$  sono marcati come bordi forti, i pixel con intensità del gradiente minore del threshold basso  $T_l$  sono eliminati, mentre tutti gli altri pixel sono segnati come bordi deboli. Applicando questo passo è possibile scartare molti falsi bordi rilevati a causa di cambi di colore all'interno dell'immagine o a causa di rumore non eliminato dal filtro Gaussiano.

### 3.2.5 Isteresi

Gli edges sono stati suddivisi in due gruppi, i pixel contenuti nel gruppo degli edges forti vengono inclusi immediatamente tra i bordi finali. Partendo da ogni pixel marcato come forte, si visitano ricorsivamente i pixel interni alla finestra di neighborhood marcati come pixel deboli includendoli tra i pixel che formano il risultato finale.

### 3.3 Hough transform

La trasformata di Hough è una tecnica che permette di isolare particolari forme all'interno di un'immagine. Vista la necessità di specificare in una qualche descrizione parametrica la forma che si vuole ricercare, la trasformata di Hough è solitamente usata per curve regolari come linee, cerchio, ellissi. Per curve non descrivibili analiticamente è possibile utilizzare la trasformata generica di Hough, ma vista la sua complessità computazionale e la necessità all'interno del problema di trovare i bordi della tavola che sono principalmente linee, non è stata presa in considerazione. La trasformata di Hough è comunque abbastanza flessibile perchè permette la presenza di gap tra una curva individuata e un'altra, oltre ad essere relativamente robusta al rumore.

Il caso più semplice della trasformata di Hough è la ricerca di linee. Una retta è descritta dall'equazione parametrica:

$$y = mx + q \quad (3.21)$$

dove  $m$  è il coefficiente angolare e  $q$  è l'intercetta.

Nella tecnica di Hough tale retta è rappresentata come un punto nel sistema di riferimento polare. La retta viene quindi descritta dalle coordinate  $r$  e  $\theta$ , dove  $r$  rappresenta la distanza del punto dall'origine e  $\theta$  rappresenta l'angolo del vettore ortogonale alla linea che si congiunge con l'origine. Utilizzando la forma parametrica la retta è rappresentata dall'equazione:

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \frac{r}{\sin\theta} \quad (3.22)$$

O equivalentemente:

$$r = x\cos\theta + y\sin\theta \quad (3.23)$$

Il piano  $(r, \theta)$  è spesso detto spazio di Hough per il set di linee in due dimensioni.

Per ogni punto di coordinate generiche  $(x_i, y_i)$  nell'immagine, passano le rette appartenenti al fascio:

$$r(\theta) = |x_i\cos\theta + y_i\sin\theta| \quad (3.24)$$

Ponendo  $r$  positivo e  $\theta \in [0, 2\pi)$ .

Il fascio appena descritto corrisponde a una curva sinusoidale nel piano  $(r, \theta)$ . L'intersezione di due sinusoidi significa che la retta rappresentata dal punto di

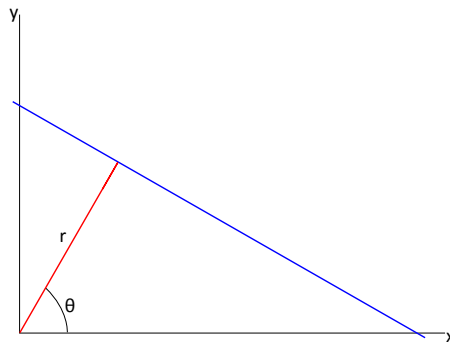


Figura 3.3: Rappresentazione di una retta nello spazio di Hough.

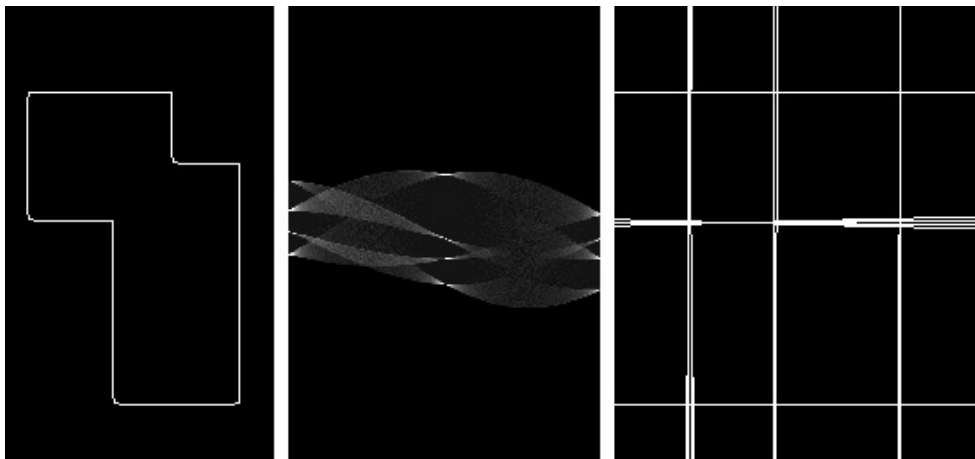


Figura 3.4: Trasformata di Hough. Da sinistra a destra: immagine di input, spazio di Hough, rette ottenute con threshold 0.4.

intersezione passa per due punti nell'immagine. Più in generale, un set di punti dell'immagine che formano una linea, produrranno un punto di intersezione forte nello spazio  $(r, \theta)$ . Il problema di individuare un set di punti allineati viene quindi convertito nel problema di ricerca di punti di intersezione.

### 3.4 Active Contour Models

Un Active contour model (o Snake) è una spline guidata da forze interne e da forze esterne che la spingono verso features come edges o linee. Lo Snake è un modello attivo, ovvero minimizza ad ogni passo la propria energia funzionale. Una volta inizializzato lo Snake nei pressi di un bordo, questo viene attratto dalla forza emanata dal bordo e una volta che uno o più punti si sono assicu-

rati a tale curva il resto dello Snake si adatta raggiungendo l'energia minima possibile.

Le forze che comandano un Active Contour Model sono di due tipi: interne ed esterne. Le forze interne agiscono sulla forma dello Snake: se sono accettati angoli acuti o se al contrario si preferisce una curva smooth, se i punti devono essere equidistanti tra loro o se possono esserci discrepanze in questo senso. Le forze esterne sono forze attrattive/repulsive derivate dalle caratteristiche dell'immagine; un esempio può essere la distanza dal punto più vicino che è visto come bordo: più lo Snake è distante e più sentirà l'attrazione verso tale bordo.

Rappresentando la configurazione dello Snake come  $v(s)$ , l'energia funzionale può essere definita come:

$$E_{snake}(v(s)) = E_{interna}(v(s)) + E_{esterna}(v(s)) \quad (3.25)$$

Dove  $v(s)$  rappresenta l'insieme delle posizioni dei punti che costituiscono lo Snake.

Per rendere lo Snake attivo, i punti che lo compongono si muovono ad ogni passo verso la posizione che minimizza l'energia totale dello Snake in quella precisa configurazione. Il processo termina quando lo Snake non si sposta tra un'iterazione e la successiva o quando viene raggiunto un numero massimo di iterazioni.

# Capitolo 4

## Estrazione del descrittore

In Computer Vision si definisce feature una porzione caratteristica dell'immagine. Un descrittore è invece un insieme di features che si desidera sia in grado di rappresentare univocamente l'immagine nello spazio delle features. Come si può evincere dalle definizioni appena date, vi sono molteplici tipi di features e la loro utilità dipende dal tipo di applicazione che si vuole sviluppare. Tra le features più utilizzate sono presenti:

- bordi e confini degli elementi presenti nell'immagine
- angoli e combinazioni di cambiamenti sul gradiente
- blobs o regioni di interesse
- keypoint estratti dalle voci precedenti

L'efficacia dell'algoritmo di riconoscimento dipende fortemente dalla qualità delle features estratte, che devono essere: ripetibili, fortemente caratterizzanti e la loro combinazione (il descrittore) deve essere il più possibile univoca. Altro fattore da considerare nella costruzione del descrittore per un'immagine è il numero di valori che lo compone. Dopo l'estrazione delle features l'immagine sarà sostituita dal descrittore calcolato in tutte le successive operazioni di matching: in generale più un descrittore è dimensionalmente esteso più le operazioni di confronto diventano computazionalmente pesanti. Diventa di essenziale importanza trovare il compromesso tra la dimensione del descrittore e il grado di caratterizzazione che è in grado di fornire.

## 4.1 Trasformate Wavelet

Dagli inizi degli anni novanta le Wavelet hanno acquisito grande interesse per matematici e ingegneri. La trasformata Wavelet (da ora WT) di un segnale non-stazionario permette di trasformarlo per darne una rappresentazione in tempo-frequenza. La WT è stata creata come alternativa alla short time Fourier Transform (STFT) per ovviare il problema dell'ampiezza di finestra non definibile a priori in quest ultimo metodo.

### 4.1.1 Trasformata Wavelet continua

La WT continua è definita come:

$$WT_{\psi}^f(b, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt \quad f \in L^2(\mathbb{R}) \quad (4.1)$$

Dove  $b$  è il parametro di traslazione e  $a$  è il parametro di scala, infatti anche la WT utilizza una finestra shiftata lungo il segnale.

Se il centro della funzione  $\psi$  è dato da  $t^*$  e il raggio è dato da  $\Delta_{\psi}$ , la funzione  $\psi_{b,a}$  è una finestra con centro  $b + at^*$  e raggio  $a\Delta_{\psi}$ . La WT continua darà informazioni locali del segnale  $f$  con una finestra temporale pari a:

$$[b + at^* - a\Delta_{\psi}, b + at^* + a\Delta_{\psi}] \quad (4.2)$$

La funzione  $\psi$  è chiamata wavelet base e deve soddisfare le condizioni:

$$\psi \in L^2(\mathbb{R}) \quad (4.3)$$

$$\int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \quad (4.4)$$

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (4.5)$$

In particolare la condizione (4.4) è necessaria per ottenere la WT inversa, che però in questo lavoro non è stata utilizzata.

### 4.1.2 Trasformata Wavelet discreta

La WT può essere definita nella propria versione discreta (DWT) per permettere un'implementazione agevolata e un'analisi veloce per segnali non continui. Applicando la DWT si ottiene una rappresentazione su tempo-scala del segnale di input correlando la wavelet a diverse scale con il segnale stesso. Nel caso



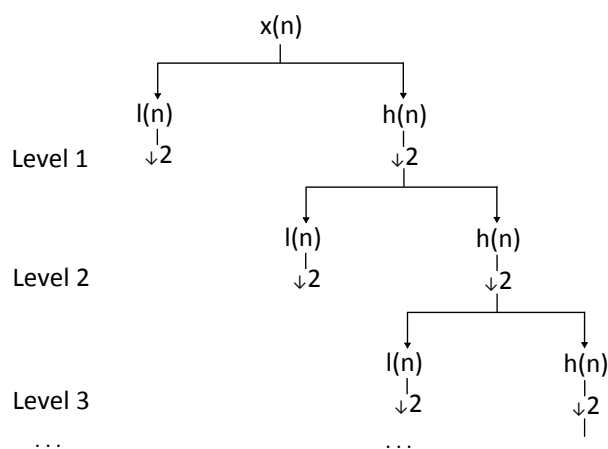


Figura 4.1: Schema di calcolo per i primi tre livelli di DWT. Dove  $l(n)$  indica il filtro passa-basso,  $h(n)$  indica il filtro passa-alto e  $\downarrow 2$  indica il downsampling di fattore 2.

discreto si utilizzano diversi filtri per analizzare il segnale alle scale desiderate: l'input è analizzato tramite una serie di filtri passa-alto per analizzare le alte frequenze e tramite una serie di filtri passa-basso per analizzare le basse frequenze. La scala è modificata tramite le operazioni di upsampling e di downsampling. Dopo aver applicato un filtro che permette il passaggio di metà della banda si può dire che il segnale ha perso metà delle informazioni, ovvero la risoluzione del segnale è dimezzata; ne deriva che un'operazione di down-sample di fattore due eliminerà i campioni ridondanti. L'applicazione della coppia di operazioni di doppio filtraggio e sampling definisce la computazione di un livello di trasformazione.

La procedura descritta è anche nota come subband coding.

### 4.1.3 Haar Wavelet

La wavelet di Haar è la prima wavelet ad essere stata formulata, nel 1909 da Alfréd Haar. La proprietà di non differenziabilità della funzione permette di ottenere vantaggi nell'analisi di segnali con forti transizioni.

La wavelet madre di Haar è la funzione:

$$\psi(t) = \begin{cases} 1 & \text{se } 0 \leq t < \frac{1}{2} \\ -1 & \text{se } \frac{1}{2} \leq t < 1 \\ 0 & \text{altrimenti} \end{cases} \quad (4.6)$$

LL <sub>3</sub>	HL <sub>3</sub>	HL <sub>2</sub>	HL <sub>1</sub>
LH <sub>3</sub>	HH <sub>3</sub>		
LH <sub>2</sub>		HH <sub>2</sub>	
LH <sub>1</sub>			HH <sub>1</sub>

Figura 4.2: Matrice di calcolo di DWT al livello tre.

Con funzione di scaling:

$$\phi(t) = \begin{cases} 1 & \text{se } 0 \leq t < 1 \\ 0 & \text{altrimenti} \end{cases} \quad (4.7)$$

La funzione di Haar è definita per ogni coppia  $n, k$  di interi in  $\mathbb{Z}$  come:

$$\psi_{n,k}(t) = 2^{\frac{n}{2}} \psi(2^n t - k) \quad t \in \mathbb{R} \quad (4.8)$$

Le proprietà delle funzioni appena definite sono le seguenti:

- dilatazioni e traslazioni della funzione di Haar formano una base ortogonale in  $L^2(\mathbb{R})$ , ovvero ogni elemento in  $L^2(\mathbb{R})$  può essere rappresentato come combinazione lineare di  $\psi$
- ortogonalità della funzione di Haar nella forma  $\int_{-\infty}^{\infty} 2^{\frac{n+n_1}{2}} \psi(2^n t - k) \psi(2^{n_1} t - k_1) dt = \delta_{n,n_1} \delta_{k,k_1}$ ; con  $\delta_{i,j}$  delta di Kronecker
- ogni funzione reale continua a supporto compatto può essere approssimata uniformemente da combinazioni lineari di  $\phi(t), \phi(2t), \dots, \phi(2^n t), \dots$  e dai loro shift.

Una delle applicazioni più interessanti della wavelet di Haar è l'immagine processing. Qualsiasi immagine rappresentata sotto forma di matrice (di dimensioni  $2^n \times 2^m$  o almeno divisibili iterativamente per 2 un numero di volte maggiore



Figura 4.3: Haar wavelet.

al livello di scomposizione che si desidera raggiungere) può essere scomposta utilizzando le funzioni wavelet.

Il processo è il seguente. Alla matrice di input vengono applicati, su ogni riga, i filtri  $L$  e  $H$  (relativamente il filtro passa-basso e il filtro passa-alto). Si ottengono due matrici  $L_1, H_1$  entrambe con numero di righe dimezzato a causa dell'operazione di downsampling. A entrambe le matrici si applicano i filtri  $L$  e  $H$  su tutte le colonne, ottenendo le matrici  $HH_1, HL_1, LH_1, LL_1$ . La matrice  $LL_1$  è la matrice di media, mentre le restanti sono dette matrici dei dettagli. Il procedimento descritto si può ripetere partendo dalla matrice di media, per computare il successivo livello della DWT.

Il descrittore è formato dall'intera matrice ottenuta al livello  $i$ -esimo di elaborazione o da una porzione di tale matrice. La scelta di non utilizzare l'intero risultato può essere dettata dal minore spazio di memorizzazione richiesto e dal livello di significatività dei dettagli evidenziati dalle sequenze di filtri applicati.

## 4.2 Filtri di Gabor

Il filtro di Gabor è un filtro lineare, costruito modulando una funzione seno/coseno con una funzione Gaussiana. La parte sinusoidale permette di localizzare il segnale in frequenza, mentre la modulazione Gaussiana permette la localizzazione nello spazio senza perdere qualità nella localizzazione in frequenza. Per la proprietà di moltiplicazione-convoluzione la risposta in frequenza di un filtro di Gabor è la convoluzione della Trasformata di Fourier della funzione armonica e la Trasformata di Fourier della funzione Gaussiana.

Daugman [2] applica i filtri bidimensionali di Gabor nel proprio sistema di riconoscimento dell'iride, demodulando successivamente l'output per comprimere il descrittore. Viene considerata solo l'informazione di fase ottenuta dal

filtro, che viene rappresentata con due bit in base al quadrante di appartenenza nel piano complesso. Il sistema di Daugman è uno dei più precisi, ottiene infatti percentuali di riconoscimento perfette nella maggior parte dei database di test.

Il descrittore viene assemblato utilizzando il risultato ottenuto dopo l'applicazione del Filtro Gabor all'immagine di input. In alcune applicazioni l'immagine viene prima scomposta in segnali monodimensionali per poi applicare il filtro su tali segnali invece di utilizzare l'intero input.

### 4.2.1 Filtro mono-dimensionale

Un filtro di Gabor per segnali mono-dimensionali è definito come:

$$g(t) = ke^{j\theta} w(at)s(t) \quad (4.9)$$

con

$$w(t) = e^{-\pi t^2} \quad (4.10)$$

$$s(t) = e^{j2\pi f_0 t} \quad (4.11)$$

$$e^{j\theta} s(t) = e^{j(2\pi f_0 t + \theta)} = (\cos(2\pi f_0 t + \theta), \sin(2\pi f_0 t + \theta)) \quad (4.12)$$

Le variabili  $k, \theta, f_0$  sono i parametri del filtro che comandano frequenza e ampiezza di banda della funzione risultante.

Si può suddividere il filtro di Gabor come due componenti allocate l'una nella parte reale e l'altra nella parte complessa dello spazio:

$$g_{Re}(t) = w(t)\sin(2\pi f_0 t + \theta) \quad (4.13)$$

$$g_{Im}(t) = w(t)\cos(2\pi f_0 t + \theta) \quad (4.14)$$

La risposta in frequenza del filtro è:

$$\hat{g}(f) = \frac{k}{a} e^{j\theta} e^{-2\pi(\frac{f-f_0}{a})} \quad (4.15)$$

### 4.2.2 Filtro bidimensionale

La versione bidimensionale del filtro è descritta da:

$$g(x, y) = s(x, y)w(x, y) \quad (4.16)$$

Dove  $s(x,y)$  è una estensione bidimensionale della funzione sinusoidale e  $w(x,y)$  è la funzione Gaussiana bidimensionale.

In particolare:

$$s(x, y) = e^{j(2\pi(u_0x + v_0y) + P)} \quad (4.17)$$

Con  $(u_0, v_0)$  e  $P$  parametri della sinusoide che comandano rispettivamente la frequenza e la fase.

Suddividendo la sinusoide nelle parti reale e immaginaria si ottiene:

$$Re(s(x, y)) = \cos(2\pi(u_0x + v_0y) + P) \quad (4.18)$$

$$Im(s(x, y)) = \sin(2\pi(u_0x + v_0y) + P) \quad (4.19)$$

La funzione Gaussiana è definita come:

$$w(x, y) = Ke^{-\pi(a^2((x-x_0)\cos\theta + (y-y_0)\sin\theta)^2 + b^2(-(x-x_0)\sin\theta + (y-y_0)\cos\theta)^2)} \quad (4.20)$$

Dove  $(x_0, y_0)$  è il punto di picco della funzione e  $a, b$  sono i parametri di scala della Gaussiana.

### 4.3 Filtri Log-Gabor

I filtri Log-Gabor sono una variante dei filtri di Gabor. Field [13] afferma che per una grande varietà di immagini i filtri di Gabor non rappresentano efficientemente l'informazione contenuta nell'input. La debolezza del filtro di Gabor è che la massima larghezza di banda è limitata a circa un octave.

Libor Masek [5] nel suo lavoro di tesi realizza un sistema di riconoscimento dell'iride umana basato sui filtri Log-Gabor. L'iride viene estratta dall'immagine utilizzando la trasformata di Hough; la regione ottenuta viene normalizzata in un'area rettangolare con dimensioni costanti; infine il descrittore viene calcolato su un numero definito di righe della regione quantizzando su quattro livelli (corrispondenti ai quadranti) l'informazione di fase di filtri monodimensionali Log-Gabor. I risultati ottenuti su un dataset di 624 immagini mostrano percentuali di FAR (False accept rate) e FRR (False reject rate) pari a 0.005% e 0.238% rispettivamente.

Field [13] propone di utilizzare una funzione la cui trasformata sia una Gaussiana sull'asse logaritmico della frequenza, la funzione Log-Gabor:

$$G(f) = e^{-\frac{[\log(\frac{f}{f_0})]^2}{2[\log(\frac{\sigma}{f_0})]^2}} \quad (4.21)$$

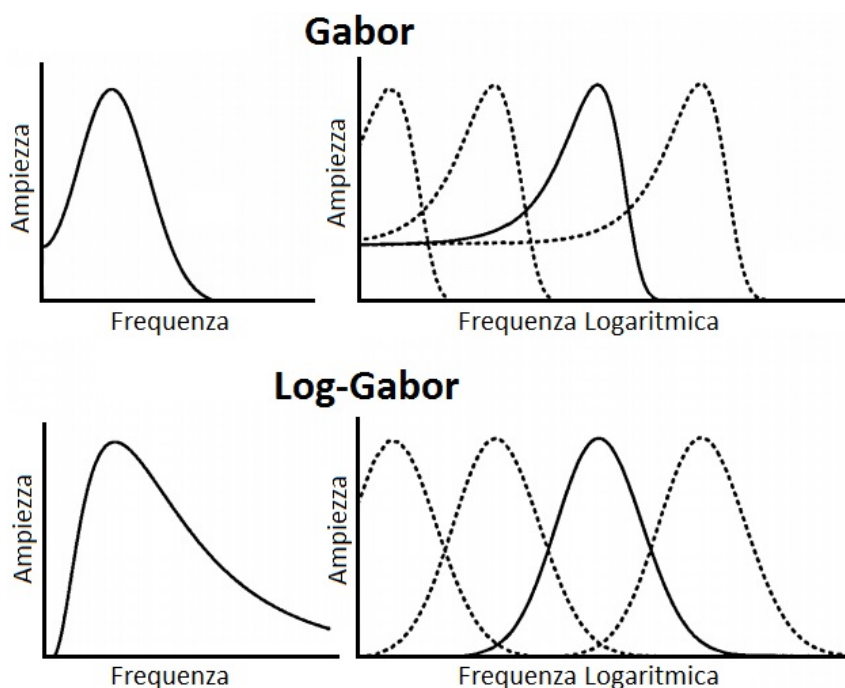


Figura 4.4: Trasformate del filtro Gabor e del filtro Log-Gabor a confronto.

Dove  $\sigma$  è la larghezza di banda e  $f_0$  è la frequenza centrale.

I studi di Field indicano che per immagini naturali la forma a coda estesa del filtro Log-Gabor dovrebbe essere in grado di rappresentare al meglio la codifica dell'input. Il filtro di Gabor al contrario potrebbe sovra-rappresentare le frequenze basse e sotto-rappresentare le alte frequenze.

## 4.4 Gaussian-Hermite Moments

I momenti sono importanti features utilizzate nel riconoscimento di pattern e nell'analisi di immagini. I Gaussian-Hermite moments sono momenti ortogonali che permettono di caratterizzare immagini o segnali con precisione anche in presenza di rumore. Un momento (monodimensionale) ortogonale è la proiezione di un segnale in uno spazio di funzione ortogonale; questa trasformazione permette di evidenziare le differenze tra i segnali stessi.

Nel caso di un'immagine di input, i momenti di Gauss-Hermite possono essere calcolati sui segnali monodimensionali orizzontali (uno per ogni riga) o sui segnali monodimensionali verticali (uno per ogni colonna). La concatena-

zione dei momenti ottenuti rappresenta il descrittore dell'input. Per ottenere descrittori di dimensioni accettabili è possibile calcolare i momenti su una dei segnali monodimensionali che formano l'immagine analizzata.

Li Ma et al. [6] hanno applicato i momenti Gaussian-Hermite al riconoscimento dell'iride umana. L'iride viene estratta dall'immagine e mappata su una regione rettangolare; l'immagine ottenuta viene migliorata al fine di evidenziare i pattern presenti e viene scomposta in segnali monodimensionali sulla direzione delle righe; su tali segnali vengono calcolati i momenti Gaussian-Hermite, che compongono un descrittore di 20480 elementi. I risultati dei test sul database CASIA (918 immagini di training e 1237 immagini di test acquisite su 213 differenti soggetti) mostrano una percentuale di riconoscimento del 99.68% dopo aver sottocampionato il descrittore a 2560 elementi.

#### 4.4.1 Momenti Gaussian-Hermite monodimensionali

Preso un segnale monodimensionale  $s(x)$  e la funzione Gaussiana con deviazione standard unitaria  $g(t)$ , il momento di ordine  $n$  del segnale è definito come:

$$M_n(x, s(x)) = \int_{-\infty}^{\infty} g(t)P_n(t)s(x+t) dx \quad n = 0, 1, .. \quad (4.22)$$

Con  $g(t) = e^{-\frac{t^2}{2}}$  e  $P(n)$  polinomio di ordine  $n$ .

Per soddisfare la condizione di ortogonalità si deve avere che:

$$\int_{-\infty}^{\infty} [g(t)P_m(t)][g(t)P_n(t)] = 0 \quad perm, n = 0, 1, ..em \neq n \quad (4.23)$$

Ovvero nel caso di  $g(t)$  funzione Gaussiana con  $\sigma = 1$ :

$$\int_{-\infty}^{\infty} e^{-t^2} P_m(t)P_n(t) = 0 \quad perm, n = 0, 1, ..em \neq n \quad (4.24)$$

Il risultato appena ottenuto mostra che per la condizione di ortogonalità si dovrebbero usare polinomiali con  $e^{-t^2}$  come funzione di pesatura. Questo tipo di polinomiali sono conosciuti come polinomi di Hermite.

Il polinomio di Hermite di ordine  $n$  è definito come:

$$H_n(t) = (-1)^n e^{t^2} \frac{d^n}{dt^n} e^{-t^2} \quad (4.25)$$

Il momento ortogonale di ordine  $n$  ottenuto fissando il polinomio appena definito diventa:

$$M_n(x, s(x)) = \int_{-\infty}^{\infty} g(t)H_n(t)s(x+t) dx \quad n = 0, 1, .. \quad (4.26)$$

Questo momento è definito come momento ortogonale Gaussian-Hermite. I momenti Gaussian-Hermite sono estesi anche alla formulazione bidimensionale in [14].

#### 4.4.2 Calcolo ricorsivo

I momenti Gaussian-Hermite possono essere calcolati in maniera diretta dalla definizione, ma la complessità computazionale di tale procedura pone una barriera all'utilizzo di questo tipo di descrittore. Diventa essenziale trovare un metodo per calcolare efficientemente tali momenti.

Una proprietà nota dei polinomi di Hermite è la loro ricorsività:

$$H_n(t) = 2tH_{n-1}(t) - 2(n-1)H_{n-2}(t) \quad n \geq 2 \quad (4.27)$$

Che porta all'equazione:

$$\begin{aligned} M_n(x, s(x)) &= \int_{-\infty}^{\infty} g(t)H_n(t)s(x+t) dt \\ &= \int_{-\infty}^{\infty} g(t)2tH_{n-1}(t)s(x+t) dt - 2(n-1) \int_{-\infty}^{\infty} g(t)H_{n-2}(t)s(x+t) dt \\ &= -2 \int_{-\infty}^{\infty} \frac{d g(t)}{dt} H_{n-1}(t)s(x+t) dt - 2(n-1)M_{n-2}(x, s(x)) dt \end{aligned} \quad (4.28)$$

Usando l'integrazione per parti:

$$\begin{aligned} &\int_{-\infty}^{\infty} \frac{d g(t)}{dt} H_{n-1}(t)s(x+t) dt \\ &= - \int_{-\infty}^{\infty} g(t)H'_{n-1}(t)s(x+t) dt - \int_{-\infty}^{\infty} g(t)H_{n-1}(t)s'(x+t) dt \end{aligned} \quad (4.29)$$

Dove  $H'_{n-1}(t) = 2(n-1)H_{n-2}(t)$  per  $n \geq 2$ .



Sostituendo all'equazione precedente si ottiene:

$$\begin{aligned} \int_{-\infty}^{\infty} \frac{d g(t)}{d t} H_{n-1}(t) s(x+t) d t \\ = -2(n-1) M_{n-2}(x, s(x)) - M_{n-1}(x, s'(x)) \end{aligned} \quad (4.30)$$

Da cui si può ricavare infine che:

$$M_n(x, s(x)) = 2(n-1) M_{n-2}(x, s(x)) + 2 M_{n-1}(x, s'(x)) \quad n \geq 2 \quad (4.31)$$

Ovvero il momento Gaussian-Hermite di ordine  $n$  del segnale  $s(x)$  è combinazione lineare del momento di ordine  $n-1$  di  $s(x)$  e del momento di ordine  $n-2$  del segnale  $s'(x)$ .

Per i momenti di ordine zero e uno è possibile usare le formule non ricorsive:

$$M_0(x, s(x)) = \int_{-\infty}^{\infty} g(t) H_0(t) s(x+t) d t = g(x) * s(x) \quad (4.32)$$

$$M_1(x, s(x)) = \int_{-\infty}^{\infty} g(t) H_1(t) s(x+t) d t = 2s(x) * \frac{d g(x)}{d x} \quad (4.33)$$

Dove l'operatore  $*$  indica la convoluzione di due segnali.

Si nota che il momento di ordine zero non è altro che il segnale convoluto con una funzione Gaussiana, mentre il momento di ordine uno è la derivata prima del segnale Gaussiano convoluta con il segnale di input moltiplicato per un fattore due.

### 4.4.3 Finestra di smoothing

I momenti Gaussian-Hermite possono essere generalizzati per utilizzarli con ampiezza di smoothing arbitraria, per un maggior controllo sul rumore o sulla specifica applicazione.

Definita la funzione Gaussiana di deviazione standard  $\sigma$  come:

$$g(x, \sigma) = \sqrt{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (4.34)$$

Si definisce il generico momento Gaussian-Hermite di ordine  $n$ :

$$M_n(x, s(x)) = \int_{-\infty}^{\infty} g(t, \sigma) H_n\left(\frac{t}{\sigma}\right) s(x+t) d t \quad (4.35)$$

La proprietà ricorsiva mantiene il suo valore:

$$M_n(x, s(x)) = 2(n-1)M_{n-2}(x, s(x)) + 2\sigma M_{n-1}(x, s'(x)) \quad n \geq 2 \quad (4.36)$$

$$M_0(x, s(x)) = \int_{-\infty}^{\infty} g(t, \sigma) H_0\left(\frac{t}{\sigma}\right) s(x+t) dt = g(x, \sigma) * s(x) \quad (4.37)$$

$$M_1(x, s(x)) = \int_{-\infty}^{\infty} g(t, \sigma) H_1\left(\frac{t}{\sigma}\right) s(x+t) dt = 2\sigma g'(x, \sigma) * s(x) \quad (4.38)$$

#### 4.4.4 Formulazione generica

La formulazione per ogni derivata del segnale di ingresso è la seguente:

$$M_n(x, s^{(m)}(x)) = 2(n-1)M_{n-2}(x, s^{(m)}(x)) + 2\sigma M_{n-1}(x, s^{(m+1)}(x)) \quad (4.39)$$

per  $m \geq 0, n \geq 2$

$$M_0(x, s^{(m)}(x)) = g(x, \sigma) * s^{(m)}(x) \quad \text{per } m \geq 0 \quad (4.40)$$

$$M_1(x, s^{(m)}(x)) = 2\sigma g'(x, \sigma) * s^{(m)}(x) \quad \text{per } m \geq 0 \quad (4.41)$$

#### 4.4.5 Significato dei momenti Gaussian-Hermite

Dalla definizione di polinomio di Hermite e di funzione Gaussiana si deduce che la funzione base  $g(t, \sigma)H_n\left(\frac{t}{\sigma}\right)$  è funzione pari per ordini  $n$  pari ed è dispari per ordini  $n$  dispari. Quindi:

$$\begin{aligned} M_n(x, s(x)) &= \int_{-\infty}^{\infty} (-1)^n g(t, \sigma) s(x-t) dt \\ &= (-1)^n [g(x, \sigma) H_n\left(\frac{x}{\sigma}\right)] * s(x) \end{aligned} \quad (4.42)$$

L'equazione appena presentata mostra che il polinomio di ordine  $n$  ha  $n$  differenti radici reali; di conseguenza il momento Gaussian-Hermite di ordine  $n$  cambierà segno  $n$  volte, con oscillazioni a frequenza sempre più alta all'aumentare dell'ordine che permettono di separare efficientemente le caratteristiche del segnale analizzato a diversi valori di banda.

L'utilizzo della derivata all'interno del calcolo dei momenti permette di caratterizzare bene il segnale, mentre lo smoothing permette l'adattamento al rumore che l'ingresso può avere in situazioni reali.

## 4.5 Zero-crossing

L'analisi del zero-crossing è una tecnica molto usata per descrivere un segnale. Associando il bit 0 ai valori sotto una soglia definita (zero solitamente) e il bit 1 ai valori superiori a tale soglia è possibile sia descrivere il segnale sia godere dei vantaggi che derivano dalla conseguente conversione in una sequenza binaria.

Nam et al. [7] applicano l'analisi del zero-crossing al riconoscimento dell'iride. Dopo aver estratto e normalizzato la regione dell'iride dall'immagine, calcolano la derivata seconda per i segnali corrispondenti ad ogni riga della regione rettangolare (tali segnali sono quindi cerchi di diverso raggio con centro il punto centrale della pupilla); successivamente assegnano il bit 1 ai pixel con derivata seconda positiva e il pixel 0 ai rimanenti. In questo modo analizzano il zero-crossing della derivata seconda dei segnali componenti l'intera iride e ottengono una matrice binaria confrontabile con le altre tramite la distanza di Hamming. I risultati ottenuti mostrano un riconoscimento del 85.08% con una varianza del 3.36% su un database di 272 immagini ottenute da 18 diverse persone.

Boles e Boashash [4] usano la tecnica zero-crossing sui segnali ottenuti applicando trasformate wavelet monodimensionali a diverse scale. La wavelet madre è definita come la seconda derivata di una funzione di smoothing  $\theta(x)$ :

$$\psi(x) = \frac{d^2\theta(x)}{dx^2} \quad (4.43)$$

La trasformata wavelet utilizzata è definita per un segnale  $f$  a scala  $s$  e posizione  $x$  come:

$$W_s f(x) = f * \psi(x) = f * (s^2 \frac{d^2\theta(x)}{dx^2})(x) \quad (4.44)$$

$W_s f(x)$  è proporzionale alla derivata seconda del segnale dopo lo smooth. Applicando l'analisi del passaggio per lo zero si può individuare i punti significativi dell'iride.

## 4.6 SIFT

Scale Invariant Feature Transform è un metodo per la creazione di descrittori per il matching basato su immagini. Il descrittore SIFT è invariante a transizioni, traslazione, rotazione e scala. Inoltre è robusto a moderate trasformazioni prospettiche e variazioni di illuminazione.

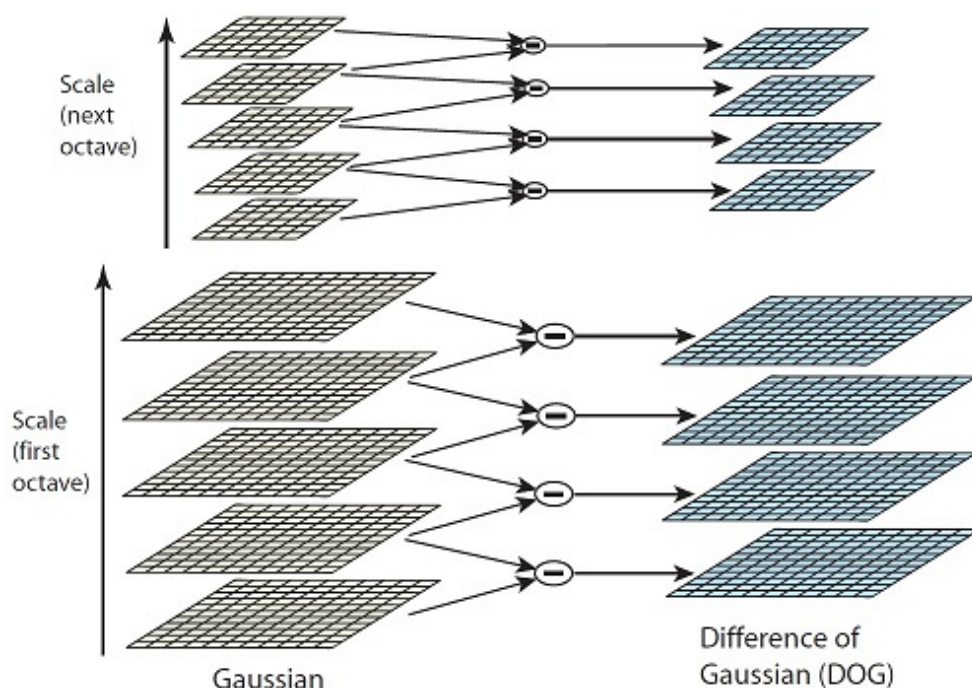


Figura 4.5: Computazione efficiente della piramide Differenza-di-Gaussiane.

Scale invariant feature transform è stata utilizzata in molteplici campi della computer vision, tra cui: matching di immagini con presenza di corrispondenze [15], unione di immagini per la creazione di una foto panoramica [16], riconoscimento biometrico [17] [18], analisi delle immagini in campo robotico.

Per individuare le locazioni stabili dei keypoints nello spazio delle scale, Lowe [19] propone l'utilizzo dei punti di estremo nel risultato  $D(x, y, \sigma)$  della convoluzione dell'immagine con una funzione differenza-di-Gaussiana.  $D(x, y, \sigma)$  può essere ottenuta dalla differenza di due scale vicine separate da un fattore moltiplicativo  $k$ :

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (4.45)$$

La funzione appena definita è scelta perchè è efficiente da computare e perchè permette di approssimare la funzione Laplaciano di Gaussiana normalizzato sulla scala.

Un approccio efficiente per calcolare  $D(x, y, \sigma)$  tramite convoluzioni ripetute è mostrato in Figura4.5.

### 4.6.1 Identificazione degli estremi locali

Allo scopo di individuare i massimi e minimi locali in  $D(x, y, \sigma)$  ogni punto candidato è confrontato con i propri otto vicini nell'immagine della scala corrispondente e con i nove vicini delle scale precedente e successiva. Un punto viene selezionato solo se è maggiore di tutti i suoi vicini o minore di tutti loro. Individuando i punti salienti in questo modo non si pone un limite maggiore alla loro quantità, quindi è necessario un metodo per campionare tale insieme. Lowe analizza i parametri ottimali per questa operazione in [20].

### 4.6.2 Localizzazione accurata dei keypoints

Questa operazione serve ad eliminare dai keypoints derivati dal punto precedente quelli con basso contrasto o che sono posizionati in prossimità di bordi.

Brown e Lowe [21] hanno proposto un metodo che sfrutta l'espansione di Taylor di  $D(x, y, \sigma)$  ai termini quadratici:

$$D(x) = D + \frac{\delta D^T}{\delta x} x + \frac{1}{2} x^T \frac{\delta^2 D}{\delta x^2} x \quad (4.46)$$

Dove  $X = (x, y, \sigma)^T$  è l'offset dal punto candidato che si sta valutando. La localizzazione dell'estremo  $\hat{x}$  è dato da:

$$\hat{x} = -\frac{\delta^2 D^{-1} \delta D}{\delta x^2 \delta x} \quad (4.47)$$

Il valore di  $D$  nel punto  $\hat{x}$  è utile per eliminare i punti che hanno un basso contrasto.

La funzione differenza-di-Gaussiana avrà risposta forte vicino ai bordi presenti nell'immagine, anche in caso di bordi non definiti o dovuti a rumore. Per scartare tali punti si procede a calcolare

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (4.48)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (4.49)$$

Dove  $H$  è la matrice Hessiana 2x2 computata sul keypoint.

I punti con  $Det(H)$  negativo vengono scartati direttamente in quanto segnale che le curvature hanno segno diverso.

I restanti punti vengono filtrati applicando la disequazione

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad (4.50)$$

Con  $r$  valore di threshold arbitrario. Vengono utilizzati solo i keypoints per cui la disequazione è vera.

### 4.6.3 Assegnamento dell'orientazione

Assegnando un orientamento consistente con le proprietà locali dell'immagine, i keypoints possono essere rappresentati relativamente a tale informazione acquisendo l'invarianza alla rotazione.

I calcoli che seguono sono eseguiti sull'immagine elaborata con la funzione Gaussiana più vicina a quella del keypoint considerato. Tale immagine viene definita con la nomenclatura  $L$ .

Per ogni keypoint  $\hat{x} = (x, y)$  vengono calcolati orientazione e intensità del gradiente:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (4.51)$$

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (4.52)$$

Dai valori calcolati viene creato un istogramma delle orientazioni del gradiente, pesando ogni entry secondo una Gaussiana sull'intensità del gradiente. L'istogramma è utilizzato per analizzare le orientazioni più frequenti del gradiente e per assegnare eventuali keypoints aggiuntivi in tali posizioni.

### 4.6.4 Descrittore dell'immagine

Per ogni keypoint viene calcolato un descrittore campionando intensità e orientamento del gradiente in una finestra circolare attorno al keypoint stesso. I valori appartenenti al vicinato vengono pesati con una funzione Gaussiana di parametro  $\sigma$  pari alla metà dell'ampiezza della finestra circolare considerata. I valori di intensità e orientamento del gradiente ottenuti vengono accumulati in istogrammi di orientamento che rappresentano l'intera informazione in un numero minore di byte.

È possibile aumentare il grado di significatività del descrittore riducendo gli effetti delle variazioni lineari di illuminazione all'interno dell'immagine. Questo risultato si può ottenere normalizzando il descrittore.

Una volta calcolati e uniti i descrittori di tutti i keypoints è possibile avere il descrittore che rappresenta l'intera immagine.

# Capitolo 5

## Implementazione

### 5.1 Introduzione

Il problema che si vuole risolvere è il matching di immagini contenenti tavole in legno. Le immagini sono suddivise in due gruppi: immagini del primo buffer e immagini del secondo buffer.

Le immagini del primo buffer sono acquisite mentre la tavola si muove longitudinalmente sul nastro della linea. Le immagini del secondo buffer sono acquisite in un momento successivo mentre la tavola si muove trasversalmente sulla linea.

L'importanza di riconoscere la tavola che passa al secondo punto tra tutte le tavole presenti nel primo buffer è la fusione di rilevazioni effettuate sulla tavola prima e dopo il passaggio attraverso le due stazioni.

#### 5.1.1 Posizionamento delle stazioni

Le immagini utilizzate dall'algoritmo sono acquisite da due Stazioni hardware poste all'interno della linea di produzione.

La prima Stazione (Stazione 1) è inserita sopra il nastro su cui le tavole scorrono longitudinalmente e scatta una fotografia della coda della tavola; la sincronizzazione tra passaggio della tavola e scatto è gestita tramite una fotocellula e un encoder.

La seconda Stazione (Stazione 2) è inserita sopra un meccanismo che muove le tavole trasversalmente; l'immagine acquisita è relativa alla coda della tavola e anche in questo caso la sincronizzazione è gestita tramite una fotocellula e un encoder.

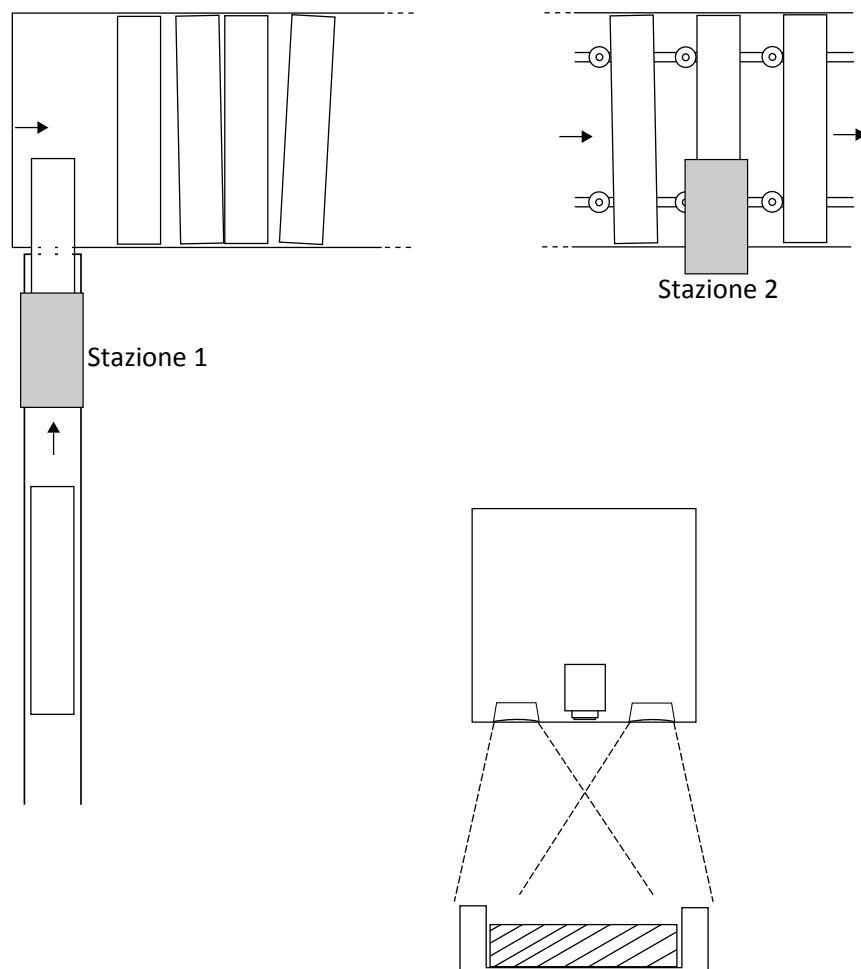


Figura 5.1: Linea di produzione schematizzata con segnalate le posizioni delle Stazioni di acquisizione immagini. Tra una Stazione e l'altra le tavole possono seguire qualsiasi tipo di percorso, che può causare variazioni nell'ordine della coda o la rimozione di alcune tavole. In basso a destra un particolare del contenuto delle Stazioni hardware (luci led e fotocamera) e prospettiva del posizionamento rispetto alla linea.



Tra la prima e la seconda Stazione le tavole possono essere tolte dalla linea ed essere reinserite in un momento successivo, possono essere rimosse definitivamente perchè danneggiate o possono essere marcate dagli operatori. Si deduce quindi che l'ordine di passaggio delle tavole tra le due Stazioni non segue il classico ordine *FIFO* che ci si aspetterebbe idealmente.

Poichè nel corso del passaggio attraverso la linea le tavole vengono analizzate da molteplici macchine, è essenziale poter assegnare un id univoco ad ogni tavola per poter riunire tutte le misurazioni una volta effettuato il passaggio alla Stazione 2.

### **5.1.2 Riconoscimento con inchiostro**

Per effettuare il riconoscimento delle tavole all'ultima stazione, al momento dell'implementazione dell'algoritmo, la tecnica utilizzata è il riconoscimento con inchiostro. Sulla tavola viene stampato un codice alla Stazione 1, il codice viene classificato e collegato all'id della tavola proprietaria. Quando le tavole passano alla Stazione 2 il codice viene analizzato nuovamente e si effettua un'operazione di matching con tutte le tavole presenti nella coda della Stazione 1.

I risultati ottenuti con questo metodo sono ottimali: si stima una percentuale di riconoscimento corretto del 99.99%. Tale risultato è prevedibile poichè si procede a stampare codici facilmente estraibili dalla facciata della tavola e creati appositamente per essere classificati univocamente all'interno dell'insieme di tavole in attesa di matching.

Nonostante i risultati dell'algoritmo in uso, vi sono degli aspetti negativi che portano a voler studiare una versione che non richieda la stampa forzata di codici:

- costo elevato dell'inchiostro e della stampante
- manutenzione frequente della stampante per mantenere puliti gli ugelli e stampare un codice riconoscibile sulle tavole
- le tavole devono essere segnate con un inchiostro difficilmente eliminabile

L'algoritmo studiato deve quindi essere in grado di riconoscere le tavole senza informazioni aggiuntive, ottenere risultati paragonabili all'algoritmo con inchiostro ed essere utilizzabile rispettando i tempi della linea di produzione.



Figura 5.2: Esempio di immagine relativa alla Stazione 1.



Figura 5.3: Esempio di immagine relativa alla Stazione 2.

### 5.1.3 Invarianza a fattori esterni

La linea di produzione in cui vengono acquisite le immagini non rispetta i canoni di idealità che ci si potrebbe aspettare, quindi l'algoritmo deve essere invariante rispetto a:

- rotazione parallela al piano di trasporto
- traslazione verticale e/o orizzontale
- scala
- luminosità
- imperfezioni nella messa a fuoco
- rumore esterno dovuto a schegge o rotture parziali delle tavole
- forma delle tavole

## 5.2 Estrazione della tavola

Per estrarre la tavola dall'immagine originale si utilizza il solo canale rosso poiché, dato il colore del legno, è il canale che mostra le maggiori variazioni di intensità tra le zone in cui la tavola è presente e quelle in cui si trova lo sfondo.

### 5.2.1 Bordo verticale

Si analizza per prima l'area  $I_v$ , dove si trova il lato verticale della tavola, in quanto essendo il più corto è il meno dispendioso da ricercare e può essere utilizzato come indicatore di presenza della tavola.

Per individuare il bordo verticale della tavola si applica il kernel di Sobel:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

In modo da evidenziare le variazioni di gradiente nella direzione orizzontale. Per eliminare dall'immagine ottenuta il rumore presente a causa di riflessi, polvere, usura del nastro e schegge si applica un filtro di Gauss di finestra  $1 \times 31$ ; il filtro permette di mantenere intensità alta solo nelle zone che presentano un bordo verticale esteso, abbassando invece il valore di energia nelle zone dove

il gradiente non è dovuto a un vero contorno verticale.

Per estrarre il bordo dall'immagine risultante dalle precedenti elaborazioni si applica un Threshold basato sulla media pesata  $k$  dell'intensità dei pixel dell'immagine, ovvero si pone intensità pari a zero per tutti i pixel con intensità sotto la soglia  $Th$ :

$$Th = \max(2k, 13 + k) \quad (5.1)$$

$$k = \frac{\sum_{i \in I_v} i^2}{\sum_{i \in I_v} i} \quad (5.2)$$

Il valore  $13 + k$ , la cui efficacia è stata verificata sperimentalmente, permette di evidenziare i bordi anche nel caso di tavole molto scure; tali tavole hanno un colore simile a quello del nastro e presentano un basso valore  $k$  poiché il gradiente presente in prossimità dei bordi non è elevato.

Nell'immagine ottenuta si procede ad una scansione orizzontale verso destra su ogni riga, salvando le coordinate del primo pixel trovato con intensità superiore a zero. La scansione si ferma ad una colonna prefissata nel caso in cui non sia stato trovato alcun pixel con intensità non-zero sulla riga considerata. Se il numero di punti trovati è maggiore ad una soglia precedentemente fissata si può garantire che la tavola è presente nell'immagine.

### 5.2.2 Bordi superiore e inferiore

Per individuare i bordi orizzontali si applica un procedimento simile a quello descritto nella sezione precedente.

Si applica all'immagine formata dal solo canale rosso il kernel di Sobel :

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Applicando l'operazione modulo nei valori risultati negativi (passaggio da intensità alta ad intensità bassa), in modo da poter individuare anche il bordo inferiore applicando una soglia positiva sul gradiente. All'immagine risultante si applica il filtro Gaussiano di finestra  $31 \times 1$  attenuando così il rumore presente nella mappa dei gradienti. Si calcola e applica infine lo stesso tipo di Threshold statistico che è stato presentato precedentemente.

Il bordo superiore viene individuato tramite un campionamento per ogni colonna dalla riga di indice zero verso il basso, salvando le coordinate dei primi

punti trovati con intensità maggiore di zero. Per il lato inferiore si procede allo stesso modo ma campionando dalla riga di indice massimo verso l'alto. Anche in queste scansioni l'algoritmo si ferma a righe prefissate nel caso in cui in una colonna non fosse presente alcun pixel con energia maggiore di zero.

Si può inserire un vincolo sul numero minimo di punti che devono essere presenti in entrambi i bordi per assicurare che la tavola sia elaborabile.

### 5.2.3 Ransac con pesatura Gaussiana

Per individuare la retta più rappresentativa passante per un set di punti è stata utilizzata una versione di Ransac che definisce una pesatura Gaussiana per assegnare lo score della retta candidata. Ransac è necessario per approssimare nel modo più preciso i bordi della tavola senza considerare eventuali punti outlier dovuti a rumore o fattori esterni di disturbo.

La pesatura Gaussiana è applicata sulla minima distanza tra la retta candidata e tutti i punti del set analizzato. Con questo metodo si possono ottenere stime più precise dell'accuratezza di una retta nell'interpolare i punti campionati rispetto alla versione più classica di Ransac dove non si considera il peso dei singoli inlier e outlier.

Definito  $P$  come l'insieme di punti da interpolare e  $ax + by + c = 0$  la retta candidata, l'algoritmo utilizzato calcola lo score  $s$  come:

$$s = \sum_{p \in P} e^{-\left(\frac{d_p}{k}\right)^2} \quad (5.3)$$

Con  $d_p = \frac{|ap(x)+bp(y)+c|}{\sqrt{a^2+b^2}}$  minima distanza tra la retta e il punto  $p$ .

Nell'interpolazione dei bordi superiore e inferiore della tavola è stata implementata un'ulteriore modifica all'algoritmo Ransac: i due punti scelti casualmente sono sempre l'uno appartenente alla prima metà del set e l'altro appartenente alla seconda metà del set. I vettori di punti dati in input all'algoritmo di Ransac sono sempre in ordine di colonna crescente per come è stato implementato il campionamento, quindi forzando i punti casuali ad essere scelti in due diverse regioni del bordo permette di stimare solo rette che sono calcolate considerando la totale curvatura della tavola di legno e non solo una sua minima parte.

Il problema della selezione non ragionata dei punti casuali può pesare molto anche in tavole che siano relativamente regolari, come si può notare in

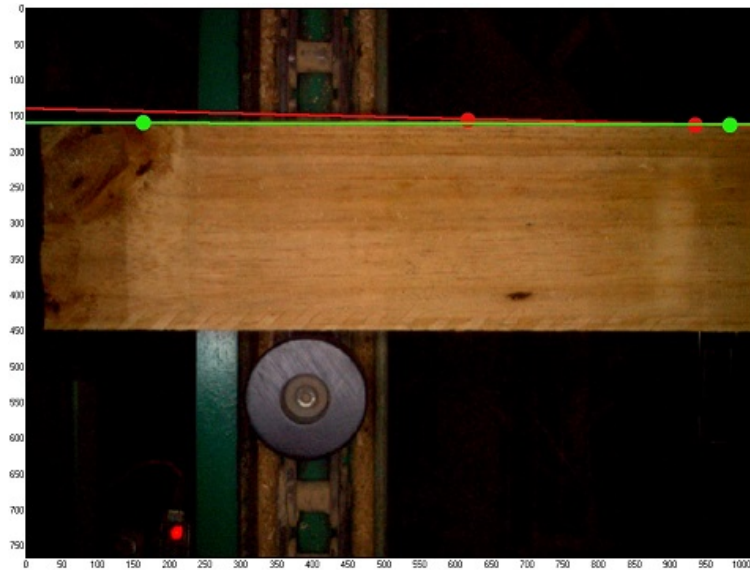


Figura 5.4: Selezione casuale dei punti per Ransac. In rosso una possibile situazione sconveniente della versione classica; in verde una possibile situazione con la modifica implementata che limita i casi indesiderati.

Figura5.4; la modifica sulla selezione casuale appena presentata ha infatti apportato notevoli miglioramenti nell'approssimazione dei bordi delle tavole.

## 5.2.4 Invarianza alla rotazione

Estrarre un descrittore invariante alla rotazione è essenziale per il riconoscimento delle tavole: rotazioni parallele all'asse di movimento della linea possono accadere spesso in quanto il legno è in parte libero di muoversi e slittare durante il trasporto nonostante le guide in ferro.

Nell'algoritmo implementato l'invarianza alla rotazione è ottenuta tramite una trasformazione geometrica affine che porta il bordo superiore di tutte le tavole ad essere parallelo al bordo dell'immagine. In particolare si individua lo spigolo in alto a sinistra  $P = (x_p, y_p)$  della tavola analizzata intersecando le rette *verticale* e *orizzontale superiore* precedentemente calcolate. Definita la retta del bordo superiore come  $ax + by + c = 0$ , si effettua quindi una rotazione di centro  $P$  e angolo

$$\alpha = \tan^{-1}\left(\frac{\frac{a(x_p+\delta)+c}{b} - y_p}{\delta}\right) \quad (5.4)$$

Con  $\delta$  valore arbitrario (nei test è stato fissato al valore 300).

Definito come  $P_1 = (x_1, y_1)$  un punto appartenente ad una delle rette stimate prima della rotazione, è possibile trovare la relativa coordinata  $y$  ruotata applicando la trasformazione:

$$y_{1rotated} = \sin(-\alpha)(x_1 - x_p) + \cos(-\alpha)(y_1 - y_p) + y_p \quad (5.5)$$

### 5.2.5 Invarianza alla scala

L'invarianza alla scala è necessaria nel caso in cui le telecamere delle Stazioni siano posizionate ad altezza diversa rispetto al nastro. Anche nel caso di un perfetto posizionamento le tavole potrebbero comunque inclinarsi o appoggiare su un elemento esterno che potrebbe causarne un avvicinamento o un allontanamento rispetto all'immagine delle altre Stazioni.

Per garantire l'invarianza alla scala si attua un resize ad un numero di righe fisso su tutte le tavole. Si estraggono le tavole precedentemente ruotate utilizzando come contorni: il bordo superiore ruotato, la retta parallela all'asse  $y$  di equazione  $x = x_p$  (coordinata  $x$  del centro di rotazione), la retta parallela all'asse  $x$  di equazione  $y = y_{min}$  dove  $y_{min}$  è il minimo valore assunto dalla retta che approssima il bordo inferiore all'interno della regione occupata dalla tavola.

Definito con  $r$  il numero di righe della regione estratta, si procede ad un resize tramite interpolazione bilineare con coefficiente di scala pari a:

$$k = \frac{\Delta}{r} \quad (5.6)$$

Dove  $\Delta$  è il numero di righe prefissato uguale per tutte le tavole.

Nelle Figure 5.5, 5.6, 5.7, 5.8 si illustrano degli esempi delle operazioni descritte finora per estrarre la regione di interesse a partire dalle immagini ricevute in input. Si può notare come l'algoritmo sia in grado di estrarre le tavole al variare della luminosità del legno, della rotazione, della forma anche in caso di crepe o porzioni mancanti e della diversa scala alle due Stazioni.

## 5.3 Creazione del descrittore

Il descrittore di una tavola deve rappresentare univocamente tale tavola all'interno del dataset di immagini. Le caratteristiche visive salienti presenti nelle

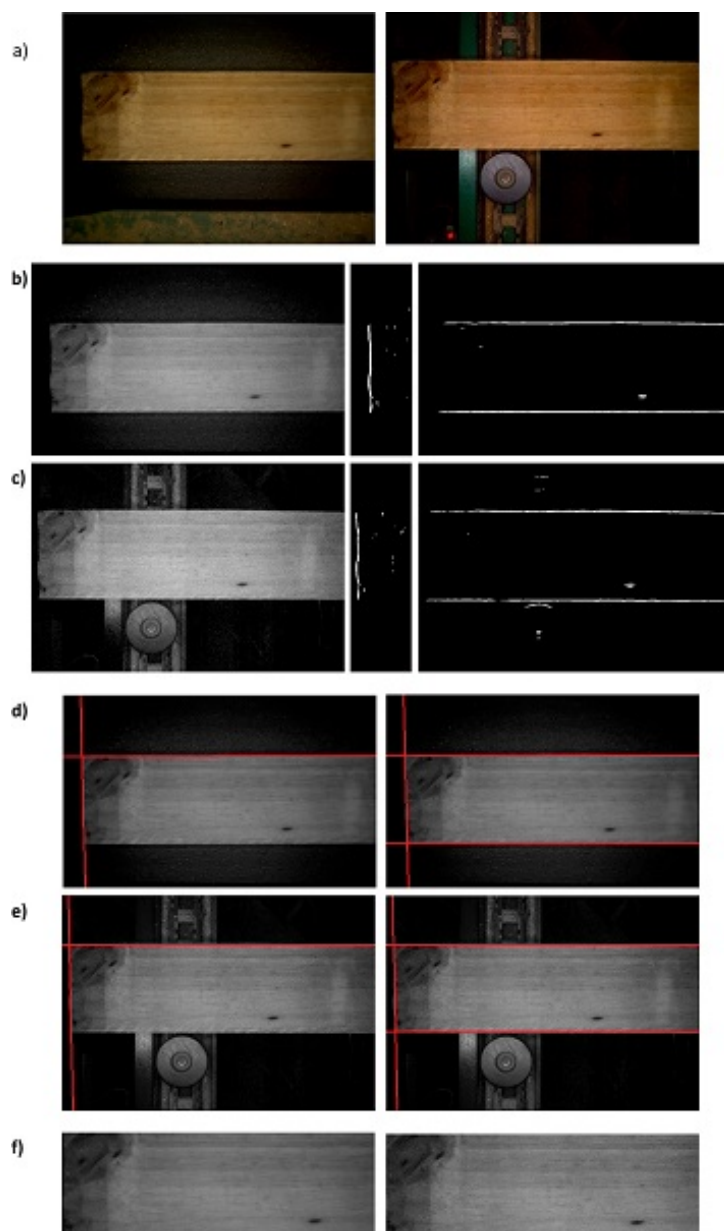


Figura 5.5: a) Immagini originali, a sinistra Stazione 1 a destra Stazione 2; b) Estrazione dei bordi Stazione 1; c) Estrazione dei bordi Stazione 2; d) Stazione 1: a sinistra computazione del punto di rotazione, a destra post-rotazione e bordi evidenziati (si ricorda che il bordo inferiore è dato dal minimo valore di riga della retta trovata sul lato della tavola); e) Stazione 2: a sinistra computazione del punto di rotazione, a destra post-rotazione e bordi evidenziati; f) tavole normalizzate estratte, a sinistra Stazione 1 a destra Stazione 2.



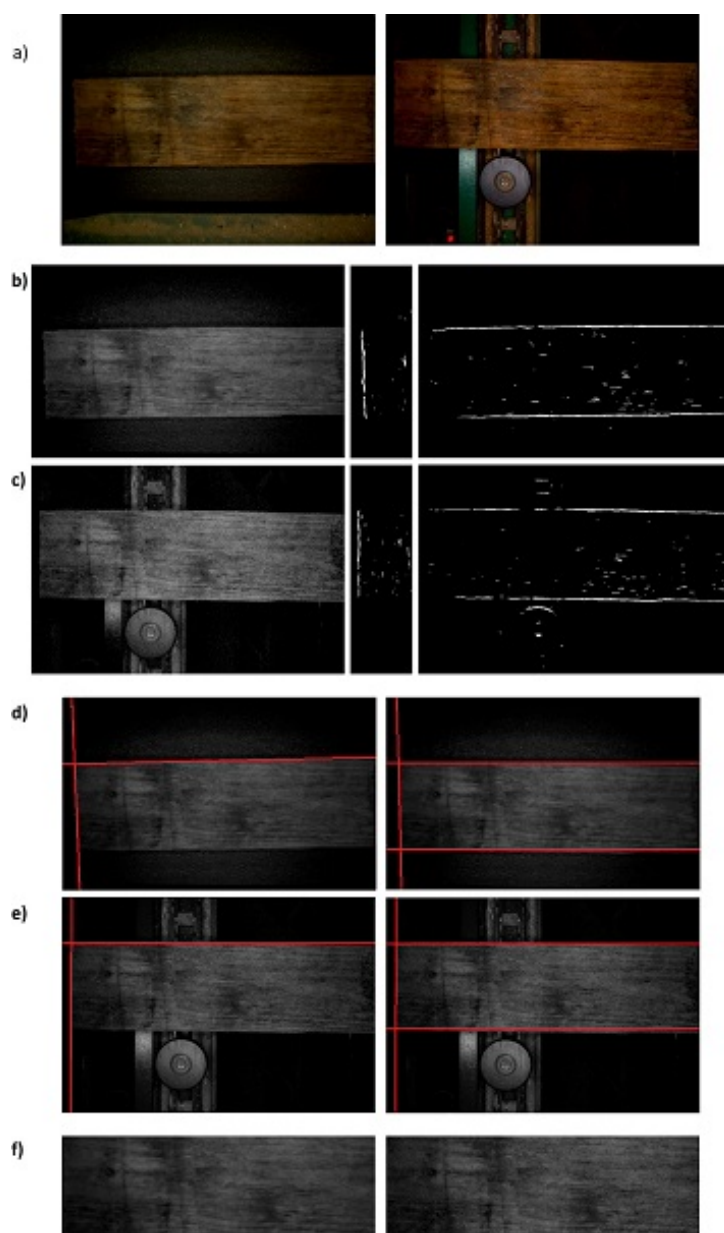


Figura 5.6: a) Immagini originali, a sinistra Stazione 1 a destra Stazione 2; b) Estrazione dei bordi Stazione 1; c) Estrazione dei bordi Stazione 2; d) Stazione 1: a sinistra computazione del punto di rotazione, a destra post-rotazione e bordi evidenziati (si ricorda che il bordo inferiore è dato dal minimo valore di riga della retta trovata sul lato della tavola); e) Stazione 2: a sinistra computazione del punto di rotazione, a destra post-rotazione e bordi evidenziati; f) tavole normalizzate estratte, a sinistra Stazione 1 a destra Stazione 2.

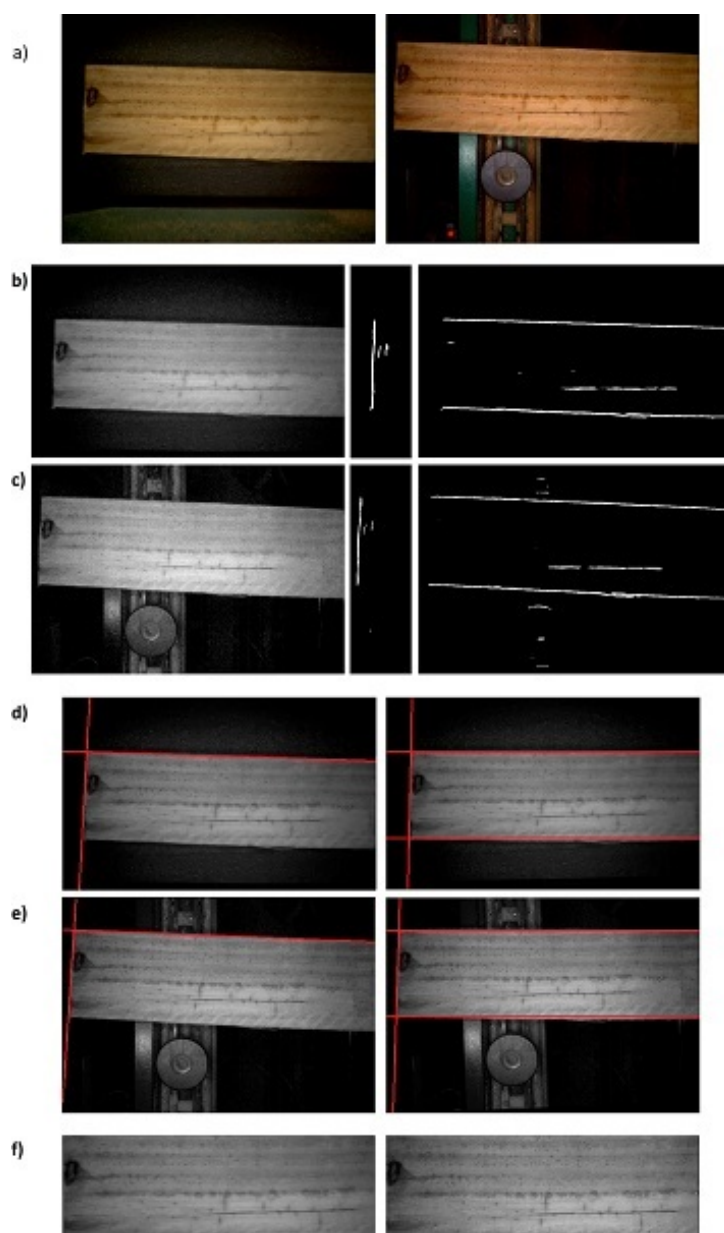


Figura 5.7: a) Immagini originali, a sinistra Stazione 1 a destra Stazione 2; b) Estrazione dei bordi Stazione 1; c) Estrazione dei bordi Stazione 2; d) Stazione 1: a sinistra computazione del punto di rotazione, a destra post-rotazione e bordi evidenziati (si ricorda che il bordo inferiore è dato dal minimo valore di riga della retta trovata sul lato della tavola); e) Stazione 2: a sinistra computazione del punto di rotazione, a destra post-rotazione e bordi evidenziati; f) tavole normalizzate estratte, a sinistra Stazione 1 a destra Stazione 2.

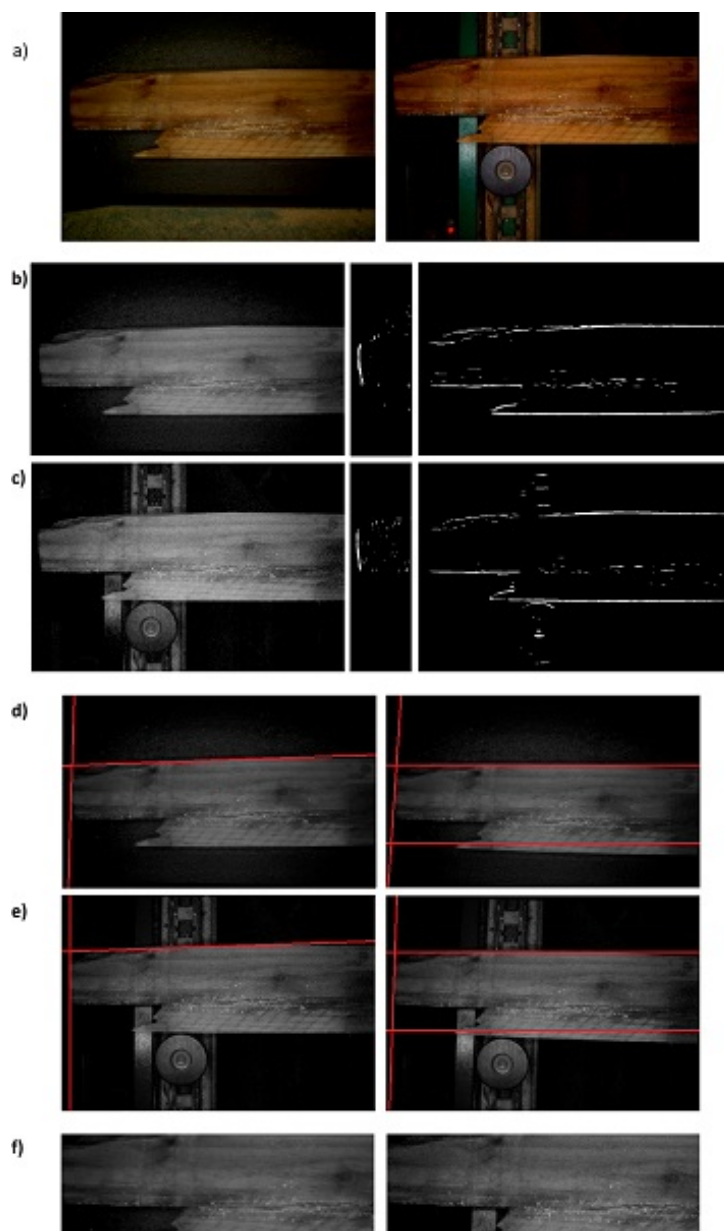


Figura 5.8: a) Immagini originali, a sinistra Stazione 1 a destra Stazione 2; b) Estrazione dei bordi Stazione 1; c) Estrazione dei bordi Stazione 2; d) Stazione 1: a sinistra computazione del punto di rotazione, a destra post-rotazione e bordi evidenziati (si ricorda che il bordo inferiore è dato dal minimo valore di riga della retta trovata sul lato della tavola); e) Stazione 2: a sinistra computazione del punto di rotazione, a destra post-rotazione e bordi evidenziati; f) tavole normalizzate estratte, a sinistra Stazione 1 a destra Stazione 2.

regioni estratte sono: venature, nodi e meno frequentemente crepe.

Al fine di individuare il miglior descrittore che permettesse di evidenziare tali elementi del legno sono stati testati i seguenti metodi: Haar Wavelet, Gaussian Hermite moments. Inoltre è stato sviluppato e testato un descrittore (VDi) basato su semplici caratteristiche statistiche dell'immagine. Tutti i descrittori sono calcolati su un'immagine ottenuta tramite combinazione lineare dei canali rosso, verde, blu dell'immagine originale.

Si procede ora a descrivere una per volta l'implementazione dei metodi.

### 5.3.1 Haar Wavelet

La wavelet di Haar permette di descrivere l'intera area estratta comprimendola incrementalmente fino al fattore desiderato pur mantenendo le principali informazioni di variazione di intensità.

Si calcola la Haar Wavelet di livello 4 sulla regione estratta della tavola dopo aver normalizzato i valori di intensità nell'intervallo  $[0,1]$  e dopo aver applicato un filtro mediano al fine di minimizzare l'entità di eventuale rumore.

Ad ogni livello si calcola prima la Wavelet monodimensionale  $I_w$  sulle righe dell'immagine  $I$  di dimensione  $C \times R$  con le formule:

$$I_w(x, y) = \frac{I(x, 2y) + I(x, 2y + 1)}{2} \quad (5.7)$$

$$I_w(x, h + y) = \frac{I(x, 2y) - I(x, 2y + 1)}{2} \quad (5.8)$$

Con  $h = R/2$ ,  $x \in [0, C - 1]$  e  $y \in [0, R/2 - 1]$ .

Successivamente si calcola la Wavelet monodimensionale  $I_w$  sulle colonne dell'immagine  $I$  con le formule:

$$I_w(x, y) = \frac{I(2x, 2y) + I(2x + 1, y)}{2} \quad (5.9)$$

$$I_w(w + x, y) = \frac{I(2x, y) - I(2x + 1, y)}{2} \quad (5.10)$$

Con  $w = C/2$ ,  $x \in [0, C/2 - 1]$  e  $y \in [0, R - 1]$ .

Il descrittore dell'immagine è composto da tutti i valori del quarto di  $I_w$  (al quarto livello) in basso a sinistra.

Per come l'algoritmo è stato implementato l'immagine  $I_w$  contiene ad ogni passo:

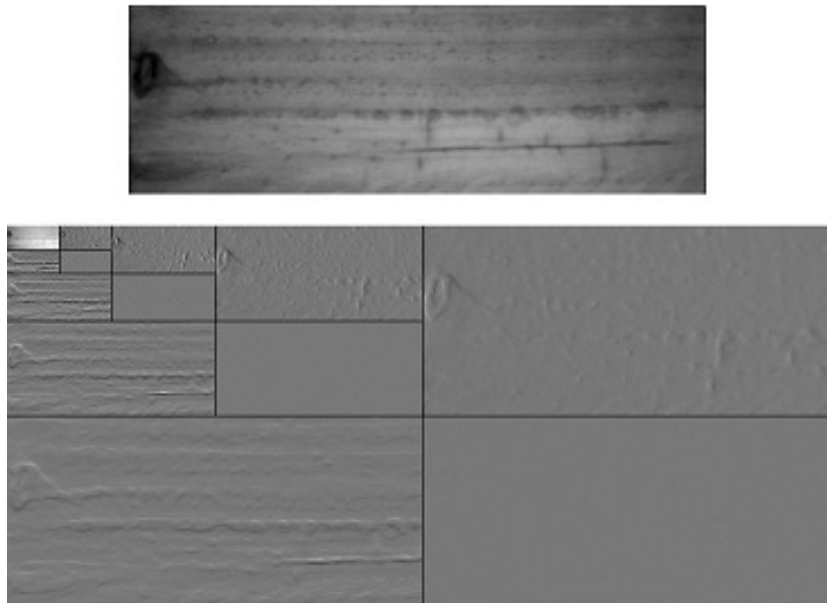


Figura 5.9: Esempio di computazione della wavelet di Haar di quarto ordine su una tavola estratta dall'immagine.

- Quadrante alto-sinistra: media su quattro punti dell'immagine di input, si ottiene l'effetto di un filtro bidimensionale passa basso (LL)
- Quadrante alto-destra: media del gradiente orizzontale, si ottiene l'effetto di un filtro passa basso verticale e di un filtro passa alto orizzontale (HL)
- Quadrante basso-sinistra: media del gradiente verticale, si ottiene l'effetto di un filtro passa basso orizzontale e di un filtro passa alto verticale (LH)
- Quadrante basso-destra: curvatura diagonale, si ottiene l'effetto di un filtro passa alto bidimensionale (HH)

Come descritto in precedenza, le caratteristiche visuali delle tavole di legno sono nodi e venature: i nodi sono solitamente di forma vagamente circolare mentre le venature sono linee di intensità inferiore rispetto al restante legno, con percorso prevalentemente orizzontale rispetto all'orientamento delle telecamere. Dopo aver elaborato l'immagine con la wavelet di Haar la porzione di immagine risultante che permette di osservare al meglio queste caratteristiche distintive è quella relativa al filtraggio LH. Il gradiente verticale evidenzia le regioni iniziali e finali delle venature, descrivendo allo stesso tempo il livello di intensità della venatura stessa; anche i nodi vengono descritti in quanto la forma tondeggiante è rilevata dal gradiente verticale. Sfruttando l'azione di

subsampling della trasformata di Haar è possibile mantenere la descrizione di tutta la regione di interesse senza ottenere un descrittore troppo pesante. Il descrittore al livello quattro è composto da  $\frac{RC}{16^2}$  valori floating point.

Per semplificare i confronti tra descrittori, si salvano i valori calcolati come vettori monodimensionali in ordine row-major e non come matrice bidimensionale, senza perdita di informazione.

### 5.3.2 VDi

VDi è l'acronimo per Variance of Difference, in quanto il metodo utilizza le operazioni differenza e varianza nella fase di matching.

Il descrittore di una tavola è calcolato scomponendo l'immagine  $I$  di dimensioni  $C \times R$  in  $N$  colonne, ognuna di larghezza  $w$ :

$$X_k = \begin{bmatrix} I(kw, 0) & I(kw + 1, 0) & \dots & I(kw + w - 1, 0) \\ I(kw, 1) & I(kw + 1, 1) & \dots & I(kw + w - 1, 1) \\ \dots & \dots & \dots & \dots \\ I(kw, C - 1) & I(kw + 1, C - 1) & \dots & I(kw + w - 1, C - 1) \end{bmatrix} \quad k = 0, \dots, N-1 \quad (5.11)$$

Per ogni colonna si estrae il segnale medio in direzione verticale utilizzando l'informazione delle prime  $\delta$  colonne e delle  $r$  righe centrali (per evitare possibile rumore nelle zone limitrofe i bordi della tavola):

$$M_k = \begin{bmatrix} \sum_{j=0}^{\delta-1} X_k(j, \frac{R-r}{2}) \\ \sum_{j=0}^{\delta-1} X_k(j, \frac{R-r}{2} + 1) \\ \dots \\ \sum_{j=0}^{\delta-1} X_k(j, \frac{R-r}{2} + r - 1) \end{bmatrix} \quad k = 0, \dots, N-1 \quad (5.12)$$

La concatenazione dei valori ottenuti compone il descrittore.

La scomposizione in segnali monodimensionali lungo le colonne dell'immagine permette di cogliere appieno le variazioni di intensità dovute a venature

e nodi. Si ha inoltre il vantaggio di caratterizzare con precisione la posizione di tali elementi all'interno della tavola analizzata. La media su  $\delta$  colonne permette di attenuare l'effetto di possibili disallineamenti sull'asse orizzontale. Le variazioni di informazione dovute a variazioni di luminosità vengono mantenute nel descrittore ma sono risolte nel particolare metodo di matching descritto in seguito. Il descrittore è composto da  $Nr$  valori.

### 5.3.3 Gaussian Hermite moments

Per calcolare il descrittore che applica i momenti Gaussian Hermite si procede a scomporre l'immagine con le stesse operazioni descritte precedentemente:

$$X_k = \begin{bmatrix} I(kw, 0) & I(kw + 1, 0) & \dots & I(kw + w - 1, 0) \\ I(kw, 1) & I(kw + 1, 1) & \dots & I(kw + w - 1, 1) \\ \dots & \dots & \dots & \dots \\ I(kw, C - 1) & I(kw + 1, C - 1) & \dots & I(kw + w - 1, C - 1) \end{bmatrix} \quad k = 0, \dots, N-1 \quad (5.13)$$

$$M_k = \begin{bmatrix} \sum_{j=0}^{\delta-1} X_k(j, \frac{R-r}{2}) \\ \sum_{j=0}^{\delta-1} X_k(j, \frac{R-r}{2} + 1) \\ \dots \\ \sum_{j=0}^{\delta-1} X_k(j, \frac{R-r}{2} + r - 1) \end{bmatrix} \quad k = 0, \dots, N-1 \quad (5.14)$$

$$M_{k \text{ norm}} = \begin{bmatrix} \frac{M_k(0) - M_{k \text{ min}}}{M_{k \text{ max}} - M_{k \text{ min}}} \\ \dots \\ \frac{M_k(r-1) - M_{k \text{ min}}}{M_{k \text{ max}} - M_{k \text{ min}}} \end{bmatrix} \quad k = 0, \dots, N-1 \quad (5.15)$$

Ottenendo  $N$  segnali di lunghezza  $r$  denominati  $M_k$  con  $k = 0, \dots, N-1$ .  $M_{k \text{ norm}}$  sono i segnali  $M_k$  normalizzati tra 0 e 1 applicando la formula riportata.

I momenti di Gauss-Hermite con parametro  $\sigma = 1$  vengono calcolati per ogni segnale  $M_k$  normalizzato seguendo la pipeline indicata in (5.10). Si segue questo modello implementativo in quanto è decisamente più veloce del modello

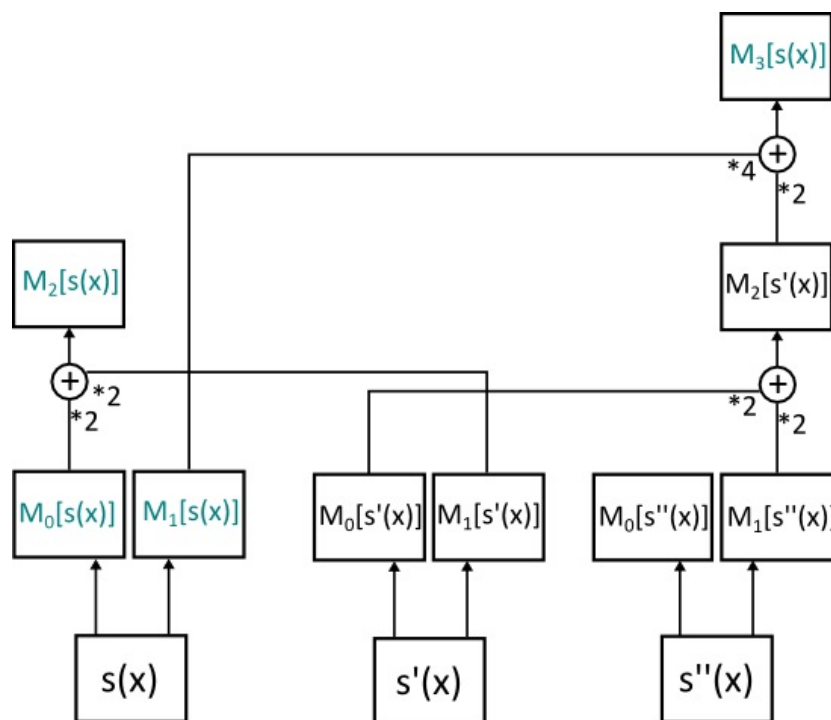


Figura 5.10: Schema per il calcolo dei momenti di Gauss-Hermite di ordine 1,2,3 con parametro  $\sigma = 1$  sul segnale  $s(x)$ .

ricorsivo, che non sfrutta le informazioni già computate ai livelli precedenti.

Anche in questo caso la scomposizione dell'immagine in segnali sulla direzione delle colonne permette di descrivere le variazioni di intensità dovute a venature e nodi. I momenti di Gauss-Hermite permettono di evidenziare ulteriormente le variazioni di energia nei segnali, caratterizzando la tavola analizzata in maniera ottimale. La media su  $\delta$  colonne permette di attenuare l'effetto di possibili disallineamenti sull'asse orizzontale. L'azione di blurring (Gaussiano) che è propria dei momenti di Gauss-Hermite permette al descrittore di adattarsi al possibile rumore presente nell'immagine: fattori esterni come schegge o sporco vengono pesati sul vicinato contenuto nella finestra di blur in modo da eliminarne in parte l'effetto sul segnale risultante, fattori intrinseci come il rumore dovuto alla telecamera vengono eliminati grazie all'alta probabilità che i pixel vicini alla zona rumorosa non siano affetti dallo stesso errore di acquisizione.



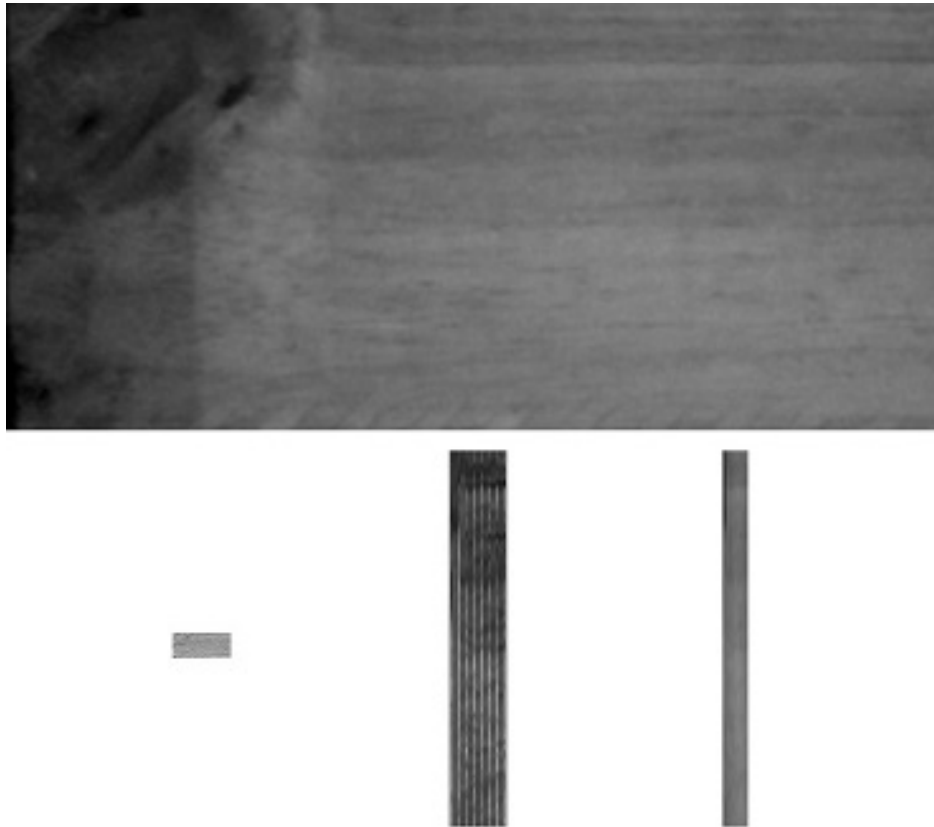


Figura 5.11: Esempio 1 di tavola estratta (normalizzata e ritagliata a dimensione 336x736 pixels) e dei tre descrittori calcolati. Le proporzioni sono state mantenute a scopo dimostrativo. Descrittori: a sinistra wavelet di Haar ordine 4 (in forma di matrice), al centro momenti Gaussian-Hermite di ordine 1,2,3 (con valori normalizzati tra 0 e 255 per una migliore visualizzazione) su quindici gruppi di colonne, a destra VDi su venti gruppi di colonne.

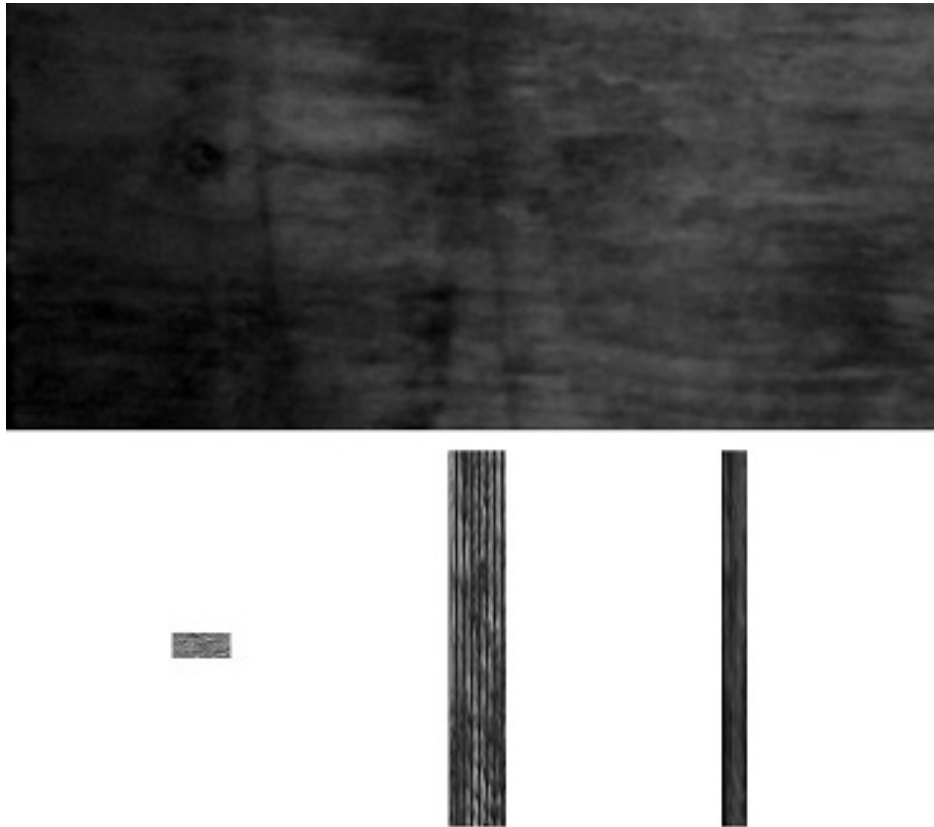


Figura 5.12: Esempio 2 di tavola estratta (normalizzata e ritagliata a dimensione 336x736 pixels) e dei tre descrittori calcolati. Le proporzioni sono state mantenute a scopo dimostrativo. Descrittori: a sinistra wavelet di Haar ordine 4 (in forma di matrice), al centro momenti Gaussian-Hermite di ordine 1,2,3 (con valori normalizzati tra 0 e 255 per una migliore visualizzazione) su quindici gruppi di colonne, a destra VDi su venti gruppi di colonne.

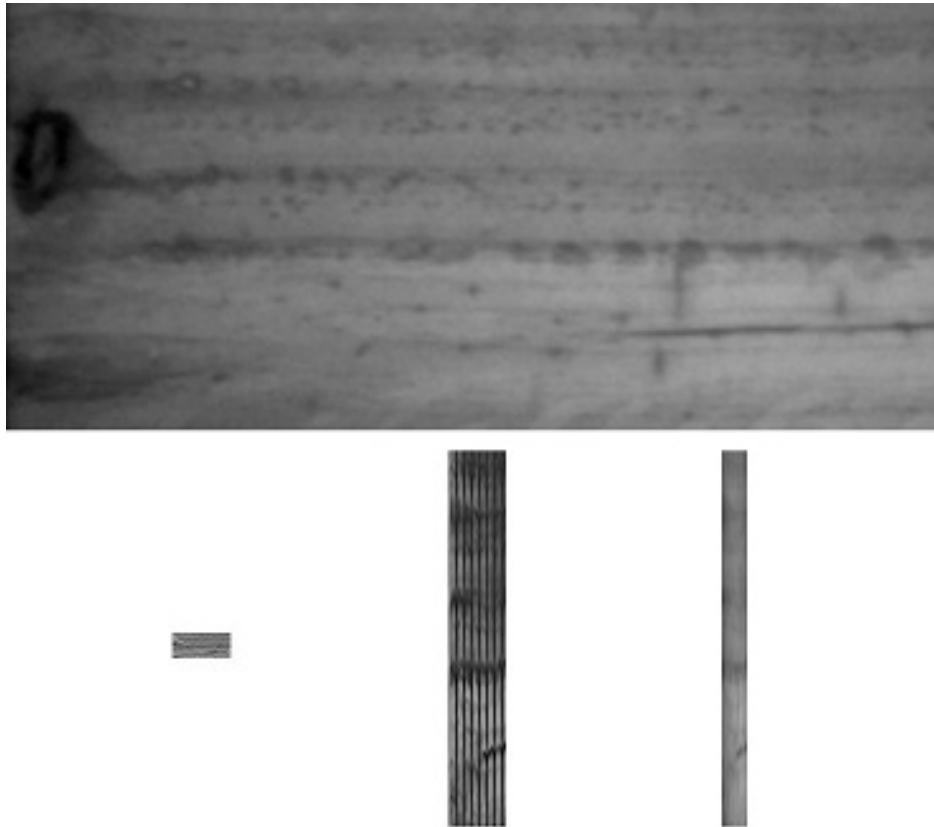


Figura 5.13: Esempio 3 di tavola estratta (normalizzata e ritagliata a dimensione 336x736 pixels) e dei tre descrittori calcolati. Le proporzioni sono state mantenute a scopo dimostrativo. Descrittori: a sinistra wavelet di Haar ordine 4 (in forma di matrice), al centro momenti Gaussian-Hermite di ordine 1,2,3 (con valori normalizzati tra 0 e 255 per una migliore visualizzazione) su quindici gruppi di colonne, a destra VDi su venti gruppi di colonne.

## 5.4 Distanza di matching

Conseguentemente la catena di operazioni descritte in precedenza, la ricerca delle corrispondenze tra immagini acquisite in tempo reale dal sistema si riduce al confronto tra descrittori. Dal confronto di due descrittori è necessario estrapolare una metrica che valuta la similarità dei due elementi in analisi. Tale metrica deve essere:

- esattamente riproducibile
- enfatizzare la separazione tra match e non-match
- i valori che la compongono devono essere ordinabili e confrontabili
- calcolata in modo coerente con il significato dei valori contenuti nel descrittore

Si ricorda che i descrittori possono avere dimensione variabile sull'asse delle colonne, dipendente dalla posizione della tavola all'interno dell'immagine, mentre la dimensione sull'asse delle righe è sempre uguale, caratteristica derivante dal ridimensionamento ad altezza fissa della regione estratta.

### 5.4.1 Haar wavelet

I descrittori calcolati tramite la wavelet di Haar sono i meno estesi dimensionalmente e i più semplici da confrontare. Si ricorda inoltre che in fase di costruzione sono salvati come vettori monodimensionali e non come matrici bidimensionali. Il subsampling di fattore due che viene applicato ad ogni livello durante il calcolo della wavelet non comporta perdita di robustezza nonostante il numero di colonne possibilmente diverso, poichè la trasformazione è comunque applicata su tutta la regione analizzata. I valori di intensità vengono normalizzati prima di procedere alla costruzione della wavelet, quindi i valori che compongono il descrittore sono confrontabili tra loro senza ulteriori trasformazioni.

Presi due generici descrittori  $D_1$  e  $D_2$  rispettivamente di dimensioni  $D_{1n}$  e  $D_{2n}$ , per ottenere una distanza di matching  $\Delta m$  che abbia le caratteristiche elencate precedentemente si individua la lunghezza minima tra i due descrittori  $N_{min} = \min(D_{1n}, D_{2n})$  e si procede a calcolare la distanza euclidea modificata

in modo da considerare  $N_{min}$ :

$$\Delta m = \sqrt{\frac{\sum_{i=0}^{N_{min}} (D_1(i) - D_2(j))^2}{N_{min}}} \quad (5.16)$$

Considerando solo i primi  $N_{min}$  valori nel calcolo della distanza di matching si tiene in considerazione il diverso numero di colonne che può formare le regioni estratte dall'immagine originale. Il salvataggio in ordine column-major permette di mantenere tale informazione anche se il descrittore è salvato in forma di vettore monodimensionale, poichè il numero di righe delle tavole estratte è costante.

#### 5.4.2 VDi

I descrittori ottenuti con il metodo VDi aumentano il proprio valore di classificatori grazie alla tecnica di matching: dati due segnali appartenenti allo stesso indice di colonna di due descrittori VDi, se appartengono alla stessa tavola la varianza della differenza dei due segnali sarà prossima al valore zero.

La differenza dei due segnali permette di valutare quanto i segnali abbiano forma simile; se gli input sono simili ci si aspetta di avere un segnale risultante che approssima una retta che per ogni indice di riga assume valore pari alla differenza lineare di luminosità tra i due segnali. La varianza del segnale ottenuto permette di ottenere l'invarianza ai cambiamenti lineari di luminosità, poichè si può vedere tale variazione come una traslazione dell'intero segnale sull'asse dei valori di intensità.

I descrittori possono avere dimensione diversa sull'asse delle colonne, dipendente dalla posizione della tavola all'interno dell'immagine, mentre la dimensione sull'asse delle righe è sempre uguale, caratteristica derivante dal ridimensionamento ad altezza fissa della regione estratta.

La distanza di matching  $\Delta m$  tra i descrittori  $D_1$  e  $D_2$  rispettivamente di dimensioni  $(D_{1r}, D_{1c})$  e  $(D_{2r}, D_{2c})$  è calcolata come:

$$\Delta m = \frac{\sum_{i=0}^{C_{min}} \sum_{j=0}^{D_{1r}} \frac{(|D_1(i,j) - D_2(i,j)| - M(i))^2}{D_{1r}}}{C_{min}} \quad (5.17)$$

Dove  $C_{min} = \min(D_{1c}, D_{2c})$  e  $M(i)$  è la media della differenza dei due descrittori alla colonna  $i$ -esima:

$$M(i) = \sum_{j=0}^{D_{1r}} \frac{|D_1(i, j) - D_2(i, j)|}{D_{1r}} \quad (5.18)$$

L'intero metodo VDi si basa su semplici coefficienti statistici della tavola estratta, nonostante questa semplicità teorica si osserveranno in seguito (6.3) dei buoni risultati nel riconoscimento.

### 5.4.3 Gaussian-Hermite moments

I descrittori calcolati applicando i momenti di Gauss-Hermite presentano tutte le caratteristiche di invarianza desiderate a prescindere dal metodo utilizzato per confrontarli.

Per ottenere la distanza di matching tra i descrittori  $D_1$  e  $D_2$  rispettivamente di dimensioni  $(D_{1r}, D_{1c})$  e  $(D_{2r}, D_{2c})$  si applica la formula:

$$\Delta m = \left[ \frac{15}{D_{1r} C_{min}} \sum_{i=0}^{C_{min}} \sum_{j=0}^{D_{1r}} (D_1(i, j) - D_2(i, j))^2 \right]^2 \quad (5.19)$$

Dove  $C_{min} = \min(D_{1c}, D_{2c})$ .

Il coefficiente  $\frac{15}{D_{1r} C_{min}}$  è inserito per considerare il numero di righe e di colonne di cui il descrittore è formato; valori entrambi variabili nel caso più generico. L'elevamento a potenza più esterno permette invece di distanziare maggiormente le distanze di vero matching da quelle di non-matching, poichè è stato osservato che nella maggioranza dei casi se due tavole corrispondono la radice della loro distanza è minore di 1; al contrario se due tavole non corrispondono la radice della loro distanza è maggiore di 1.

# Capitolo 6

## Testing dell'algoritmo

In questo capitolo si esaminano le prestazioni dell'algoritmo. Si analizzano: la precisione nel riconoscimento, la soglia ottimale per il miglior compromesso tra falsi riconoscimenti e falsi negativi, i tempi di esecuzione di ogni fase.

### 6.1 Dataset

Allo scopo di verificare la precisione degli algoritmi implementati è stato raccolto un dataset apposito. Il dataset è composto da 15191 coppie di immagini acquisite all'interno di una linea di produzione in condizioni equivalenti a quelle di un'installazione definitiva. Ogni coppia di immagini riguarda la stessa tavola di pino radiata: la prima immagine è scattata alla Stazione 1 mentre la seconda immagine è scattata alla Stazione 2.

Le immagini analizzate presentano tutte le variazioni che l'algoritmo deve essere in grado di superare. Tali fattori si riscontrano sia singolarmente sia combinati:

- rotazioni sul piano di trasporto
- variazioni di luminosità
- messa a fuoco non equivalente
- scala non equivalente
- traslazioni
- tavole rotte o con presenza di rumore
- nastro di scorrimento sporco (causa di rumore nella segmentazione)

- guide in ferro della linea di produzione che causano riflessi e rumore
- tavole di legno con diversa intensità di colore
- tavole senza presenza di nodi e/o venature evidenti
- tavole inclinate rispetto al piano di trasporto

Le immagini che compongono il dataset hanno dimensioni 1024x768 con i canali rosso, verde e blu per ogni pixel.

A differenza dei dataset riguardanti il riconoscimento dell'iride dove per ogni individuo si hanno a disposizione un certo numero di immagini di training, nel caso studiato si ha a disposizione una sola immagine corrispondente alla tavola in analisi. Questo comporta un'ulteriore fattore di difficoltà poichè avendo più di un'immagine appartenente allo stesso elemento sarebbe possibile aumentare la probabilità di essere in grado di riconoscerlo univocamente utilizzando la maggiore quantità di informazione.

## 6.2 Definizione dei test

Sono stati effettuati due test per verificare l'efficacia dei tre metodi realizzati:

- One vs All

In questo test per ogni immagine del dataset relativa alla Stazione 1 si calcolano e si salvano in memoria i descrittori. Successivamente per ogni immagine relativa alla Stazione 2 si calcolano i tre descrittori e si esegue il matching con i relativi descrittori di tutte le immagini relative alla Stazione 1. La tavola con distanza di matching più bassa è identificata come la corretta corrispondente.

- One vs 500

In questo test per ogni immagine del dataset relativa alla Stazione 1 si calcolano e si salvano in memoria i descrittori. Successivamente per ogni immagine relativa alla Stazione 2 si calcolano i tre descrittori e si esegue il matching con i relativi descrittori corrispondenti della Stazione 1 e con i descrittori di altre 499 tavole scelte casualmente tra l'intero dataset della Stazione 1. La tavola con distanza di matching più bassa è identificata come la corretta corrispondente.



Il test *One vs All* è pensato in modo da rendere il più difficile possibile l'identificazione della corrispondenza corretta: maggiore è il numero di tavole confrontate e maggiore è la probabilità di un errore dovuto alla presenza di una tavola molto simile a quella analizzata. Rispetto ai possibili scenari in cui l'algoritmo sarà inserito, questo test crea un livello di difficoltà decisamente più alto.

Il test *One vs 500* simula uno scenario applicativo reale: la coda di tavole in attesa di matching nella linea di produzione varia tra le 0 e le 500 tavole. Il fattore di casualità con cui si confronta l'elemento in analisi permette di verificare che l'algoritmo è affidabile anche senza continuità nella coda FIFO della produzione.

Allo scopo di determinare la soglia ottimale per separare i riconoscimenti corretti da quelli non corretti si calcolano i coefficienti FAR (False Acceptance Rate) e FRR (False Reject Rate), che permettono rispettivamente di stimare la probabilità di un falso riconoscimento e la probabilità di scartare un riconoscimento corretto.

$$FAR = \frac{\int_0^k P_{negative}(x) dx}{\int_0^\infty P_{negative}(x) dx} \quad (6.1)$$

$$FRR = \frac{\int_k^\infty P_{positive}(x) dx}{\int_0^\infty P_{positive}(x) dx} \quad (6.2)$$

## 6.3 Risultati

### 6.3.1 Haar Wavelet

I descrittori calcolati applicando la wavelet di Haar sono stati ottenuti dopo aver: ridimensionato la regione di interesse estratta a dimensioni di  $300 \times C$  pixel dove  $C$  è il valore minimo tra 900 e il numero di colonne ottenute dopo il ridimensionamento, applicato un blur mediano con finestra di ampiezza  $11 \times 11$  all'immagine per eliminare possibili elementi di disturbo e forti variazioni di luminosità, normalizzato tra 0 e 1 i valori contenuti nella matrice ottenuta, calcolato la wavelet fino al livello quattro.

Nel test *One vs All* la percentuale di riconoscimento incondizionato è pari a 97.2824%, ovvero non vengono correttamente riconosciute 413 tavole sull'intero dataset.

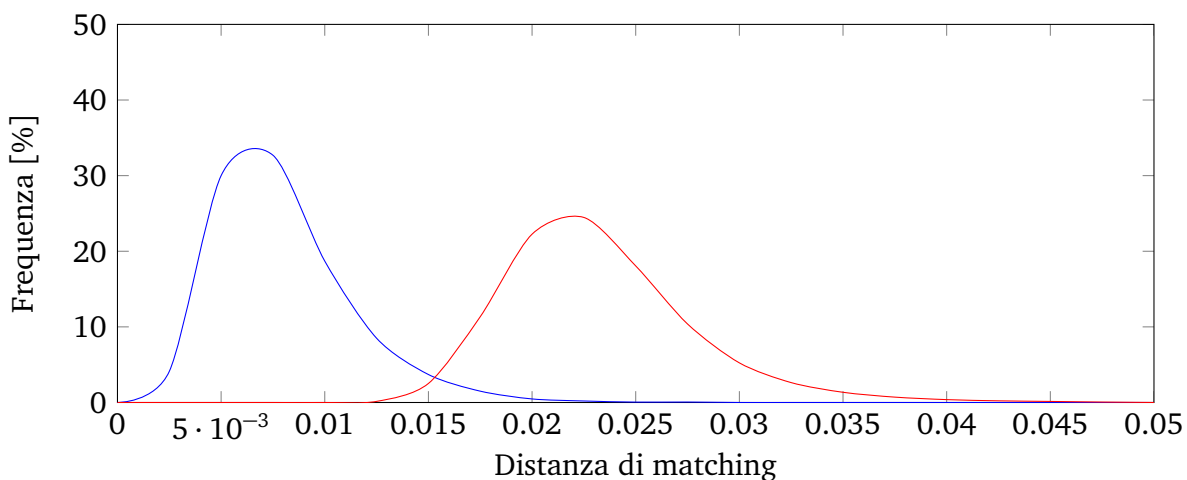


Figura 6.1: Distribuzione della distanza di matching per i descrittori Haar wavelet. In blu la distribuzione delle distanze nei riconoscimenti esatti. In rosso la distribuzione delle distanze nei non-match.

Nel test *One vs 500* la percentuale di riconoscimento incondizionato è pari a 98.5034%, ovvero non vengono correttamente riconosciute 228 tavole sull'intero dataset.

Le distribuzioni delle distanze di matching sono presentate in Figura 6.1 mentre i coefficienti FAR e FRR sono esposti in Tabella 6.1. Si nota che le due distribuzioni non sono nettamente separate; questo comporta una difficoltà nel trovare la soglia che permette di mantenere basso sia il FAR sia il FRR, ovvero il sistema non sarà in grado di accettare un'alta porzione di identificazioni corrette senza favorire i falsi positivi. La soglia migliore per questo tipo di descrittore è 0.0175, con percentuale 2.6640 di falsi positivi accettati e una percentuale 2.2400 di identificazioni corrette scartate. Tali prestazioni sono comunque troppo poco robuste per poter essere utilizzate nel sistema di riconoscimento realizzato.

### 6.3.2 VDi

I descrittori calcolati applicando VDi sono stati ottenuti dopo aver ridimensionato la regione di interesse estratta a dimensioni di  $300 \times C$  pixel dove  $C$  è il valore minimo tra 900 e il numero di colonne ottenute dopo il ridimensionamento. Il descrittore è calcolato utilizzando l'area ottenuta, scomponendola in colonne di larghezza 40 pixel e utilizzando per il calcolo del segnale verticale medio le prime 30 colonne di ogni gruppo. Nel calcolo del descrittore vengono

<b>Soglia</b>	<b>FAR [%]</b>	<b>FRR [%]</b>
0	0	100
0.0025	0	100
0.005	0	95.8855
0.0075	0	65.8659
0.01	0	33.1190
0.0125	0.0007	14.4414
0.015	0.1466	5.9146
0.0175	2.6640	2.2400
0.02	14.0077	0.7850
0.0225	36.2679	0.3181
0.025	60.7417	0.1150
0.0275	78.7982	0.0677
0.03	89.1767	0.0068
0.0325	94.4169	0.0068
0.035	97.0437	0
0.0375	98.3954	0
0.04	99.1094	0
0.0425	99.4871	0
0.045	99.7071	0
0.0475	99.8532	0
0.05	99.9109	0
oltre	100	0

Tabella 6.1: Coefficienti FAR e FRR per i descrittori Haar wavelet.

tralasciate porzioni della regione estratta perchè spesso rumorose.

Nel test *One vs All* la percentuale di riconoscimento incondizionato è pari a 99.2320%, ovvero non vengono correttamente riconosciute 117 tavole sull'intero dataset.

Nel test *One vs 500* la percentuale di riconoscimento incondizionato è pari a 99.7768%, ovvero non vengono correttamente riconosciute 34 tavole sull'intero dataset.

Le distribuzioni delle distanze di matching sono presentate in Figura 6.2 mentre i coefficienti FAR e FRR sono esposti in Tabella 6.2. La distribuzione delle distanze dei riconoscimenti corretti è caratterizzata da un'alta compattezza, gran parte dell'area sottesa alla curva è a sinistra della soglia 30; questa è una caratteristica apprezzabile in un descrittore in quanto dimostra che lo stesso elemento viene rappresentato in modo molto simile. La distribuzione delle distanze dei falsi è invece molto dispersa, soprattutto sul lato destro. La sovrapposizione tra le due distribuzioni non è molto estesa, soprattutto se confrontata con quella dei descrittori Haar wavelet. Le caratteristiche esposte si riscontrano anche osservando la Tabella 6.2, dove si può identificare come apprezzabile la soglia al valore 35 che comporta un coefficiente FAR di 0.3421 e un coefficiente FRR di 1.2605.

Nonostante il fattore di miglioramento rispetto al metodo presentato precedentemente, anche questa configurazione presenta una percentuale troppo alta di possibili falsi positivi. Per diminuire questo errore a un valore accettabile si scarterebbe una porzione troppo alta di riconoscimenti corretti.

<b>Soglia</b>	<b>FAR [%]</b>	<b>FRR [%]</b>
0	0	100
5	0	96.5769
10	0	59.0089
15	0	25.5805
20	0.0011	11.0521
25	0.0158	4.8096
30	0.0946	2.3949
35	0.3421	1.2605
40	0.9081	0.7364
45	2.0024	0.4843
50	3.6835	0.3251
55	6.0998	0.2057
60	9.2187	0.1592
65	12.9362	0.1327
70	17.2152	0.0929
75	21.8749	0.0862
80	26.8165	0.0730
85	31.8246	0.0597
90	36.8855	0.0597
95	41.8019	0.0464
100	46.5671	0.03317

Tabella 6.2: Coefficienti FAR e FRR per i descrittori VDi.

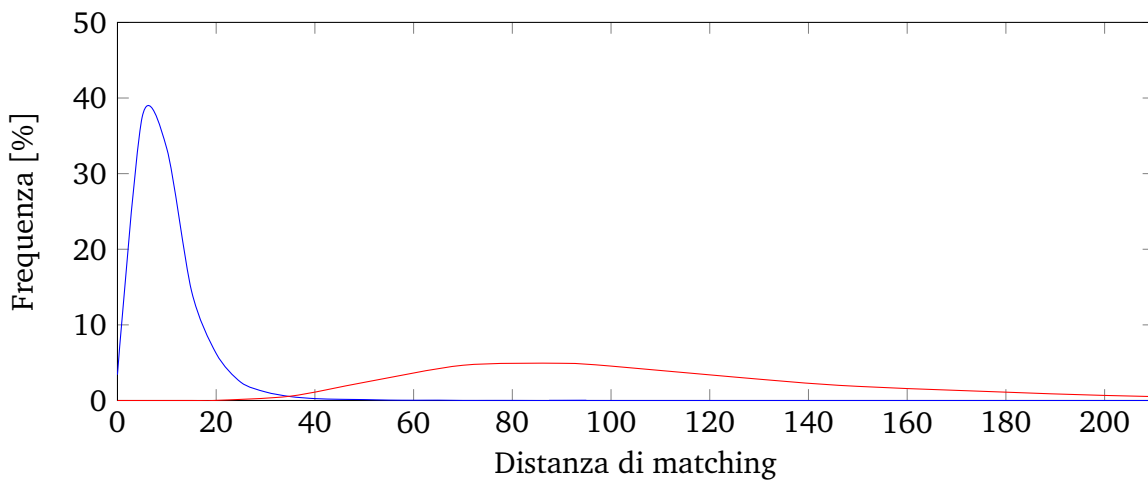


Figura 6.2: Distribuzione della distanza di matching per i descrittori VDi. In blu la distribuzione delle distanze nei riconoscimenti esatti. In rosso la distribuzione delle distanze nei non-match.

### 6.3.3 Gaussian Hermite moments

I descrittori calcolati applicando i momenti di Gauss-Hermite sono stati ottenuti dopo aver ridimensionato la regione di interesse estratta a dimensioni di  $300 \times C$  pixel dove  $C$  è il valore minimo tra 900 e il numero di colonne ottenute dopo il ridimensionamento. Il descrittore è stato calcolato sui segnali verticali medi ottenuti su gruppi di trenta colonne di un pixel, separati tra loro da dieci colonne di un pixel. Le aree dell'immagine che presentavano ripetutamente rumore o problemi di illuminazione non sono state considerate nel calcolo dei descrittori. Su ogni segnale medio verticale normalizzato tra 0 e 1 sono stati computati i momenti di ordine 0,1,2,3 e il descrittore è stato formato utilizzando i momenti di ordine 1,2,3.

Nel test *One vs All* la percentuale di riconoscimento incondizionato è pari a 99.7899%, ovvero non vengono correttamente riconosciute 32 tavole sull'intero dataset.

Nel test *One vs 500* la percentuale di riconoscimento incondizionato è pari a 99.9606%, ovvero non vengono correttamente riconosciute 6 tavole sull'intero dataset.

Analizzando le distribuzioni delle distanze in Figura 6.3 si nota che l'area di sovrapposizione delle due funzioni è notevolmente ridotta rispetto ai metodi precedenti. I coefficienti FAR e FRR evidenziano come questo vada anche a in-

fluire positivamente sul grado di affidabilità del sistema. L'ampia dispersione della distribuzione delle distanze dei non-match, che ha centro in 5.4, permette di avere una crescita notevolmente bassa del fattore FAR anche quando ci pone in prossimità di valori FRR ottimali. Una possibile configurazione che presenta valori soddisfacenti per l'uso in situazioni reali è con soglia 1.1.

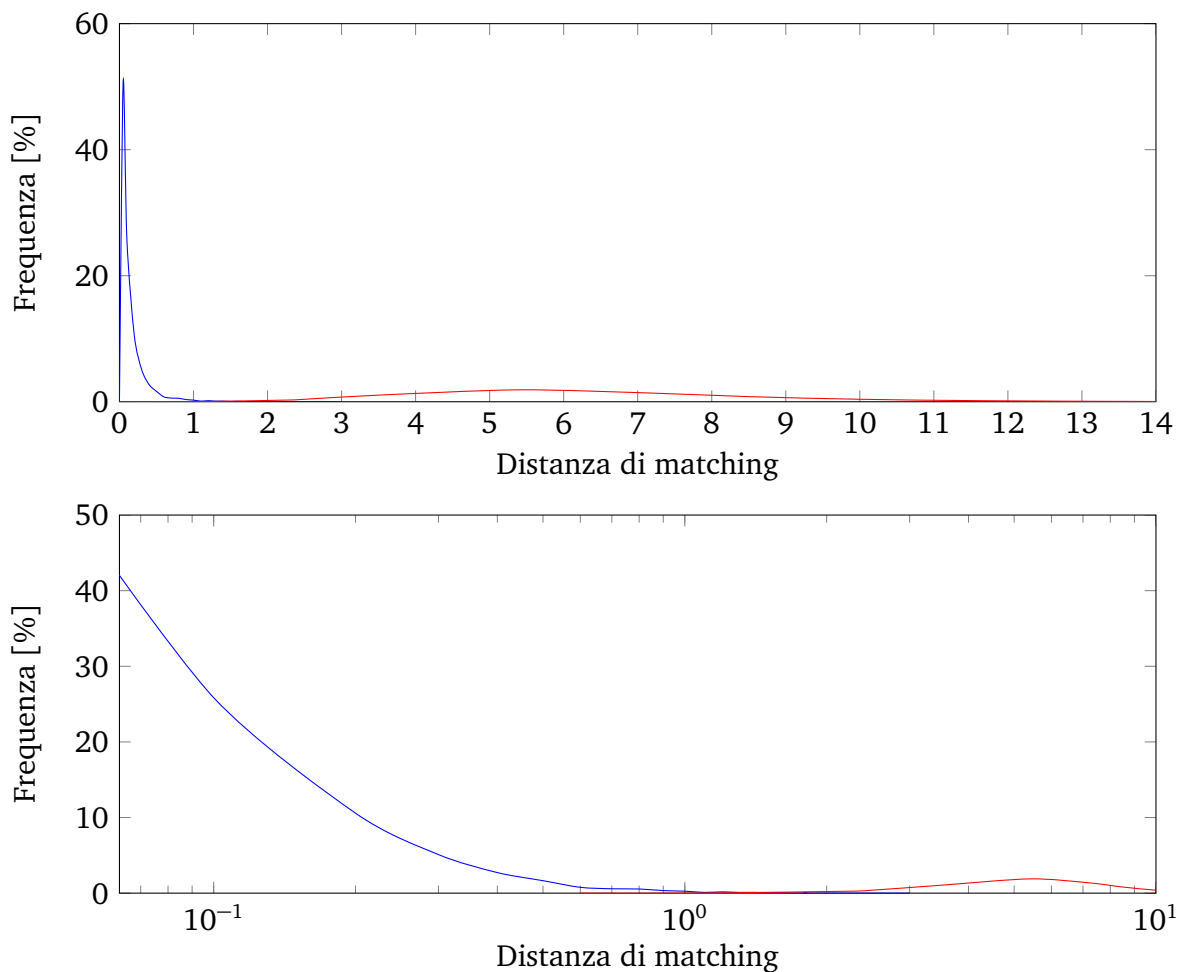


Figura 6.3: Distribuzione della distanza di matching per i descrittori Gaussian-Hermite. In blu la distribuzione delle distanze nei riconoscimenti esatti. In rosso la distribuzione delle distanze nei non-match. Nel grafico in basso, particolare della sovrapposizione tra le due distribuzioni.

<b>Soglia</b>	<b>FAR [%]</b>	<b>FRR [%]</b>
0	0	100
0.1	0	49.1853
0.2	0	23.3393
0.3	0	12.7647
0.4	0.0002	7.6588
0.5	0.0004	4.9344
0.6	0.0009	3.2852
0.7	0.0022	2.5134
0.8	0.0061	1.9328
0.9	0.0114	1.3787
1.0	0.0166	1.0423
1.1	0.0282	0.7784
1.2	0.0470	0.6795
1.3	0.0753	0.509
1.4	0.1063	0.3958
1.5	0.1499	0.2969
1.6	0.2120	0.2441
1.7	0.2864	0.1715
1.8	0.3748	0.1583
1.9	0.4753	0.1253
2	0.6344	0.1121
2.2	1.0083	0.0726
2.4	1.5219	0.0528
2.6	2.2683	0.0396
2.8	3.2051	0.0198
3.0	4.3931	0.0066

Tabella 6.3: Coefficienti FAR e FRR per i descrittori Gaussian-Hermite.



## 6.4 Tempi computazionali

L'analisi dei tempi computazionali dell'algoritmo è determinante per individuare l'effettiva possibilità di installazione all'intero di un impianto di produzione. Per rispettare la velocità della linea è necessario che l'intero matching sia eseguito in meno di 250ms; per intero matching si intende l'estrazione della tavola dall'immagine della Stazione 2 e il confronto con una coda di cinquecento descrittori già calcolati della Stazione 1.

Si può definire il tempo dell'intero matching come:

$$T_{tot} = T_{estr} + T_{desc} + 500 * T_{match} \quad (6.3)$$

Dove  $T_{estr}$  è il tempo di estrazione della tavola,  $T_{desc}$  è il tempo necessario a calcolare il descrittore,  $T_{match}$  è il tempo necessario ad effettuare il matching tra due descrittori.

Si può osservare che in  $T_{estr}$  si eseguiranno tutte le operazioni di elaborazione dell'immagine che sono le più pesanti tra quelle di cui l'algoritmo è composto, in  $T_{desc}$  si eseguono tutte le operazioni matematiche necessarie alla descrizione della tavola tramite il descrittore, in  $T_{match}$  si eseguono operazioni allo scopo di stimare la distanza tra due descrittori. Si può quindi prevedere che il peso computazionale di  $T_{estr}$  sarà elevato in proporzione agli altri tempi. Si vuole inoltre che  $T_{match}$  sia ridotto poichè è pesato con fattore moltiplicativo cinquecento.

I test sono stati effettuati su un calcolatore con scheda madre SUPERMICRO X9DRG-QF equipaggiato con due processori Intel Xeon E5-2620 @ 2.00 GHz, 32 GB di RAM e Windows 7 64 bit. I tempi computazionali riportati in Tabella 6.4 sono ottenuti con codice sequenziale.

I metodi implementati rispettano ampiamente il limite temporale imposto. In particolare, visti i risultati ottenuti si conferma la possibilità di utilizzare nel sistema definitivo i momenti di Gauss-Hermite.

## 6.5 Conclusioni sui risultati

Dai risultati riportati si conclude che i descrittori calcolati utilizzando i momenti Gauss-Hermite sono i migliori per gli scopi prefissati del sistema. In particolare presentano le seguenti caratteristiche preferenziali rispetto agli altri descrittori:

Estrazione Stazione 1 [ms]	Estrazione Stazione 2 [ms]
64.476	57.372

Metodo	Calcolo descrittore [ms]	Matching 1vs1 [ms]	Intero matching [ms]
Gaussian-Hermite	1.227	0.0564	86.799
VDi	0.191	0.0461	80.613
Haar Wavelet	24.423	0.0026	83.095

Tabella 6.4: In alto: tempi computazionali per l'estrazione delle tavole dalle immagini acquisite; media su duecento esecuzioni. In basso: tempi computazionali per il calcolo dei descrittori e per il matching; media su quattrocento esecuzioni per il calcolo del descrittore, media su quarantamila esecuzioni per il matching 1vs1.

- massime percentuali di riconoscimento incondizionato (99.7899% *One vs All*, 99.9606% *One vs 500*)
- distribuzione dei falsi centrata in un valore di soglia distante dall'intervallo preferenziale per la distribuzione dei veri riconoscimenti
- coefficienti FAR e FRR che permettono molteplici configurazioni di riconoscimento apprezzabili, in base al sistema da realizzare
- possibilità di utilizzare tutta la tavola estratta per calcolare il descrittore; in questo modo tutte le informazioni visibili saranno contenute nel descrittore

Nel sistema finale quindi i descrittori utilizzati sono quelli calcolati con i momenti di Gauss-Hermite.

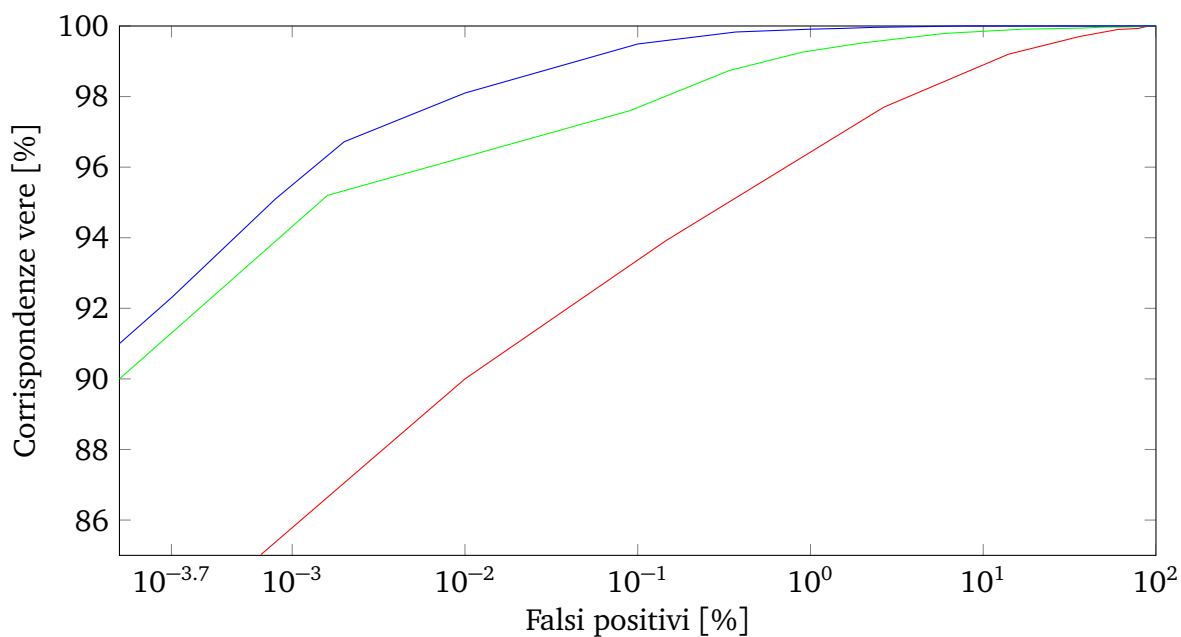


Figura 6.4: Curva ROC dei tre metodi testati. In rosso Haar wavelet, in verde VDi, in blu Gaussian-Hermite moments.

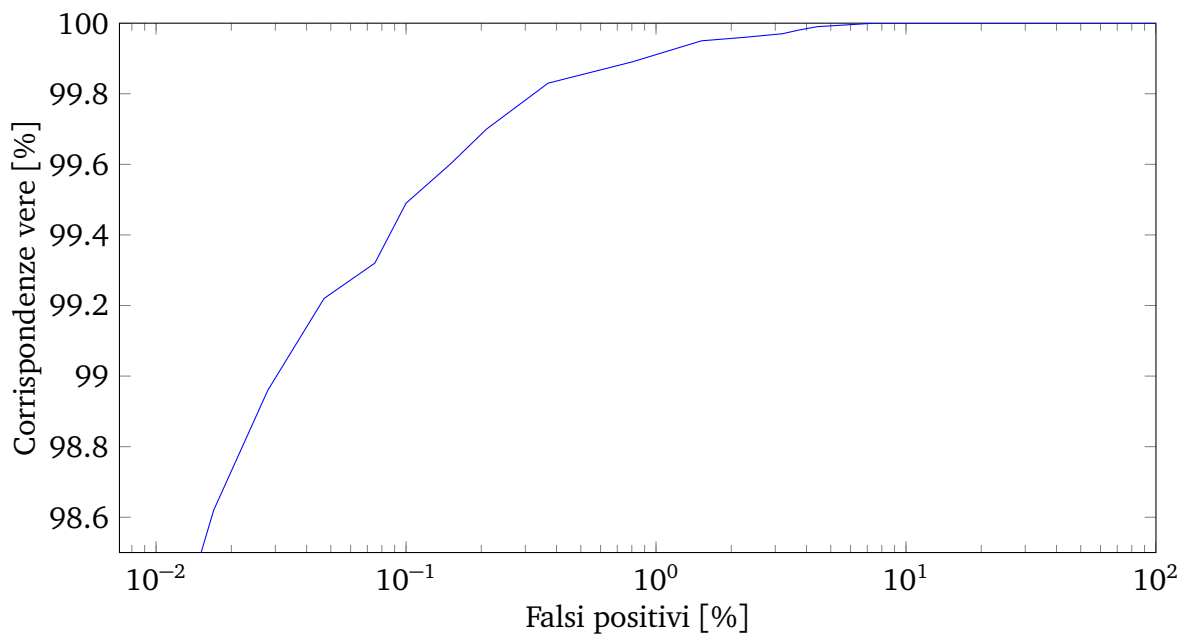


Figura 6.5: Particolare della curva ROC ottenuta con i descrittori di tipo Gaussian-Hermite moments.



# Capitolo 7

## Installazioni

### 7.1 iNKLESS iDSCAN

A scopo di installazione, l'algoritmo denominato iNKLESS iDSCAN è stato completamente re-implementato utilizzando la libreria proprietaria della ditta MICROTEC. L'estrazione delle tavole e i descrittori sono stati ottenuti con gli algoritmi presentati in precedenza. In particolare i descrittori sono calcolati utilizzando i momenti di Gauss-Hermite.

L'algoritmo è stato reso più preciso e ripetendo il test sul database di 15191 tavole di pino radiata sono stati raggiunti i risultati di riconoscimento incondizionato *One vs All* 99.8556% e *One vs 500* 99.9869%.

Il sistema è stato inoltre modificato per considerare la possibilità di analizzare gruppi di tavole di diversa larghezza durante la giornata di produzione senza dover modificare il parametro di ridimensionamento.

### 7.2 Berneck (Brasile)

Dal 30 Giugno 2014 il sistema è in funzione presso l'impianto *Berneck S. A. Painis e Serrados* a Curitiba, Brasile. Le Figure 7.1, 7.2 mostrano il setup funzionante con la linea di produzione attiva.

### 7.3 Hasslacher (Austria)

Al fine di acquisire un database riguardante tavole di abete è stata eseguita un'installazione temporanea nell'impianto *HASSLACHER NORICA TIMBER* a Sachsenburg, Austria. Durante la raccolta delle immagini il sistema è stato



Figura 7.1: Sistema iNKLESS iDSCAN in funzione. Berneck Stazione 1



Figura 7.2: Sistema iNKLESS iDSCAN in funzione. Berneck Stazione 2

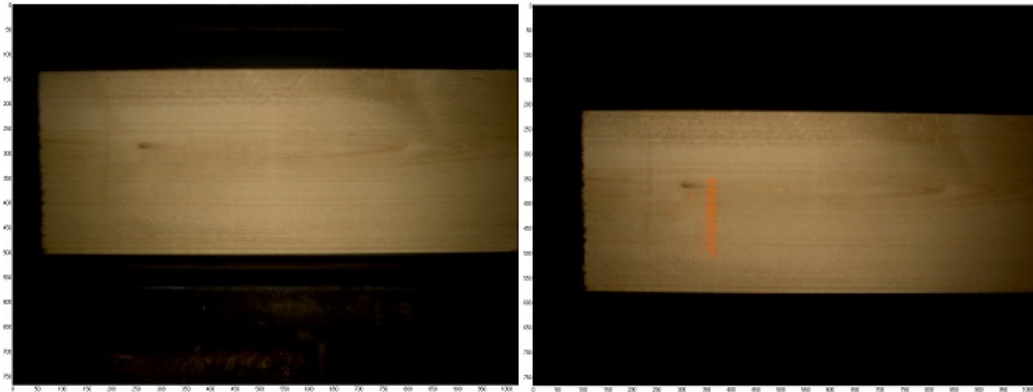


Figura 7.3: Esempio di immagini nel setup Hasslacher. A sinistra Stazione 1, a destra Stazione 2.

testato in tempo reale, con risultati soddisfacenti ma non stimabili precisamente a causa della particolare configurazione della linea.

Dai test effettuati sulle immagini raccolte (5295 coppie di tavole) sono state ottenute le seguenti percentuali di riconoscimento incondizionato: *One vs All* 99.5090% e *One vs 500* 99.9433%.

Le percentuali sono inferiori rispetto ai test effettuati su tavole di pino radiata, perchè come osservato in precedenza il legno di abete presenta elementi distintivi meno evidenti sulla superficie delle tavole. Il test è comunque molto soddisfacente sia considerando il tipo di legno sia considerate le condizioni notevolmente diverse rispetto ai test precedenti: luminosità diversa, dimensione delle tavole maggiore, installazione con fotocamera a distanza e configurazione differenti.

Un ulteriore elemento di disturbo per l'algoritmo è la presenza di marcature eseguite dall'operatore al fine di contrassegnare particolari caratteristiche presenti sulla tavola. Un esempio di tavole corrispondenti è mostrato in Figura 7.3.





## Capitolo 8

### Conclusioni

In questo lavoro di tesi è stato realizzato un sistema di reidentificazione di tavole in legno con descrittori basati sui momenti di Gauss-Hermite e utilizzabile all'interno di una linea di produzione con acquisizione immagini in tempo reale. Il sistema definitivo ha ottenuto percentuali di riconoscimento incondizionato *One vs All* 99.8556% e *One vs 500* 99.9869% in un database di 15191 coppie di immagini di tavole in pino radiata e percentuali di riconoscimento incondizionato *One vs All* 99.5090% e *One vs 500* 99.9433% in un database di 5295 coppie di immagini di tavole in abete. I risultati ottenuti sono molto soddisfacenti, considerando inoltre che: le immagini presentano tutti gli elementi di disturbo verso i quali è richiesto che il sistema si invariante, nel database è presente una sola corrispondenza esatta a differenza del riconoscimento dell'iride che viene eseguito su database contenenti un gruppo di immagini relative alla stessa persona e il match viene ritenuto esatto se viene scelta una qualsiasi di tali immagini.

Sono stati presentati diversi metodi per segmentare oggetti in immagini generiche ed è stata analizzata nel dettaglio la sequenza di operazioni realizzata per estrarre e normalizzare le tavole nel sistema implementato; sono stati testati, analizzati e confrontati tre metodi per il calcolo dei descrittori: Gaussian-Hermite moments, Haar wavelet e VDi; sono stati proposti i metodi per il confronto dei relativi descrittori, analizzando nel dettaglio le distribuzioni delle distanze di matching e le possibili configurazioni applicabili all'algoritmo di reidentificazione.

Il sistema realizzato è stato installato e testato in due distinti impianti ed è attualmente in funzione in uno di questi. Si può quindi affermare che l'algoritmo realizzato è in grado di soddisfare gli obiettivi prefissati.

## 8.1 Problemi attuali

I problemi attualmente presenti nelle tavole non correttamente riconosciute sono principalmente di origine meccanica. Se la tavola è particolarmente storta o non è ben posizionata nel nastro può inclinarsi e questo causa una segmentazione errata per la presenza della facciata laterale all'interno dell'immagine acquisita (tali tavole sono state escluse dai test presentati precedentemente). Nel caso di malfunzionamenti nel nastro o nel posizionamento delle tavole possono essere acquisite immagini contenenti due tavole in parte sovrapposte o porzioni di tavola rotta. Nel caso di rotture evidenti nelle tavole durante il periodo di tempo tra una Stazione e l'altra il riconoscimento diventa difficoltoso a causa della mancanza di corrispondenza tra le informazioni acquisite.

Un ulteriore problema riscontrato riguarda i riflessi che si presentano sulle tavole piattate e in generale con superficie particolarmente liscia. La luce emessa dalle lampade led presenti nell'alloggiamento della camera viene riflessa, causando la perdita di informazione nell'area interessata. Se la regione con presenza di riflesso non è estesa, la perdita di informazione relativa a eventuali nodi e/o venature viene resa influente dalla pesatura in fase di matching, ma nel caso di riflessi particolarmente estesi o che causano la perdita di particolari fortemente caratterizzanti per la tavola l'algoritmo estrae un descrittore che non è totalmente rappresentativo.

## 8.2 Sviluppi futuri

Nonostante il sistema presenti ottime percentuali di riconoscimento, c'è ancora margine di miglioramento. In particolare un aspetto che può migliorare le prestazioni del sistema è la minimizzazione dell'area di sovrapposizione tra le distribuzioni delle distanze di corrispondenza esatta e delle distanze di non-matching; a questo scopo potrebbe rivelarsi necessaria la modifica dell'intero metodo di computazione della distanza di matching.

L'acquisizione in ogni Stazione di due immagini, una per ogni faccia, permetterebbe di poter aumentare la quantità di informazione disponibile in ogni descrittore oltre a risolvere i riconoscimenti falliti a causa del capovolgimento della tavola. Attuare questa modifica comporta notevoli problemi dal punto di vista della progettazione dell'hardware poichè le tavole scorrono su un nastro non trasparente. Una possibile alternativa è l'inserimento della seconda telecamera sono nelle Stazioni trasversali dove non è presente il nastro che occlude il lato inferiore della tavola; si risolve così il problema del capovol-

gimento raddoppiando il numero di confronti effettuati, senza modificare la struttura meccanica dell'intera linea.

Nel caso di legni con informazioni visuali poco evidenti o nel caso più generico di tavole senza presenza di nodi e venature sono state proposte a livello teorico due soluzioni: allargare l'area inquadrata per aumentare la probabilità che siano presenti dei nodi nella regione di acquisizione, inserire una seconda fotocamera che acquisisca la porzione di tavola precedente per ottenere lo stesso risultato della proposta precedente. La prima proposta ha l'effetto indesiderato di diminuire la definizione dei particolari acquisiti a causa dell'allontanamento della camera e la possibile produzione di descrittori che siano troppo poco rappresentativi. La seconda proposta comporta l'inserimento e la sincronizzazione di una seconda telecamera, l'aumento dei tempi computazionali sia per la fase di estrazione sia per la fase di matching.

Per le esigenze attuali i tempi computazionali richiesti dall'algoritmo sono ampiamente accettabili. Nell'ottica di possibili applicazioni che richiedano tempi computazionali più brevi può essere interessante cercare di ottimizzare ulteriormente l'algoritmo convertendolo per l'esecuzione su Gpu o sfruttando la possibilità di parallelizzare su Cpu porzioni di codice adatte.



# Capitolo 9

## Appendice

### 9.1 Test di robustezza

Allo scopo di stimare la robustezza dei metodi analizzati per ottenere i descrittori sono stati realizzati quattro test, applicati a duemila coppie di tavole, in cui le regioni estratte vengono alterate nelle seguenti modalità:

- aggiunta di un errore di traslazione orizzontale
- aggiunta di un errore di traslazione verticale
- aggiunta di un errore di rotazione
- aggiunta di una combinazione degli errori sopra elencati

Gli errori inseriti dopo l'estrazione delle tavole hanno intensità scelta casualmente all'interno di un intervallo definito in precedenza. Ogni tipologia di errore è stata testata aumentando gradualmente il limite superiore di tale intervallo, fino ad arrivare ad un punto di evidente degrado delle prestazioni da parte di almeno uno dei metodi analizzati. I risultati ottenuti sono consultabili in Tabella9.1.

Si nota che il metodo generalmente più robusto è quello che utilizza i momenti di Gauss-Hermite. L'elemento di errore che più influisce sui risultati di riconoscimento è la rotazione, seguita dalla traslazione verticale che è meno evidente utilizzando i momenti di Gauss-Hermite grazie all'acquisito effetto di smoothing. L'errore dovuto alla traslazione orizzontale viene reso meno influente nei metodi VDi e Gauss-Hermite con la computazione dei segnali verticali mediati su un gruppo di colonne di ampiezza predefinita; il metodo Haar wavelet risente invece più pesantemente tale variazione.

<b>Traslazione orizz. [pix]</b>	<b>Traslazione vert. [pix]</b>	<b>Rotazione [°]</b>	<b>Risultati HW [%]</b>	<b>Risultati VDi [%]</b>	<b>Risultati GH [%]</b>
-	-	-	96.65	99.55	100
0 - 5	-	-	94.3	99.35	99.95
0 - 10	-	-	93.75	99.35	99.95
0 - 20	-	-	93.45	99.05	99.95
0 - 30	-	-	91.75	97.90	99.8
-	0 - 5	-	94.75	99.40	99.95
-	0 - 10	-	92.05	98.70	99.90
-	0 - 20	-	73.10	89.15	98.50
-	0 - 30	-	54.90	75.45	94.10
-	-	0 - 1	91.75	96.45	99.95
-	-	0 - 2.5	60.85	74.55	95.10
-	-	0 - 5	48.65	34.45	72.35
0 - 5	0 - 5	0 - 1	90.25	96.65	99.95
0 - 10	0 - 10	0 - 1.5	68.65	82.70	97.55
0 - 20	0 - 20	0 - 2	49.95	72.10	94.25

Tabella 9.1: Risultati dei test di robustezza eseguiti su un campione di due-mila tavole. Gli acronimi sono: HW Haar Wavelet, VDi il metodo presentato precedentemente, GH Gaussian-Hermite moments.

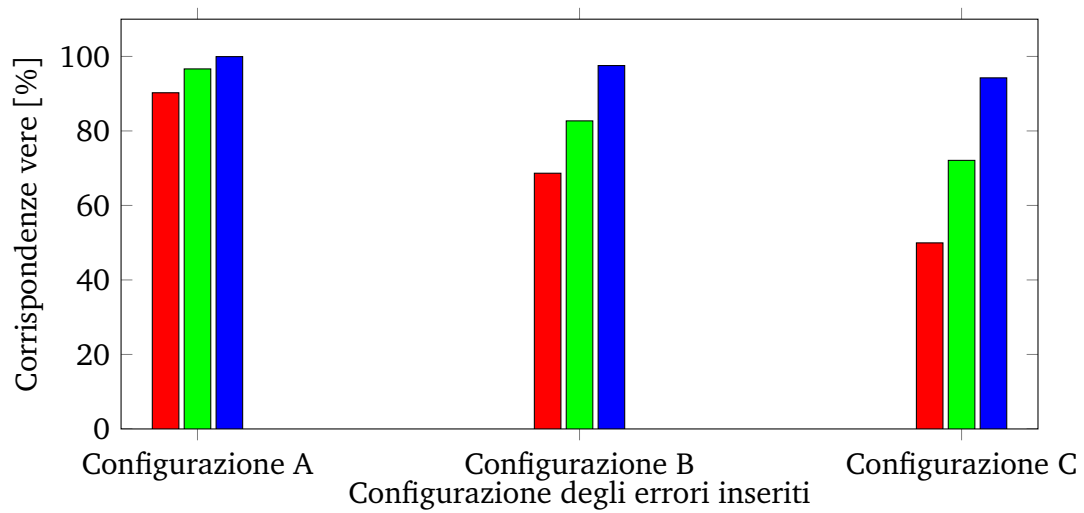


Figura 9.1: Grafico rappresentate le prestazioni di riconoscimento sul dataset di 2000 immagini dopo l'inserimento di errori combinati. *Configurazione A* indica traslazione orizzontale 0-5, traslazione verticale 0-5, rotazione 0-1; *Configurazione B* indica traslazione orizzontale 0-10, traslazione verticale 0-10, rotazione 0-1.5; *Configurazione C* indica traslazione orizzontale 0-20, traslazione verticale 0-20, rotazione 0-2. In rosso HW, in verde VDi, in blu GH.





# Bibliografia

- [1] Tobias Pahlberg, Erik Johansson, Olle Hagman and Matthew Thurlley, “Wood fingerprint recognition using knot neighborhood k-plet descriptors,” February 2014.
- [2] John Daugman, “How iris recognition works,” *IEEE Transactions on circuits and systems for video technology*, vol. 14 n71, January 2004.
- [3] Richard P Wildes, “Iris recognition: an emerging biometric technology,” *Proceedings of the IEEE*, vol. 85 n79, pp. 1348–1363, 1997.
- [4] W. W. Boles and B. Boashash, “A human identification technique using images of the iris and wavelet transform,” *IEEE Transactions on signal processing*, vol. 46 n4, pp. 1185–1188, April 1998.
- [5] Libor Masek, “Recognition of human iris patterns for biometric identification,” 2003.
- [6] Li Ma, Tieniu Tan., Yunhong Wang and Dexin Zhang, “Local intensity variation analysis for iris recognition,” *Pattern Recognition*, vol. 37, pp. 1287–1298, February 2004.
- [7] KyongWoo Nam, Kyong Lok Yoon, Jun Sung Bark and Woo S. Yang, “A feature extraction method for binary iris code construction,” *Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004)*, vol. 13 n76, pp. 284–287, 2004.
- [8] Shinyoung Lim, Kwanyong Lee, Okhwan Byeon and Taiyun Kim, “Efficient iris recognition through improvement of feature vector and classifier,” *ETRI Journal*, vol. 23 n72, pp. 61–70, June 2001.
- [9] Jafar M. H. Ali and Aboul Ella Hassanien, “An iris recognition system to enhance e-security environment based on wavelet theory,” *Advanced Modeling and Optimization*, vol. 5 n72, pp. 93–104, 2003.

- [10] Mahmoud Elgamal and Nasser Al-Biqami, “An efficient feature extraction method for iris recognition based on wavelet transformation,” *International Journal of Computer and Information Technology*, vol. 2, pp. 521–527, May 2013.
- [11] Ayra Panganiban, Noel Linsangan and Felicito Caluyo, “Wavelet-based feature extraction algorithm for an iris recognition system,” *Journal of Information Processing Systems*, vol. 7 n73, pp. 521–527, September 2011.
- [12] Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang, “Efficient iris recognition by characterizing key local variations,” *IEEE Transactions on image processing*, vol. 13 n76, pp. 739–750, June 2004.
- [13] David J. Field, “Relations between the statistics of natural images and the response properties of cortical cells,” *J. Opt. Soc. Am. A*, vol. 4 n712, pp. 2379–2394, August 1987.
- [14] Jun Shen, “Orthogonal gaussian-hermite moments for image characterization,” *SPIE*, vol. 3208, pp. 224–233, 1997.
- [15] Ce Liu, Jenny Yuen and Antonio Torralba, “Sift flow: Dense correspondence across scenes and its applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33 n5, pp. 978–994, 2011.
- [16] Matthew Brown and David G. Lowe, “Automatic panoramic image stitching using invariant features,” *International Journal of computer vision*, vol. 74, pp. 59–73, 2007.
- [17] J.-G. Wang; J. Li; W.-Y. Yau and E. Sung, “Boosting dense sift descriptors and shape contexts of face images for gender recognition,” *Proc. Computer Vision and Pattern Recognition Workshop*, pp. 1–8, 2010.
- [18] Manuele Bicego; Andrea Lagorio; Enrico Grosso and Massimo Tistarelli, “On the use of sift features for face authentication,” *Proc. Computer Vision and Pattern Recognition Workshop*, 2006.
- [19] David G. Lowe, “Object recognition from local scale-invariant features,” *Proc. of the International Conference on computer vision*, September 1999.
- [20] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60 n72, pp. 91–110, 2004.

- [21] Matthew Brown and David G. Lowe, "Invariant features from interest point groups," *British Machine Vision Conference*, pp. 656–665, September 2002.
- [22] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biological Cybernetics*, vol. 61, pp. 103–113, 1989.
- [23] Jun Shen and Youfu Wu, "Properties of orthogonal gaussian-hermite moments and their applications," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 588–599, 2005.
- [24] Pierre Moreels and Pietro Perona, "Evaluation of features detectors and descriptors based on 3d objects," *Computer Vision, ICCV Tenth IEEE International Conference*, vol. 1, pp. 800–807, October 2005.
- [25] Krystian Mikolajczyk and Cordelia Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, pp. 1615–1630, 2005.
- [26] Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool, "Surf: speeded up robust features," *Computer Vision and image understanding*, vol. 110, pp. 346–359, 2008.
- [27] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Graphics and image processing*, vol. 24 n76, pp. 381–395, 1981.
- [28] J. Kittler, J. Illingworth and J. Foglein, "Threshold selection based on a simple image statistic," *Computer Vision, graphics and image processing*, vol. 30 n76, pp. 125–147, December 1985.