

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
SEDE DI CESENA
SECONDA FACOLTÀ DI INGEGNERIA CON SEDE A CESENA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

TITOLO DELLA TESI

**SVILUPPO DI ALGORITMI PER L'AUTOMAZIONE DI
MISURE INDUSTRIALI**

Tesi in
Elaborazione delle immagini LM

Relatore:
Prof. ALESSANDRO BEVILACQUA
Co-relatore:
Ing. STEFANO ANDREINI

Presentata da:
MANUEL BARTOLINI

Sessione III
Anno Accademico 2010-2011

PAROLE CHIAVE

Elaborazione di immagini

Visione industriale

Sistema automatico

Metrologia

Imaging quantitativo

Ciò che ieri era fantascienza, stasera potrebbe essere già
storia.

Neil Armstrong

Indice

Introduzione	xi
1 Machine vision	1
1.1 Machine Vision	1
1.1.1 Vantaggi	3
1.1.2 Componenti	3
1.1.3 Difficoltà	4
1.1.4 Applicazioni	4
1.2 Metrologia	5
1.3 Ottica	6
1.3.1 Da Pinhole all'introduzione delle lenti	6
1.3.2 Proprietà e terminologia del modello a lente sottile convessa	8
1.3.3 Ottiche telecentriche	11
1.3.4 Lenti speciali	13
1.4 Sensori e acquisizione dell'immagine	14
1.4.1 Sensori sensibili alla luce	15
1.4.2 Sensori CCD e CMOS	16
1.4.3 Tipologie di utilizzo dei sensori	17
1.4.4 Campionamento e quantizzazione	17
1.4.5 Immagini digitali	19
1.5 Illuminazione	19
1.5.1 Sorgenti luminose	20
1.5.2 Tecniche di illuminazione	21
1.6 Motivazioni	23
2 Materiale	25
2.1 MTL	25
2.1.1 Meccanica	26
2.1.2 Illuminatore	27
2.1.3 Ottica e telecamera	27

2.1.4	Calibrazione	28
2.1.5	Software	31
2.2	Pezzi utilizzati	33
3	Misura industriale automatica con macchine di visione	37
3.1	Acquisizione dell'immagine	38
3.1.1	Dimensione e variabilità del pixel	39
3.1.2	Determinazione dei bordi	39
3.1.3	Riflettività	40
3.1.4	Artefatti dovuti a polvere, etc	42
3.2	Dimensione delle immagini e tempistiche	42
4	Tecniche di image processing	45
4.1	Morfologia	46
4.1.1	Dilatazione	47
4.1.2	Erosione	47
4.1.3	Opening e closing	48
4.1.4	Algoritmi morfologici	50
4.2	Segmentazione	51
4.2.1	Thresholding	51
4.2.2	Edge detection	54
4.3	Rappresentazione	60
4.3.1	Chain code	60
4.4	Interpretazione	62
4.4.1	Analisi sintattica	62
5	Algoritmo	65
5.1	Progettazione	66
5.2	Riduzione del rumore	66
5.3	Estrazione dei bordi	70
5.4	Rappresentazione	71
5.5	Interpretazione	72
6	Risultati sperimentali	77
6.1	Valutazione dei risultati	78
6.1.1	Diametri	79
6.1.2	Piani (Spallamenti)	81
6.1.3	Lati obliqui	82
6.1.4	Archi di circonferenza	82
	Conclusioni e sviluppi futuri	85

Bibliografia	86
Ringraziamenti	89

Introduzione

Oggi grazie al continuo progredire della tecnologia, in tutti i sistemi di produzione industriali si trova almeno un macchinario che permette di automatizzare determinate operazioni. Alcuni di questi macchinari hanno un sistema di visione industriale (*machine Vision*), che permette loro di osservare ed analizzare ciò che li circonda, dotato di algoritmi in grado di operare alcune scelte in maniera automatica.

D'altra parte, il continuo progresso tecnologico che caratterizza la realizzazione di sensori di visione, ottiche e, nell'insieme, di telecamere, consente una sempre più precisa e accurata acquisizione della scena inquadrata. Oggi, esigenze di mercato fanno sì che sia diventato necessario che macchinari dotati dei moderni sistemi di visione permettano di fare misure morfometriche e dimensionali non a contatto. Ma le difficoltà annesse alla progettazione ed alla realizzazione su larga scala di sistemi di visione industriali che facciano misure dimensioni non a contatto, con sensori 2D, fanno sì che in tutto il mondo il numero di aziende che producono questo tipo di macchinari sia estremamente esiguo.

A fronte di capacità di calcolo avanzate, questi macchinari necessitano dell'intervento di un operatore per selezionare quali parti dell'immagine acquisita siano d'interesse e, spesso, anche di indicare cosa misurare in esse.

Questa tesi è stata sviluppata in sinergia con una di queste aziende, la *Vici & Co.*, che produce alcuni macchinari per le misure automatiche di pezzi meccanici. Attualmente, nell'immagine del pezzo meccanico vengono manualmente indicate le forme su cui effettuare misure.

Lo scopo di questo lavoro è quello di studiare e prototipare un *algoritmo che fosse in grado di rilevare e interpretare forme geometriche note, analizzando l'immagine acquisita dalla scansione di un pezzo meccanico.*

Le difficoltà affrontate sono tipiche dei problemi del “mondo reale” e riguardano tutti i passaggi tipici dell'elaborazione di immagini, dalla “pulitura” dell'immagine acquisita, alla sua binarizzazione fino, ovviamente, alla

parte di analisi del contorno ed identificazione di forme caratteristiche. Nonostante le soluzioni proposte sfruttino tecniche di elaborazione di immagini consolidate, il prototipo ottenuto ha consentito di raggiungere pienamente gli obiettivi preposti.

Nel primo capitolo verranno introdotti alcuni concetti propri della *Machine Vision*: che cos'è, alcune applicazioni, i vantaggi, i problemi e alcune delle tecniche utilizzate per costruire sistemi di visione industriale.

Nel secondo capitolo verranno presentati il sistema di visione utilizzato in questa tesi e alcuni dei pezzi meccanici utilizzati come *benchmark*. Successivamente, nel terzo capitolo saranno, discussi i problemi incontrati nell'analisi del problema.

Nel quarto capitolo verranno introdotte alcune delle tecniche di *image processing* utilizzate nello sviluppo dell'algoritmo e nel capitolo cinque saranno presentate le scelte effettuate, spiegando in quale modo le tecniche del precedente capitolo sono state utilizzate.

I risultati sperimentali saranno presentati e discussi nel capitolo sei che sarà seguito dalle conclusioni tratte e dai possibili sviluppi futuri.

Buona lettura.

Capitolo 1

Machine vision

La vista è forse uno dei sensi più importanti degli esseri viventi, infatti permette loro di cacciare per procurarsi cibo, trovarsi un posto al sicuro da pericoli e di cercarsi un compagno o compagna per procrearsi. Animali e umani fanno ampiamente uso della vista. Delle macchine che hanno la capacità di “osservare” ciò che li circonda, fornisce a quest’ultime un importante funzionalità.

Oggi si parla di *Machine Vision* ovvero, quelle macchine che grazie a dispositivi di visione offrono assistenza nell’area della produzione industriale, nel campo dell’agricoltura/allevamento, videosorveglianza e in ambito medico.

1.1 Machine Vision

Non è semplice fornire una completa e soddisfacente definizione di *Machine Vision*, perché racchiude tanti e disparati insiemi di tecnologie e applicazioni. Si può comunque sintetizzare il concetto nel seguente modo:

Il termine *Machine Vision* (MV) riguarda la progettazione di sistemi composti da parti meccaniche, ottiche, elettroniche e software per esaminare oggetti naturali o materiali, processi industriali con l’obiettivo di determinare difetti e migliorarne quindi la qualità, operando efficientemente e garantendo la sicurezza dei prodotti e del processo[7].

La progettazione di MV non è semplice a causa della grande varietà di oggetti che si possono esaminare. Per questo motivo la MV si è sviluppata maggiormente nel settore riguardante l’ispezione e il controllo di prodotti. In questi casi l’utente conosce perfettamente i vincoli che un determinato bene

deve avere, e si avvale di macchine di visione per identificare quei prodotti che non rispettano le date tolleranze.

Si faccia attenzione a non confondere il termine *Machine Visione* e *Computer Vision*, perché la prima è una branca dell'ingegneria, non una scienza. Mentre la *Computer Visione* è la scienza che studia metodi di acquisizione, di processo e di analisi di immagini tramite computer.

Si può quindi affermare, che la Machine Vision si avvale di tecniche di Computer Vision nel momento che, acquisita un'immagine, ne ricava il contenuto informativo.



Figura 1.1: Sia umani che sistemi di visione necessitano degli stessi componenti: gestione meccanica degli oggetti che devono essere esaminati, lente ottica, sensori di immagine (occhi o telecamera), sistema di processo e analisi dell'immagine e infine una base di conoscenza per decidere se accettare o rifiutare il bene in esame[7].

1.1.1 Vantaggi

L'inserimento delle prime macchine di visione all'interno dell'industria risale circa agli anni '70 in concomitanza alla terza rivoluzione industriale. L'approdo di questi macchinari all'interno della catena di produzione ha portato notevoli vantaggi, facendo così aumentare la domanda degli stessi e consentendo alla MV di evolversi. Alcuni dei vantaggi sono:

- **Riduzione dei costi:** le macchine di visione vanno a sostituire la figura degli operatori adibiti al controllo, i quali essendo soggetti a distrazione, stanchezza etc, non garantiscono un controllo affidabile e totale. La MV invece riesce ad effettuare un controllo perpetuo 24 ore al giorno per 365 giorni, assicurando così all'azienda un controllo completo e permettendo un'ottimizzazione delle risorse.
- **Aumento competitività:** l'introduzione di MV porta un livello di innovazione che va ad incrementare il valore aggiunto del prodotto.
- **Automazione del controllo qualità processo/prodotto:** permette una precisa analisi statistica del processo produttivo fornendo quindi il dato di ripetibilità dell'andamento della produzione e garantisce un controllo qualità continuo e costante per ogni prodotto.

1.1.2 Componenti

Un tipica macchina di visione è composta da diverse parti, le quali:

- *sensori di immagine*, sono composti da due elementi. Il primo è un dispositivo fisico che cattura l'energia irradiata dall'oggetto. Il secondo è chiamato *digitalizer* e ha il compito di convertire la grandezza fisica percepita dal sensore in una grandezza digitale;
- *hardware specializzato*, può essere pensato come *digitalizer* più potente. Questo componente ha il compito di effettuare operazioni primitive, aritmetiche e logiche sull'immagine andando così a ridurre il rumore generatosi nell'acquisizione dell'immagine;
- *computer*, può essere sia un semplice PC che un supercomputer;
- *software* dedicato al *image processing and analysis*, è il cuore della MV ovvero è il componente che estrae le informazioni dell'immagine;
- *dispositivo di archiviazione di massa*, che ha il compito di immagazzinare la grande mole di immagini catturate dai dispositivi di visione in attesa di elaborazione.

1.1.3 Difficoltà

L'elemento che sta alla base del processo di analisi delle MV è un'immagine. L'elaborazione di un'immagine porta con sé notevoli difficoltà e necessita di un buon grado di esperienza nel settore.

Alcuni dei problemi di questo settore possono essere:

- *perdita di informazioni*: la maggior parte delle macchine di visione utilizzano sensori che trasformano la scena reale osservata 3D in una immagine 2D. Venendo a meno una dimensione un quantitativo non indifferente di informazioni vengono perse, come angoli e linearità;
- *interpretazione*: un uomo che analizza un'immagine, grazie alla propria capacità di ragionamento e facendo uso di una *knowledge base* che continua a riempire dalla nascita, riesce ad interpretare senza eccessivo sforzo la scena rappresentata. Non si può dire la stessa cosa per una MV che può analizzare solo immagini per le quali è stata progettata;
- *rumore*: una MV come ogni dispositivo meccanico ed elettronico è soggetto ad errore, quindi è necessario che una MV abbia tutta una gestione di attenuazione del rumore;

Questi elencate sopra sono solo alcuni dei problemi che si affrontano nello sviluppo di una MV. Altri problemi verranno affrontati nei prossimi capitoli in relazione al macchinario utilizzato.

1.1.4 Applicazioni

Oggi le macchine di visione sono utilizzate in larga scala, perché grazie alla possibilità di “osservare” il mondo che le circonda offrono notevoli vantaggi a chi decide di farne uso.

Controllo di processo Uno dei primi utilizzi delle MV si è avuto in campo industriale, dalla necessità delle aziende di poter controllare e automatizzare il loro flusso di produzione. Per controllo e automatizzazione si intende l'utilizzo di guide robot, ovvero tutti quei sistemi il cui scopo è di guidare un robot industriale per prendere oggetti da posizione non note a priori, come potrebbero succedere su un nastro trasportatore, oppure robot che devono eseguire lavorazioni su pezzi la cui posizione nello spazio non è determinata a priori. Il principale vantaggio dell'ausilio di sistemi di visione a robot industriale è la capacità di correggere in tempo reale la loro posizione.

Per controllo di processo può anche essere considerato il controllo del traffico, delle infrazioni sulle autostrade, il controllo dei mezzi presenti sulle piste di un aeroporto. Recentemente sistemi di visione sono installati nei centri commerciali, su zone pedonali per fare sorveglianza oppure per determinare certi tipi di comportamenti.

Metrologia Le MV sono oggi utilizzate anche per compiere misure dimensionali non a contatto di ogni tipo di oggetti (pezzi meccanici, frutti ect). L'utilizzo di MV per fare misure consiste nella possibilità di ricavare dall'immagine derivante dalla scena acquisita dal dispositivo di visione, i bordi della scena il profilo dell'oggetto e da questo calcolare le grandezze sul sistema di riferimento immagine, per poi trasformarle in grandezze del mondo reale.

Questo tipo di utilizzo fornisce garanzie per quel che riguarda l'accuratezza e la ripetibilità delle misure.

Controllo qualità Le MV vengono utilizzate in campo industriale anche per effettuare il controllo qualità. Definite certe tolleranze che il processo di produzione deve mantenere, è possibile demandare il controllo a delle macchine di visione industriale che permettono, rispetto ad un controllo umano, una ispezione totale, una ripetibilità del controllo sempre uguale, garantendo quindi un rilevamento sistematico di difetti.

Questa tesi è stata sviluppata lavorando su una macchina, che sarà presentata nel capitolo 2, che compie misure automatiche su pezzi meccanici.

1.2 Metrologia

Per il VIM 3¹, la metrologia comprende tutti gli aspetti, sia quelli teorici che quelli pratici, della misurazione, qualunque sia l'incertezza di misura e il campo d'applicazione. Per misurazione si intende il processo volto a ottenere sperimentalmente uno o più valori che possono essere ragionevolmente attribuiti a una grandezza. A sua volta una grandezza è la proprietà di un fenomeno, corpo o sostanza, che può essere espressa quantitativamente mediante un numero e un riferimento[8].

Nel recente passato un pezzo veniva misurato alla fine della propria produzione (post-processo), oggi generalmente si effettuano misure durante la produzione tramite l'inserimento di sensori ad hoc a seconda della grandezza che si vogliono determinare.

¹International vocabulary of metrology.

Nel caso della MV la misurazione avviene in modo indiretto, in quanto i sensori di visione acquisiscono l'*immagine* dell'oggetto sulla quale poi, grazie alle tecniche offerte dalla Computer Vision di *image processing* e *analysis*, si vanno a ricavare le grandezze desiderate.

1.3 Ottica

Dai paragrafi precedenti, si intuisce che l'*immagine*, gioca un ruolo decisamente importante nella MV, ma cos'è un'immagine?

Al giorno d'oggi tutti hanno scattato almeno una fotografia, ma non tutti sanno come la scena fotografata si trasforma in un'immagine: stampata su una pellicola fino a una decina di anni fa e oggi immagazzinata su un hard disk come un insieme di valori detti pixel.

L'acquisizione e relativa formazione di un'immagine può essere riassunta con una semplice definizione:

un'immagine viene generata quando un *sensore* registra *radiazioni* che hanno interagito con un *oggetto fisico*.

1.3.1 Da Pinhole all'introduzione delle lenti

La generazione di un'immagine derivante da una scena reale ebbe i suoi albori già nella lontana Grecia con Aristotele ed Euclide, passando poi per la *Camera Obscura* di Leonardo da Vinci e arrivando così alla prima fotografia nel 1837, ad opera di *L. J. M. Daguerre*.

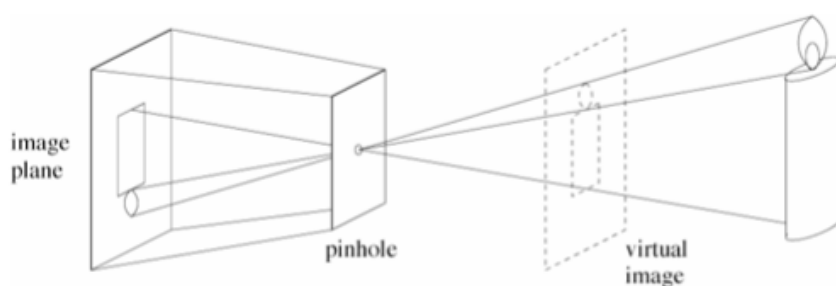


Figura 1.2: Pinhole camera.

Il modello che sta alla base della fotografia è la *Pinhole Camera*. La Pinhole camera è una *black box*, con un piccolo buco da una parte e un sensore dalla parte opposta.

La definizione matematica è la seguente: di tutti i raggi di luce del mondo, si campionano solo quelli che passano attraverso il piccolo foro della *black box* e vengono proiettati su un piano.

L'immagine di figura 1.2 può essere vista dal punto di vista geometrico andando così a definire alcune proprietà.

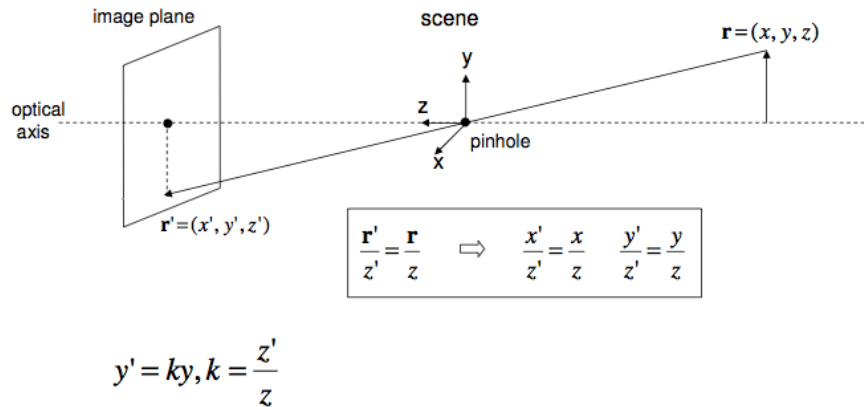


Figura 1.3: Proiezione prospettica Pinhole.

Dalla figura 1.3 si nota che la proiezione sul piano immagine è invertita rispetto alla sorgente dei raggi di luce che passano per il foro. Mentre dall'equazioni si evince che la dimensione della scena proiettata è inversamente proporzionale alla distanza della sorgente dei raggi di luce.

Il modello Pinhole presenta però dei problemi, quali:

- oscurità dell'immagine: il foro è oltrepassato solo da pochi raggi;
- distanza: la proiezione è inversamente proporzionale alla distanza ;
- diametro del foro: è necessario definire un buon compromesso tra apertura del foro e nitidezza dell'immagine. Infatti, se il foro è troppo grande passano troppi raggi e la proiezione risulterà rumorosa, se troppo piccolo il foro è attraversato solo da pochi raggi con il risultato di una proiezione sfocata.

Per i motivi sopra il foro del modello Pinhole è stato sostituito con una lente sottile convessa.

L'introduzione della lente nel modello di Pinhole, cambia la geometria di acquisizione della scena e grazie alla l'uso di una lente *sottile* convessa si possono ignorare gli effetti ottici dovuti a quest'ultima.

Le equazione che definiscono il nuovo modello sono:

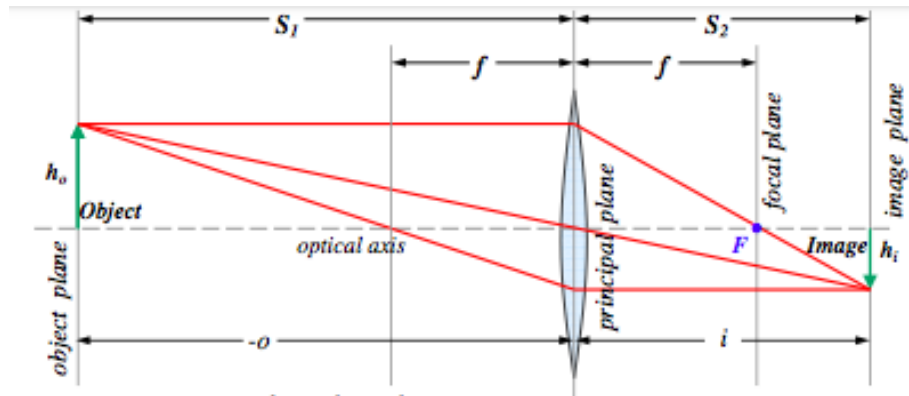


Figura 1.4: Modello a lente sottile.

- Formula Gaussiana:

$$\frac{1}{S_1} + \frac{1}{S_2} = \frac{1}{f}$$

- Formula Cartesiana:

$$\frac{1}{i} - \frac{1}{o} = \frac{1}{f}$$

- Proporzionalità:

$$M = \frac{f}{f - o} = -\frac{i}{o} = \frac{h_i}{h_o}$$

1.3.2 Proprietà e terminologia del modello a lente sottile convessa

Nel modello a lente sottile la proiezione della scena sul piano immagine è funzione di diversi parametri. Il più importante di tutti è sicuramente la *distanza focale* f . Questa è la distanza che c'è tra la lente e il punto F del modello, dove passano tutti quei raggi che sono paralleli all'*asse ottico*.

Al variare di f varia in modo inversamente proporzionale il *field of view*, fov . Gli estremi della fov non sono altro che quei punti della scena dai quali vengono riflessi i raggi luminosi che colpiscono la lente perpendicolarmente.

Un altro parametro che influenza la proiezione, è funzione del diametro della lente D e f . Si parla di *f-number*:

$$f/\# = \frac{f}{D}$$

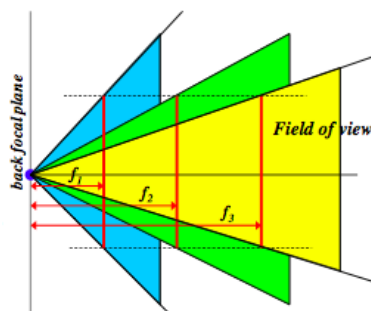


Figura 1.5: Field of View.

più questo valore è alto più raggi di luce raggiungono il piano immagine.

Lo *shutter speed*, un altro parametro, indica l'ammontare di tempo in cui il piano immagine è esposto alla luce. Anche questo parametro ha un'importanza determinante sulla proiezione, infatti l'eccessiva esposizione del piano immagine ai raggi comporta un'immagine confusa.

Un altro fattore riguarda il *focus* e *defocus*, o meglio la *messa a fuoco* della proiezione. Per spiegare questo concetto si parte da questa affermazione: *è a fuoco solo una scena posta ad una distanza tale che rispetta la condizione:*

$$\frac{1}{S_1} + \frac{1}{S_2} = \frac{1}{f}$$

Succede però, che un sensore che sta sul piano immagine non abbia una dimensione puntiforme, quindi sarà colpito da più raggi provenienti sia da piani in *focus* che da piani in *defocus*. Questo fattore è dovuto alla risoluzione del sensore, proprio come quella dell'occhio umano che riesce a mettere a fuoco, dentro certi limiti, più oggetti posti su piani diversi.

Definiamo così il *blur circle* l'intervallo nel quale i raggi di luce caduti al suo interno saranno approssimati ad un punto della scena. Nella figura 1.6 è mostrato anche un altro fattore il *depth of field*, il quale è funzione del blur circle e del diametro della lente. Al diminuire di D il DOF aumenta.

Nel modello di Pinhole, l'ingrandimento dell'oggetto è inversamente proporzionale alla sua distanza dalla dispositivo di visione. Un problema dei sistemi a lente sottile è la *distorsione geometrica* che equivale ad un allungamento dell'immagine non uniforme, o più intuitivamente come una diverso fattore di scala tra il centro dell'immagine e i bordi. Per risolvere

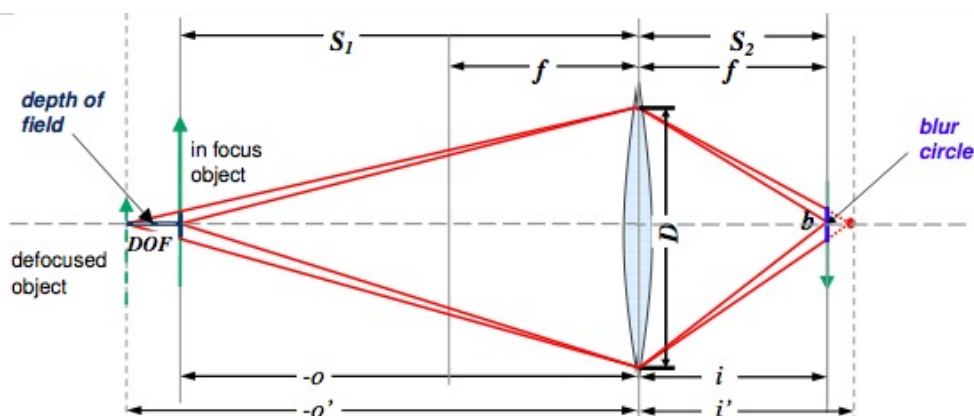


Figura 1.6: Focus e defocus.

la distorsione geometrica, è necessario sottoporre il sistema ad una fase di calibrazione.

Fatto un accenno dell'ottica del dispositivo di visione, si può affermare che il modello ideale presentato è un sistema lineare, o meglio *Linear Shift Invariant System*. I raggi di luce in ingresso saranno perfettamente proiettati sul piano di proiezione. Il modello ideale quindi potrebbe essere approssimato alla funzione delta di Dirac. Considerando però l'idealità solo un concetto teorico, un sistema ottico *non* è mai ideale, quindi si afferma che, dato in ingresso al sistema ottico un punto, si avrà in uscita una *Point Spread Function* del valore in ingresso.

La PSF di un dispositivo ottico è la proiezione di un singolo punto di un oggetto. L'ammontare del rumore (*blurring*) dell'immagine è dato dalla qualità del dispositivo.

Dopo queste digressioni di carattere tecnico si definisce la funzione di formazione dell'immagine nel seguente modo:

$$g(x, y) = f(x, y) * h(x, y) + n(x, y)$$

dove:

- $g(x, y)$ è l'immagine risultate;
- $f(x, y)$ è l'immagine ideale della scena;
- $h(x, y)$ è la PSF;
- $n(x, y)$ è il rumore dovuto all'acquisizione.

1.3.3 Ottiche telecentriche

Le MV sono utilizzate al giorno d'oggi per svolgere operazioni che debbano garantire una certa accuratezza e una ripetibilità. Questi obiettivi non sono raggiungibili con le lenti sottili presentate sopra, infatti sono soggette a diversi problemi:

- ingrandimento dell'oggetto inversamente proporzionale alla distanza;
- distorsione geometrica;
- errori di prospettiva;
- risoluzione dell'immagine si abbassa lungo il field of view;
- i bordi degli oggetti osservati hanno un livello di incertezza causato dal variare della geometria dei raggi che li colpiscono

Per ovviare a questi problemi si ricorre all'utilizzo di lenti *telecentriche*. Le lenti telecentriche rispetto alle lenti sottili permettono solo ai raggi paralleli all'asse ottico di raggiungere il piano immagine. L'immagine quindi sarà il risultato dei soli raggi che passano per il punto F di figura 1.4. L'utilizzo di lenti telecentriche porta diversi vantaggi rispetto alla lente convessa classica:

No ingrandimento La particolare proprietà delle lenti telecentriche fa sì che la grandezza proiettata sul piano immagine non sia inversamente proporzionale alla distanza del piano immagine dalla lente. Infatti proiettando solo i raggi paralleli all'asse ottico, questi rimarranno tali anche allontanando o avvicinando il piano immagine.

Riduzione della distorsione La distorsione è uno dei peggiori problemi che limita l'accuratezza della misura. Tutte le lenti sono soggette a distorsione e spesso il cambiamento di un pixel dell'immagine reale rispetto a quella attesa può comportare dei problemi.

La distorsione è definita come la differenza percentuale tra la distanza del centro dell'immagine rispetto a un punto nell'immagine reale e la distanza del centro dell'immagine sempre con lo stesso punto che potremmo calcolare in assenza di distorsione.

Ci sono due tipi di distorsione, una che ha segno positivo è chiamata *pincushion* e una con segno negativo chiamata *barrel*.

Le lenti comuni hanno una percentuale di distorsione ottica che può arrivare fino a qualche decina, le lenti telecentriche offrono una percentuale di

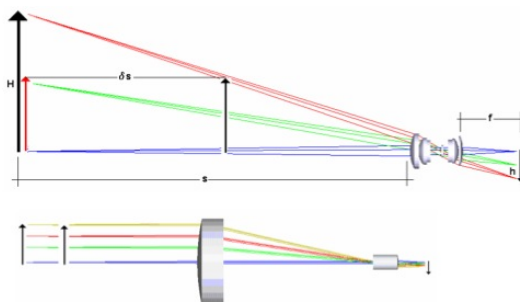


Figura 1.7: Nell'immagine sopra è rappresentata la geometria di una classica lente con le differenze di proiezione. Sotto è rappresentato il sistema telecentrico che mostra che al variare della distanza la dimensione rimane invariata.

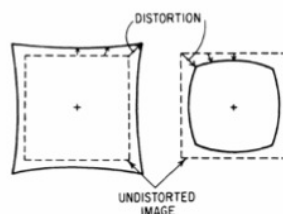


Figura 1.8: A sinistra la distorsione pincushion, a destra la barrel.

distorsione sotto lo 0.1%. Questo significa che il massimo errore causato dalla distorsione potrebbe essere inferiore di un pixel nei dispositivi di visione ad alta risoluzione.

Riduzione dell'errore prospettico Quando vengono acquisiti oggetti con una cavità, l'immagine risultante presenterà il foro in tutta la sua interezza come mostrato in figura 1.10.a, mentre una lente telecentrica non è soggetta a questo fenomeno come dimostra la figura 1.10.b.

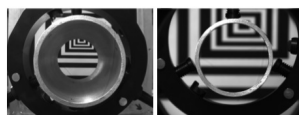


Figura 1.9: a) Acquisizione con lente classica. b) Acquisizione con lente telecentrica.

Quest'effetto è causato dal mancato filtraggio dei raggi che vengono proiettati sul piano immagine. Infatti se l'asse della cavità è parallelo all'asse ottico, sul piano immagine verranno proiettati solo i raggi ortogonali alla cavità.

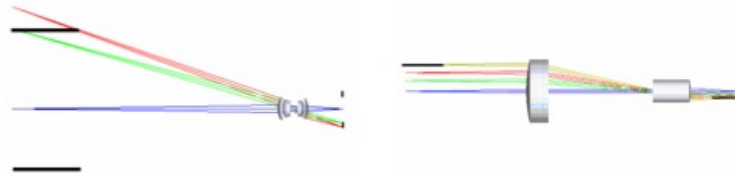


Figura 1.10: a) Geometria dei raggi con lente classica. b) Geometria dei raggi con lente telecentrica.

Determinazione certa dei bordi Quando si acquisisce l'immagine di un oggetto tramite una lente comune tutti i raggi riflessi dall'oggetto vengono proiettati sul piano immagine. Potrebbero raggiungere il piano immagine dei raggi riflessi da un punto nell'intorno del bordo dell'oggetto e allo stesso tempo il raggio riflesso dal bordo non raggiungere il piano immagine. Questo porta alla rappresentazione di un bordo non reale. Utilizzando lenti telecentriche come mostrato in figura 1.11.b verranno acquisiti in prossimità dei bordi quei raggi che sono tangenti all'oggetto.

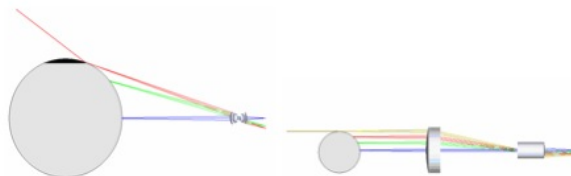


Figura 1.11: a) Geometria dei raggi con lente classica. b) Geometria dei raggi con lente telecentrica.

1.3.4 Lenti speciali

Lenti Zoom Questo particolare tipo di lenti hanno delle lunghezze focali che sono regolabili su alcuni intervalli. Vengono utilizzate nei prototipi dove la lunghezza focale non è ancora determinata. Le lenti Zoom sono larghe, poco robuste e molto costose. Soffrono di molta distorsione.

Lenti Macro Questo tipo di lenti sono progettate per lavorare con un fattore di ingrandimento vicino a 1. Le lenti Macro forniscono una miglior qualità dell'immagine rispetto alle lenti classiche.

Lenti Close-up Le lenti Close-up sono formate da una lente classica, preceduta da una seconda lente, l'insieme delle due ottiche riduce la distanza dell'oggetto inquadrato rispetto alle lenti classiche. L'ingrandimento nominale è:

$$m = \frac{f_{lens}}{f_{attachment}}$$

in cui f_{lens} è la lunghezza focale della lente principale, mentre $f_{attachment}$ è la lunghezza focale della seconda lente. La distanza dell'oggetto è approssimativamente uguale alla lunghezza focale della seconda lente $f_{attachment}$.

Teleconvertitori I teleconvertitori sono sistemi in cui una seconda ottica è posizionata tra la lente principale e il piano immagine con l'obiettivo di aumentarne la lunghezza focale. La difficoltà nell'utilizzo di questo tipo di ottica è l'alta distorsione a cui è soggetto.

Reverse Mouting Questo sistema ha la lente montata in modo inverso, con l'obiettivo di ridurre l'ingrandimento. In questo caso la distanza dell'oggetto rispetto alla al sistema di visione deve essere corta, mentre la distanza tra la lente e il piano immagine deve essere lunga.

1.4 Sensori e acquisizione dell'immagine

Secondo la norma tecnica UNI4546² un sensore è un particolare trasduttore³, che si trova in diretta interazione con il sistema misurato.

In ambito *metrologico*, il sensore è quel componente che effettua fisicamente la trasformazione della grandezza percepita in ingresso in un segnale di altra natura, generalmente elettrico.

I sensori possono essere utilizzati in modo *diretto* ed in modo *indiretto*. Per modo diretto si intende quando l'output di un sensore viene letto direttamente senza necessità di processamento (es. termometri a mercurio).

² Misure e misurazioni, Termini e definizioni fondamentali (link)

³Il trasduttore è un dispositivo che ha il compito di trasmettere dell'energia da un punto ad un altro punto. I trasduttori prendono in ingresso un segnale e riportano in uscita un altro segnale funzione di quello in ingresso. $s_{output} = f(s_{input})$

Mentre l'utilizzo indiretto, impiegato al giorno d'oggi per la maggiore, avviene quando l'output di un sensore è dato in ingresso ad un'altra macchina che ha il compito di processare ed interpretare la grandezza.

Nella Machine Vision i sensori di visione vengono utilizzati in maniera indiretta, in quanto catturata la scena osservata, questa viene data in input a un computer, più precisamente al software di *image processing e analysis*.

I sensori di visione si differenziano dal modo in cui acquisiscono la scena:

- *sensibili alla luce*: sono i sensori più comuni, presenti in fotocamere e telecamere e catturano la scena inquadrata grazie alle variazioni luminose prodotte da essa;
- *tomografici*: sono sensori che sfruttano i raggi X⁴ per ricostruire l'immagine. Questo tipo di sensori permettono la ricostruzione 3D della scena inquadrata, si pensi alle macchine per la TAC⁵ usate in ambito medico;
- *radar*: ricostruiscono la scena tramite l'utilizzo di onde elettromagnetiche;
- *ultrasuoni*: utilizzano onde meccaniche sonore.

Come sarà ampiamente presentata nel capitolo 2 in questa tesi è stata utilizzata una macchina di visione che si avvale di sensori sensibili alla luce per acquisire la scena.

1.4.1 Sensori sensibili alla luce

I sensori di immagine sensibili alla luce, d'ora in avanti in questa tesi sensori d'immagine, sono composti da un array di *pixels* che raccolgono fotoni⁶. Il numero di fotoni raccolti in ogni pixel è convertito in una carica elettrica da un fotodiodo⁷. Questa carica è trasformata in un voltaggio che poi sarà

⁴È nota come raggi X quella porzione dello spettro elettromagnetico con una lunghezza d'onda compresa approssimativamente tra 10 nanometri (nm) e 1/1000 di nanometro (1 picometro).

⁵Tomografia Assiale Computerizzata.

⁶Il fotone in fisica è secondo il modello standard una particella elementare.

⁷In optoelettronica un fotodiodo è un particolare tipo di diodo fotorilevatore che funziona come sensore ottico sfruttando l'effetto fotoelettrico, in grado cioè di riconoscere una determinata lunghezza d'onda dell'onda elettromagnetica incidente (assorbimento del fotone) e di trasformare questo evento in un segnale elettrico di corrente applicando ai suoi estremi un opportuno potenziale elettrico. Esso è dunque un trasduttore da un segnale ottico ad un segnale elettrico.

amplificato e convertito in un valore digitale tramite un convertitore analogico digitale.

Generalmente, ogni pixel del sensore ha un fotodiodo il quale poi corrisponderà con un pixel⁸ dell'immagine.

Si noti che il numero effettivo di pixel di un'immagine non corrisponde precisamente al numero pixel dei sensori che producono l'immagine. Ad esempio per una fotocamera standard da 5 megapixel che dovrebbe produrre immagini composte da 5 milioni di pixel, produce immagini da 2560×1920 che precisamente corrispondono a 4.9 milioni. I pixel mancanti sono utilizzati dalla fotocamera per compiere operazioni di formazione dell'immagine.

1.4.2 Sensori CCD e CMOS

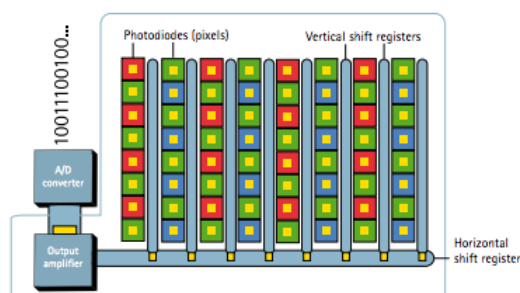


Figura 1.12: Sensore CCD.

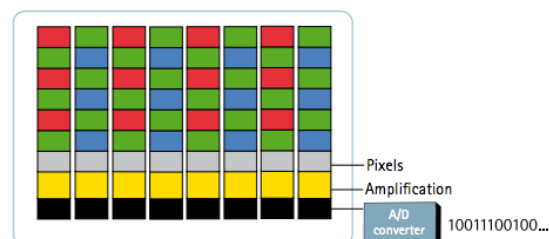


Figura 1.13: Sensore CMOS.

Nel sensore CCD (Coupled Charge Device), la luce (carica) che va a colpire un pixel del sensore viene trasformata tramite un nodo esterno al sensore. Le cariche che colpiscono il pixel vengono trasformate in una tensione, la quale poi viene bufferizzata e inviata all'esterno come un segnale analogico. Generalmente il segnale prima di entrare in ingresso ad un convertitore analogico/digitale viene amplificato.

I vantaggi di questo sensore rispetto ai sensore CMOS sono una miglior sensibilità alla luce e sono poco soggetti a rumore. Mentre gli svantaggi rispetto all'altra tipologia sono la necessità di avere componenti elettronici ai morsetti del CCD, la loro produzione è dispendiosa e, il loro consumo di energia elevato di circa 100 volte ai sensori CMOS.

⁸con il termine pixel (contrazione della locuzione inglese picture element) si indica ciascuno degli elementi puntiformi che compongono la rappresentazione di una immagine raster nella memoria di un computer.

I sensori CMOS presenti sul mercato da non tanti anni hanno il vantaggio di ridurre i tempi di *readout*, i consumi, e le dimensioni rispetto ai sensori CCD. Questo perché la logica di trasformazione della carica analogica in ingresso in grandezze digitali è tutta interna al chip. Per contro però questo tipologia di sensori essendo completamente integrati soffrono maggiormente di rumore strutturato.

1.4.3 Tipologie di utilizzo dei sensori

Sensore singolo Per acquisire un'immagine 2D usando un singolo sensore è necessario che questo abbia la possibilità di muoversi sia sull'asse delle x e delle y sull'area occupata dalla scena da acquisire. Un unico sensore con due gradi di libertà viene utilizzato per fare scansione ad alta precisione, generalmente è usato per la *microdensitometria*⁹.

Sensore in linea In questo caso si parla di linea ottica, in cui N sensori sono disposti in linea. La linea scansiona la scena in una direzione, mentre il movimento in modo perpendicolare ad essa permette la scansione dell'intera scena. Il risultato dell'acquisizione è un'immagine formata dalle N linee scansionate dalla linea ottica. Questa tipologia di utilizzo viene generalmente utilizzate nell'aziende per acquisire immagini di componenti di grandi dimensioni, infatti facendo muovere la linea ottica è possibile scansionare zone di vasta dimensione con un unico dispositivo di visione.

Sensore matriciale questo tipo di sensore il più comune viene montato sulle macchine fotografiche, su telecamere etc. Sono tutti quei dispositivi di visione che permettono con un'unica acquisizione di ricavare un'immagine in due dimensioni. Le principali applicazioni che utilizzano questo tipo di sensori sono la videosorveglianza e monitoraggio del traffico.

1.4.4 Campionamento e quantizzazione

Una scena del mondo reale può essere vista come un'immagine continua, $f(x, y)$, la quale ha delle coordinate x, y e un'ampiezza. Quando tramite un dispositivo di visione una scena viene proiettata su un piano immagine composto da sensori, si parla di *digitalizzazione*. La conversione di un'immagine continua in una digitale è composta da due parti: un *campionamento*

⁹La microdensitometria è la misura della densità ottica di materiali, misurata tramite l'ammontare di luce che è riflessa o trasmessa.

che ha il compito di digitalizzare le coordinate x,y e una *quantizzazione* che digitalizza le ampiezze.

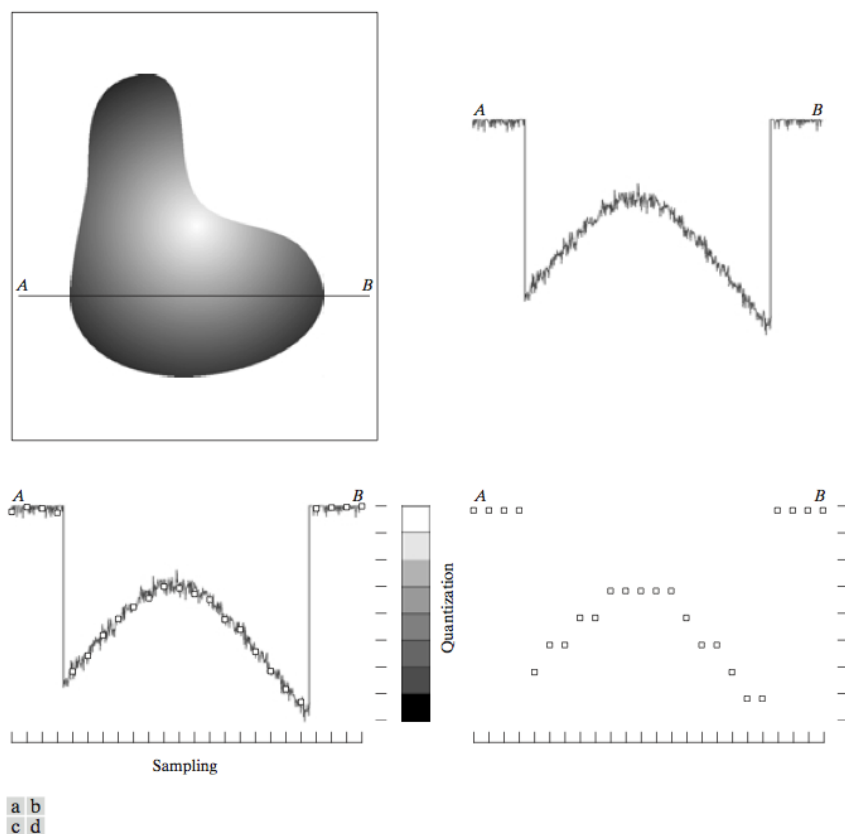


Figura 1.14: Digitalizzazione di un'immagine [6].(a) Immagine continua. (b) Scan line dell'immagine continua da A a B. (c) Campionamento e quantizzazione. (d) Scan line del punto b digitalizzato.

Se il campionamento delle coordinate è di facile intuizione, è necessario porre più attenzione sulla quantizzazione. La quantizzazione è il processo che ha il compito di andare a percepire la tonalità del punto dell'oggetto emettitore del raggio che ha colpito il sensore. Un esempio di ampiezza unidimensionale è mostrato in figura 1.14.b in cui è graficato il variare della tonalità del segmento AB dell'oggetto di figura 1.14.a. In figura 1.14.c è mostrata invece la quantizzazione della tonalità del segmento AB e il suo campionamento. Si noti l'oscillazione del segnale dovuta all'errore introdotto dalla strumentazione.

Il risultato della digitalizzazione del segmento AB è mostrato in figura 1.14.d, dove il segnale analogico in ingresso è stato campionato e quantizzato dando così in uscita il segnale digitale del segmento.

1.4.5 Immagini digitali

Dopo aver accennato come un dispositivo di visione acquisisce e digitalizza una scena si può dare la definizione di *immagine digitale* come un insieme di *pixel* che sono la digitalizzazione della PSF di un punto dell'oggetto della scena. Un pixel oltre ad avere coordinate spaziali x,y ha una profondità che rappresenta la tonalità del punto dell'oggetto della scena quantizzata.

Le principali categorie dell'immagini digitali sono:

- immagini binarie: sono le immagini in bianco e nero, un pixel è codificato con un solo bit, che ha valore 0 se la tonalità è nulla, mentre 1 se la tonalità è massima.
- immagini in scala di grigio: sono immagini che hanno i pixel codificati in n bit, ovvero possono assumere 2^n tonalità di grigio.
- immagini RGB: sono immagini a colori che hanno 3 canali per ogni pixel, ognuno codificato con n bit, che rappresentano rispettivamente la tonalità di rosso, verde e blu.

1.5 Illuminazione

Nell'esperienze di tutti i giorni, i nostri occhi per osservare ciò che ci circondano, utilizzano la luce ambiente. Nella MV l'utilizzo della luce ambiente non è una buona scelta perché è soggetta a tanti fattori, ad esempio una nuvola che oscura il sole, lo sfarfallio di una luce al neon possono portare a delle variazioni dell'illuminazione che alterano l'acquisizione della scena. Nelle applicazione di MV c'è quindi la necessità di poter controllare la luce per permettere un accurata acquisizione delle caratteristiche desiderate evitando di imbattersi in falsi positivi.

Gli oggetti riflettono i raggi in due modi. Si parla di riflessione speculare quando i raggi di luce che colpiscono l'oggetto vengono riflessi in una singola direzione, si pensi ad uno specchio. Mentre si definisce riflessione diffusa, quando i raggi che colpiscono l'oggetto vengono riflessi in tutte le direzioni.

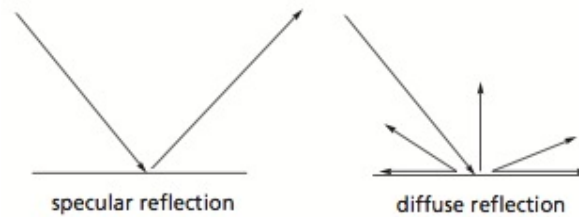


Figura 1.15: Tipologie di riflessione

In realtà, a seconda del materiale di cui è fatto l'oggetto, la riflessione dei raggi è soggetta sia a fenomeni diffusivi che speculari.

1.5.1 Sorgenti luminose

Nella MV vengono utilizzate diverse tipi di sorgenti luminose, la scelta di una rispetto ad un'altra è guidata dalla necessità di quali caratteristiche si vogliono estrapolare dalla scena.

- **LED:** è composto da un array di diodi ad emissione luminosa.
 - *Vantaggi:* l'array di diodi può assumere diverse configurazioni; se viene utilizzato come luce stroboscopica garantisce un'alta emissione ad alte velocità; emette luce di un solo colore che è utilizzato in molte applicazioni.
 - *Svantaggi:* non può essere usato dove si richiedono sorgenti luminose che emettano diversi colori; la dimensione è direttamente proporzionale all'area da illuminare.
- **Illuminatori a fibra ottica:** sono lampade incandescenti in custodia, dove la luce è trasportata dal fascio di fibre ottiche.
 - *Vantaggi:* il fascio di fibre può avere molte configurazioni; il generatore elettrico di carica può essere lontano dall'applicazione.
 - *Svantaggi:* le lampade ad incandescenza hanno un basso rendimento soprattutto per la luce di colore blue.
- **Lampade fluorescenti:** sono tubi con materiale fluorescente ad alta intensità.
 - *Vantaggi:* generano luce diffusiva; possono avere sia un ampio che stretto campo spettrale; sono efficienti e hanno lunga durata.

- *Svantaggi*: hanno una limitata possibilità di configurazione.
- **Strobo**: sono particolari lampade ad arco che utilizzano gas xeno¹⁰
 - *Vantaggi*: garantisce alti picchi di illuminazione.
 - *Svantaggi*: richiede delle particolari tempistiche di attivazione dei sensori che devono acquisire la luce. È dannoso per l'occhio umano, sono quindi necessarie protezioni per chi lavora nei pressi della macchina.

1.5.2 Tecniche di illuminazione

Per scegliere il tipo di illuminazione che più si appropria al tipo di applicazione che si sta progettando è bene tenere in considerazione quali tipi di oggetti e caratteristiche voglio essere acquisite. I due tipi di illuminazione sono la luce *puntuale* e *diffusiva*.

- **Luce puntuale**, sono facili da implementare perché gli illuminatori sono piccoli e possono essere posti a distanza dall'oggetto. Generalmente possono essere utilizzate tutte le tipologie di sorgenti luminose elencate sopra. L'illuminazione puntuale è utilizzata per acquisire oggetti con bordi affilati, determinati tipi di ombre e caratteristiche superficiali dell'oggetto. Utilizzando questa tecnica con materiali speculari si corre il rischio di saturare i sensori di acquisizione, con la conseguenza che parti dell'oggetto non direttamente colpite appaiono scure.
- **Luce diffusiva**, questo tipo di illuminazione ha una maggior complessità di installazione rispetto a quella precedente causata dalle dimensioni dell'emettitore e dalla necessità che quest'ultimo sia vicino alla scena. Se viene utilizzata con materiali speculari non è soggetta al problema della tecnica esposta sopra. Al contrario però è meno sensibile ai particolari superficiali. Questo tipo di illuminazione può anche causare del rumore in prossimità dei contorni dell'immagine.

Dopo aver scelto che tipo di raggi l'illuminatore deve emettere è necessario scegliere dove posizionare l'illuminatore.

- *Back lighting*, l'illuminatore si trova dietro l'oggetto. Questo tipo di illuminazione, che può essere sia puntuale che diffusivo, genera un alto

¹⁰Lo xeno o xenon è l'elemento chimico di numero atomico 54. Il suo simbolo è Xe. È un gas nobile incolore, inodore e molto pesante.

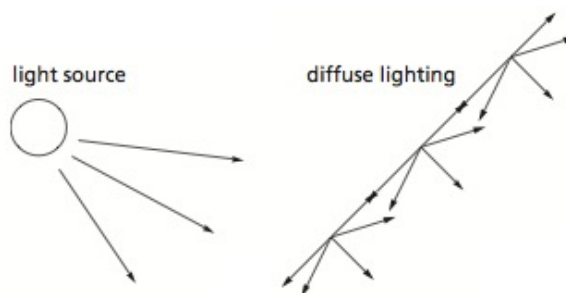


Figura 1.16: Tipi di illuminazione

contrasto tra uno scuro profilo dell'oggetto e un chiaro background. È generalmente utilizzata per rilevare l'assenza o la presenza di fori, orientamento o per fare misure sull'oggetto. Generalmente la sorgente luminosa è monocromatica, rossa, verde o blu.

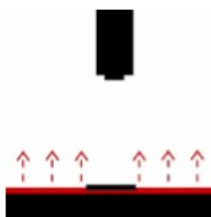


Figura 1.17: Dark lighting

- *Full bright field*, questo tipo di illuminazione consiste nell'illuminare in maniera diffusiva tutta la scena. Ci sono molti modi per creare questo tipo di illuminazione, ma i due principali sono presentati in figura 1.18. La diffusione tramite cupola è particolarmente indicata per oggetti con faccie speculari. Gli on-axis lavora su oggetti speculari piatti. Questa tipologia di utilizzo è molto indicata per acquisire in modo dettagliato differenti angolature, dettagli superficiali, o caratteristiche topografiche negli oggetti piatti. Da tenere presente però è che per avere dei buoni risultati è necessario che gli illuminatori siano vicini al campione.
- *Partial Bright Field*, è una delle tecniche più usate, si distingue da quella sopra perché qui la sorgente di luce è puntuale e non diffusiva. Viene utilizzata per generare contrasto tra dettagli topologici.

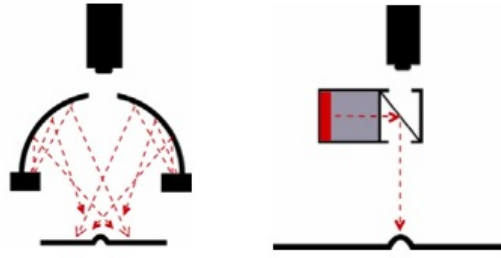


Figura 1.18: a) Hemispheric dome. b) On-axis.

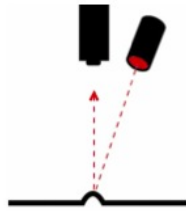


Figura 1.19: Partial Bright Field

1.6 Motivazioni

In questi primi paragrafi è stato fatto un accenno sulla vasta branca della *Machine Vision* per poter fornire al lettore una base per la comprensione dei problemi affrontati nello sviluppo della tesi.

Questa tesi è nata dall'esigenza di un'azienda del riminese, *Vici & C.*, di automatizzare una funzionalità della loro macchina di visione.

La MV in questione, che sarà presentata in modo dettagliato nel capitolo 2, ha il compito di scansionare un pezzo meccanico e compiere sul risultato della scansione, delle misure per controllare se le tolleranze date in input sono rispettate.

Per determinare se le tolleranze sono soddisfatte è necessario dare all'immagine derivante dalla scansione un preciso significato geometrico. Oggi è necessario che un operatore, data l'immagine scansionata, vada a definire quali sono e come misurare le forme geometriche presenti nell'immagine.

L'obiettivo della tesi è nato con l'idea di sfruttare l'esperienza del gruppo *Computer Vision Group* di Bologna coordinato dal Prof. *Alessandro Bevilacqua* per creare un algoritmo che data in ingresso l'immagine derivante dalla scansione, definisca in modo automatico e indipendente dall'operatore,

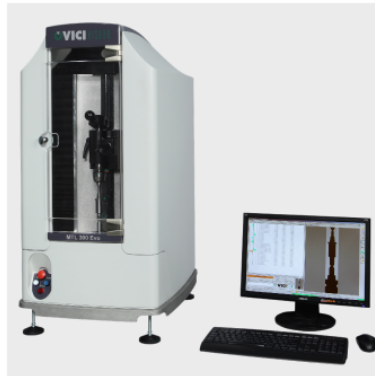


Figura 1.20: MTL 300 EVO

le forme geometriche note di cui è composto il pezzo. Restituendo quindi in output la divisione dell'immagine in tanti blocchi quante sono le forme note misurabili.

Capitolo 2

Materialiale

2.1 MTL

La macchina di visione utilizzata, chiamata MTL, esegue il controllo dimensionale mediante un sistema non a contatto per la determinazione delle caratteristiche geometriche di componenti cilindrici. L'intera gestione del sistema è affidata all'applicativo software MTL che funge da interfaccia tra operatore e macchina. La macchina è stata realizzata per l'impiego in ambiente industriale con particolare attenzione alla robustezza, alla sicurezza nel lavoro. La macchina è costituita da una carpenteria rigida e da un basamento con sistema di assorbimento delle vibrazioni tramite piedi di appoggio a terra a livellamento.

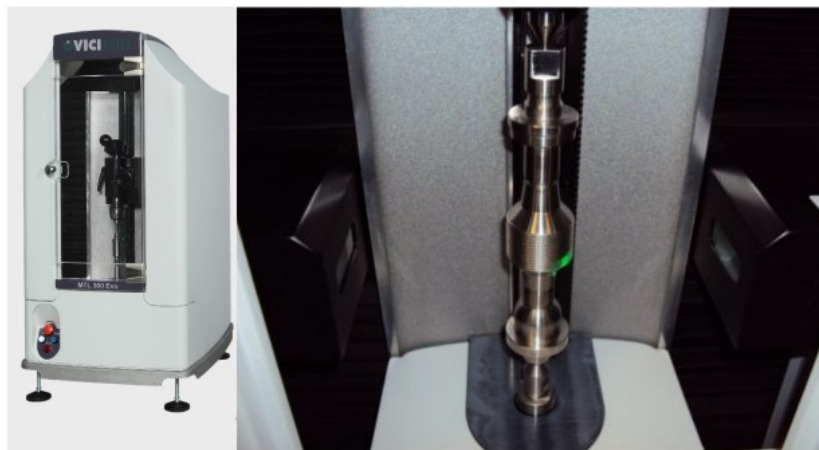


Figura 2.1: a) MTL. b) Fotografia scattata mentre MTL acquisisce il profilo del pezzo.

La macchina è stata progettata e costruita per eseguire un controllo dimensionale di componenti di fabbricazione a tornio, con la possibilità di essere fissati fra due contropunte. Il componente caricato viene sottoposto ad una scansione che genera un'immagine da cui si rileva il profilo del pezzo; tale profilo è utilizzato per il calcolo delle analisi impostate in fase di configurazione del particolare. L'applicativo software MTL consente l'impostazione delle analisi da effettuare e la visualizzazione dei risultati ottenuti.

2.1.1 Meccanica

Facendo riferimento all'immagine di figura 2.2 la parte cerchiata in rosso, evidenzia il carrello che muovendosi verticalmente permette la scansione del pezzo. Il carrello è a forma di cavallo, al centro gli viene applicata la forza di trazione che gli permette il movimento e nelle due braccia laterali c'è da una parte il dispositivo di visione e dall'altra il proiettore.

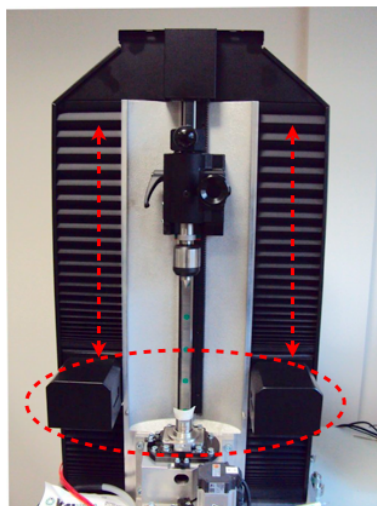


Figura 2.2: MTL fotografata senza carter per metter in evidenza (cerchio rosso) il carrello che muovendosi verticalmente permette la scansione del pezzo.

MTL per acquisire l'immagine muove il carrello dal basso verso l'alto dando alla telecamera il comando di acquisizione ogni $10\mu m$.

L'acquisizione avviene tramite il principio della *back illumination* nel seguente modo: l'emettitore emette un fascio di luce collimato verso il pezzo e i raggi che non sono ostacolati dal profilo del pezzo vanno ad irradiare i sensori del dispositivo di visione.

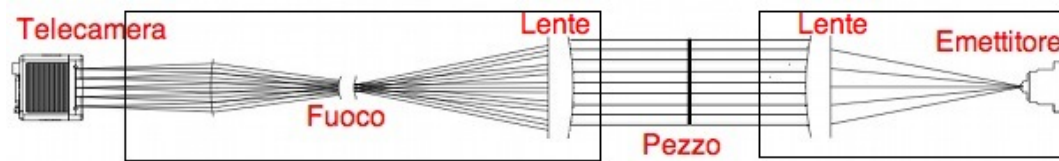


Figura 2.3: Sistema di acquisizione.

2.1.2 Illuminatore

L'illuminatore di MTL è a LED, prodotto da terzi specificatamente per la retro illuminazione di oggetti. L'illuminatore può emettere 3 tonalità di colori rosso, verde e blu. In MTL è stata scelta l'emissione della luce verde, perché ha rispetto agli altri colori un maggiore ampiezza della lunghezza d'onda.

L'emettitore è presentato dalla ditta produttrice come un *illuminatore collimato (telecentrico)*. Il motivo di questo nome è dato dalla presenza di una lente telecentrica a valle dello stesso che ha il compito di collimare la luce emessa. Considerando infatti le proprietà di questo tipo di lenti esposte nel paragrafo 1.3.3, se utilizzate in modo inverso i raggi luminosi che colpiscono il retro della lente con una qualsiasi inclinazione, vengono diffusi dalla parte opposta in modo parallelo tra loro. Questo fa in modo che i raggi che colpiscono l'oggetto siano tutti paralleli tra loro andando così a ridurre fenomeni di riflessione in prossimità dei bordi. Inoltre grazie alla lente telecentrica tutti i raggi emessi dall'emettitore sono raccolti dalla lente e direzionati verso la scena.

2.1.3 Ottica e telecamera

A monte della telecamera c'è un'ottica composta da due lenti telecentriche. L'inserimento di un'altra lente telecentrica prima del piano immagine genera una telecentricità sia lato oggetto sia lato sensore, si parla quindi di obiettivi *bi-telecentrici*.

L'utilizzo delle normali lenti telecentriche può portare ad una non uniforme costanza dell'ingrandimento dell'oggetto sul piano immagine rispetto alla scena acquisita. Questo è causato dal fatto che i raggi una volta attraversata la lente vengono deviati verso il piano immagine e l'inclinazione con cui lo colpiscono dipende dal campo angolare. Inoltre il sistema ottico risulta asimmetrico e quindi meno stabile. Questo fenomeno porta ad avere dimensioni e forme differenti tra il centro immagine e i bordi. L'introduzione di una lente telecentrica prima dei sensori, permette di catalizzare tutti i raggi presenti

nell'obiettivo e dirigerli verso i sensori in modo tale che colpiscano il piano immagine in modo ortogonale.

Un altro errore di proiezione generato dalle lenti telecentriche rispetto a quelle bi-telecentriche, riguarda il *depth of field* (1.3.2). La profondità di campo dipende dall'F-number: maggiore è l'F-number (l'apertura ottica diminuisce) maggiore è anche la profondità di campo, e queste quantità variano quasi linearmente. Questo avviene perché la profondità di campo è il massimo spostamento accettabile dell'oggetto dalla posizione di miglior fuoco. Oltre questo limite la risoluzione non è più accettabile perché i raggi che colpiscono i sensori hanno uno spot non sufficientemente piccolo, irradiando così più pixel con la stessa informazione e quindi generando un'immagine con una cattiva messa a fuoco.

L'effetto di chiudere l'apertura dell'obiettivo, cioè aumentare l'F-number, è quello di diminuire la divergenza dei raggi, i quali saranno meno sparpagliati, permettendo di avere uno spot sufficientemente piccolo sul rivelatore. Oltre ad un certo valore per l'F-number, però la risoluzione peggiora invece che migliorare; questo è dovuto alla diffrazione, che limita la minima apertura consentita per un sistema, quando è richiesto un buon contrasto.

La telecentricità lato immagine, o bi-telecentricità, fa sì che sia mantenuto un buon contrasto, anche quando vengono osservati oggetti molto spessi; la ragione di questo è che la simmetria del sistema ottico aiuta a mantenere la simmetria dello spot, e quindi a contenere il defocusing. Il risultato è una profondità di campo maggiore del 20-30% rispetto ad un'ottica non telecentrica.

La telecamera utilizzata, come anticipato, è un dispositivo lineare con una risoluzione di $4096px$, i sensori sono CCD di dimensione $10\mu m$, con un solo canale a 8 bit.

2.1.4 Calibrazione

Hardware La calibrazione della meccanica di MTL viene fatta su quei componenti che permettono il movimento del carrello che trasporta il sistema di visione. Un'altra calibrazione riguarda l'ottica della macchina, quindi è necessaria la configurazione dei giusti parametri presentati nel capitolo precedente. Prima fra tutti è la configurazione della lunghezza focale in funzione della distanza tra l'obiettivo e il centro del pezzo che secondo lo scopo della macchina deve essere a fuoco.

Un'altro tipo di calibrazione deve essere fatta sulla telecamera, come ad esempio la correzione del *dark noise*.

Software La calibrazione software ha il compito di andare a definire le tabelle di conversione che trasformano le grandezze calcolate nel sistema di riferimento immagine in valori reali (metri). Le operazioni di calibrazione software vengono effettuate tramite degli appositi pezzi, in cui tutte le grandezze che possono essere misurate sono certificate da un'azienda certificatrice.

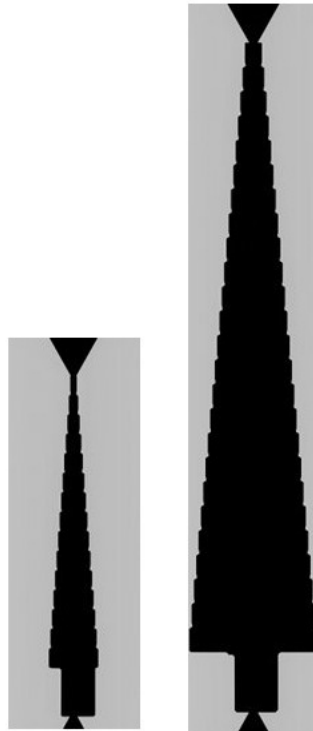


Figura 2.4: Pezzi certificati utilizzati per la calibrazione software.

La MTL ha tre tabelle di calibrazione, due per le grandezze orizzontali che derivano dalla stessa *look up table* e una per quelle verticali. La necessità di aver due tabelle di calibrazione per le grandezze orizzontale nasce dalla necessità dell'accuratezza di misura dettata dal *business* della macchina.

La LUT orizzontale viene create associando per ogni grandezza orizzontale certificata il rispettivo intervallo in pixel dell'immagine,

```
Step0 = px_left, px_right, nominal_distance_0  
Step1 = ...  
...  
StepN = ...
```

La LUT verticale è costruita definendo la quota y in pixel in corrispondenza del punto di riferimento delle altezze certificate e per ogni altezza nominale viene inserita la rispettiva quota in pixel.

```
Step0 = px_y0, px_y1, nominal_distance_0
Step1 = px_y0, px_y2, nominal_distance_1
...
StepN = px_y0, px_yN+1, nominal_distance_N
```

Per ogni tabella il coefficiente di conversione sarà calcolato come il rapporto tra la distanza in pixel e quella nominale in *mm*.

$$coeff_n = \frac{|px_{right} - px_{left}|}{nominal_distance_n}$$

Calibrazione diametri MTL è stata realizzata per fare metrologia su pezzi cilindrici supponendo che l'asse pezzo consista con il centro dei raggi delle varie circonferenze.

Tutte le immagini digitali derivate dall'acquisizione di una scena tramite un dispositivo di visione sono soggette a distorsione che aumenta all'avvicinarsi dei bordi della stessa. Per questo motivo l'aberrazione¹ sarà tanto minore al centro rispetto ai bordi dell'immagine, quindi per avere una maggiore accuratezza nella misura dei diametri è stata creata una tabella di calibrazione dedicata.

Per ogni riga della LUT orizzontale si calcola il coefficiente di conversione e si va ad inserire nel grafico un punto alle coordinate $(px_{right} - px_{left}, coeff_n)$.

Dopo aver inserito tutti i punti del grafico questi vengono interpolati tramite una curva polinomiale a tratti che passa per tutti i punti del grafico così da avere per ogni diametro in pixel il suo relativo coefficiente di conversione.

Con questo procedimento è garantita maggiore accuratezza nella misura dei diametri.

Calibrazione orizzontale Questo tipo di calibrazione viene utilizzata per il calcolo delle distanze punto punto. Partendo sempre dalla LUT orizzontale viene calcolata per ogni riga il rispettivo coefficiente di conversione ma questa volta nel grafico vengono inseriti due punti, uno in corrispondenza della coordinata $(px_{left}, coeff_n)$ e un'altro in $(px_{right}, coeff_n)$. Il risultato è un grafico simmetrico con asse di simmetria precisamente sulla x equivalente alla metà della risoluzione della telecamera.

¹L'aberrazione di un sistema ottico è la differenza tra l'immagine effettiva, reale o virtuale, formata dal sistema e l'immagine che si voleva ottenere

Come sopra l'insieme dei punti del grafico viene interpolato con una curva polinomiale a tratti così da definire il coefficiente di conversione per ogni distanza punto punto.

Calibrazione verticale La calibrazione verticale viene fatta per dare ai pixel dell'immagine l'equivalente altezza in mm e per compensare l'errore che si genera durante il movimento del carrello.

Il carrello durante il suo movimento verticale fornisce alla telecamera lineare il comando di acquisizione ogni $10\mu m$, questo idealmente significa che un pixel dell'immagine ha un'altezza pari a $0.010mm$. Quest'affermazione è valida solo idealmente, infatti il carrello è soggetto ad un errore sistematico dovuto alle vibrazioni che si generano nella macchina, in più il tempo di apertura e chiusura della telecamera non è nullo.

La creazione del grafico di conversione è fatta nel modo seguente: partendo dal basso dell'immagine si considera l'altezza di tutti i pixel fino ad arrivare alla quota y_0 pari a $10\mu m$. Quindi l'altezza reale dalla base della scena acquisita al primo punto certificato è pari a $\#pixel \times 10\mu m$ (es. $0.010mm \times 10000px = 100mm$).

Successivamente viene calcolato il coefficiente di conversione in modo incrementale rispetto alla quota precedente, ad esempio se rispetto alla quota certificata y_0 , y_1 è distante $1cm$ significa che:

$$\frac{100mm + 10mm}{10000px + 100px} \simeq 108mm$$

Quest'operazione è fatta per tutte le altezze certificate e per ognuna di esse viene messo nel grafico un punto nell'ascissa equivalente alla differenza in pixel e in ordinata in corrispondenza della variazione percentuale dell'altezza dei pixel calcolata rispetto al valore ideale di $10\mu m$.

2.1.5 Software

Il software MTL si compone essenzialmente di due parti:

- gestione ricette,
- esecuzione delle misure.

Una ricetta MTL è il programma che definisce quali operazioni la macchina di visione deve compiere sul pezzo. Una ricetta è sempre formata da tre operazioni fondamentali, quali:

1. scansione pezzo: genera l'immagine del pezzo:

2. selezione inizio pezzo: si definisce qual'è il punto basso più estremo. Questa misura potrà essere utilizzata come punto di riferimento;
3. selezione fine pezzo: si definisce qual'è il punto alto più estremo. Questa misura potrà essere utilizzata come punto di riferimento.

Una volta definite le tre operazioni è possibile aggiungere nuove funzionalità alla ricetta, come ad esempio il calcolo di un diametro, di un'altezza, di un angolo, la rotazione del pezzo etc... Ad esempio quando si desidera aggiungere al programma un'operazione di *calcolo diametro*, l'operatore deve disegnare tramite un tool grafico, sull'immagine a monitor del pezzo scansionato, un rettangolo che contenga i due profili sinistro e destro del particolare da misurare. Determinata la regione d'interesse, l'operatore deve configurare la lunghezza nominale del diametro e la tolleranza accettata.

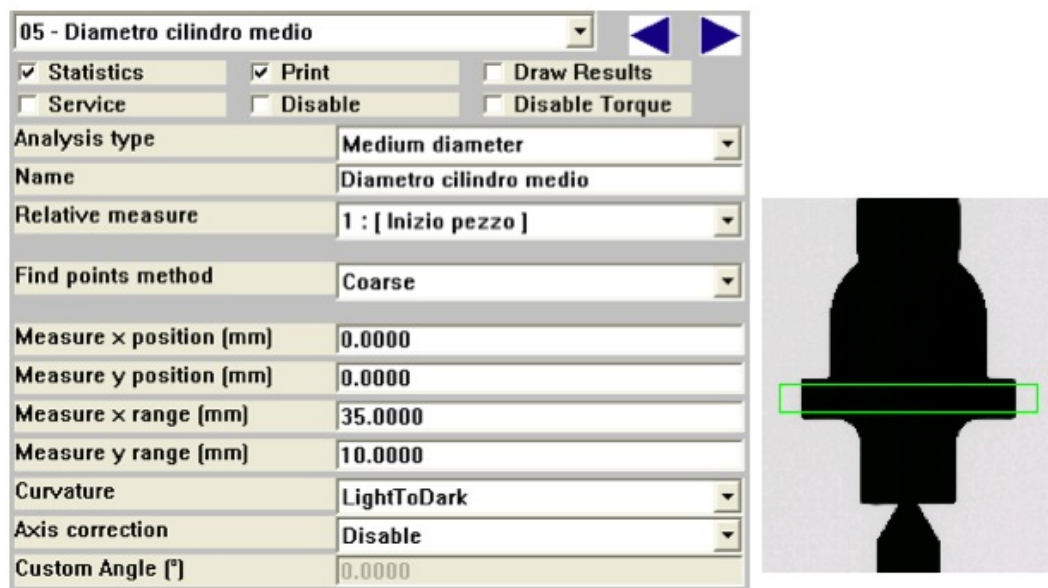


Figura 2.5: Configurazione dell'analisi di diametro medio: a sinistra c'è la scheda per configurare i parametri, a destra c'è il tool grafico.

Dopo aver creato la ricetta si può passare alla fase vera e propria di controllo dei pezzi meccanici della stessa fattura di quello in oggetto al programma. Quando viene lanciato il comando che determina l'inizio del controllo, la MTL esegue tutti i passi presenti nella ricetta e restituisce in uscita la tabella delle misure effettuate, indicandone se rispettano o meno le tolleranze.

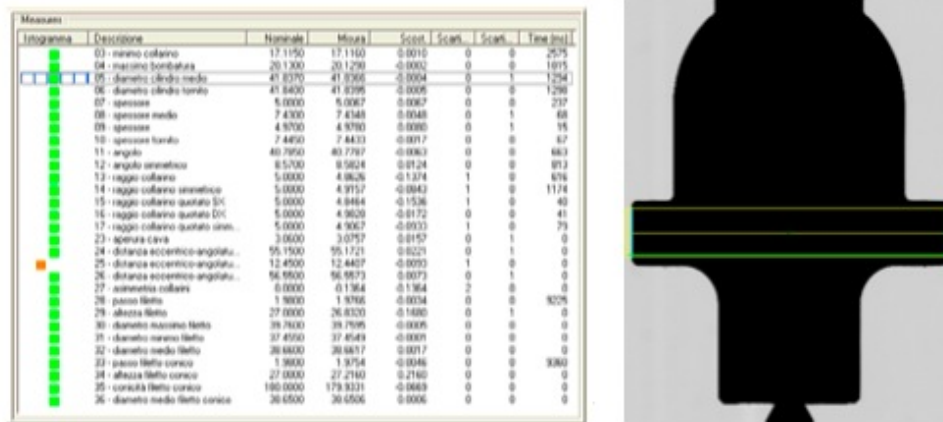


Figura 2.6: Risultati dell'analisi diametro.

2.2 Pezzi utilizzati

La macchina MTL è utilizzata principalmente per il calcolo di:

- diametri: l'operatore deve creare una *bounding box* indicando gli estremi del diametro;
- distanza tra piani (spallamenti) orizzontali: è necessario che l'operatore definisca i piani tra i quali verrà calcolata la distanza;
- angoli formati tra lati obliqui e asse pezzo: l'operatore deve selezionare tramite la *bounding box* la zona dell'immagine in cui è presente il lato obliquo;
- raggio di archi di circonferenza: come sopra l'operatore deve selezionare la regione in cui si trova l'arco di circonferenza;
- etc..

L'elenco sopra indica le principali forme geometriche che l'operatore in fase di creazione della ricetta deve selezionare nel tool grafico. L'algoritmo che è stato sviluppato deve *trovare ed interpretare* queste forme geometriche in modo automatico. Per questo motivo sono stati scelti due pezzi, da utilizzare come *benchmark*, che sono composti dalla principali forme geometriche elencate sopra.

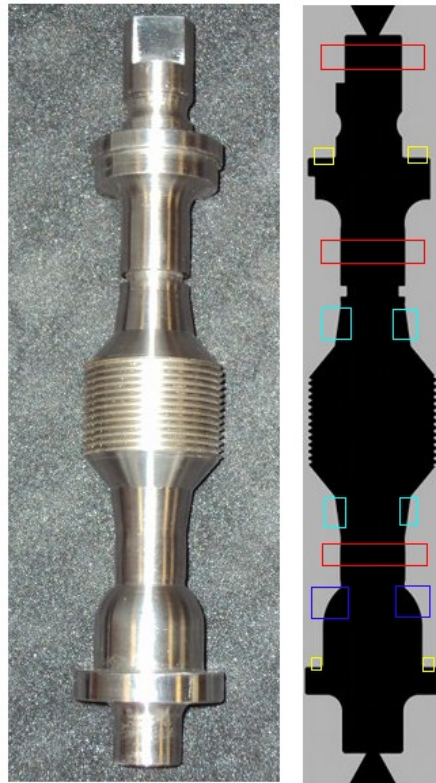


Figura 2.7: Pezzo 1: a) originale; b) scansione ed interpretazione manuale del pezzo.

Pezzo 1 Il pezzo di figura 2.7.a è stato fabbricato ad hoc dall'azienda con le principali forme geometriche, per mostrare le funzionalità offerte dalla MTL. In figura 2.7.b sono selezionate:

- rettangoli rossi: selezionano le parti del pezzo in cui è possibile calcolare un diametro;
- rettangoli gialli: indicano la presenza di piani (spallamenti) del pezzo;
- rettangoli azzurri: selezionano le zone in cui ci sono lati obliqui;
- rettangoli blu: indicano le parti in cui c'è un arco di circonferenza.

Pezzo 2 Il secondo pezzo è stato scelto perché presenta a vista delle possibili difficoltà in fase di analisi. Come mostrato in figura 2.8.b ci sono i rettangoli colorati nello stesso precedente modo, mentre contornate da un rettangolo

bianco sono indicate particolari zone del pezzo in cui sono presenti sottili “incisioni”.

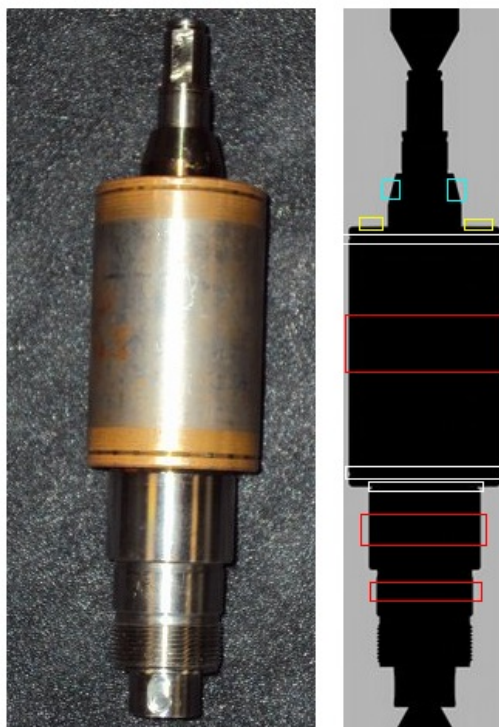


Figura 2.8: Pezzo 2: a) originale; b) scansione ed interpretazione manuale del pezzo.

Capitolo 3

Misura industriale automatica con macchine di visione

MTL presentata nel capitolo 2 appartiene alla categoria di macchine di visione che fanno *metrologia*, ovvero effettuano misure non a contatto.

Fare misura su un'immagine significa: dati due punti, contare il numero di pixel che li separa. Questa definizione è sicuramente la punta dell'iceberg.

Qualsiasi programma di grafica ha la capacità, dati due estremi su un'immagine, di calcolarne la distanza in pixel. Nasce però un problema: l'unità di misura è in pixel, la quale ha significato nel mondo digitale, ma perde tutto il proprio valore quando si torna nel mondo reale.

Le macchine di visione che fanno misure non a contatto con sensori ottici, necessitano di una fase di calibrazione che, come presentato per la MTL nel paragrafo 2.1.4, va a definire *quantitativamente il significato di un pixel nel mondo reale*.

Ponendo ora l'attenzione sulla macchina di visione utilizzata in questa tesi, successivamente alla fase di calibrazione è necessario configurare una ricetta di controllo in modo tale che ad ogni scansione di un dato pezzo meccanico si vada a calcolare e controllare solo le lunghezze d'interesse. Per far ciò è necessario che un operatore definisca quali misure il sistema debba controllare.

L'obiettivo di questa tesi è sviluppare un algoritmo che in modo automatico vada a definire quali misure compiere su un dato pezzo in ingresso. Per far ciò è necessario che il sistema *interpreti* l'immagine derivante dalla scansione.

Per permettere al software MTL di interpretare l'immagine è necessario che questa sia ben formata. Prima che l'immagine sia elaborata è neces-

sario sottoporla ad una fase di *preprocessing* per ridurre gli errori dovuti all'acquisizione.

3.1 Acquisizione dell'immagine

Il corretto funzionamento di un sistema di visione industriale dipende dall'accuratezza con cui si acquisisce l'immagine. Quando si parla di accuratezza in questo settore si intende:

- *misura*: corrispondenza tra il valore misurato digitalmente e la grandezza reale;
- *forma*: acquisizione della reale forma del pezzo, evitando la generazione di caratteristiche non veritiere nell'immagine;
- *mancato rilevamento*: perdita di informazioni e caratteristiche della scena acquisita.

L'acquisizione dell'immagine gioca un ruolo fondamentale nella macchine di misura industriale, produce infatti l'input per il software di elaborazione. Bisogna però tenere in considerazione che le MV sono formate da componenti meccanici, ottici ed elettronici ognuno dei quali è soggetto ad errori più o meno rilevamenti.

I problemi di acquisizione che sono stati riscontrati in MTL che quindi influiscono sull'accuratezza del sistema sono i seguenti:

- dimensione e variabilità del pixel,
- determinazione certa del bordo,
- riflettività
- artefatti dovuti a polvere, etc,

L'insieme degli errori generati nella fase di acquisizione si ripercuote sull'immagine digitale da analizzare.

Nei paragrafi seguenti è spiegato come gli errori introdotti dai componenti sopra elencati si ripercuotono sull'immagine da elaborare.

3.1.1 Dimensione e variabilità del pixel

Un pixel è un elemento puntiforme che compone l'immagine digitale, ha una forma quadrata e una dimensione unitaria in ogni punto dell'immagine. Un pixel di un'immagine risultate dall'acquisizione di una scena equivale però a una porzione di mondo reale. La dimensione di questa porzione di mondo reale è determinata dalla risoluzione del sensore utilizzato e dai possibili effetti distorcenti dovuti dall'ottica. Per determinare questo valore è necessario in fase di configurazione del sistema definire una tabella di calibrazione che determina per ogni pixel la propria grandezza reale.

In MTL un'immagine acquisita è il risultato di N scansioni fatte dalla linea ottica durante il proprio moto verticale, quindi la dimensione di un pixel è determinata da due fattori:

- *statico*: definisce la larghezza del pixel in base alla risoluzione del sensore e dagli effetti distorcenti dell'ottica;
- *dinamico*: definisce l'altezza del pixel in base al movimento.

Quando un'immagine è il risultato sia di componenti ottiche che meccaniche è necessario determinare la dimensione e la variabilità dei pixel all'interno dell'immagine. In MTL durante il moto verticale del carrello, viene dato alla telecamera il comando di acquisizione ogni $10\mu m$, e idealmente l'altezza di un pixel può essere considerata tale. Il movimento verticale del carrello però è soggetto a degli errori sistematici e accidentali, corretti in parte dall'operazione di calibrazione, che ne alterano il movimento. A causa di questi errori può succedere che la telecamera non inizi ad acquisire esattamente ogni $10\mu m$ oppure possono verificarsi situazioni in cui il l'emettitore e la linea ottica non siano perfettamente allineate allungando o appiattendolo la porzione di mondo reale che il pixel irradiato rappresenta digitalmente.

3.1.2 Determinazione dei bordi

La corretta determinazione del bordo è uno dei principali problemi delle applicazioni di visione industriale. La precisa determinazione nell'immagine del punto in cui si trova il bordo del pezzo reale va a inficiare sull'accuratezza di misura. In prossimità di un bordo dell'immagine non si verifica mai una "secca" transizione chiaro scuro o viceversa, ma quell'effetto di sfocatura mostrato in figura 3.1.

Quest'effetto è causato dai raggi di luce che colpendo il pezzo in prossimità di un bordo vengono deviati e fanno ad irradiare sensori già caricati.

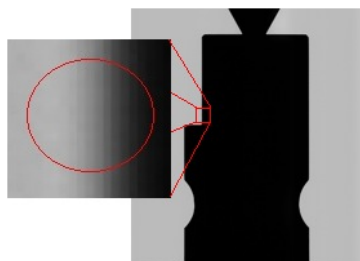


Figura 3.1: Transizione dei livelli di grigio in prossimità di un bordo dell'immagine.

Questo problema, noto anche come *problema del gradiente*, non permette una semplice e accurata individuazione del bordo.

Ulteriormente come spiegato sopra, il pixel quantitativamente rappresenta una porzione del mondo reale e quindi una parte non unitaria della scena acquisita. Il reale bordo del pezzo equivale ad un punto unitario che però nell'immagine digitale sarà approssimato ad un pixel. Per i motivi del paragrafo precedente è necessario uno studio del sistema per l'esatta determinazione del bordo del pezzo che sarà realmente all'intero del pixel; si parla quindi di precisione *sub-pixel*.

3.1.3 Riflettività

L'immagine acquisita dalla MTL si presenta con una tonalità alta ai bordi e una tonalità circa nulla che rappresenta il foreground dell'immagine. La conformazione del background dell'immagine è data quindi dai raggi emessi dall'emettitore (vedi 2.1.1) che riescono a raggiungere i sensori del dispositivo di visione, mentre il foreground corrisponde a quei sensori che *non* vengono irradiati.

La procedura sopra discussa è prettamente teorica, infatti fenomeni fisici di riflessione¹ vanno ad alterare l'immagine acquisita.

Uno dei problemi di riflessione è dato dal fascio di luce piatto solo idealmente. Ad esempio quando il dispositivo di visione è ad una certa quota h , anche il centro dell'emettitore si trova a tale quota ma il fascio generato non sarà realmente piatto, ma bensì occuperà un intervallo $h \pm \Delta$.

I raggi di luce che, non essendo allineati con il dispositivo di visione colpiscono il pezzo, verranno riflessi in qualche direzione. I raggi riflessi però possono

¹La riflessione è il fenomeno per cui un'onda, in questo caso luminosa, cambia di direzione a causa di un impatto con un materiale riflettente.

andare ad irradiare un sensore già colpito da un raggio luminoso, andando così ad aumentare la tonalità del pixel risultante. L'effetto dovuto a questo fenomeno è mostrato in figura 3.2.

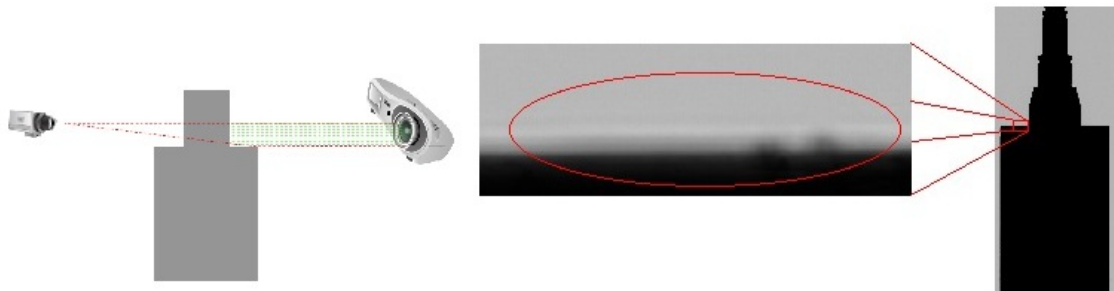


Figura 3.2: Il raggio che colpisce lo spigolo del pezzo, va ad irradiare un sensore già caricato. L'effetto risultante è presentato nel quadrato.

Un altro problema causato dalla riflessione non noto a priori è dovuto alla conformazione di certi pezzi. Soprattutto nei pezzi meccanici composti può succedere che in certi punti si verifichi una situazione in cui una conformazione di diverse angolature, facciano rimbalzare più volte i raggi al loro interno prima che questi riescono ad arrivare al dispositivo di visione. L'effetto di questo problema si riscontra nell'immagine acquisita nel modo presentato in figura 3.3.



Figura 3.3: Il pezzo in questione è un pezzo composto, composto da un albero e un cilindro inserito sopra. La fessura che c'è tra l'unione dei due pezzi, racchiusa nel rettangolo rosso, crea degli angoli che deviano in modo non regolare i raggi che li colpiscono.

3.1.4 Artefatti dovuti a polvere, etc

La prima cosa che salta all'occhio quando si guarda un'immagine di un pezzo scansionata dalla MTL è la presenza di forme "strane" sui bordi del pezzo. Queste forme irregolari sono semplicemente dei granelli di polvere, impercettibili all'occhio umano, che si depositano sul pezzo. Il rilevamento di questi granelli di polvere è in funzione dell'alta risoluzione della riga ottica.

Questo problema risulta di poco conto quando l'interpretazione di un pezzo è di competenza di un operatore che all'eventuale riscontro di queste irregolarità può pulire il pezzo. Mentre, quando l'interpretazione dell'immagine è a carico di un sistema di elaborazione automatico il deposito di polvere sui bordi del pezzo genera non pochi problemi.

La possibile alterazione dell'immagine è mostrata in figura 3.4.

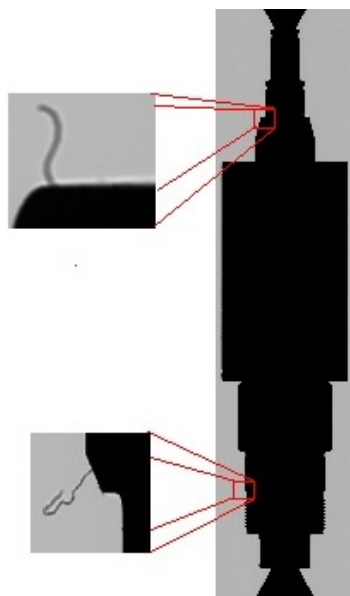


Figura 3.4: Granelli di polvere depositati sul pezzo ne alterano il contorno.

3.2 Dimensione delle immagini e tempistiche

Un requisito di questa tesi è la possibilità di integrare l'algoritmo sviluppato nel software di elaborazione di immagine della MTL. Le funzionalità sviluppate dovranno guidare l'operatore nella creazione delle ricette (vedi 2.1.5) interpretando in modo autonomo le forme geometriche note misurabili e lasciando all'operatore il compito di inserire le dimensioni nominali e le

tolleranze. L'*improvement* apportato dall'algoritmo deve anche permettere un risparmio di tempo nella creazione delle ricette.

Definito questo vincolo temporale, bisogna tenere conto che le immagini acquisite dal MTL sono formate da diverse migliaia di milioni di pixel. Si consideri che un'acquisizione della linea ottica forma un'immagine lineare di $4096px$ mentre l'altezza varia a seconda del pezzo. Se un oggetto è alto $30cm$ e il dispositivo di visione acquisisce una linea ogni $10\mu m$ significa che l'immagine sarà formata dall'unione di 30000 righe. L'immagine risultante sarà formata da più di $120Mpx$ e i tempi di elaborazione aumentano in maniera lineare all'aumentare dei pixel. Nello sviluppo della tesi è stato necessario prendere delle decisioni per riuscire a soddisfare il requisito temporale.

Capitolo 4

Tecniche di image processing

L'*image processing* è una branca della *Computer Vision* ma non esiste nella letteratura una definizione che indica i confini dell'*image processing*. A volte la distinzione è data definendo l'*image processing* come una disciplina in cui sia l'input che l'output sono immagini, però questa definizione si dimostra essere una limitazione, infatti ad esempio il semplice conteggio della media dei pixel di un'immagine non ricadrebbe in quest'ambito. D'altra parte la *Computer Vision* è anche vista come la scienza che studia soluzioni per emulare la vista umana, che permette l'apprendimento e l'interazione con il mondo circostante, andando così a richiamare principi d'intelligenza artificiale. Anche in questo caso la parte che riguarda l'interpretazione dell'immagine ricade tra l'*image processing* e la *computer vision*.

Non c'è un confine certo tra l'*image processing* e le altre branche della *Computer Vision*, è però possibile utilizzare un paradigma che definisce tre livelli di elaborazione dell'immagini: *low, mid* e *high level*.

Nel primo livello, *low*, sono racchiuse tutte le operazioni primitive di *pre-processing* che consistono nella riduzione del rumore dovuto all'acquisizione, nel migliorare il contrasto e nel rendere più nitida possibile l'immagine. Queste operazioni hanno come peculiarità che sia l'input che l'output sono immagini.

Il livello medio, *mid*, consiste nell'operazione di segmentazione dell'immagine, quindi partizionare e classificare l'immagine in regioni d'interesse. In questo livello l'input è un'immagine mentre l'output è generalmente un insieme di informazioni estratte dall'immagine.

L'ultimo livello, *high*, interpreta l'immagine, riconoscendo oggetti, analizzandoli, svolgendo quelle funzioni cognitive che un essere umano tipicamente

associa alla vista di un'immagine.

In questo capitolo verranno presentate nell'ordine, low, mid e high le tecniche di image processing che sono stata prese in considerazione nello sviluppo della tesi.

4.1 Morfologia

La parola *morfologia* viene associata a quella branca della biologia che studia le forme e le strutture di animali e piante. I concetti della *morfologia matematica* possono essere utilizzati come strumento nell'elaborazione delle immagini per estrarre informazioni da esse.

La morfologia matematica offre una potente approccio per i numerosi problemi del *image processing* e permette di estrarre il contenuto informativo dell'immagini.

Un'immagine è formata da un insieme di pixel che ne definiscono globalmente il contenuto informativo ma ogni pixel assieme al proprio vicinato ne fornisce informazioni locali. L'elaborazione dell'immagine tramite operatori morfologici si avvale di *elementi strutturati* che definiscono la forma elementare del contenuto informativo che si vuole estrarre. L'elemento strutturante è una matrice, in cui il pixel centrale è chiamato origine e identifica il pixel d'interesse che dovrà essere processato. Gli altri pixel definiscono il vicinato rispetto all'origine.

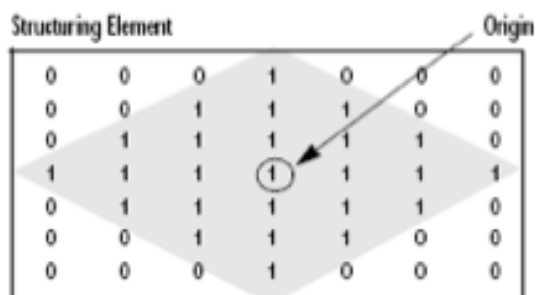


Figura 4.1: Elemento strutturante a diamante.

L'introduzione dei concetti di morfologia nell'elaborazione delle immagini, cambia l'approccio classico in cui si aveva in ingresso un'immagine e l'uscita era l'immagine elaborata. In questo caso l'output è l'insieme di attributi

estratti dall'immagine. Per questo motivo si dice che la morfologia estrae il significato dall'immagine.

Nei prossimi paragrafi verranno introdotte le tecniche che sono state utilizzate nello sviluppo della tesi, per tutte le altre nozioni riguardanti l'utilizzo della morfologia matematica nell'elaborazione dell'immagine si richiama a [6].

4.1.1 Dilatazione

Considerando A e B , l'immagine e l'elemento strutturante rispettivamente, appartenenti al campo Z^2 , la *dilatazione* di A tramite B è definita come:

$$A \oplus B = \{z | [(\hat{B})_z \cap A] \subseteq A\}.$$

Il risultato della dilatazione è l'insieme di tutti i *displacement*, z , tale che B e A hanno sovrapposto almeno un elemento. $(\hat{B})_z$ è la versione speculare di B , traslata di z . L'elemento strutturato viene posizionato su ogni punto dell'immagine d'ingresso e, se il punto equivale all'origine di B , si esegue l'OR logico fra l'elemento strutturante e l'immagine.



Figura 4.2: Dilatazione con un elemento strutturante 3×3 con tutti i valori 1. a) immagine originale; b) immagine risultate dopo la dilatazione.

La dilatazione viene utilizzata generalmente:

- espande i componenti;
- ingrandisce delle caratteristiche;
- riempie i buchi non appartenenti alla regione d'interesse.

4.1.2 Erosione

Considerando l'immagine e l'elemento strutturato A e B , si definisce *erosione*:

$$A \ominus B = \{z | (B)_z \subseteq A\}.$$

L'equazione mostra che l'erosione di A tramite B è l'insieme di tutti i punti z tale che B , traslata di z è contenuta in A . L'elemento strutturante viene posizionato su ogni punto dell'immagine A e, se l'AND logico fra tutti gli elementi a '1' dell'elemento strutturante e l'immagine in ingresso vale '1', il punto corrispondente dell'immagine in uscita viene messo a '1'.

L'erosione può essere vista come la sottrazione degli elementi dei due insiemi A e B .



Figura 4.3: Erosione con un elemento strutturante 3×3 con tutti i valori 1. a) immagine originale; b) immagine risultate dopo l'erosione.

L'erosione viene utilizzata generalmente:

- erodere i componenti connessi;
- ridurre delle caratteristiche;
- rimuovere collegamenti leggeri fra due oggetti;
- ridurre il rumore ai bordi.

4.1.3 Opening e closing

Come spiegato sopra, la dilatazione espande un'immagine mentre l'erosione la riduce. La combinazione di questi due operatori elementari definisce altri due operatori morfologici: l'*opening* e il *closing*.

L'*opening* di un'immagine A con un elemento strutturato B , è definita:

$$A \circ B = (A \ominus B) \oplus B.$$

L'*opening* è quindi composta da una prima erosione di A con B , per poi dilatare il risultato sempre con lo stesso B . L'operazione di *opening* rimuove tutti i pixel nelle regioni che sono troppo piccole per contenere l'elemento strutturante mantenendo però la forma.

Proprietà dell'operazione di *opening*:

- rimuove le regione che non contengono l'elemento strutturato;
- fanno lo *smooth* dei contorni;
- eliminato le connessioni sottili tra gli oggetti;

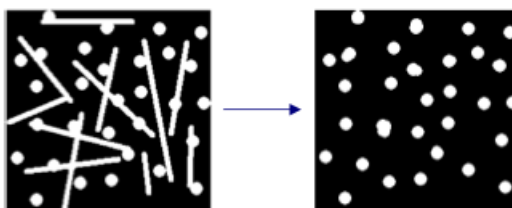


Figura 4.4: Erosione dell'immagine di sinistra con un elemento strutturante di forma circolare.

L'operazione di closing di un insieme A tramite l'elemento strutturante B , è definita:

$$A \bullet B = (A \oplus B) \ominus B.$$

La closing è quindi composta da una dilatazione, seguita dall'erosione. L'operazione di closing connette gli oggetti che sono vicini tra loro, riempie i buchi non appartenenti alla regione d'interesse e mantiene la forma originale.

Proprietà dell'operazione di closing:

- fanno lo *smooth* dei contorni;
- uniscono gli oggetti vicini tra loro;
- riempiono i buchi che sono più piccoli dell'elemento strutturato.



Figura 4.5: Dilatazione dell'immagine di sinistra con un elemento strutturante di forma circolare.

4.1.4 Algoritmi morfologici

La combinazione di due o più delle operazioni presentate sopra da vita a diverse funzioni con determinate proprietà, in questo paragrafo ne verranno presentate solo alcune.

- *Morphological boundary extraction*: facendo la sottrazione dell'immagine originale A con il risultato ottenuto erodendo A , è possibile ricavare i bordi dell'immagine.

$$A - (A \ominus B).$$



Figura 4.6: Estrazione del contorno dell'immagine di sinistra utilizzando un elemento strutturato 3×3 .

- *Hit and Miss*: è uno strumento utilizzato per riconoscere le forme. Si avvale di due elementi strutturati uno che identifica la forma del foreground ($SE1$) e l'altro la forma del background ($SE2$). L'operatore hit-miss preserva i pixel che *matchano* con la forma espressa in $SE1$ e non *matchano* con la forma $SE2$.

$$\begin{aligned} hitmiss(I, SE) &= (I \ominus SE1) \& (\neg I \ominus SE2), \\ SE &= (SE1, SE2), SE1 \cap SE2 = \emptyset \end{aligned}$$

- *Thinning*: come dice il nome, questa operazione viene utilizzata successivamente ad un *boundary extraction* per snellire i bordi dell'oggetto. Questa operazione utilizza la trasformata hit and miss.

$$thin(I, SE) = I - hitmiss(I, SE)$$

Generalmente per fare lo thinning dell'immagine si utilizza un insieme di elementi struttura che definiscono le regioni d'interesse:

$$\{SE\} = \{SE^1, SE^2, \dots, SE^n\}.$$

- *Smoothing*: questa tecnica è utilizzata per attenuare o rimuovere i massimi in prossimità dei bordi o gli effetti causati dal rumore.

$$smooth(I, SE) = ((I \circ SE) \bullet SE)$$

Si noti che tutte le operazioni descritte sono applicabili sia con immagine binarie che con immagine a livelli di grigio.

4.2 Segmentazione

La segmentazione si inserisce tra i livelli dell'immagine processing al secondo posto, ovvero *mid level*. Questo perché al contrario delle operazioni morfologiche presentate sopra, dato in input un'immagine, l'output è un insieme di attributi estratti. La segmentazione di un'immagine consiste nel suddividere l'immagine in regioni d'interesse o oggetti a seconda del problema da risolvere. Questo step dell'immagine processing è forse il più difficile, infatti una segmentazione accurata determina il successo o il fallimento delle procedure di analisi.

Gli algoritmi di segmentazione dell'immagini sono generalmente basati su due proprietà riguardanti l'intensità dei pixel: uguaglianza e discontinuità. Nella prima categoria l'approccio consiste nella suddivisione dell'immagine in regioni che hanno determinati valori di uguaglianza, in questo caso si parla di *thresholding*. Mentre il principale approccio utilizzato nella seconda categoria è basato sull'analisi della variazione d'intensità delle tonalità dei pixel, come accade in prossimità dei bordi dell'immagine (*edge detection*).

4.2.1 Thresholding

Con l'espressione sogliare un'immagine si intende l'operazione con la quale si determina un valore soglia T , da cui viene generata una nuova immagine dove tutti i pixel con valore inferiore alla soglia valgono 0, mentre quelli maggiori valgono 1.

Il thresholding di un'immagine è un semplice modo per separare gli oggetti d'interesse dell'immagine, *foreground*, da quelli appartenenti al *background*.

La scelta della soglia da applicare all'immagine, supponiamo a 8 bit, si basa sull'analisi dell'istogramma a livelli di grigio. Nel caso più semplice se l'istogramma è bimodale la determinazione della soglia non comporta troppo sforzo, infatti T viene scelta tra le due mode, mentre per determinare la soglia di immagini che hanno l'istogramma di non facile analisi è necessario ricorrere a degli appositi algoritmi.

Metodo P-Tile Questo metodo viene utilizzato quando la tonalità della zona d'interesse è approssimativamente conosciuta e si va quindi a definire una soglia T in modo empirico. La conoscenza a priori del valore della zona d'interesse rende questo metodo scarsamente automatizzabile.

Peakiness Detection L'analisi delle "piccosità" dell'immagine è forse uno dei metodi migliori per la determinazione del valore di soglia. Questo tipo di algoritmo viene utilizzato in immagini con istogrammi N -modali. L'algoritmo funziona nel seguente modo:

1. trovare nell'istogramma due massimi locali che abbiano la minima distanza (g_i e g_j di figura 4.7);

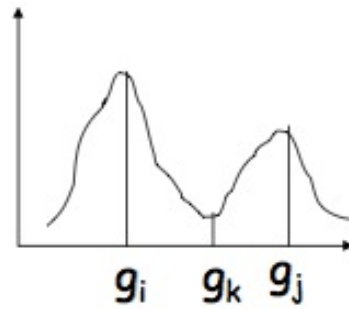


Figura 4.7: Istogramma

2. trovare tra i due picchi il punto più basso tra loro (g_k);
3. calcolare la *peakiness* come:

$$\max \left(H(g_i), \frac{H(g_j)}{H(g_k)} \right)$$

4. Calcolare e scegliere la g_k che ha piccosità maggiore.

Metodo Isodata Questo algoritmo permette di trovare il valore di soglia in modo iterativo e robusto. L'algoritmo è così formato:

1. l'istogramma originale è diviso in due parti definendo una soglia iniziale pari a $T_0 = 2^{B-1}$, in cui B è la profondità dell'immagine.

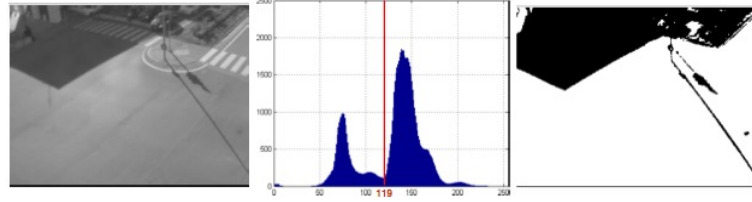


Figura 4.8: a) Immagine originale. b) Istogramma dell'immagine in cui la riga rossa determina la sogliatura, calcolata con l'algoritmo di peakiness, di valore 119. c) Immagine risultate dalla segmentazione.

2. Si calcola la media semplice dei valori appartenenti al foreground, $m_{f,0}$, e quella del background, $m_{b,0}$.
3. Un nuovo valore di soglia, T_k viene calcolato come la media delle medie dei campioni calcolate al punto 2.

$$T_k = \frac{m_{f,k-1} + m_{b,k-1}}{2}$$

4. il punto 2 e il punto 3 vengono ripetuti fino a che la soglia converge $T_k = T_{k-1}$.

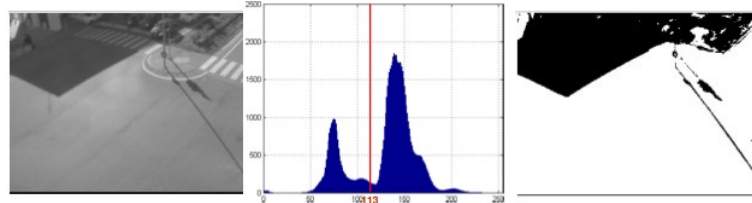


Figura 4.9: a) Immagine originale. b) Istogramma dell'immagine in cui la riga rossa determina la sogliatura, calcolata con l'algoritmo di isodata, di valore 113. c) Immagine risultate dalla segmentazione.

Otsu Il metodo di Otsu permette di effettuare una sogliatura automatica dell'immagine determinando la soglia T che minimizza la varianza σ_{within} all'interno delle due classi separate da T . Se si definisce N come il numero massimo di pixel dell'immagine:

$$\sigma_{within}^2(T) = \frac{n_1(T)}{N} \sigma_1^2(T) + \frac{n_2(T)}{N} \sigma_2^2(T)$$

dove:

$$n_1(T) = |\{I_{(x,y)} < T\}|, \quad n_2(T) = |\{I_{(x,y)} \geq T\}|$$

La stessa formula può essere riscritta come la massimizzazione della varianza tra le classi, $\sigma_{between}$:

$$\sigma_{between}^2(T) = \sigma^2 - \sigma_{within}^2(T) = \frac{n_1(T)n_2(T)}{N^2} [\mu_1(T) - \mu_2(T)]^2$$

L'algorithmo iterativamente va a calcolare al soglia che massimizza $\sigma_{between}^2$.

$$T^* = \operatorname{argmax}[\sigma_{between}^2(T)]$$

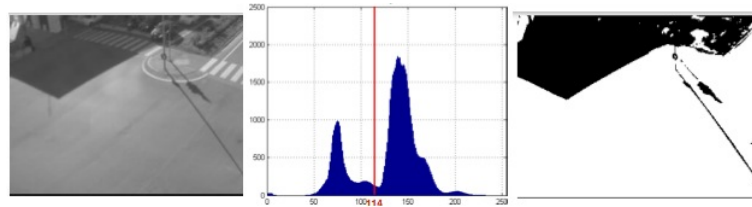


Figura 4.10: a) Immagine originale. b) Istogramma dell'immagine in cui la riga rossa determina la sogliatura, calcolata con l'algorithmo di otsu, di valore 114. c) Immagine risultate dalla segmentazione.

4.2.2 Edge detection

Le principali tecnica di *edge detection* che permettono di rilevare punti, linee e bordi di un'immagine analizzando la discontinuità si basano sull'utilizzo di maschere. Generalmente viene utilizzata una maschera 3×3 , in cui per ogni pixel viene calcolata la somma dei prodotti, dei coefficienti con i livelli di grigio dell'immagine, che vengono sovrapposti dalla maschera.

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 = \sum_{i=1}^9 w_i z_i$$

Un bordo si trova in una zona dell'immagine in cui c'è una forte transizione dei livelli di grigio, idealmente si verifica la situazione di figura 4.11.a in cui c'è una transizione secca che determina il bordo. In pratica però, l'ottica, la quantizzazione e tutte le imperfezioni derivanti dall'acquisizione portano ad avere dei bordi rumorosi. Per questo motivo la transizione dei livelli di grigio in prossimità di un bordo è paragonabile alla rampa di figura 4.11.b.

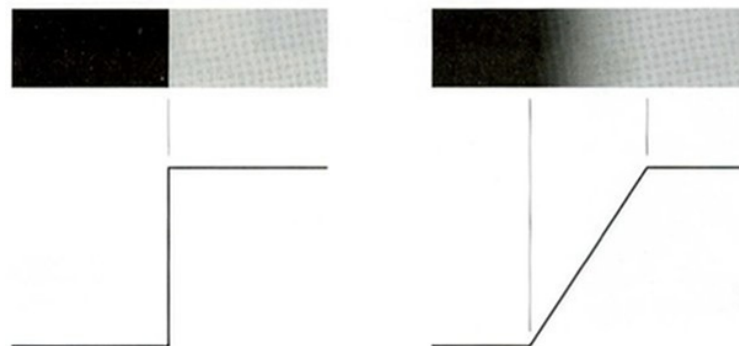


Figura 4.11: a) Profilo ideale della variazione dei livelli di grigio in prossimità di un bordo. b) Profilo reale della variazione dei livelli di grigio in prossimità di un bordo senza rumore.

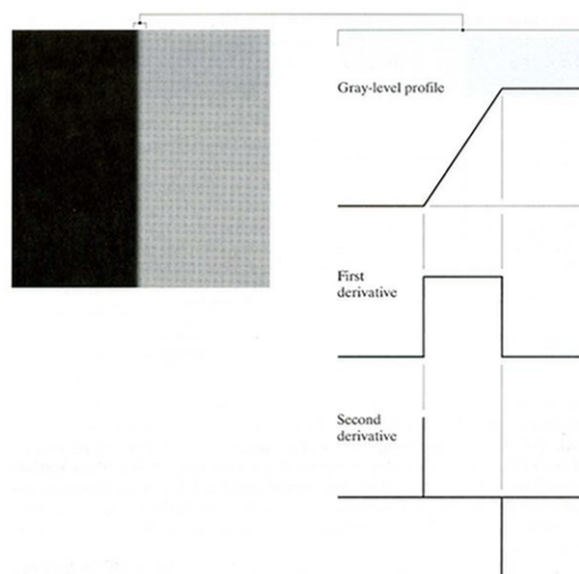


Figura 4.12: Il primo grafico rappresenta il profilo in prossimità di un bordo. Il grafico in mezzo è la derivata prima del grafico sopra. L'ultimo grafico rappresenta la derivata seconda.

La pendenza della rampa è inversamente proporzionale al grado di rumore presente sul bordo.

Facendo la derivata prima della rampa, come si evince dalla figura 4.12, questa è maggiore di zero in tutti dell'intervallo in cui la funzione rampa non

è costante. Mentre facendo la derivata seconda il segno è positivo solo nell'istante in cui inizia la transizione scuro-chiaro e negativo alla fine. La derivata prima ci fornisce un'informazione approssimativa ma robusta della zona dell'immagine in cui è presente il bordo nell'immagine, mentre la derivata del secondo ordine fornisce informazioni puntuali che però sono maggiormente influenzate dal rumore.

Gradiente Le derivate prime di un'immagine digitale sono basate su un'approssimazione del gradiente 2D. Il gradiente di un'immagine $f(x, y)$ nella posizione (x, y) è definito come un vettore:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

L'ampiezza del gradiente è dato da:

$$\nabla f = \text{mag}(\nabla f) = \sqrt{G_x^2 + G_y^2}$$

Questa quantità fornisce il massimo indice di incremento di $f(x, y)$ per unità nella direzione di ∇f .

La direzione del gradiente dell'immagine è una grandezza molto importante, perché è sempre perpendicolare alla direzione del bordo.

$$\alpha(x, y) = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Figura 4.13: Regione 3×3 dell'immagine in cui i valori z rappresentano il livello di grigio.

Il calcolo del gradiente di un'immagine consiste nell'ottenere le derivate parziali $\frac{\partial f}{\partial x}$ e $\frac{\partial f}{\partial y}$ per ogni pixel. Considerando una porzione d'immagine come

quella in figura 4.13 è possibile calcolarne la derivata tramite l'operatore *Robert cross-gradient*:

$$G_x = (z_9 - z_5)$$

$$G_y = (z_8 - z_6)$$

L'utilizzo di una maschera 2×2 non restituisce buoni risultati per il fatto che non si ha un chiaro centro della maschera. Per questo motivo, per i calcoli delle derivate parziali, vengono utilizzate maschere che occupano porzioni dell'immagine 3×3 :

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

In questa formula, la differenza tra la terza e la prima riga della regione 3×3 dell'immagine, approssima la derivata rispetto all'asse delle x . Analogamente la differenza fra la terza e la prima colonna approssimano la derivata rispetto alla direzione y . Le formule sopra che permettono di calcolare le derivate parziali dell'immagine prendono il nome di operatore di *Prewitt*.

Un'altro modo per ottenere le derivate parziali è utilizzando l'operatore *Sobel*:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

L'utilizzo della maschera di Prewitt è più semplice di quella di Sobel, anche se soffre il rumore presente nell'immagine, al contrario la maschera Sobel raddoppiando i primi vicini (*4-connectivity*) introduce uno *smooth* per dare più importanza al pixel centrale. Si noti che l'assenza nelle formule sopra di alcuni coefficienti dei livelli di grigio appartenenti al vicinato, significa che questi danno risposta nulla nelle area in cui si ha un livello di grigio costante.

Gli operatori Prewitt e Sober permettono quindi di ricavare le derivate parziali dall'immagine. Il calcolo del gradiente come mostrato, essendo formato da una radice quadrata, a volte non è di facile utilizzo nel sistema di riferimento discreto dell'immagini. Per questo motivo il gradiente dell'immagine viene approssimato nel seguente modo:

$$\nabla f \approx |G_x| + |G_y|$$

Canny edge detector Lo scopo del *edge detection* in generale è la riduzione significativa dell'ammontare di dati in un'immagine, preservando però le proprietà strutturali. Per raggiungere questo obiettivo esistono molti algoritmi ma uno in particolare è stato sviluppato da *John F. Canny* nel 1986. Anche se datato l'algoritmo *Canny edge detector* è lo standard nel rilevamento dei bordi, utilizzato sia a livello industriale che di ricerca.

L'algoritmo di Canny si basa su cinque criteri per rilevare nel modo migliore i bordi.

1. **Riduzione rumore:** il primo step consiste nella massima riduzione del rumore presente nell'immagine e per far ciò si ricorre all'utilizzo di un filtro Gaussiano. Il filtro Gaussiano ha il compito di fare uno *smooth* dell'immagine eliminando il rumore in prossimità dei bordi e appiattendolo i contrasti pochi rilevanti. Il filtro viene calcolato tramite il principio della convoluzione¹:

$$g(i, j) = \frac{1}{2\pi\sigma^2} \sum_{m=1} \sum_{n=1} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma^2}\right)} f(i-m, j-n)$$

2. **Calcolo gradiente:** in questo secondo step viene calcolato il gradiente utilizzando l'operatore Sobel:

$$|G| = |G_x| + |G_y|$$

3. **Direzione del bordo:** la direzione del bordo viene calcolata utilizzando il gradiente in x e y ricavati al punto precedente:

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

4. **Risoluzione della direzione del bordo:** una volta a conoscenza della direzione del bordo, la si relaziona con la direzione reale che il bordo ha nell'immagine. Un'immagine digitale è una griglia di valori quindi un pixel appartenente al bordo può raggiungere un suo vicino, sempre facente parte del bordo, muovendosi al massimo in quattro direzioni:

- 0 gradi: direzione orizzontale;
- 45 gradi: lungo la diagonale positiva;
- 90 gradi: direzione verticale;
- 135 gradi: lungo la diagonale negativa.

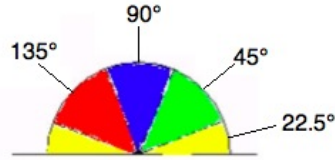


Figura 4.14: Semicerchio diviso in 5 parti che rappresentano i range di appartenenza degli angoli dei bordi.

Si può quindi ricavare la direzione digitale del bordo secondo che l'angolo calcolato per ognuno appartenga ad uno degli spicchi del semicerchio di figura 4.14.

- giallo: $-0.4142 \leq \tan\theta < 0.4142$
- verde: $0.4142 \leq \tan\theta < 2.4142$
- blu: $|\tan\theta| \geq 2.4142$
- rosso: $-2.4142 \leq \tan\theta < -0.4142$

5. **Soppressione falsi positivi:** dopo essere venuti a conoscenza di tutte le direzioni dei bordi, si applica il *nonmaximun suppression*. Questa tecnica è usata per scansionare il bordo lungo la sua direzione ed eliminare tutti quei pixel che non sono considerati tali.

6. **Edge linking** l'output dell'algoritmo è un'immagine binaria in cui tutto è background ad eccezione dei bordi, foreground, per far ciò è necessario fare il trshesholing con una soglia T . La decisione del valore di soglia non è un'operazione semplice considerando che l'immagine risultato del punto precedente non è ancora esente da possibili errore.

La scelta di un valore di soglia troppo basso andrebbe anche ad accettare quei massimi causati dall'errore, mentre una soglia alta potrebbe andare ad eliminare anche massimi reali. Per risolvere questo problema viene fatto una thresholding utilizzando due soglie.

- sogliatura bassa t_l ;
- sogliature alta t_h .

¹la convoluzione è un'operazione tra due funzioni che genera una terza funzione:
 $g(x) = \int_{-\infty}^{\infty} f(\tau)h(x - \tau)dx \quad g = f * h.$

Generalmente $t_h \approx 2t_l$.

Questa tecnica a doppia sogliatura è chiamata *hysteresis thresholding*:

$$t_0 \geq \begin{cases} |\nabla f(x, y)| \geq t_1 & \text{è sicuramente un bordo} \\ |\nabla f(x, y)| \leq t_1 & \text{potrebbe essere un bordo, a seconda del contesto} \\ |\nabla f(x, y)| < t_0 & \text{non è un bordo} \end{cases}$$

Il *potrebbe essere* significa che il pixel è considerato un bordo se i suoi vicini sono bordi *strong*, ovvero con un livello di grigio elevato.

4.3 Rappresentazione

L'output proveniente da una prima fase di segmentazione, ottenuta ad esempio con gli algoritmi presentati nel paragrafo precedente, è un'immagine in cui le regioni d'interesse sono separate dal background. Questa separazione però è solo a livello immagine e quindi fine a se stessa, bisogna infatti rappresentare e descrivere le regioni d'interesse in modo che possono essere nuovamente processate.

La rappresentazione può essere fatta in funzione delle caratteristiche esterne, i bordi, oppure rispetto a quelle interne, i pixel compresi tra i bordi. Dopo essersi focalizzati sul tipo di rappresentazione è necessario descrivere la regione d'interesse. Per esempio, una regione può essere *rappresentata* dal proprio bordo, e il bordo può essere *descritto* rispetto alle sue caratteristiche, come la lunghezza, la direzione e il vicinato.

Quando si sceglie di utilizzare una rappresentazione esterna allora l'obiettivo è l'estrazione delle caratteristiche di forme, al contrario se si utilizza una rappresentazione interna, allora ci si concentra su colore e texture.

4.3.1 Chain code

Il *Chain Code* è utilizzato per rappresentare i contorni tramite una sequenza connessa di segmenti con un data lunghezza e direzione. Questa rappresentazione si basa sul principio della *4-connectivity* o *8-connectivity* mostrata in figura 4.15.

L'algoritmo Chain Code riceve generalmente in ingresso il risultato dell'edge detection e il verso di navigazione, orario o antiorario. L'immagine in ingresso, è quindi una griglia di valori, dove trovato un pixel appartenente al bordo, lo si etichetta con una delle direzioni di figura 4.15 a seconda del verso di percorrenza. Successivamente si naviga il bordo etichettando ogni

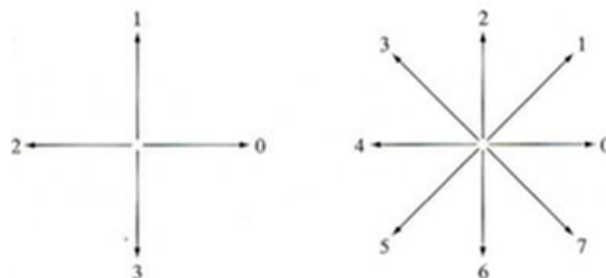
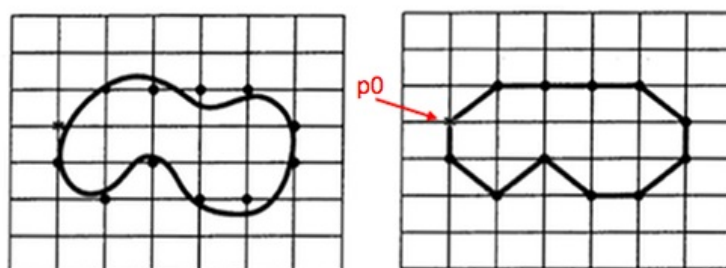


Figura 4.15: a) 4-connectivity. b) 8-connectivity.

pixel fino a ritornare nel punto di partenza. Il risultato del chain code di un'immagine è in funzione del pixel di partenza e del verso scelto.

Il Chain Code presenta però dei difetti: i tempi di esecuzione aumentano a seconda della dimensioni dell'immagine e anche una piccolo disturbo lungo il bordo a causa del rumore può cambiare il risultato. Per arginare i problemi sopra generalmente si procede ad un sottocampionamento dell'immagine a scapito del contenuto informativo del bordo.



Rappresentazione: 100076543432

Figura 4.16: a) Risultato del edge detection. b) Sottocampionamento di a. c) Risultato.

Come rappresentato in figura 4.16.c il risultato dell'algoritmo di Chain Code è una stringa, con questo procedimento infatti la regione d'interesse di un'immagine 2D viene mappata come un'informazione in una sola dimensione, riducendo così la complessità.

Un potenziamento del Chain Code può essere realizzato in modo tale che runtime l'algoritmo riesce a raccogliere anche informazioni sulla descrizione della regione d'interesse.

4.4 Interpretazione

L'interpretazione dell'immagine è l'ultimo livello del image processing, in questo step vengono utilizzare tecniche e algoritmi che hanno il compito di utilizzare le informazioni descrittive estrapolate dalla segmentazione per interpretare e quindi riconoscere gli oggetti nell'immagini.

Quest'approccio chiamato *pattern recognition* è diviso in due aree principali: *decisione teorica* e *strutturale*. La prima categoria lavora con descrittori quantitativi come la lunghezza, l'area e la struttura. La seconda categoria invece tratta descrittori qualitativi.

Un pattern è un disposizione di descrittori che definiscono certe caratteristiche. Le tre disposizione comuni usate nella pratica sono vettori, utilizzati nel pattern recognition quantitativo, e stringhe e alberi utilizzati in quello qualitativo.

4.4.1 Analisi sintattica

L'analisi sintattica dei descrittori di una regione d'interesse fornisce una metodologia per gestire problemi di riconoscimento strutturale. L'idea che è alla base del *syntactic pattern recognition* è la definizione di un insieme di pattern primitivi (grammatica), un insieme di regole che li gestiscono e un riconoscitore (automa) la cui struttura è determinata dai due insiemi precedenti.

Grammatiche I primi che formalizzarono i concetti di grammatica furono Chomsky e Bar-Hillel a cavallo degli anni '50 e '60, con l'obiettivo di formalizzare una struttura matematica che aiutasse a capire le proprietà basilari del linguaggio naturale.

Il concetto di struttura grammaticale della frase è nato facendo il *parsing* di una semplice frase inglese, "The girl walks gracefully" [4]. Nella frase citata "the girl" è la parte sostantiva utilizzato come soggetto, e "walks gracefully" è la parte verbale composta dal verbo "walks" e l'avverbio "gracefully".

Si può definire la produzione della frase nel seguente modo:

```

<frase> -> <parte sostantiva> <parte verbale>
<parte sostantiva> -> <articolo> <sostantivo>
<parte verbale> -> <verbo> <avverbio>
<articolo> -> The
<sostantivo> -> girl
<verbo> -> walks
<avverbio> -> gracefully

```

Il simbolo “ $->$ ” significa che la proprietà a sinistra può essere riscritta nel modo di destra.

Una grammatica G è tupla $G = (V_N, V_T, P, S)$:

- V_T è un termine terminale, come ad esempio articolo, sostantivo dell'esempio sopra;
- V_N è un simbolo non terminale, come frase;
- P è un'insieme finito di regole di riscrittura o produzioni;
- $S \in V_N$ è il simbolo di partenza della frase.

Le grammatiche sono caratterizzate dalla forma delle loro produzioni. Nel image processing sono di particolare interesse le grammatiche *regolari* (tipo 3) e quelle *context free* (tipo 2). Le prime possono essere risolte tramite l'utilizzo di un automa a stati finiti, le seconde tramite gli algoritmi stack base.

L'idea alla base dell'analisi sintattica dell'immagine processing è che dato in ingresso una stringa rappresentante la regione d'interesse, la generazione di una grammatica ad hoc e il relativo parser, permette di fare interpretazione sulla regione d'interesse.

Automa come riconoscitore di stringhe Effettuare il pattern recognition utilizzando una grammatica di tipo 3 può essere risolto tramite l'utilizzo di un automa a stati finiti. Data in input una stringa, l'automa a stati finiti è in grado di riconoscere se questa appartiene ad un pattern (linguaggio) o meno. Un automa a stati finiti è definito da una tupla:

$$A_f = (Q, \Sigma, \delta, q_0)$$

dove:

- Q è un insieme finito e non vuoto di stati;
- Σ è un alfabeto finito;
- δ è l'insieme delle transizioni, $Q \times \Sigma$;
- q_0 è lo stato iniziale;

Capitolo 5

Algoritmo

L'algoritmo che è stato sviluppato durante il lavoro di tesi è il risultato della risoluzione dei sotto problemi del image processing descritti nel capitolo precedente. La sequenzialità che c'è tra i vari blocchi dell'algoritmo determina una stretta relazione fra essi, in cui un errore di valutazione o una scelta sbagliata si ripercuote sul risultato dell'intero lavoro. Nello sviluppo di questa tesi è stato utilizzato un *modello a spirale* per valutare quantitativamente quali scelte influivano maggiormente sul risultato dell'algoritmo. L'obiettivo è quindi stato quello di riuscire ad arrivare ad una rapida prototipazione che desse dei risultati e successivamente fare quanti giri di spirale erano necessari per affinare i risultati dei vari blocchi.

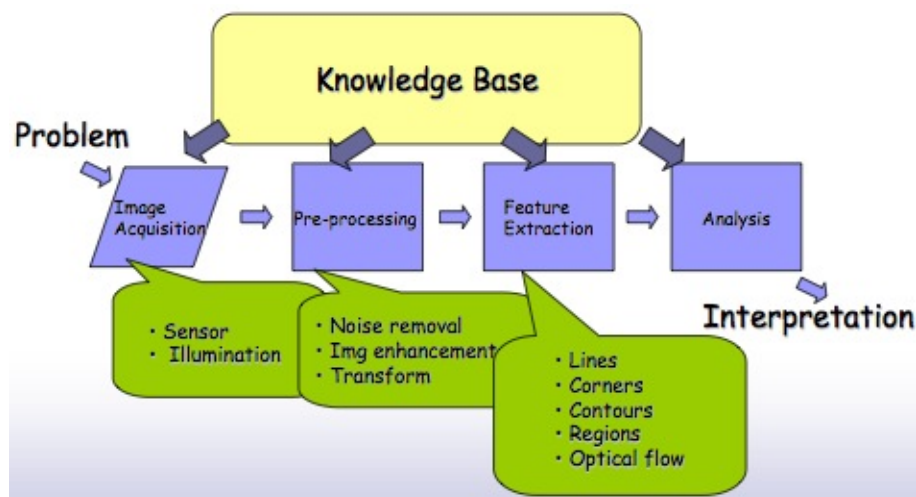


Figura 5.1: Algoritmo di risoluzione di un problema di image processing

L'obiettivo della tesi è la realizzazione di un algoritmo che preso in ingresso un'immagine di un pezzo meccanico acquisito dalla MTL, interpreti le forme geometriche dell'oggetto e restituisca in uscita l'insieme delle grandezze misurabili sul pezzo.

L'algoritmo ha avuto una prima prototipazione fatta nell'ambiente di sviluppo *MatLab* che mette a disposizione un tool apposito per fare elaborazione dell'immagine. Dopo essere giunti al primo risultato, per affinare i vari blocchi, l'algoritmo è stato riscritto in *C++* sfruttando le potenzialità della libreria *OpenCV* che è sviluppata da *Intel* e attualmente distribuita con licenza open source BSD.

5.1 Progettazione

Come già ampiamente presentato nei capitoli precedenti l'ambiente in cui questa tesi è stata sviluppata è nell'ambito della *Machine Vision*. La risoluzione di un problema di MV comporta la risoluzione di quattro sottoproblemi rappresentati in figura 5.1. La progettazione dell'algoritmo quindi consiste nello sviluppo di ogni sotto problema, tenendo però in considerazione che l'obiettivo del lavoro di tesi non è quello di fare misura accurate sui pezzi ma bensì interpretarne le forme geometriche.

1. *Image denoise*: riduzione del rumore in prossimità dei bordi dell'immagine, in modo tale che la determinazione dei bordi sia il più possibile vicina alla realtà.
2. *Boundary extraction*: effettuare la segmentazione dell'immagine definendo così la regione d'interesse e quindi l'estrazione puntuale dei bordi del pezzo.
3. *Rappresentazione e descrizione*: prendendo in ingresso il risultato del punto precedente analizzare le continuità e discontinuità nell'immagine.
4. *Interpretazione*: definire un modello per ogni forma geometrica che si intende interpretare e trovare in quali zone dell'immagine questi pattern sono presenti, con l'obiettivo di restituire in output l'insieme delle forme geometriche composte da cui è formato il pezzo dato in input alla MTL.

5.2 Riduzione del rumore

La fase di *denoise* dell'immagine è una delle fasi più delicate nel image processing e quindi anche nello sviluppo dell'algoritmo. La buona riuscita del

denoise dell'immagine, essendo alla radice, determina l'accuratezza dei risultati. In questa fase la scelta di tecniche di riduzione del rumore troppo "pesanti" potrebbe comportare il rischio che particolari dell'immagine vengano persi, al contrario se troppo "leggere" ci si può imbattere poi nelle fasi successive nella presenza di falsi positivi.

Per fare una riduzione del rumore ad hoc in modo tale che vada a rimuovere ciò che è effettivamente *noise* e lasci invariato ciò che appartiene al pezzo sarebbe necessario utilizzare tecniche di *denoise conservativo*. Questo tipo di tecniche però necessitano della consapevolezza della dimensione e variabilità del pixel dell'immagine rispetto alla scena reale acquisita e per risolvere il problema dimensionale del pixel è necessario svolgere un lavoro di analisi dell'errore generato sull'immagine acquisita rispetto alla scena reale e ricorrere quindi ad una calibrazione, diversa da quella presente oggi, che sia indipendente da oggetti e forme acquisite.

Tra i problemi elencati nel capitolo tre, fedelmente allo scopo della tesi che consiste nell'interpretare forme geometriche e non di misurarle, non sono stati presi in considerazione i problemi riguardanti la dimensione e variabilità del pixel, considerando corretti i valori dimensionali forniti dalla calibrazione esistente, ma sono stati considerati tutti quegli errori in prossimità dei bordi del pezzo generati in fase di acquisizione, da fenomeni di riflettività e dalla presenza di artefatti sull'oggetto. In prima battuta osservando l'immagine di un pezzo su un monitor non viene evidenziato alcun tipo di rumore sui bordi a causa dal sotto campionamento che viene fornito a video dell'immagine, ma facendo lo zoom dell'immagine in prossimità dei bordi si sono riscontrati i problemi presentati in 3.1.3 e 3.1.4.

Come primo passo si è cercato di ridurre le imperfezioni presenti ai bordi tramite un'operazione di erosione (vedi 4.1.2) utilizzando la classica maschera *flat* 5×5 mostrata sotto.

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Con la maschera sopra si voleva andare a "spegnere" tutti quei pixel il cui vicinato non era completamente connesso, però in alcune situazioni le immagini presentavano errori dovuti a granelli di polvere di dimensione maggiore

della maschera, rendendo così inefficiente questa operazione. Una soluzione poteva essere quella di aumentare la dimensione della maschera ma si correva il rischio di andare ad eliminare dei particolari del pezzo. Dopo questi primi test è stata cambiata la maschera, utilizzandone una che rappresentava una circonferenza di raggio 5. I risultati sono stati migliori dei precedenti ma nelle zone dell'immagine che rappresentavano i piani del pezzo, il rumore causato dalla polvere andava a sommarsi a problemi di riflessione rendendo così quasi nullo il denoise fatto.

Facendo altri test è stato trovato un buon compromesso tra l'attenuazione del rumore dovuto alla polvere e agli effetti di riflessione facendo uno *smoothing* morfologico. Questa tecnica infatti va ad attenuare in prossimità dei bordi il contrasto dovuto agli effetti di riflessione e allo stesso tempo ad "appiattire" i minimi introdotti dalla polvere.

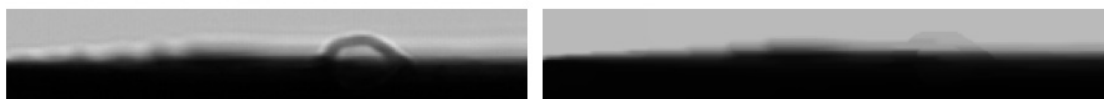


Figura 5.2: a) Piano del pezzo corrotto da effetti di riflessione e polvere. b) Risultato del denoise tramite smoothing morfologico.

L'approccio dello smoothing morfologico è stato utilizzato nella prima fase di prototipazione ma successivamente dopo essere arrivati alla produzione dei primi risultati, sono stati testati e realizzati altri metodi più intelligenti per fare il denoise dell'immagine. Nelle fasi successive specialmente nella prossima il problema dovuto al *blur* sui bordi dell'immagine non permette un'accurata estrazione dei bordi. Per questo ragione la soluzione è stata quella di effettuare una binarizzazione dell'immagine per attenuare gli effetti causati da fenomeni di riflessione in prossimità dei bordi.

Considerando che gran parte dell'immagine è occupata dall'ombra del pezzo e stato scelto di utilizzare l'algoritmo di Otsu per determinare un valore di soglia ottimale.

Dopo aver fatto il thresholding dell'immagine e quindi riducendo in maniera consistente la presenza dell'effetto "fumo" sui bordi sono state applicate operazioni di erosione con maschere fatte ad hoc che definiscono la conformazione del vicinato che i pixel in prossimità dei bordi devono avere, per eliminare gli artefatti causati dalla polvere. Le maschere utilizzare contengono sia informazioni riguardanti il foreground che il background, infatti per questo motivo possiamo considerare l'approccio utilizzato come se fossero delle operazioni di *hit & miss*.



Figura 5.3: a) Piano del pezzo corrotto da effetti di riflessione e polvere. b) Risultato del denoise dopo aver effettuato il thresholding dell'immagine con elementi strutturati ad hoc.

Con le maschere di figura 5.4 si va a definire la conformazione del vicinato che ogni pixel in prossimità di piani dell'immagine deve avere. Infatti considerando 5.4.a e la sua origine in rosso, tutti i pixel a sinistra, destra e sottostanti devono appartenere al pezzo, mentre quelli superiori devono fare parte del background.

```

0 0 0 0 0 0 0 1 1 1 1 1 1
0 0 0 0 0 0 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 0 0 0 0 0 0

```

Figura 5.4: Maschera ad hoc per eliminare il rumore sui piani orizzontali: a) Piano rivolto verso l'alto. b) Piano rivolto verso il basso.

Con lo stesso principio appena spiegato le maschere di figura 5.5 sono state utilizzate per erodere gli errori sui bordi verticali dell'immagine del pezzo.

```

0 0 1 1 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 1 0 0

```

Figura 5.5: Maschera ad hoc per eliminare il rumore sui bordi verticali: a) Bordo rivolto verso sinistra. b) Bordo rivolto verso il destra.

5.3 Estrazione dei bordi

Successivamente alla fase di riduzione del rumore, si è proceduto con la segmentazione dell'immagine per estrarre la regione d'interesse ovvero i bordi del pezzo. Restando sempre fedeli all'obiettivo della tesi in cui non è necessaria un'accuratezza di misura è stato possibile utilizzare tecniche di estrazione dei bordi approssimative, nel senso che viene individuato il pixel che al suo interno conterrà il reale bordo del pezzo.

Per estrarre il bordo dell'immagine è stato utilizzato l'algoritmo *Canny edge detector*, quindi è stata derivata l'immagine rispetto a x e y tramite le maschere di *Sobel* di dimensione 3×3 , successivamente calcolato il gradiente e utilizzati i criteri esposti in 4.2.2 per rendere il risultato dell'estrazione robusto.

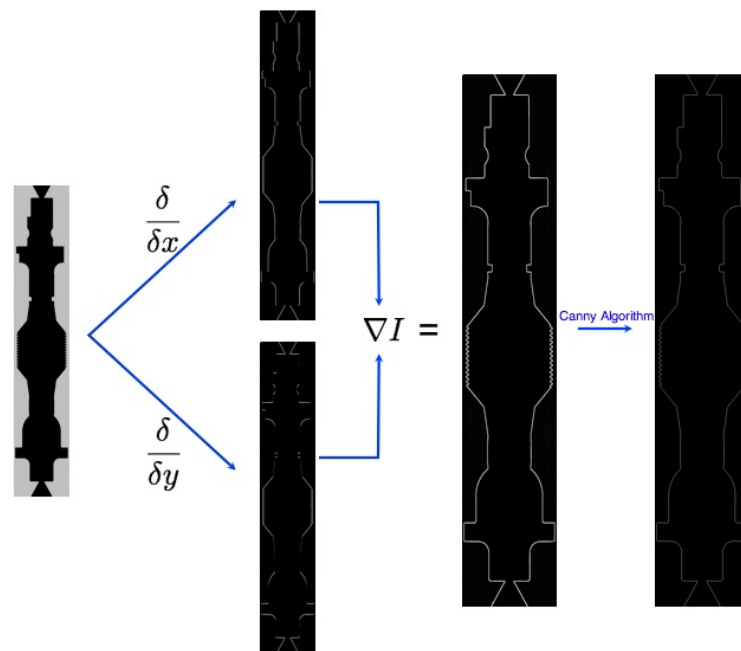


Figura 5.6: Algoritmo Canny Edge Detector.

Dopo le operazioni di estrazione si ha a disposizione un'immagine a due livelli in cui tutto è background e solamente i bordi del pezzo appartengono al foreground. Prima di effettuare operazioni di rappresentazione dell'immagine è necessario che i bordi dell'immagine siano *thin* ovvero non devono essere mai presenti due pixel contigui in x alla stessa quota y . Per far ciò è stata operata una operazione morfologica di *thinning*.

5.4 Rappresentazione

Nel image processing la fase di segmentazione è seguita dalla fase di rappresentazione dell'immagine, quindi in questa tesi consiste nell'estrapolare le informazioni sui bordi. Questo step dell'algoritmo riceve in ingresso un problema in due dimensioni e restituisce un problema in una sola dimensione. Per ridurre la dimensione del problema il primo approccio è stato l'utilizzo dell'algoritmo di *Chain Code* che come spiegato in 4.3.1 naviga i bordi dell'immagine e restituisce una stringa ordinata contenente la codifica di ogni pixel appartenente al bordo.

L'idea che sta alla base di questa fase dell'algoritmo è di utilizzare il concetto del Chain Code *8-connectivity*, ovvero navigare e codificare il bordo dell'immagine, e salvare in un'apposita struttura dati solo quei pixel che determinano una variazione della continuità del bordo. Ad esempio se si naviga un lato verticale verrà tenuta nota solo del pixel e dalla rispettiva direzione, che corrisponde al vertice iniziale del segmento, poi quando navigando il bordo si riscontra un cambio di direzione sarà tenuta nota del pixel corrispondente come vertice di un nuovo segmento, e così via.

Il risultato di questo algoritmo è la rappresentazione del pezzo come una insieme di vettori, i quali parametri vengono salvati in apposite strutture dati che ne determinano il punto di applicazione mentre la direzione e il verso sono implicite nel valore di codifica del Chain Code.

```
CornerPoint {
    x;
    y;
    code;
}
```

L'algoritmo di Chain Code spiegato in 4.3.1 ha lo scopo di andare a codificare la direzione del passo che bisogna compiere da un pixel appartenente ad bordo per approdare al successivo facente sempre parte del foreground. Per questo si considera una delle codifiche 4.15 e si ricerca il successore in un intorno 3×3 in modo tale che la distanza fra i due sia unitaria. Però per evitare problemi introdotti da falsi positivi causati dal rumore che non è stato possibile eliminare, l'algoritmo di Chain Code è stato sviluppato in modo tale che la ricerca del vicino di un pixel possa avvenire o con una maschera classica 3×3 , oppure con una 5×5 o una 7×7 .

Sempre secondo il principio dello sviluppo a spirale i test compiuti hanno dimostrato che il Chain Code con una maschera 5×5 garantisce risultati migliori all'algoritmo sviluppato.

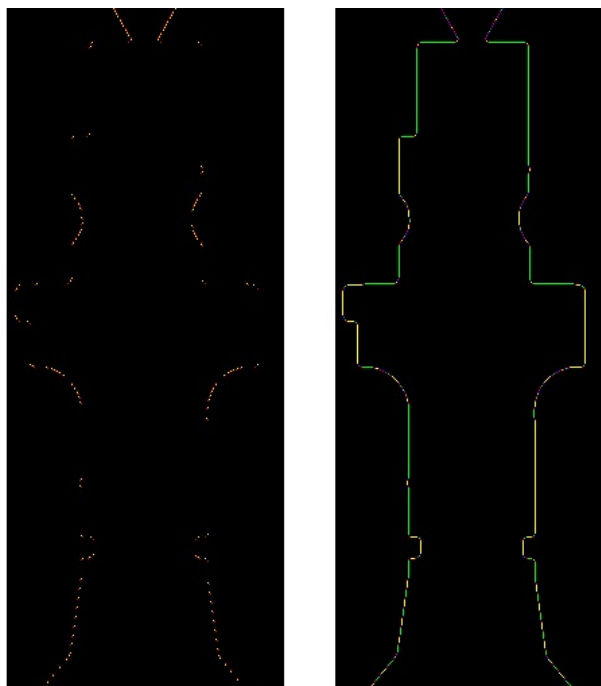


Figura 5.7: a) Rappresentazione grafica del risultato del Chain Code (discontinuità). b) Rappresentazione grafica del risultato complementare del Chain Code (continuità).

5.5 Interpretazione

Le funzionalità principali messe a disposizione dal software MTL, presentato in 2.1.5, devono permettere all'utente di calcolare la dimensione dei diametri del pezzo, la distanza tra piani del pezzo, gli angoli formati tra lati obliqui e l'asse pezzo e il raggio degli archi di circonferenza presenti nel pezzo. L'obiettivo dell'algoritmo sviluppato deve quindi indicare all'utente le zone dell'immagine in cui è possibile calcolare una delle seguenti forme geometriche:

- diametri;
- piani (spallamenti);
- lati obliqui;
- archi di circonferenza.

Grazie agli step precedenti dell'algoritmo in questa fase il problema bidimensionale in ingresso è stato trasformato in un problema mono dimensionale fornendo la rappresentazione dell'immagine nel seguente modo:

$Image = \{P_1(x_1, y_1, code_1), P_2(x_2, y_2, code_2), \dots, P_n(x_n, y_n, code_n)\}$ tale che:

$$code_i = \{0, \dots, 7\}, \quad i = 1, \dots, n, \quad n = \# \text{ punti di discontinuità}$$

In questo modo l'immagine può essere vista come un insieme di vettori, in cui è fornito in modo esplicito il punto di applicazione e in modo implicito, tramite la codifica di figura 4.15.b, direzione, verso e modulo. Grazie al codice si riesce già ad estrapolare delle informazioni sui lati verticali e orizzontali del pezzo, infatti i punti di discontinuità codificati con codice 0 o 4 rappresentano i bordi orizzontali del pezzo, mentre quelli con valori 2 o 6 determinano i lati verticali. La determinazione di lati obliqui e archi di circonferenza però non è di facile derivazione, perché a causa del campionamento, questi nell'immagine digitale sono formati dall'unione di piccoli segmenti.

L'idea che è stata utilizzata per interpretare le forme geometriche note di cui è composto un pezzo è: *scansionare tutti i vettori dell'insieme e raggruppare tra loro quei vettori che assieme determinano una forma composta.*

Per far ciò è stata utilizzata la tecnica di *syntactic pattern recognition* definendo per ognuna delle seguenti forme geometriche:

- lati orizzontali,
- lati verticali,
- lati obliqui,
- archi di circonferenza,

le regole di produzione che determinano di quali vettori e con che regolarità questi possono essere riscritti.

La generazione di regole di produzione per lati verticali e orizzontali è di facile fattura, infatti essendo questi segmenti non composti possono essere riscritti nel seguente modo:

Lato_orizzontale -> 0 | 4

Lato_verticale -> 2 | 6

in cui si vuole esplicitare che durante la scansione della rappresentazione dell'immagine quando si incontra uno dei codici sopra, si etichetta il segmento che va dal punto di applicazione del vettore al punto di applicazione del successivo, come un lato orizzontale o verticale del pezzo.

Mentre la generazione delle regole di riscrittura per le altre due tipologie di forme non è banale. Infatti a causa del campionamento queste forme in un'immagine digitale sono composte da più vettori dell'insieme.

Studiando le immagini di figura 5.7 sono state determinati due tipi di regolarità che caratterizzano il campionamento di lati obliqui del pezzo.

1. se il lato obliquo forma con l'asse verticale del pezzo un angolo maggiore o uguale a 45° , figura 5.8, il lato obliquo è campionato come una sequenza che alterna un vettore obliquo di modulo maggiore o uguale a 1, un vettore lineare di modulo unitario e un'altro vettore obliquo codificato come il precedente di modulo maggiore o uguale a uno.

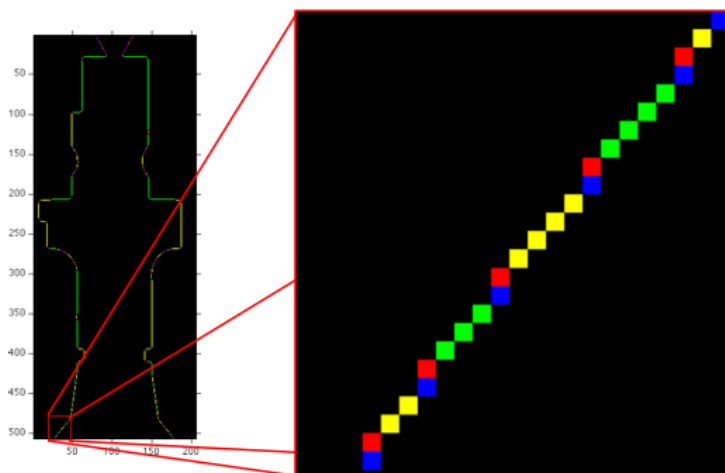


Figura 5.8: Campionamento di un lato obliquo che forma con l'asse verticale del pezzo un angolo maggiore o uguale a 45°

2. se il lato obliquo forma con l'asse verticale del pezzo un angolo inferiore a 45° , figura 5.9, il lato obliquo è campionato come una sequenza che

alterna un vettore obliquo di modulo unitario, un vettore lineare di modulo maggiore di uno e un'altro vettore obliquo codificato come il precedente di modulo unitario.

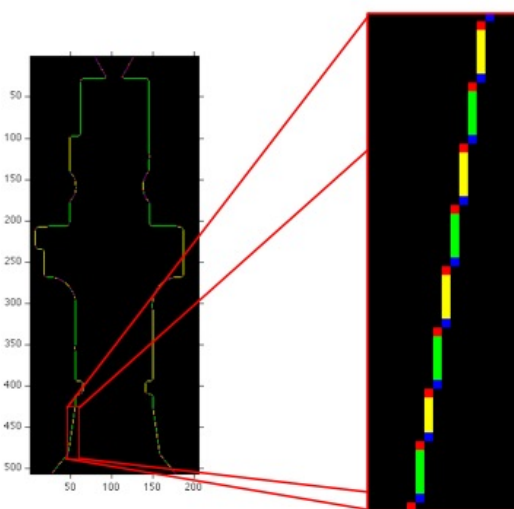


Figura 5.9: Campionamento di un lato obliquo che forma con l'asse verticale del pezzo un angolo minore a 45°

Grazie alla regolarità riscontrata nel campionamento dei lati obliqui è possibile quindi definire queste regole di produzione:

Lato_obliquo1 -> Lato_Obliquo * n Lato_Lineare * 1 Lato_Obliquo * m

Lato_obliquo2 -> Lato_Obliquo * 1 Lato_Lineare * n Lato_Obliquo * 1

Lato_Lineare -> 0 | 2 | 4 | 6

Lato_Obliquo -> 1 | 3 | 5 | 7

in cui

- il simbolo “*” indica il modulo del vettore,
- $m, n \geq 1$.

Per la generazione della regola di produzione dell'arco di circonferenza, sempre dallo studio del campionamento è stato riscontrato che questa forma geometrica non rispetta nessuna delle regolarità esposte sopra, ma bensì le alterna in modo del tutto aleatorio:

Arco \rightarrow Lato_Obliquo * n Lato_Lineare * m Lato_Obliquo * q

$$m, n, q \geq 1.$$

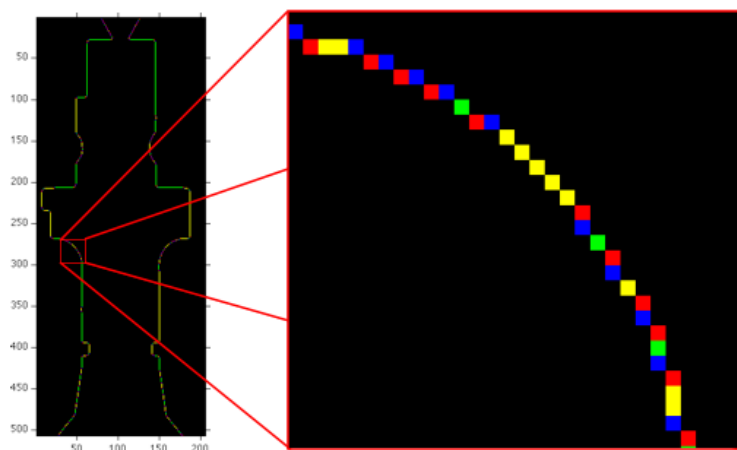


Figura 5.10: Campionamento di un arco di circonferenza.

Grazie all'analisi sintattica della rappresentazione dell'immagine si è in possesso di tutte le forme geometriche semplici che compongono il pezzo scansionato. Le forme interpretate soddisfano l'obiettivo al 75%, infatti oltre ai segmenti obliqui e gli archi di circonferenza, i segmenti orizzontali corrispondono ai piani dell'immagine. Con l'analisi sintattica non è possibile determinare in prima battuta i diametri essendo a loro volta in un'immagine una forma composta. Per estrapolare i diametri dall'immagine sono stati semplicemente accoppiati segmenti verticali che occupavano almeno per un pixel la stessa quota y .

Capitolo 6

Risultati sperimentali

Lo sviluppo dell'algoritmo, come ampiamente spiegato nel capitolo precedente, ha avuto un approccio modulare, in cui ogni modulo risolve un sotto problema del image processing. L'immagine che il primo blocco riceve in ingresso è il risultato dell'acquisizione di un pezzo meccanico effettuata tramite il sistema di visione industriale presentato nel capitolo 2. Quindi tutte le scelte, riguardanti le tecniche e i metodi di denoise dell'immagine, sono state prese in riferimento ai problemi esposti nel capitolo 3.

Mentre l'approccio risolutivo che è stato impiegato negli altri moduli fa sì che questi siano riutilizzabili, perché non è stato direttamente vincolato ai problemi emersi nello studio del sistema MTL.

L'algoritmo una volta ricevuta in ingresso un'immagine restituisce in uscita l'elenco delle forme riscontrate nella figura, dandone la porzione dell'immagine che le contiene. Per dare un risultato visivo dei risultati ottenuti sono state evidenziate con colori diversi, le porzioni dell'immagine in ingresso che contengono le forme geometriche interpretate:

- *verde*: inizio o la fine del pezzo;
- *rosso*: diametri;
- *giallo*: piani (spallamenti);
- *azzurro*: lati obliqui;
- *blu*: archi di circonferenza.

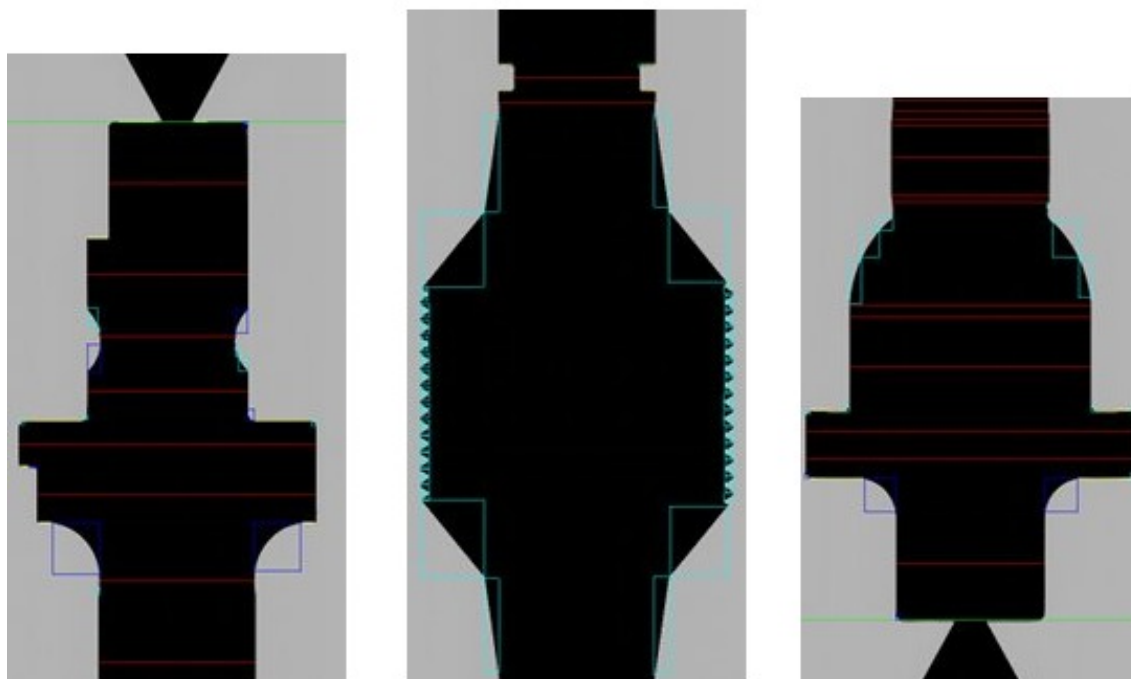


Figura 6.1: Risultato dell'interpretazione del pezzo 1.

6.1 Valutazione dei risultati

La valutazione dell'algoritmo è stata compiuta rispetto alla realtà oggettiva (detta anche *ground truth*) rilevata da un'analisi visiva dei pezzi. Quindi per determinare l'efficacia dell'algoritmo è necessario confrontare per ogni forma rilevata visivamente se questa è stata interpretata correttamente non solo per tipo di forma ma anche per porzione d'immagine occupata.

Per far ciò è necessario definire la seguente tabella:

Tabella 6.1: Tabella di valutazione

Verità / Interpretazione	Forma	No Forma
Forma	Vero positivo	Falso negativo
No Forma	Falso positivo	Vero negativo

Nelle righe della tabella è indicata la verità che scaturisce dall'analisi visiva del pezzo, in cui nella prima riga definiamo la forma d'interesse del pezzo, mentre nella seconda riga si indica tutto ciò che non è d'interesse.

Nelle colonne invece è indicata l'interpretazione che l'algoritmo fornisce o meno alla forma d'interesse.

Le celle della tabella sono etichettate nel seguente modo:

- *Vero positivo*: significa che l'algoritmo interpreta correttamente la forma d'interesse nella porzione dell'immagine in cui si trova.
- *Falso negativo*: l'algoritmo non determina dove è presente, la forma d'interesse.
- *Falso positivo*: l'algoritmo determina una forma d'interesse dove non è presente.
- *Vero negativo*: l'algoritmo non determina forme d'interesse ove queste non siano realmente presenti.

Per forme di non interesse si intende l'insieme di tutte le forme dei segmenti che compongono il pezzo che sono contate manualmente.

La valutazione dell'algoritmo è tanto maggiore tanto quanto il valore nelle celle di *falso positivo* e *falso negativo* sia vicino allo zero.

L'algoritmo è stato valutato per entrambi i benchmark mostrati in 2.2 rispetto alle quattro forme geometriche trattate durante lo svolgimento della tesi. Le informazioni per ciascun pezzo sono presentate in 6.2 e 6.3.

Tabella 6.2: Forme d'interesse del pezzo 1.

Forme	Numero
Forme segmenti	139
Diametri	15
Piani	18
Lati Obliqui	70
Archi	10

Tabella 6.3: Forme d'interesse del pezzo 2.

Forme	Numero
Forme segmenti	120
Diametri	12
Piani	16
Lati Obliqui	63
Archi	0

6.1.1 Diametri

La valutazione dell'interpretazione dei diametri è mostrata nella tabelle 6.4 e 6.5, in entrambe si evince che l'algoritmo riconosce correttamente tutti i *diametri* presenti realmente sul pezzo, ma interpreta anche delle zone dell'immagine in cui non sono presenti dei diametri. Si noti che nella valutazione

dei diametri essendo questi forme composti da due lati verticali paralleli tra loro, le forme di non interesse sono tutti i segmenti che compongono il pezzo.

Tabella 6.4: Tabella di valutazione diametri pezzo 1

Verità / Interpretazione	Diametri	No Diametri
Diametri	15	0
No Diametri	7	132

Tabella 6.5: Tabella di valutazione diametri pezzo 2

Verità / Interpretazione	Diametri	No Diametri
Diametri	12	0
No Diametri	4	116

La generazione di *falsi positivi* è causata dal rumore presente sui bordi che non si è riusciti ad eliminare. Succede infatti che un reale lato verticale viene campionato come l'unione di più di uno che differiscono tra loro di un pixel. Quindi l'interprete, come mostrato in figura 6.4, invece che trovare in una porzione dell'immagine un solo diametro ne trova più, in cui la loro ampiezza differisce di qualche pixel.

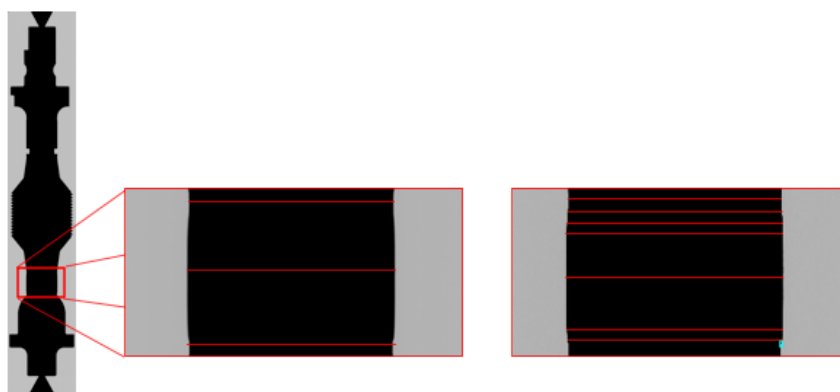


Figura 6.2: Porzione d'immagine in cui a causa del rumore presente sui bordi l'algoritmo invece di rilevare i tre diametri raffigurati nella figura di mezzo, ne rileva quattro in più.

6.1.2 Piani (Spallamenti)

Anche per i piani sono state effettuate le stesse operazioni sopra tenendo però in considerazione che ora le forme di non interesse sono tutti i segmenti che compongono il pezzo esclusi i segmenti orizzontali che di fatto rappresentano i piani dell'immagine.

Tabella 6.6: Tabella di valutazione piani pezzo 1

Verità / Interpretazione	Piani	No Piani
Piani	18	0
No Piani	3	111

Tabella 6.7: Tabella di valutazione piani pezzo 2

Verità / Interpretazione	Piani	No Piani
Piani	16	0
No Piani	4	108

Dai risultati delle tabelle 6.6 e 6.7 anche in questo caso tutti i piani dell'immagine vengono trovati e come sopra l'algoritmo trova falsi positivi in corrispondenza di piani affetti da un elevato rumore facendo sì che venga rilevato più di un piano.

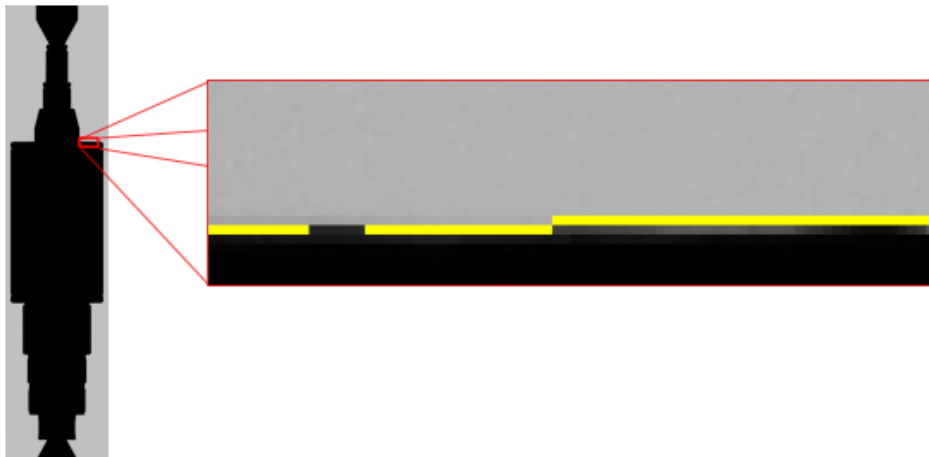


Figura 6.3: Un piano del pezzo in cui a causa del rumore presente sul bordo, l'algoritmo invece di rilevare un unico piano ne rileva due in più.

6.1.3 Lati obliqui

L'efficacia dell'algoritmo nell'interpretazione dei lati obliqui non si è dimostrata robusta come le precedenti, infatti lati obliqui di dimensioni ridotte come quelli che compongono da esempio un filetto o uno smusso non sempre vengono interpretati correttamente. Come sopra nella tabella le non forme, ovvero i veri negativi, consistono nell'insieme di tutte i segmenti che compongono il pezzo ad eccezione di quelli obliqui.

Tabella 6.8: Tabella di valutazione lati obliqui pezzo 1

Verità / Interpretazione	Lati Obliqui	No Lati Obliqui
Lati Obliqui	66	4
No Lati Obliqui	8	69

Tabella 6.9: Tabella di valutazione lati obliqui pezzo 2

Verità / Interpretazione	Lati Obliqui	No Lati Obliqui
Lati Obliqui	45	18
No Lati Obliqui	0	57

6.1.4 Archi di circonferenza

L'interpretazione degli archi di circonferenza si è dimostrata la meno robusta delle quattro a causa della non regolarità del campionamento di questa forma.

Tabella 6.10: Tabella di valutazione archi di circonferenza pezzo 1

Verità / Interpretazione	Archi	No Archi
Archi	6	4
No Archi	4	125

Tabella 6.11: Tabella di valutazione archi di circonferenza pezzo 2

Verità / Interpretazione	Archi	No Archi
Archi	0	0
No Archi	0	120

Come è mostrato in figura 6.5 un arco di circonferenza con un raggio di dimensioni non marginali a causa del campionamento è riconosciuto dall'analizzatore sintattico come la sequenza di segmenti obliqui.

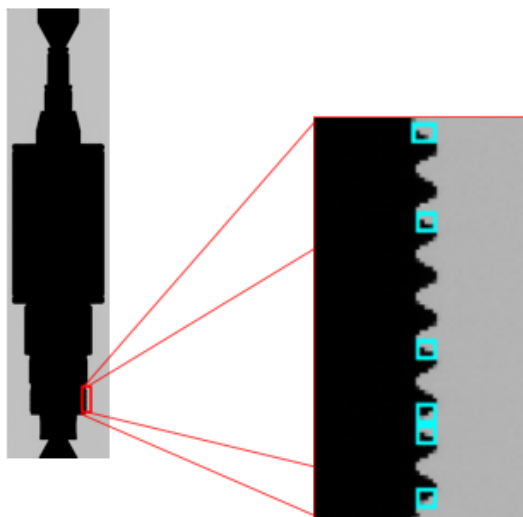


Figura 6.4: Zona dell'immagine in cui è presente un filetto dove l'algoritmo non interpreta tutti i lati obliqui perché di dimensioni ridotte

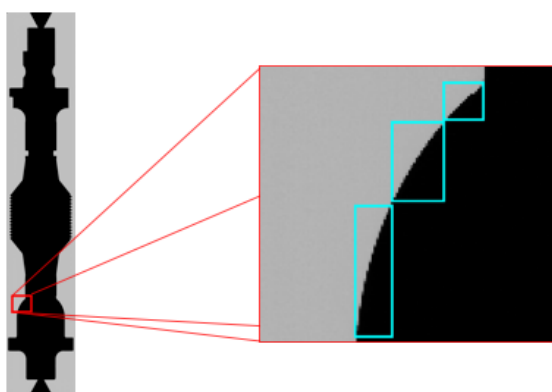


Figura 6.5: Arco di circonferenza del pezzo che non viene interpretato correttamente.

Conclusioni e sviluppi futuri

Nel corso della presente tesi è stato progettato ed implementato un algoritmo in grado di interpretare forme geometriche note, in un'immagine acquisita utilizzando un macchinario di produzione industriale.

Nella prima fase dell'algoritmo, che consiste nella riduzione del rumore nell'immagine derivante dall'acquisizione, sono state prese decisioni e utilizzate tecniche dedicate proprio all'attenuazione degli errori introdotti dal dispositivo di visione utilizzato. Mentre nella successiva fase di elaborazione dell'immagine che idealmente riceve in ingresso un'immagine priva di rumore, le scelte e le tecniche utilizzate sono indipendenti dall'acquisizione. Questo significa che l'algoritmo, con il solo adattamento della parte di *denoising*, potrebbe essere utilizzato per l'interpretazione di forme geometriche note su altri sistemi di visione industriale. L'analisi dei risultati ottenuti su pezzi meccanici usati come *benchmark* consente di verificare che tutte le forme che ci si proponeva di identificare sono state trovate e caratterizzate.

Tra i problemi evidenziati e discussi nel capitolo 3, di comune accordo con l'azienda, non sono stati affrontati né quelli riguardanti dimensione e variabilità del pixel né quelli concernenti il rilevamento del profilo dell'immagine tramite l'analisi del gradiente.

Uno sviluppo futuro dell'algoritmo potrebbe essere quello di andare a studiare in modo approfondito i problemi che si verificano durante la fase di acquisizione e, tramite un'analisi statistica, ricavare forma e dimensione di ogni pixel dell'immagine. Dopo aver ricavato queste informazioni, è possibile utilizzare tecniche di *denoising* conservativo in grado di rimuovere nell'immagine ciò che è veramente errore, senza alterare il reale profilo del pezzo acquisito.

Un secondo interessante sviluppo riguarda lo studio del gradiente dell'immagine. In questa tesi, dato che l'obiettivo non era misurare ma interpretare, sono state utilizzate tecniche di estrazione del bordo che restituivano solamente il pixel nel quale effettivamente è presente il reale bordo del pezzo.

Tramite lo studio del gradiente è possibile ottenere una precisione sub-pixel ovvero sapere in quale posizione, all'interno del pixel, si trova il bordo reale. Avendo a disposizione anche la reale posizione del bordo, l'algoritmo sarebbe in grado senza modifiche sostanziali, di misurare in modo accurato le grandezze fisiche delle forme geometriche note interpretate nell'immagine.

Bibliografia

- [1] A. Bevilacqua, A. Gherardi, and L. Carozza. An automatic system for the real-time characterization of vehicle headlamp beams exploiting image analysis. *IEEE Transactions on Instrumentation and Measurement*, 99(10), October.
- [2] A. Bevilacqua, A. Gherardi, and L. Carozza. An industrial vision-based technology system for the automatic test of vehicle beams. *IEEE International Symposium on Industrial Electronics*, July 2009.
- [3] A. Bevilacqua, L. D. Stefano, and A. Lanza. A simple self-calibration method to infer a non-parametric model of the imaging system noise. *Proceedings of the IEEE Workshop on Motion and Video Computing*.
- [4] K. S. Fu. *Syntactic Pattern Recognition and Application*.
- [5] H. Golnabi and A. Asandpour. Design and application of industrial machine vision systems. *Robotics and Computer-Integrated Manufacturing*, (23), 2007.
- [6] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Upper Saddle River, New Jersey 07458, second edition.
- [7] M. Graves and B. G. Batchelor. *Machine vision for the inspection of natural products*. Springer, 2003.
- [8] J. C. on Guides for Metrology (JCGM). *International vocabulary of metrology — basic and general concepts and associated terms (3 ed.)*, 2008.
- [9] P. M. Pietikäinen and P. J. Rönning. Machine vision and intelligent systems. *INFOTECH OULU*, 2002.
- [10] D. Schonfeld. Optimal structuring elements for the morphological pattern restoration of binary images. *IEE Transactions on pattern analysis and machine intelligence*, 16(6), June 1994.

- [11] R. Szeliski. *Computer Vision: Algorithms and Applications*. AM draft, 2008.
- [12] L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEE Transactions on Image Processing*, 2(2):176 201, 1993.
- [13] N. Zuech. *Understanding and Applying Machine Vision*. Marcel Dekker, Inc, second edition, 2000.

Ringraziamenti

Il primo ringraziamento non può non andare al Professore Alessandro Bevilacqua che per la seconda volta mi ha seguito nella preparazione della Tesi, generando in me un profondo interesse nell'elaborazione d'immagini e dandomi una base nell'approccio risolutivo di questo tipo di problemi. Ha colto il mio desiderio di "concretezza", permettendomi per la seconda volta, di lavorare all'interno di un'azienda facendo sì che i risultati del mio lavoro avessero risvolti immediati su macchinari di tipo industriale. Non voglio dimenticarmi anche che lui indirettamente fu tra quelli che al termine della laurea triennale mi diedero gli stimoli per dedicare altri due anni alla mia formazione professionale, e grazie a lui oggi ho avuto l'occasione di firmare un contratto di lavoro ancora prima della laurea. Grazie Prof.

Un altro ringraziamento va all'azienda Vici & Co che mi ha ospitato per sei mesi mettendomi a disposizione una scrivania e i mezzi che mi hanno permesso di realizzare questa tesi. Un grazie va anche a tutto il personale dell'azienda e in particolare al Ing. Stefano Andreini, mio tutor, che con immensa pazienza mi ha fornito tutte le informazioni necessarie alla comprensione di tutti i fattori riguardanti il macchinario utilizzato.

Il ringraziamento più grande va a chi mi è stato vicino lungo tutta la mia carriera universitaria: i miei genitori, Nerio e Marisa, mio fratello Maicol e soprattutto la mia amata Mery, i suoi genitori, Oriano e Maurizia, i cognati Andrea e Michela e infine la piccola Vanessa venuta alla luce 11 giorni prima dell'atteso traguardo. Dato che non sono uno di tante parole preferisco ringraziare tutti singolarmente.

Un ultimo ringraziamento va agli autori dei testi presi a riferimento e non espressamente citati, per lo svolgimento di questo lavoro.¹

¹[3], [1], [2], [5], [9], [11], [13], [12], [10]