

POLITECNICO DI MILANO

FACOLTA' DI INGEGNERIA
SCUOLA DI INGEGNERIA DELL'INFORMAZIONE
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA ELETTRONICA



SISTEMA DI VISIONE ARTIFICIALE PER IL RICONOSCIMENTO, LA
MISURA E LA VETTORIALIZZAZIONE AUTOMATICA DI POLIGONI
IRREGOLARI IGNOTI

Relatore: Chiar.mo Prof. Franco Zappa

Tesi di Laurea di:

Giovanni Genovese

Matr. 740052

Anno Accademico 2013 - 2014.

*... “Alla macchina, il lavoro della macchina,
all’uomo, il brivido di nuove scoperte
e di nuove creazioni” ...
[Kazuma Tateishi]*

RINGRAZIAMENTI

Per primo, non per ordine di importanza, vorrei ringraziare il monaco William of Ockham, che mi ha permesso di guardare alla vita con semplicità.

L'Imperatore e filosofo Marco Aurelio, che mi ha permesso di vedere la vita con ottimismo, tenacia, apprezzamento alla vita stessa ed alle piccole cose.

Fabrizio de Andrè mi ha accompagnato durante la stesura di questo elaborato.

Il product engineer dei sistemi di visione artificiale della Omron Electronics S.P.A., mi ha guidato nello sviluppo di questo studio, e di questo elaborato.

I miei genitori, che oltre a darmi i natali, hanno permesso la realizzazione di questo enorme progetto che si chiama "vita".

Mio fratello, Francesco, unico punto di riferimento in diversi momenti e contesti, nei momenti brutti come nei momenti belli.

Sommario

INDICE DELLE FIGURE:	6
ELENCO DELLE TABELLE:	8
ABSTRACT	9
NOTE PER LA LETTURA DELL'ELABORATO	11
INTRODUZIONE	12
1.1 Specifiche di progetto	14
1.2 I sistemi di visione artificiale nel campo dell'automazione industriale	14
1.3 "Spese, guadagni, ricavi,..". La scelta dell'investimento	16
1.4 I sistemi di illuminazione.....	17
1.5 Le ottiche	20
1.6 I sensori di immagine.....	22
1.7 Gli algoritmi di analisi ed elaborazione delle immagini	24
2. IL SISTEMA DI VISIONE ARTIFICIALE – LO STATO DELL'ARTE.	27
2.1 Il sistema di visione Omron FH, analisi delle caratteristiche e degli strumenti fondamentali per lo sviluppo dell'applicazione	28
2.1.1 CPU	29
2.1.2 Le telecamere	30
2.1.3 L'ambiente di sviluppo	31
2.2 Gli algoritmi di elaborazione utilizzati.....	33
2.2.1 Camera image Input FH	34
2.2.2 Precise Calibration	35
2.2.3 Advanced Filter	36
2.2.4 Gravity and Area	38
2.2.5 Edge Position	39
2.2.6 Scan Edge Position	39
2.2.7 Intersection	40
2.2.8 Color data	40
2.2.9 Macro e Calculation	41
2.3 Il setup di acquisizione, misura e test.....	42
SEZIONE II	45
3. ELABORAZIONE DELLE IMMAGINI	45
3.1 Il metodo "dei raggi" – settore applicativo	45

3.2	Il metodo “dei raggi”- sviluppo dell’algoritmo di elaborazione.....	46	
3.2.1	“Start”	48	
3.2.2	“Gravity and Area”	51	
3.2.3	“Scan Edge Position, Edge position”	53	
3.2.4	Meccanismo di trasformazione degli angoli	54	
3.2.5	Storage	55	
3.3	Macro: Angle analisys & Intersection until convergence.....	55	
3.3.1	Adattamento delle regioni dei sensori.	63	
4.	IL METODO “AD INSEGUIMENTO”.....	64	
4.1	Settore applicativo	64	
4.2	Sviluppo dell’algoritmo di elaborazione	65	
4.2.1	“Start”	67	
4.2.2	Multi scan edge position	68	
4.3	Macro: Angle analisys, Intersection, lost point, o continuous following	73	
4.3.1	Color Data	74	
4.3.2	“Continuous following”	77	
4.4	Il metodo ad inseguimento - analisi di immagini nel mondo della micro e nano elettronica		77
4.5	Misure e controlli sui lati dei poligoni rilevati.....	78	
4.6	Applicazione al mondo della micro e nano elettronica	79	
5.	CONCLUSIONI E SVILUPPI FUTURI.....	80	
	BIBLIOGRAFIA.....	81	
	APPENDICE A. TECHNOLOGY OVERVIEW PATTERN MATCHING ALGORITHM SHAPE SEARCH III.....	82	
	APPENDICE B. FH-3050 VISION SYSTEM INSTRUCTION SHEET ...	82	
	APPENDICE C. FH-SC04 CAMERA INSTRUCTION SHEET	82	
	APPENDICE D. VISION PLATFORM FH DATA SHEET.....	82	

Indice delle figure:

- Figura 1. Macro struttura di un sistema di visione artificiale
- Figura 2. Sistema di illuminazione *Dome*.
- Figura 3. Sezione di un sistema di illuminazione *Dome*.
- Figura 4. Acquisizioni con illuminazione diretta e con sistema *Dome*.
- Figura 5. Sezione del sistema di illuminazione *Low-angle*.
- Figura 6. Acquisizione con sistema d'illuminazione *Low Angle*.
- Figura 7. Effetto della distorsione ottica.
- Figura 8. Grafico per la scelta della focale dell'ottica
- Figura 9. Profondità di campo.
- Figura 10. Sezione di un sensore CMOS APS.
- Figura 11. Bayer Pattern Color Filter Array.
- Figura 12. Breve storia del "pattern matching".
- Figura 13. Esempi di campioni reali sui quali vengono effettuati controlli e riconoscimento delle informazioni criptate.
- Figura 14. Connessione del sistema di visione artificiale.
- Figura 15. Modelli unità centrale di elaborazione CPU FH.
- Figura 16. Modelli telecamere con sensore CMOS
- Figura 17. Caratteristiche geometriche ed elettriche dei sensori.
- Figura 18. Interfaccia utente di sviluppo del sistema di visione FH-3050
- Figura 19. Interfaccia operatore implementata su sistema di visione FH-3050.
- Figura 20. Interfaccia "Edit flow".
- Figura 21. Meccanismo di ottimizzazione del range dinamico *multi slope*.
- Figura 22. Impostazione dei parametri di impulso per lo *strobe*.
- Figura 23. Immagine acquisita ed immagine elaborata dall'algoritmo *Precise Calibration*.
- Figura 24. Errori residui di calibrazione.
- Figura 25. Interfaccia di scelta del tipo di filtro.
- Figura 26. Applicazione di filtri su campioni reali.
- Figura 27. Interfaccia di scelta delle gamme cromatiche.
- Figura 28. Flusso di impostazione e di azione dell'algoritmo *Gravity and Area*.
- Figura 29. Regione di elaborazione dell'algoritmo *Edge Position*.
- Figura 30. Esempio di applicazione dell'algoritmo *Scan Edge Position*.
- Figura 31. Applicazione dell'algoritmo *Intersection*.
- Figura 32. Applicazione dell'algoritmo *Color Data*.
- Figura 33. Interfaccia di programmazione *Macro*.
- Figura 33A. Setup del sistema di misura.
- Figura 33B. Setup del sistema con retro illuminazione.
- Figura 34. Foto del setup di misura.
- Figura 35. Esempio di visualizzazione dei risultati dell'algoritmo di pattern matching Shape search III.
- Fig. 36. Diagramma di flusso del metodo "dei raggi".
- Figura 37. Diagramma di flusso implementato all'interno del software applicativo FH ver.5.20.
- Figura 38. Interfaccia per l'impostazione dei dati di partenza dell'algoritmo dei raggi.
- Figura 39. Layout dell'algoritmo *2.Data Exchange*.
- Figura 40. Effetto del filtraggio. Immagine originale.
- Figura 41. Effetto del filtraggio. Immagine filtrata.
- Figura 42. Effetto del filtraggio. Immagine estratta.
- Figura 43. Interfaccia di selezione della regione di ispezione dell'algoritmo *Gravity and Area*.

- Figura 44. Particolare di impostazione dell' algoritmo "*edge position*".
- Figura 45. Particolare di impostazione dell' algoritmo "*scan edge position*".
- Figura 46. Impostazione delle regioni di scansione.
- Figura 47. Regioni di impostazione oltre limite.
- Figura 48. Algoritmo *Intersection*.
- Figura 49. Impostazione regione dello scan edge position 0 o 1 dell' Intersection. Caso 1: angolo memorizzato $< 90^\circ$.
- Figura 50. Impostazione regione dello scan edge position 0 o 1 dell' Intersection. Caso 2: angolo memorizzato $> 90^\circ$.
- Figura 51. Impostazione regione dello scan edge position 0 o 1 dell' Intersection. Caso 3: angolo memorizzato $> 90^\circ$ - emisfero inferiore della regione di misura.
- Figura 52. Visualizzazione dei risultati del metodo dei raggi.
- Figura 53. Particolare del funzionamento del metodo dei raggi.
- Figura 54. Particolare di un dispositivo con 8 livelli di "*metal*".
- Figura 55. Diagramma di flusso del metodo ad inseguimento.
- Figura 56. Diagramma di flusso implementato all' interno del software applicativo FH ver.5.20 per il metodo ad inseguimento.
- Figura 57. *Scan Edge Position* di partenza.
- Figura 58. *Scan Edge Position* numero 1.
- Figura 59. *Scan Edge Position* numero 2.
- Figura 60. *Scan Edge Position* numero 3.
- Figura 61. *Scan Edge Position* numero 4.
- Figura 62. *Scan Edge Position* numero 5.
- Figura 63. *Scan Edge Position* numero 6.
- Figura 64. *Scan Edge Position* numero 7.
- Figura 65. *Scan Edge Position* numero 8.
- Figura 66. Risultato dell' esecuzione dei primi 8 *scan edge position*.
- Figura 67. Particolare della misura di colore - sfondo.
- Figura 68. Particolare della misura di colore - poligono.
- Figura 69. Visualizzazione dei risultati del metodo ad inseguimento.
- Figura 70. Particolare di *etching* eccessivo.
- Figura 71. Analisi immagine *STI*.
- Figura 72. Particolare - analisi immagine *STI*.

Elenco delle tabelle:

Tabella 1. Estratto del data sheet della CPU FH-3050

Tabella 2. Valori ammissibili di “*shutter speed*”.

Tabella 3. External Reference Tables – Calculation.

Tabella 4. External Reference Tables – Gravity and Area.

Tabella 5. External Reference Tables – *Edge Position*.

Tabella 6. External Reference Tables – *Scan Edge Position*.

Tabella 7. External reference tables – *Intersection*.

Abstract

Il presente elaborato è frutto di uno studio interdisciplinare per lo sviluppo e la realizzazione di un sistema di visione artificiale atto al riconoscimento e la digitalizzazione automatica di poligoni irregolari.

Nel settore automazione industriale ed automatizzazione dei processi di misura, al fine della minimizzazione degli scarti di produzione, nasce l'esigenza di un sistema automatico in grado di rilevare, i vertici ed i valori dei lati, dei vertici e degli angoli di scarti di produzione sotto forma di poligoni.

Eseguito uno studio dello stato dell'arte tra i sistemi di visione artificiale attualmente in commercio, troviamo per tecnologia e caratteristiche fisiche e prestazionali, diverse soluzioni in grado di acquisire immagini con alto range dinamico, indispensabile per rendere il sistema immune al cambiamento di condizioni di illuminazione ambientale. Tra gli elementi di elaborazione delle immagini troviamo invece, ai fini del nostro studio, degli algoritmi basati sul riconoscimento dei bordi di oggetti. Combinando tali algoritmi possiamo ottenere automaticamente le caratteristiche salienti delle regioni delle immagini sulle quali vengono effettuate le misurazioni.

La parte maggiore di tale lavoro consiste quindi nello sviluppo di una procedura automatica, un nuovo algoritmo di misura, capace di rilevare in automatico i vertici di un poligono ignoto. Sono state quindi tracciate due strade, due filosofie di architettura del sistema e del software.

Il sistema sviluppato è anche in grado di effettuare la verifica di allineamento con degli assi di riferimento, fornire le misure rilevate secondo il sistema di riferimento desiderato (mm, inches, etc.), verifica di linearità dei poligoni, e la misura puntuale dei lati al fine del rilevamento di conformità, ed il baricentro. Viene utilizzato un sistema di illuminazione a led bianchi, per ogni poligono verrà acquisita un'immagine, a seguire una pre-elaborazione ed un filtraggio multiplo, e l'immagine verrà così elaborata utilizzando i nuovi algoritmi sviluppati. Il sistema progettato e sviluppato nasce quindi per un'esigenza nel settore dell'automazione industriale, potrebbe trovare campo di applicazione nella misura ed il rilevamento automatico delle caratteristiche dei dispositivi elettronici acquisiti mediante acquisizione delle immagini con metodo SEM (Scanning Electron Microscopy), TEM (Transmission Electron Microscopy), etc..

Parole chiave: visione artificiale, intelligenza artificiale, algoritmi elaborazione immagine

Abstract (english)

This, is a multidisciplinary work, developing in order to give a solution to a concrete and real customer. He needs finding a artificial vision solution system, capable auto recognizing corner and vertices from unknown shapes travelling on a conveyor. The aim of the customer is to recycle these shapes, in order to make a "puzzle" using these waste pieces of production. Then the reason pushing in the developing is economic but also ecologic. The first because the customer goes to save money, the second is obvious. Then the job order comes from the Industrial automation world, and we are talking about artificial vision systems into this world.

First work done, the scouting of the products within what offer the world market. And we choose the best vision system following our criteria, like High Dynamic Range in terms of acquisition, useful in order to give more efficiency to the vision system. Looking at the processing vision Items, or algorithms, we can find a lot of instruments, and some of them do the *edge position* job.

Working into a prefixed region, there is an internal scan where matrix images values are read in order to find the first derivate. Combining what we found in terms of instruments, we develop two

new algorithms, with two different working philosophies. We will also explain what good and bad of the two possible paths, and we will also speak about other measures required on the specs, like shape sides alignments, output parameters in mm or inches, area, etc.

In order to build the best setup, we used a white bar LED light model, we put some filtering and calibration into the image pre elaboration phase, after that we applied separately the two developed algorithms in order to tests their performances, and reporting them respect to what we evaluate before.

We are looking now at the future, and we think that we can for sure apply these developed techniques in order to make directly measuring into SEM (Scanning Electron Microscopy), and TEM acquired images (Transmission Electron Microscopy).

Keywords: machine vision systems, artificial intelligence, image algorithms

Note per la lettura dell'elaborato

Nella stesura del seguente elaborato, si è volutamente adottato un linguaggio semplice per un'agevole lettura. Non si sono volutamente riportate formule se non quelle ottenute dallo sviluppo dell'applicazione. Per le relazioni che legano fenomeni fisici trattati, si rimanda ai testi di riferimento, verranno sottolineate di volta in volta invece le dipendenze. Tuttavia, la chiave della semplicità risiede nel possedere già delle conoscenze di base in merito al mondo della visione artificiale, dell'elettronica, della fisica di base, ed elemento basilare, la fantasia. Infatti per lo studio di problematiche di visione artificiale, spesso il compito più arduo è quello di riuscire a compiere un'attenta analisi di un fenomeno. Non verranno altresì riportati tutti i passi del codice elaborato, gli schemi, ed altri documenti prodotti per lo sviluppo del sistema, in prima istanza per esigenze di spazio. Verranno spesso fatte delle citazioni in merito a modalità di funzionamento dei sistemi utilizzati di cui non verrà approfondito l'aspetto tecnico della scelta del percorso intrapreso per l'implementazione. Nonostante tutto, l'elaborato è di facile comprensione se si intraprende la lettura gradualmente e progressivamente, mostrando e rispettando un momento di riflessione durante l'introduzione di nuovi elementi. È frutto di un viaggio, di un percorso, e per questo si è voluto dar maggior peso al percorso intrapreso per la scelta delle metodologie adottate, piuttosto che ai risultati, che ovviamente, sono stati ottenuti come previsto. Si fornirà talvolta, l'idea da cui è partita la decisione nell'intraprendere un percorso, o delle osservazioni che hanno portato all'utilizzo di uno strumento a discapito di un altro. Nelle speranze dell'autore, se non altro la possibilità che tale elaborato sia letto e compreso, che la lettura sortisca un effetto piacevole, capace di indurre curiosità, dubbi, e che sia analizzato con criticità, con uno sguardo anche ai diversi punti di vista che non sono stati trattati nella stesura dell'elaborato.

Introduzione

Noi, esseri umani abitanti del pianeta terra, esseri dotati di intelligenza, di infinite qualità come difetti, e soprattutto, di organi sensoriali, viviamo un'epoca in cui il risparmio energetico, il tema dell'ecologia, la salvaguardia dell'ambiente, sono tra i temi di maggiore dibattito ed interesse. Il lavoro che verrà qui presentato, nasce da una idea, da un'esigenza, di un'azienda operante nel settore dell'automazione industriale. Per questioni di segretezza e di accordi di riservatezza, non verrà citato il nome dell'azienda in questione, né verranno mostrate immagini della stessa. Tale esigenza, è stata recepita da un'azienda che opera nel campo della produzione di prodotti di automazione industriale, in qualità di consulente e fornitore dei prodotti, e della soluzione. Questo studio è stato quindi sviluppato per assolvere questa prima esigenza, ma dall'incipit, dal dovere, nasce poi una fantasia dell'autore. Ed infatti i metodi e gli strumenti sviluppati ed implementati, potranno di sicuro trovare collocazione in settori extra contestuali a questo progetto. Verrà poi mostrata la strada percorsa per lo sviluppo che assolve alle specifiche in oggetto, e verranno poi fornite le coordinate per uno sviluppo intelligente e per una possibile contestualizzazione ad altri mondi. Un'esigenza, un risparmio, economia. Seppur vogliamo considerare anche un aspetto che, ahimè, non verrà trattato in questo tomo, è quello dell'ecologia. Poiché se nella fantasia dell'autore risiede la possibilità del riutilizzo di una tecnica sviluppata, nel sogno dello stesso risplende l'immagine che quanto sviluppato possa sul serio esser di supporto per una maggiore attenzione al pianeta che ci ospita.

Nell'ottica di riutilizzare sfridi provenienti da processi di produzione, viene progettato un setup di acquisizione e misura, e vengono sviluppati gli strumenti inerenti l'elaborazione e l'interfaccia utente.

Verrà quindi trattato nel primo capitolo una panoramica su ciò che compone l'universo della visione artificiale, le peculiarità. Nel secondo capitolo verranno poi definite le specifiche del sistema utilizzato, dando spazio alle peculiari funzionalità che hanno permesso lo sviluppo di questo lavoro. Terzo e quarto capitolo: il sistema di visione artificiale implementato, il setup di misura, gli algoritmi di elaborazione sviluppati. Per finire, uno sguardo al futuro.

Volgendo lo sguardo invece, all'aspetto economico, la giustificazione di un investimento atto al fine della ricomposizione automatica di sfridi, va necessariamente comparata al risparmio che ne deriva. Questo elaborato, è stato concepito nell'ottica di fornire uno strumento di ausilio per l'analisi, lo studio, e la progettazione di un sistema di visione artificiale in ambito industriale. Per questo motivo viene altresì riportato un paragrafo con cenni su l'aspetto economico della tecnologia adottata, ma non solo. Infatti, un ingegnere è portato non soltanto ad applicare dei metodi nell'ottica di trovare soluzioni, ma è spesso portato a trovare delle soluzioni che prima di tutto siano economiche.

Delineato lo spazio di lettura. La prima scelta e la prima strada intrapresa è ovviamente ricaduta su un sistema di visione artificiale. Viste le specifiche, in prima battuta si è passati ad immaginare il sistema di illuminazione ideale, la scelta della risoluzione del sensore, e si è poi passati allo studio, allo sviluppo del programma di elaborazione.

Nel settore di utilizzo, i sistemi di illuminazione giocano un ruolo fondamentale nella riuscita di una applicazione di visione artificiale. Basti pensare che con una telecamera catturiamo immagini, e su di essa vengono effettuate delle elaborazioni. Il caso peggiore è l'analisi che si serve di misure del colore. Il colore di una immagine, varia infatti in base alla sorgente luminosa, ciò che capteremo è il riflesso della sorgente luminosa sulla scena. In ambito industriale infatti, oltremodo per la maggior parte dei controlli di visione effettuati, vi sono delle condizioni di illuminazione che variano nell'arco della giornata. È questo un primo collo di bottiglia per i sistemi di visione. Nell'ottica poi di riuscire a carpire dei particolari riguardo un oggetto, esiste poi una folta libreria di sistemi di illuminazione che variano per geometria e colore. Definito il sistema di illuminazione si è passati poi alla scelta della risoluzione secondo la precisione richiesta. In merito alla precisione richiesta, visto che non si trattavano di semplici requisiti su dimensioni spaziali, si è passati allo studio della correlazione tra deviazioni d'angolo e spostamento in pixel, per la scelta ottima della risoluzione.

Dallo studio fisico, a quello elettronico, si è passati a quello informatico, ed infine a quello “telecomunicazionista”. Come dire, uno studio completo di *fisica per l’informazione*.

Per lo sviluppo degli algoritmi si è dovuti un po’ attingere a delle conoscenze di informatica, elaborazione, ed intelligenza artificiale. Di grande utilità sono risultate tutte le materie di studio del corso di laurea intrapreso. In ultima battuta sono stati analizzati anche aspetti derivanti dallo studio di metodi matematici per l’ingegneria.

Oltre al lavoro svolto per lo sviluppo vero e proprio di questo lavoro, si è voluti fornire in questa sede un elaborato scelto appositamente per descrivere i punti chiave del percorso. Tutto nell’ottica di fornire in questo elaborato non la soluzione pronta, ma la ricetta, la mappa, la strada da percorrere. Adottando tale presunzione, si offre il percorso che porta a qualsiasi traguardo, basta chiamarlo ogni volta con un nome diverso.

Buona lettura.

1.1 Specifiche di progetto

Lo studio che ha dato spunto per lo sviluppo dell'applicazione di visione artificiale oggetto di tale elaborato, deriva da una azienda operante nel settore dell'automazione industriale, di cui non verranno citati i dettagli per motivi di segretezza e di accordi di riservatezza stipulati. Verranno ad ogni modo definite le caratteristiche e le specifiche, punto di partenza per lo sviluppo, ma non verranno mostrate immagini relative all'applicazione reale, bensì immagini provenienti da simulazioni su di un setup di misurazione reale in laboratorio. L'esigenza per la realizzazione di questo progetto è di natura economica, ma si può trovare anche una sistemazione morale nello studio di applicazioni derivanti dal sistema di visione sviluppato. Questo perché, nell'ottica del riciclo di sfridi di produzione, lo studio in oggetto tratta di un sistema che automaticamente rilevi i vertici degli sfridi, che si possono presentare sotto forma di poligoni irregolari, depositi su un nastro trasportatore che li dirige da una parte di un macchinario di produzione, ad una parte terminale della linea di produzione. Una volta rilevati i vertici, forniti sotto forma di coordinate coerenti con il sistema di riferimento scelto (nel nostro caso è stata prevista la possibilità di ottenere simultaneamente i risultati in millimetri ed in pollici), verranno altresì effettuate delle misure e dei rilievi sui poligoni rilevati. Una volta effettuati i rilievi e le misure di conformità richieste, il sistema sviluppato deve essere in grado di trasferire i risultati dell'elaborazione, composto da un report sulle misurazioni effettuate, e l'immagine dello sfrido, ad una sistema di elaborazione esterna. Il *feedback* fornito dal sistema di visione in oggetto servirà quindi ad una ricomposizione automatica degli sfridi per un riutilizzo degli stessi. E da qui la sistemazione economica, ma con uno sguardo al futuro per un riciclo di qualsiasi materiale, e da qui la sistemazione morale nel rispetto dell'ambiente e della natura. Al fine di un "*re incollaggio*" efficace delle forme degli sfridi, da specifica, vengono altresì richiesti i seguenti controlli sui lati dei poligoni rilevati: verifica allineamento della forma rispetto ad un asse di riferimento con una precisione di $\pm 1^\circ$, verifica linearità dei lati del poligono, verifica che ciascun punto del bordo sia all'interno di una fascia di spessore 2 mm, a cavallo della retta passante per i due relativi vertici rilevati. Il sistema di visione deve avere un campo inquadrato di $200 \times 200 \text{ mm}^2$, deve fornire risultati con una precisione di $\pm 0.5 \text{ mm}$, e non vi sono vincoli in merito alla distanza di posizionamento della telecamera e del sistema di illuminazione.

Effettuato quindi uno studio di fattibilità dell'applicazione, si è passati in seguito allo sviluppo, utilizzando un sistema di visione, un sistema di illuminazione opportunamente scelto, ed implementando una logica al sistema fine alla risoluzione del problema, avendo a disposizione gli strumenti che verranno mostrati in questo elaborato. Dallo sviluppo, alla realizzazione, è stato facile il passaggio all'idea di future e possibili applicazioni del sistema, e della metodologia di elaborazione sviluppata. Verrà infatti mostrata nell'ultima parte di questo elaborato, l'idea della possibile applicazione al settore del controllo automatico su scansioni e acquisizioni di immagini provenienti dal mondo della micro e nano elettronica. Ma come citato in questa prima parte, vi è la prospettiva dell'adattamento di tale tecnica e del sistema sviluppato, in altri settori oltre quelli per cui viene presentato in questo studio.

1.2 I sistemi di visione artificiale nel campo dell'automazione industriale

Laddove è necessario implementare un controllo che richieda il senso visivo ed una conseguente elaborazione delle informazioni acquisite, entrano in gioco i sistemi di visione artificiale. Nella loro accezione più generica, o almeno nel settore merceologico di riferimento oggetto di questo studio (automazione industriale), un sistema di visione artificiale è composto da strumenti atti al rilevamento di un soggetto in una scena opportunamente illuminata, che lavorano in simbiosi con strumenti di elaborazione capaci di generare dei risultati del controllo. Il processo implementato in tali sistemi può esser quindi scomposto nelle sue fondamentali componenti: la sorgente di illuminazione, la scena, un sistema di lenti, un sensore di immagine, degli strumenti di elaborazione

di immagine capaci di processare l'immagine ottenuta, ed un processore che gestisca tutte le operazioni affini richieste al sistema di visione. In figura 1 troviamo un esempio schematico della catena di processo di un sistema di visione artificiale.

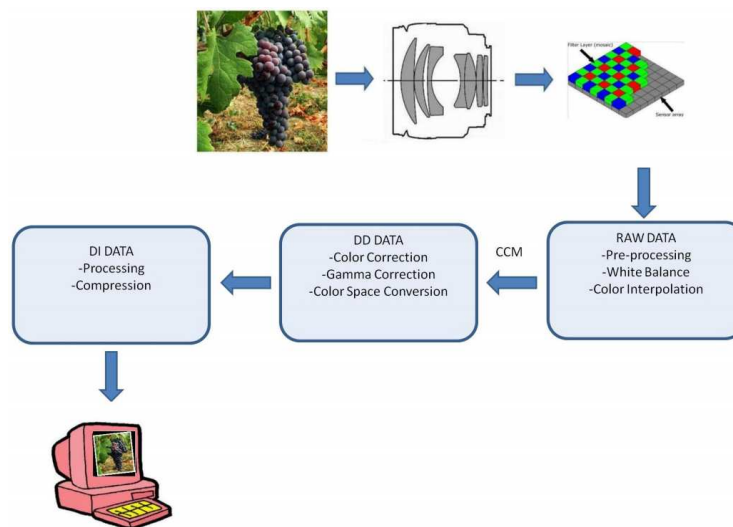


Figura 1. Macro struttura di un sistema di visione artificiale

Storicamente, nel settore dell'automazione industriale, i sistemi di visione artificiale vengono utilizzati al fine del controllo della qualità dei prodotti ottenuti e dei processi, ed al fine dell'automatizzazione degli stessi. In entrambe le macro categorie delineate, ritroviamo il meccanismo che richiede l'intervento di tali sistemi nell'ottica dei sistemi automatici: il meccanismo della controreazione [1]. Vediamo così che tale principio, alla base del funzionamento di numerosissimi sistemi automatici ed elettronici, sia intravedibile anche nell'utilizzo dei sistemi di visione artificiale come meccanismo di *feedback* all'interno dell'anello di reazione. Il feedback più semplice che viene richiesto ad un controllo di visione artificiale è il classico risultato di conformità, il controllo è andato a buon fine o meno, seguendo le elaborazioni prefissate. Ma immaginiamo adesso di non eseguire una semplice operazione di comparazione tra un oggetto che sia conforme ed uno che non lo sia. Al fine di ottenere dai sistemi di visione artificiale dei meccanismi di *feedback* più reattivi, e con una mole di informazioni consistente, è necessario oggi conferire ai sistemi di visione artificiale un grado di "intelligenza" superiore. Immaginiamo adesso, di ricreare il meccanismo di retroazione che intrinsecamente adottiamo nell'atto della presa di un qualsiasi oggetto. Il nostro braccio può essere interpretato da un robot antropomorfo (per trovare una collocazione nel contesto dello sviluppo di questo lavoro). Nell'atto della presa di un oggetto, la nostra vista continuamente capta delle immagini, che vengono elaborate dal nostro cervello, che comanda l'azionamento del braccio nell'atto dell'avvicinamento dello stesso all'oggetto. La nostra mente si serve del feedback dell'elaborazione dell'immagini dei nostri sensori per direzionare il braccio. La capacità di analisi, e di scomposizione del processo visivo e mentale, permettono una corretta analisi per le problematiche di visione artificiale. Visto questo esempio, è facile adesso immaginare verso dove si stia andando, e quali sono i requisiti del mondo dell'automazione industriale per i sistemi di visione artificiale.

L'utilizzo di sistemi di visione artificiale, permettono la possibilità di ottenere dei controlli oggettivi, che non soffrono della variabilità del "sistema umano". Con questa definizione non si intende classificare una macchina come superiore rispetto all'uomo, ma si vuole collocare come di ausiliare utilità alle attività ripetitive, tediose e faticose degli esseri umani.

1.3 “Spese, guadagni, ricavi,...”. La scelta dell’investimento

Nell’ottica dell’adozione di un sistema di visione artificiale, a prescindere dalla natura dell’esigenza, un qualsiasi soggetto è portato¹ ad uno studio ed ad una analisi dei costi che giustificano l’adozione del sistema. Per fare un esempio, uno dei primi controlli di visione artificiale implementato nel settore di automazione industriale è stato il controllo qualità su dei soggetti prodotti. La necessità che ha spinto l’adozione di tali sistemi è stata di duplice natura: tecnica e produttiva. La natura tecnica, risiede nella richiesta di sensori, di sistemi, capaci di rilevare non più la semplice presenza o assenza dei prodotti, ma di parti di essi, e/o la qualità degli stessi. La natura produttiva, o commerciale, risiede nella esigenza di assicurare la qualità del prodotto finito non soltanto sotto il profilo cosmetico, ma soprattutto funzionale del prodotto. La mancanza di una parte di un prodotto può causare una perdita ad una azienda che esula dal solo aspetto dell’immagine dell’azienda, visto che può praticamente creare dei danni all’utente o utilizzatore del prodotto. Un investimento visto con questa chiave di lettura, di sicuro giustificerebbe qualsiasi spesa per il conferimento di qualità ad un prodotto. Dobbiamo purtroppo (in questo caso) avere a che fare col mondo reale, dove vigono delle regole e soprattutto delle consuetudini, che mal si sposano col concetto di sicurezza e dell’attenzione al prossimo a discapito del beneficio del singolo. Detta questa prima nota, se vogliamo dire l’aspetto e la collocazione filosofica dell’elaborato, ogni qualvolta si deve approssicare lo studio di fattibilità per una qualsiasi applicazione, si devono fare i conti con l’investimento per far fronte alla necessità del controllo di visione artificiale. Nel computo dell’investimento, vanno valutati diversi aspetti che vanno ben oltre la natura economica dei dispositivi in se. Il primo aspetto – tecnico - da tener in conto, è l’efficienza in termini di durata dei dispositivi, al fine anche di poter organizzare anche un piano di manutenzioni programmate. A tal fine, i parametri discriminanti nel mondo dei dispositivi elettronici sono l’MTTF (*Mean Time To Failure*) e/o l’MTBF (*Mean Time Between Failure*). Un fattore discriminante in termini produttivi è anche l’efficienza energetica della macchina e del sistema. Ma di sicuro sotto il profilo tecnico, è importante che i sistemi siano robusti, efficienti, e manifestino durabilità nel funzionamento, fondamentale per uno stabile funzionamento della macchina nel suo complesso.

Dal punto di vista economico invece, dall’esempio riportato, il primo fattore da considerare è, banalmente, la spesa rispetto al ricavo derivante dall’assunzione di tale sistema. Il ricavo può essere quindi anche inteso come eliminare da una intera produzione un prodotto piuttosto che l’intera produzione.

Il sistema di visione è però chiamato a non fornire una gran percentuale di “falsi scarti” (giudicare non conformi dei prodotti che lo sono), e di fornire il 100% degli scarti realmente esistenti. Da questa assunzione quindi, deriveranno quindi delle scelte tecniche che di sicuro andranno a incidere sulla spesa del sistema adottato. I danni di immagine invece, sono difficilmente quantificabili, o per lo meno a breve termine.

Spesso si ha a che fare con il computo della spesa derivante dai “fermo macchina”. A causa del malfunzionamento dei macchinari industriali, ci sono dei momenti in cui viene fermata la produzione. Spesso accade che sia richiesto una figura esperta per la risoluzione del problema, e questo tempo di fermo della produzione genera una perdita per l’azienda. A tal fine, ciò che viene richiesto ai sistemi di visione, non è la ovvia richiesta di una fermata “nulla” dei sistemi in qualsiasi condizione, ma che siano altresì dotati di strumenti atti alla semplice risoluzione di problemi riscontrabili. In tal modo, non è richiesto l’intervento di una figura specializzata per la risoluzione di un problema che potrebbe creare dei danni all’azienda.

Infine, nell’ottica di compiere uno “scouting” per la scelta di una piattaforma di visione artificiale, la *soluzione* visione artificiale deve avere altre caratteristiche. In termini in questa sede tecnico-economici, la piattaforma scelta deve avere le seguenti peculiarità:

¹ Al minimo, per il fine più ovvio.

- *esser di facile interpretazione*: al fine di ridurre il tempo di sviluppo delle applicazioni, e per lo studio necessari; non richiedere l'intervento di personale specializzato; essere impartito in breve tempo a personale neofita.
- *Comunicazione di semplice ed efficace impostazione*: le anomalie e le perdite economiche derivanti dalla inefficiente comunicazione tra dispositivi, sono tra le spese maggiori nello sviluppo e nella diagnostica delle problematiche a livello macchina. A tal fine, ad un sistema di visione artificiale viene richiesto che sia in grado di interfacciarsi con i bus di uso comune, o almeno dell'uso nella macchina stessa. L'impostazione della comunicazione deve essere di facile comprensione, e di fruibile conoscenza approfondita.
- *Assistenza rapida ed efficace*: Sempre nell'ottica di non generare perdite, nel caso si verificasse un'anomalia non gestibile dall'utilizzatore, e già in sede di sviluppo di un'applicazione, di una macchina, di un sistema complesso, è necessario che il produttore della piattaforma sia in grado di fornire informazioni facilmente fruibili, e di un alto grado di soddisfazione delle richieste, una efficiente rete di assistenza.
- *Fruibilità in tempi brevi*: nel caso in cui vi sia la necessità di mandare in riparazione dei componenti, è necessario che un eventuale ricambio giunga in tempi brevi in qualsiasi parte si trovi il prodotto. E da qui la necessità di *globalità* del produttore o della rete di distribuzione di ricambi e assistenza, qualora si pensi ad un'azienda che agisce su mercati extra territoriali.

Tutti i fattori visti, nonché altri non meno importanti non trattati in questa sede, devono esser parte dello studio di fattibilità di una soluzione di visione artificiale. Non prescindono da questi per finire, gli aspetti della formazione del personale, sui dispositivi e sull'utilizzo, il capitale umano da investire, che definiscono poi il *time to market* per una macchina o applicazione sviluppata, fase nel ciclo di vita di un prodotto (inteso qui come macchinario o sistema sviluppato) in cui il capitale viene solo e soltanto investito, speso, a meno che non si intraprendano delle eventuali azioni di marketing, *partnership*, richieste di collaborazioni, finanziamenti esterni, etc.

1.4 I sistemi di illuminazione

In ambito visione artificiale per automazione industriale, i sistemi di illuminazione più utilizzati variano per geometrie, spettro di emissione, direzionalità, ed ovviamente, costo. L'unico punto fermo, a meno che non si utilizzino sorgenti coerenti², è che le sorgenti di radiazione utilizzate sono LED. Grazie alla stabilità nel tempo, alla lunga durata, e all'efficienza energetica, tali dispositivi hanno preso piede sui dispositivi di radiazione utilizzati in precedenza. Il mercato ha fatto sì che il prezzo di questi dispositivi sia, al giorno della stesura di questo elaborato, comparabile a comuni dispositivi semplici. Vigge altresì una necessità nel campo della visione artificiale come nel campo della fotografia, in un modo o nell'altro, la sorgente di radiazione dovrà esser sincronizzata con il sistema di cattura dell'immagine. Ovvero, il sensore della telecamera verrà esposto alla radiazione per un tempo pari allo *shutter speed*³, tipicamente con durata variabile tra pochi microsecondi, e centinaia di millisecondi. Per tutta la durata dell'esposizione alla radiazione, è quindi necessario che la sorgente di radiazione sia stabile, o al meno al limite del poterla considerare come tale. Questo significa che all'interno della durata minima dello *shutter speed*, la pulsazione della radiazione da parte della sorgente della radiazione debba esser tale da poter esser considerata continua, o almeno, ad una frequenza opportunamente più alta, vale in questo caso l'applicazione della teoria del campionamento [2]. Per questo motivo quindi vengono utilizzati i LED come sorgenti di illuminazione, tipicamente alimentati in PWM da un controllore che può trovarsi all'interno o separatamente all'illuminatore. I sistemi di illuminazione si differenziano dunque in termini di "colore", tipicamente UV, blu, verdi, rossi, IR, bianchi; e secondo le loro geometrie. La scelta di un

² Laser.

³ Verrà approfondito il meccanismo di funzionamento al paragrafo 2.3.1.

colore piuttosto che un altro, è da relegare al tipo di materiale e di superficie da analizzare. Un'altra discriminante, se si riesce piuttosto a carpire un effetto generato da una sorgente con radiazione più o meno energetica⁴. Esistono in commercio svariati sistemi di illuminazione per applicazioni di visione artificiale. Vengono riportati a titolo di esempio due esempi di sistemi di illuminazione adottati per controlli di visione artificiale in ambito industriale: la *Dome*, e la *Low-angle*. La prima viene utilizzata per rilevare immagini di oggetti senza ombre.

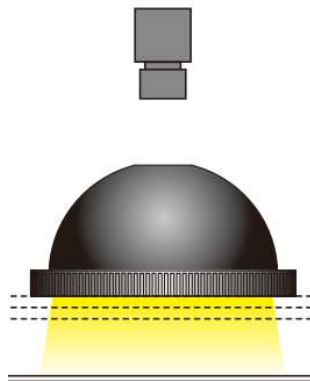


Figura 2. Sistema di illuminazione *Dome*.

Vista la direzionalità della sorgente, sugli oggetti giungono raggi provenienti idealmente da tutti i raggi della circonferenza della “campana *Dome*”, come visibile nella figura 2, che mostra la sezione del sistema di illuminazione [3]. Ed è ideale per superfici che non sono lisce.

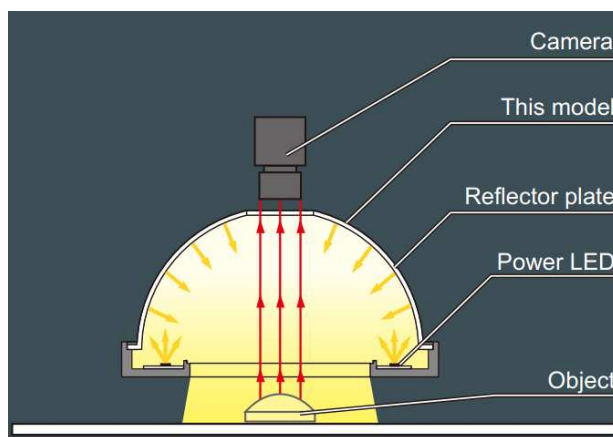


Figura 3. Sezione di un sistema di illuminazione *Dome*.

In figura 4, sulla sinistra viene visualizzata la cattura di una immagine con un comune sistema di illuminazione, una barra LED, mentre sulla destra è possibile notare l'effetto dell'utilizzo di un sistema di illuminazione *Dome*.

⁴ $E = h\nu$.



Figura 4. Acquisizioni con illuminazione diretta e con sistema *Dome*.

Come si vede dalla figura 4, in prossimità delle zone dove troviamo delle curve, mediante l'utilizzo di questo sistema riusciamo a catturare un'immagine insensibile alle "rugosità" delle superfici analizzate.

Il secondo esempio invece è un sistema di illuminazione che per la sua geometria è adatto alla cattura dei profili delle immagini. Viene riportata in figura 5 la sezione del sistema di illuminazione *Low-angle*. Esso è composto da una corona circolare di n strisce di LED, con asse ottico di emissione focalizzato verso il centro della corona circolare, ad un piano inferiore rispetto al piano della corona.

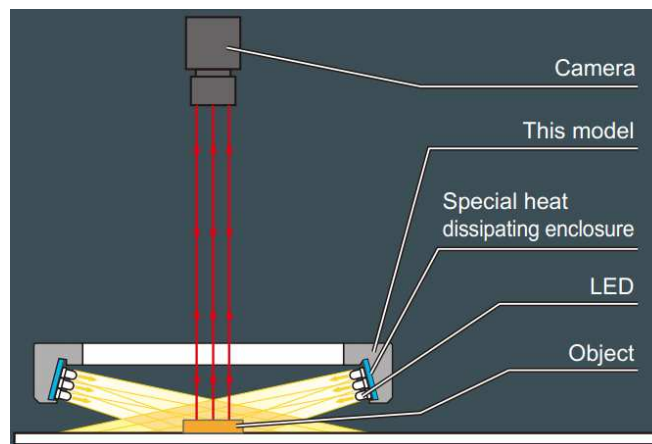


Figura 5. Sezione del sistema di illuminazione *Low-angle*.

Vista la direzionalità degli assi ottici, tale sistema risulta ideale per l'estrazione di profili degli oggetti, anche di materiale trasparente, o è utile nel caso si vogliano evidenziare delle zone di discontinuità di superfici lisce. In figura 6 è riportato a titolo di esempio un campione di cattura di una immagine relativa alla parte inferiore di una comune batteria. I caratteri impressi, sono come "scavati" all'interno del metallo. Grazie quindi alla direzionalità dei LED, tali zone di discontinuità risaltano alla cattura.



Figura 6. Acquisizione con sistema d'illuminazione *Low Angle*.

1.5 Le ottiche

L'utilizzo delle ottiche nelle applicazioni di visione artificiale in ambito automazione industriale, è spesso limitato alla scelta di ottiche a *bassa distorsione*, o *standard*. Infatti, viste le dimensioni dei sensori in gioco, da 1/3 a 1 pollice, vengono utilizzate le cosiddette ottiche a "*passo C*", nome dettato dalla dimensione e dal passo della tipologia dell'innesto. Utilizzando delle ottiche dobbiamo tenere in considerazione degli effetti non lineari che possono inficiare il controllo, la misura, etc. Infatti, a causa delle aberrazioni cromatiche, zone di discontinuità potrebbero esser male interpretate a causa del rumore in termini di un colore aggiunto, della distorsione generata dall'ottica. A causa delle diverse distorsioni, rilievi e misure potrebbero soffrire di un'imprecisione tale da non poter ottenere un sistema stabile di misura.

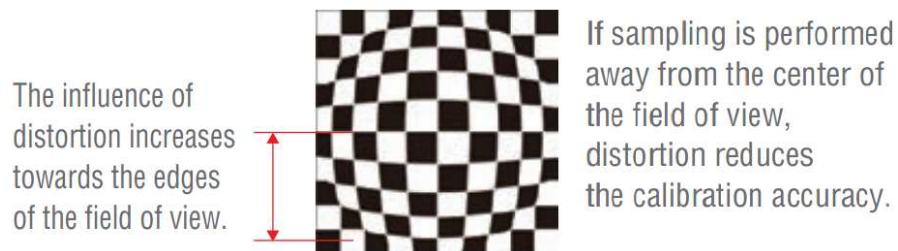


Figura 7. Effetto della distorsione ottica.

Per ovviare a tutti problemi derivanti dall'utilizzo delle ottiche, ad oggi esistono numerosi centri di ricerca e sviluppo che lavorano nello sviluppo di nuovi materiali, ideali al nostro scopo sotto il profilo ottico. All'interno dei sistemi di visione, troviamo altresì numerosi strumenti di elaborazione software capaci di compensare in parte gli effetti indesiderati delle ottiche. Le compensazioni valgono, come sempre, al limite di definite e precise condizioni di utilizzo. Sempre nel settore industriale, troviamo poi dei sistemi ottici che discostano dal semplice principio di compensazione di effetti indesiderati, ma che implementano filosofie di osservazione innovative [4]. Troviamo ad esempio ottiche telecentriche, che non soffrono dell'approssimazione "dell'arco di circonferenza" rispetto alla focale; ottiche *Poliview*, Macro, pericentriche, multicentriche, etc. Per gli scopi del lavoro svolto, sono state scelte delle ottiche a bassa distorsione, visto che abbiamo a che fare con delle analisi dimensionali. La scelta della focale della tipologia di ottiche utilizzate, è agevole mediante l'utilizzo di grafici presenti all'interno dei data sheet delle relative telecamere. Si riporta a titolo di esempio, il grafico per la scelta della focale dell'ottica per una telecamera con sensore da 2/3 di pollice:

High-speed CMOS Camera (Standalone) FH-S□04 (Using 3Z4S-LE VS-H1)

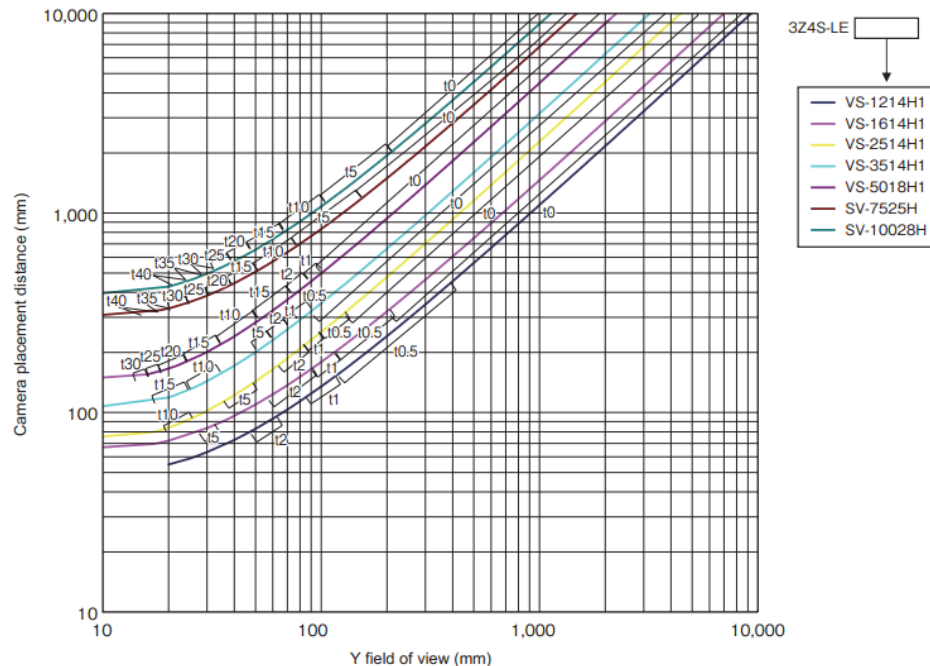


Figura 8. Grafico per la scelta della focale dell'ottica

Una volta definita da specifica la distanza alla quale può esser posta la telecamera, e la superficie da inquadrare, è possibile individuare un punto su una delle rette presenti nel grafico in figura 8. Sulle ascisse troviamo il lato Y della superficie da inquadrare, mentre sulle ordinate troviamo la distanza. Per ottenere la dimensione del lato X che verrà inquadrato in tali condizioni, basta impostare la proporzione utilizzando i dati relativi alla risoluzione del sensore.

Sulle ottiche troviamo, o almeno sulle ottiche convenzionali, due meccanismi di regolazione meccanica: la ghiera della messa a fuoco, ed la ghiera del diaframma. Con la prima è possibile regolare la messa a fuoco, ovvero il piano alla quale si riesce ad ottenere un immagine nitida. Il diaframma è invece un otturatore meccanico, che regola la quantità di luce che viene trasferita dalla scena al sensore. Il suo funzionamento è analogo a quello della pupilla dell'occhio umano. Modificando quindi l'impostazione del diaframma, modificheremo quindi la quantità di luce che "impatterà" sul sensore, ma non solo. Con esso, infatti, cambia anche la *profondità di campo* (*DOF* o *Depth of Focus*). L'occhio umano è un organo formidabile, e ci mostra immagini con una profondità di campo sbalorditiva. Altro non è che il range di distanza all'interno della quale riusciamo a vedere nitidamente, ovvero a mettere a fuoco. "Aprendo" il diaframma, otterremo una maggiore quantità di luce, ma una minore profondità di campo, e viceversa. Quindi laddove è necessario mettere a fuoco diversi piani di un oggetto che non risiedono nello stesso, è necessario chiudere il diaframma, e magari "aprire" lo *shutter speed*⁵ (se è possibile!). Un esempio visivo può esser d'ausilio per la comprensione del concetto della profondità di campo:

⁵ Il funzionamento verrà chiarito al paragrafo 2.3.1.



Figura 9. Profondità di campo.

1.6 I sensori di immagine

Il dispositivo atto alla trasformazione dell'immagine della scena in una matrice di valori – numeri – è il sensore. Esso è composto da una matrice di elementi sensibili alla radiazione luminosa. Il numero degli elementi, le dimensioni della matrice, definisce la risoluzione del sensore. All'atto pratico, un sensore con una risoluzione maggiore consente di effettuare più controlli o misure con il medesimo scatto, e consente di ottenere una maggiore precisione nelle applicazioni di misura. All'atto invece di una prima calibrazione del sistema di visione artificiale, per avere un'informazione di massima in merito alla precisione dell'acquisizione, è possibile impostare la proporzione tra lato inquadrato, e numero di pixel sul lato inquadrato. Da questa è possibile ottenere a quanti millimetri (ad esempio), corrisponde uno scostamento di un pixel nel sensore. Pixel è il più piccolo elemento della matrice del sensore, interpretato dal fotodiodo e dal circuito di interfaccia al bu di comunicazione (per quanto concerne i sensori CMOS trattati in questo elaborato). Secondo la tecnologia, troviamo nel nostro settore di applicazione telecamere con sensori CCD o C-MOS. Compiendo un'analisi sull'evoluzione del mercato dei sensori dell'ultimo ventennio, possiamo facilmente renderci conto che, almeno per gli utilizzi fini a questo studio, si è raggiunti una maturità progettuale e processuale tale per cui i sensori C-MOS hanno delle caratteristiche superiori ai sensori CCD. Le caratteristiche salienti e discriminanti per la scelta di un sensore in ambito acquisizione immagini con telecamere industriali sono:

- *Fill-Factor (FF)*: ovvero, la percentuale di area del pixel occupata dal dispositivo di foto rivelazione (fotodiodo) rispetto all'area totale, ove è presente l'elettronica di interfaccia. Attraverso l'utilizzo di microlenti, processi di *scaling* adottati per le tecnologie in oggetto, tecnologie di fabbricazione alternative (front e back illuminated devices), il *Fill Factor* può in prima approssimazione ad oggi non essere considerato un fattore rilevante. Viene di seguito riportato un esempio di uno spaccato di un sensore C-MOS APS (*Active pixel sensor*):

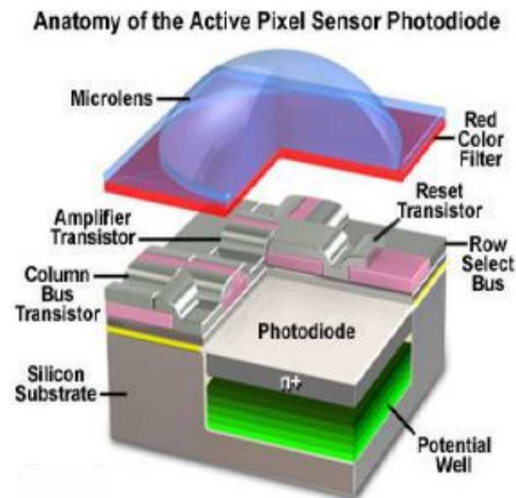


Figura 10. Sezione di un sensore CMOS APS

- *Range dinamico*: da esso dipende la capacità del sensore nel rilevare radiazioni all'interno di un range più o meno esteso tra i due estremi che sono la saturazione, e la corrente di buio. Saturazione è la massima quantità di fotoni oltre la quale non sarà più generata carica derivante da effetto fotoelettrico. Mentre la corrente di buio è la carica generata dal fotorelevatore in assenza di luce incidente, la carica di rumore equivalente (NEP). Il rapporto tra le quantità citate – in termini logaritmici di rapporto segnale rumore – definisce il range dinamico. Esso è quindi per definizione, dipendente dalla radice quadrata dell'area del fotorelevatore[5]. Per fare un po' di *reverse engineering*, è curioso notare come l'area dei pixel dei sensori utilizzati in ambito industriale siano di dimensioni ben superiori a quelli utilizzati in ambito *consumer*. Una "fotografia" del mercato attuale, fornisce un fattore che varia da 10 a 6, per quanto concerne i sensori C-MOS. Ad oggi, vi sono anche delle tecniche di polarizzazione e di lettura dei fotodiodi, atte all'estensione del range dinamico dei sensori⁶.
- *Velocità*: nell'ottica della richiesta di una velocità sempre crescente nell'ambito dei controlli di visione artificiale, il fattore velocità di trasferimento è parte attiva nel computo del tempo totale richiesto al controllo di visione. Mentre i tempi di lettura dei sensori CCD risultavano molto lenti rispetto ai C-MOS (a meno di particolari implementazioni circuitali, in via del tutto generale), questi ultimi permettono ad oggi velocità di acquisizione che si attestano intorno ai 350 *fps* (frame per secondo), equivalenti a poco più di 3ms. Tempo a cui va sommato il tempo dello *shutter speed*⁷ per ottenere il tempo di acquisizione totale, per una telecamera con sensore a colori risoluzione VGA 640x480 pixel.

I sensori vengono ulteriormente differenziati in base alla capacità di acquisire l'immagine a colori, o in scala di grigi. Oltre all'intuitiva distinzione in termini della possibilità di discriminare colori, è lecito pensare già in prima approssimazione che le informazioni da gestire in caso piuttosto che nell'altro sarà nel rapporto 1:3. Questo perché, in ambito acquisizione da parte dei sensori, si è soliti interpretare e ricostruire i colori mediante composizione delle tre componenti cromatiche pure RGB. È lecito altresì sottolineare in questa sede, che i sensori a colori, per via della tecnologia costruttiva, introducono una ulteriore distorsione nell'acquisizione dell'immagine. Con uno sguardo al mercato, a meno di un utilizzo di sensori con tecnologia Foveon X3[6] o TFD [7] (che non esistono ancora in telecamere industriali!), i sensori a colori vengono prodotti depositando (in linea di principio e per un esempio visivo) un filtro come quello mostrato in figura 11 su un sensore *monochrome*.

⁶ Si veda anche il paragrafo 2.3.1. *Multi slope acquisition*

⁷ Il significato verrà chiarito al paragrafo 2.2.1. *Camera Image Input*.

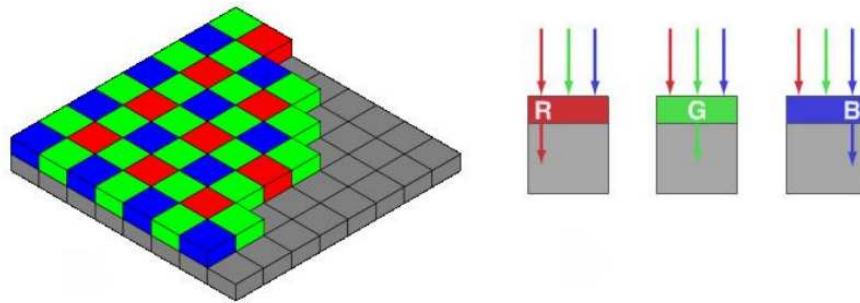


Figura 11. Bayer Pattern Color Filter Array.

Il primo deleterio effetto nel campo della visione artificiale (o almeno in particolari condizioni) è che a causa di una ricostruzione dell'informazione relativa alla gamma non realmente interpretata per pixel, (mediante l'utilizzo di approssimazioni per interpolazione di informazioni relative a pixel adiacenti), l'informazione sul colore per singolo pixel soffre già di una intrinseca distorsione.

1.7 Gli algoritmi di analisi ed elaborazione delle immagini

Una volta acquisita l'immagine ed una volta trasferita all'unità di elaborazione l'immagine, essa è quindi pronta per essere analizzata. Storicamente, per assistere alla prima comparsa dei sistemi di visione nel campo dell'automazione industriale, dobbiamo guardare agli anni '80 del secolo passato. I primi controlli implementati, in termini di elaborazione dell'immagine, soddisfacevano ai requisiti del "controllo qualità" sui prodotti delle linee di processo. In termini di elaborazione quindi, il controllo che veniva effettuato era un confronto dell'immagine acquisita al controllo x rispetto all'immagine master X . Questa operazione veniva e viene implementata nei cosiddetti algoritmi di "pattern matching". In verità, a causa dello scarso potere computazionale dei sistemi di elaborazione a quegli anni⁸, non vi era la concreta possibilità di effettuare un confronto tra matrici ad alta velocità. Il *pattern matching* degli anni '80 si serviva quindi di una pre-binarizzazione dell'immagini, ed altro non effettuava se non un conteggio di pixel. Pertanto, il controllo qualità effettuato, viene tradotto in una semplice controllo presenza / assenza di una parte dell'immagine acquisita (del colore o scala di grigi prescelta). In termini "sensoristici"[Cit.], non più che una fotocellula capace di rilevare colori, focalizzata su un'area inquadrata, e con una "buona sensibilità". Abbiamo assistito negli ultimi trent'anni ad una vera e propria rivoluzione nel campo della elaborazione di immagini. Oggi i sistemi di elaborazione ci forniscono un alto potere computazionale, grazie allo sviluppo di sistemi *Multi processor* e distribuiti, che lavorano a velocità sempre più spinte. Negli anni '90 del secolo trascorso, con l'avvento di nuove tecnologie nel campo dei sensori applicate all'universo industriale, si è passati poi all'elaborazione delle immagini in scala di grigi, piuttosto che binaria. Si è passati quindi alla realizzazione di algoritmi di elaborazione capaci di effettuare il vero e proprio confronto tra le matrici riportanti i valori del master e dell'acquisizione x .

Il "criterio di giudizio", è il criterio di accettazione, la soglia di commutazione di un giudizio *buono* o *non buono*, conforme o non conforme. Possiamo immaginarlo in quest'ambito come la percentuale - in termini di *pattern* - oltre il quale il controllo deve risultare conforme. Al di sotto di questo valore di soglia di accettazione, il controllo deve fornire un esito negativo. Ogni algoritmo di elaborazione, agirà all'interno dello spazio dell'immagine prefissato, elaborerà l'immagine secondo quanto compete, e fornirà un giudizio sul compito assegnato, basandosi sulle soglie di giudizio inserite. Nel caso del *pattern matching*, l'algoritmo effettuerà una comparazione tra il modello assegnato, all'interno della regione di azione prefissata, e basandosi sui criteri di giudizio assegnati

⁸ Si deve sempre pensare che lo *stato dell'arte* approda al mondo industriale dopo più di una decade.

fornirà il giudizio del controllo. Troviamo poi altri algoritmi di elaborazione, atti al rilevamento di uno o più bordi, zone di discontinuità all'interno di una zona prefissata. Essi forniranno come risultati delle informazioni in termini di coordinate, e stabilirà il giudizio del controllo in base alle coordinate di conformità assegnate. Una panoramica sulle peculiarità degli strumenti di elaborazione verrà affrontata nel prossimo capitolo, e sarà sicuramente di ausilio alla contestualizzazione di quanto fin qui affermato.

Nell'ottica di rendere gli strumenti di elaborazione più veloci e precisi, e con l'obiettivo di conferire più robustezza rispetto alle variazioni di condizioni di cattura dell'immagine, ad oggi troviamo algoritmi atti alla misura sui pattern, capaci di fornire con un grado di estrema precisione, posizione, angolo, degli n oggetti che si trovano all'interno dell'immagine acquisita. Tali algoritmi combinano una tecnologia del rilevamento di contributo altamente innovativo, derivante dallo sviluppo di algoritmi di elaborazione immagine nel campo dei rilevamenti facciali⁹. In figura 12 troviamo un breve riassunto grafico dell'evoluzione nella storia del *pattern matching*.

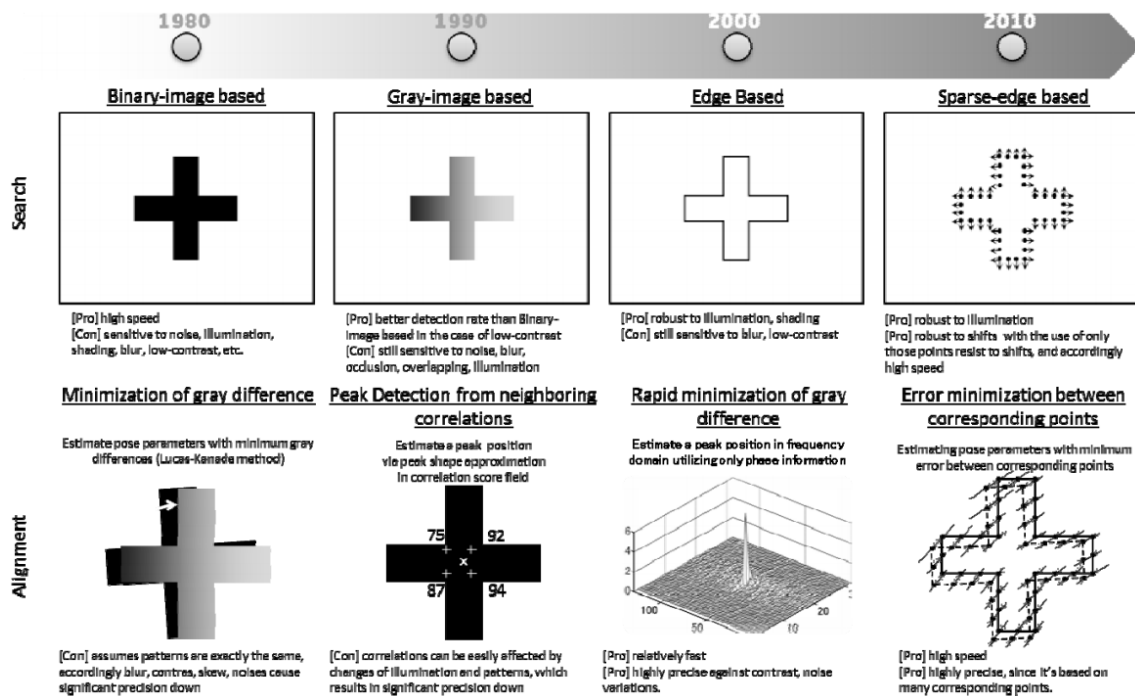


Figura 12. Breve storia del “*pattern matching*”.

Dall'evoluzione degli algoritmi di *pattern matching*, tra i quali troviamo diverse varianti per potenza di elaborazione, grado di precisione, misure effettuate [8], troviamo poi lo sviluppo di numerosi algoritmi dedicati ad altre misure e controlli sulle immagini acquisite. I primi esempi di algoritmi entrati a far parte delle librerie per automazione industriale, sono gli algoritmi “dimensionali”, che si servono del rilevamento di zone di discontinuità (da intendere all'interno della matrice bitmap dell'immagine lungo una direzione di ricerca) per fornire dati in merito alle coordinate dei punti rilevati. Su questa base esistono poi algoritmi più complessi in grado di fornire dei calcoli su un set di coordinate rilevate secondo le zone di ricerca assegnate. Oltre a questi, le esigenze di mercato hanno dato vita ad algoritmi di elaborazione, all'interno del campo trattato, specializzati in dei rilevamenti dedicati, anche nell'ottica dell'automatizzazione dei processi, della tracciabilità dei prodotti, del feedback sempre più attivo per sistemi robotizzati. Troviamo quindi in

⁹ Si veda Appendice A. Technology overview Pattern Matching Algorithm Shape search III.

commercio, ad esempio, algoritmi dedicati per la lettura di codici 1D e 2D come *barcode*, *datamatrix*. Algoritmi atti al riconoscimento (OCR) e la verifica (OCV) di caratteri e stringhe. Algoritmi via via sempre più complessi, dedicati a funzioni studiate appositamente per il campo di interesse. È interessante notare che tali algoritmi di elaborazione, sfruttano principi derivanti dal mondo della matematica, dell'ingegneria, della fisica, e per lo sviluppo degli stessi, spesso viene richiesta la conoscenza di tutto il sistema di visione artificiale, e non solo dell'immagine già acquisita.

Agli algoritmi che si occupano dell'elaborazione dell'immagine vera e propria, si affiancano algoritmi di supporto, di composizione, compensazione. È possibile altresì effettuare dei calcoli utilizzando i risultati degli algoritmi di elaborazione. L'universo dell'elaborazione delle immagini è così vasto che richiederebbe almeno un capitolo a sé, tuttavia in questo paragrafo si è voluto introdurre il principio di elaborazione, che sarà via via più chiaro nella lettura di questo elaborato.



Figura 13. Esempi di campioni reali sui quali vengono effettuati controlli e riconoscimento delle informazioni criptate.

2. Il sistema di visione artificiale – lo stato dell’arte.

Volgendo uno sguardo al mercato dei produttori di piattaforme di visione artificiale, e volendo compiere un’analisi sulla tendenza e sullo stato dell’arte a livello globale¹⁰, possiamo eseguire una prima macro classificazione per potenzialità - in termini di elaborazione (hardware e software), grado di connettività con i dispositivi esterni, capacità di adattamento rispetto a delle variazioni di qualsiasi natura, e quindi possibilità di conferire un grado di intelligenza superiore al controllo visivo. Gli ultimi due aspetti citati ben si sposano tra loro per dar vita a sistemi di rilevamento 3D che eseguono una composizione tra un immagine catturata da un sensore 2D, ed una mappa di profilo della profondità ottenuta mediante uno scanner laser[9]. Come combinare la vista al tatto per ottenere un sistema 3D. O come combinare le informazioni ottenute da due telecamere per simulare una vista stereoscopica... L’universo visione artificiale come si sarà inteso, è vasto, e a oggi intravediamo numerose e vastissime terre inesplorate.

In termini di potenzialità di elaborazione, notiamo quindi che lo sviluppo di piattaforme di visione artificiale, utilizzano sistemi hardware (industriali) simili alle piattaforme che troviamo nel campo informatico. La tendenza è quindi quella dell’utilizzo di sistemi *PC-based*, ai quali vengono connesse telecamere, vengono interfacciati i dispositivi esterni, quali altri sistemi di elaborazione, HMI (*Human Machine Interface*), PLC, dispositivi di archiviazione, *DataBase*, etc.. E la tendenza segue quella dello sviluppo dei processori, quindi sistemi a velocità sempre più elevate, sistemi *Multi processor*, capaci di fornire una potenza di calcolo superiore grazie all’elaborazione in parallelo di più processori. Le telecamere seguono anch’esse lo sviluppo e la *roadmap* dello sviluppo dei sensori. Ad oggi troviamo nel mercato sensori che variano per risoluzione, qualità - in termini di range dinamico ed altri fattori citati. Un modello di telecamera differente, che troviamo nell’ambito controllo visivo ad alta velocità, o per oggetti soggetti a rotazione, è la telecamera con sensore “lineare”. In queste non abbiamo un sensore matriciale, bensì un sensore composto da una sola stringa di pixel. Un aspetto molto importante nello sviluppo di applicazioni di visione artificiale è la connettività del sistema. Potrà sembrare strano, ma le difficoltà maggiori nascono nel momento in cui si devono far dialogare tra loro dei dispositivi. Questa è una nota critica che esula dall’aspetto commerciale, poiché tali difficoltà vengono riscontrate anche con dispositivi dello stesso produttore che comunicano con la stessa modalità. Differenziamo quindi le comunicazioni in termini fisici, quindi di *bus* utilizzato. Tra questi troviamo le classiche interfacce digitali (tensioni logiche nell’universo industriale 0 – 24 Vdc), analogiche (in tensione o corrente), comunicazioni basate su *bus* di comunicazione industriale *EtherNET*, e che utilizzano diversi protocolli di comunicazione, come *EtherCAT*[10], *TCP/IP*, *ASCII on Eth*, etc..

La tendenza, sotto il profilo dei bus di comunicazione, è quella di fornire il maggior numero di connessioni possibili, nell’ottica del *revamping* di macchinari industriali, e nell’ottica del collegamento con dispositivi di produttori diversi. Mentre la tendenza per gli sviluppi futuri, verte sulla necessità di avere una comunicazione che sfrutti un livello fisico sicuro, ed un protocollo ed una modalità di comunicazione semplice e veloce. Ad oggi, dando uno sguardo sul mercato, pare che in ambito industriale il bus di comunicazione candidato a diventare lo standard nei prossimi anni sia *EtherCAT*. Verranno in questo capitolo esaminate le caratteristiche salienti del sistema di visione Omron FH, utilizzato per lo sviluppo dell’oggetto di questo elaborato.

¹⁰ Non saranno riportati in questa sede i particolari dello studio di mercato effettuato, si mostrano volutamente le caratteristiche e le tendenze con aspetto critico sotto il profilo tecnico sul mercato.

2.1 Il sistema di visione Omron FH, analisi delle caratteristiche e degli strumenti fondamentali per lo sviluppo dell'applicazione

La piattaforma di visione artificiale Omron FH, offre un portfolio dispositivi completo per lo sviluppo e la realizzazione della quasi totalità delle applicazioni di visione artificiale in ambito industriale. L'unità centrale di elaborazione basata su piattaforma *PC-based*, viene interfacciata alle telecamere mediante collegamento *Camera Link*, ed è possibile connetterla al mondo esterno mediante l'utilizzo di diversi *bus* di comunicazione, che vengono riassunti in figura 14. Troviamo all'interno un ambiente di sviluppo, già completo di una libreria di algoritmi di elaborazione di immagine, di strumenti per lo sviluppo di interfacce personalizzate, e di ausilio alla programmazione. Esiste anche un ambiente di simulazione, che replica le funzionalità del sistema, e sul quale è possibile lavorare *off-line* su dei campionamenti acquisiti, o in *real time* con gestione da remoto del sistema. È possibile interagire attivamente col sistema di visione mediante l'utilizzo dei *bus* di comunicazione, comandando da remoto il funzionamento anche a livello locale del singolo strumento di elaborazione, nonché a livello di sistema. Per un'analisi più approfondita delle caratteristiche del sistema, si rimanda il lettore alla lettura del manuale del sistema di visione¹¹. Verranno in questo capitolo mostrate le caratteristiche utili al nostro scopo dei dispositivi utilizzati.

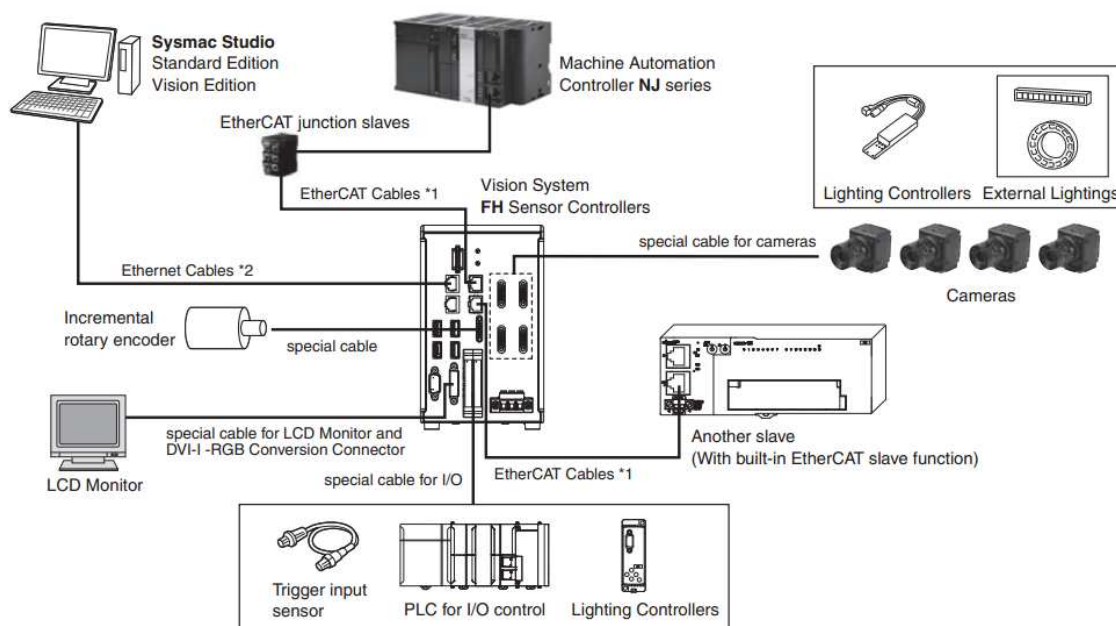


Figura 14. Connessione del sistema di visione artificiale

In Tabella 1, vengono invece riportate le caratteristiche elettriche e di interfacciamento del sistema di visione Omron FH.

¹¹ Z340-E1-01. FH/FZ5 Series Vision System – User's Manual.

Tabella 1. Estratto del data sheet della CPU FH-3050

Type		High-speed Controllers (4 core)			Standard Controllers (2 core)				
Model	NPN PNP	FH-3050	FH-3050-10	FH-3050-20	FH-1050	FH-1050-10	FH-1050-20		
External interface	Serial communications	RS-232C: 1 CH							
	EtherNet communications	No-protocol (TCP/UDP) 1000BASE-T							
	EtherNet/IP communications	1 port	2 port	2 port	1 port	2port	2port		
	EtherCAT communications	Ethernet port baud rate: 1 Gbps (1000 BASE-T) EtherCAT protocol (100BASE-TX)							
	Parallel I/O	(In the 2-line random trigger mode) 17 inputs (STEP0/ENCTRIG_Z0, STEP1/ENCTRIG_Z1, ENCTRIG_A0 to 1, ENCTRIG_B0 to 1, DSA0 to 1, DIO0 to 7, DI_LINE0) 37 outputs (RUN0 to 1, READY0 to 1, BUSY0 to 1, OR0 to 1, ERROR0 to 1, GATED to 1, STGOUT0/SHTOUT0, STGOUT1/SHTOUT1, STGOUT2 to 7, DOP to 15, ACK) (In the 5-line to 8-line random trigger mode) 19 inputs, STEP0 to 7, DI_LINE0 to 2, DIO to 7) 34 outputs (RFADY0 to 7, BUSY0 to 7, OR0 to 7, ACK, FROR, STGOUT/SHTOUT to 7)							
	Encoder interface	RS422-A line driver level, Phase A/B: single-phase 4MHz (multiplying phase difference of 1MHz by 4 times), Phase Z: 1MHz							
	Monitor interface	DVI-I output IF × 1ch							
	USB interface	4 channels (supports USB 1.1 and 2.0)							
	SD card interface	SDHC card of Class4 or higher rating is recommended.							
	Power supply voltage	20.4 to 26.4 VDC							
Ratings	Current consumption (at 24.0 VDC) *2	When connected to a intelligent compact camera	Connected to 2 cameras	5.0 A max.	5.4 A max.	6.4 A max.	4.7 A max.	5.0 A max.	5.9 A max.
		When connected to a 300,000-pixel camera, 2 million-pixel camera, 4 million-pixel camera, 5 million-pixel camera or 12 million-pixel camera	Connected to 4 cameras	—	7.0 A max.	8.1 A max.	—	6.5 A max.	7.5 A max.
			Connected to 8 cameras	—	—	11.5 A max.	—	—	10.9 A max.
			Connected to 2 cameras	4.1 A max.	4.2 A max.	5.2 A max.	3.6 A max.	3.7 A max.	4.5 A max.
			Connected to 4 cameras	—	4.8 A max.	5.6 A max.	—	4.3 A max.	5.0 A max.
Connected to 6 cameras	—	—	6.8 A max.	—	—	6.2 A max.			
Insulation resistance	Between DC power supply and controller FG: 20 MΩ or higher (rated voltage 250 V)								

2.1.1 CPU

L'unità centrale di elaborazione si basa su piattaforma *PC-based* con processori *double* e *quad core* Intel™, e sistema operativo Microsoft Windows 7™. I modelli di CPU si differenziano sia sotto l'aspetto del modello del processore *double* o *quad* core, sia per il numero di telecamere supportate, rispettivamente 2, 4, o 8 telecamere. In Fig. 15 viene riportata la tabella riassuntiva dei modelli di CPU della piattaforma di visione. Le uscite digitali sono di tipo a BJT *open collector*, con uscita a transistor NPN o PNP selezionabile da connessione del segnale COMIN¹². Oltre alle modalità di connessione contemplate riportate al paragrafo precedente, è presente uno slot per memorie SD Card, per il salvataggio ed il caricamento di *backup* di sistema, o di immagini. Ogni modello possiede due distinte schede di rete, ed è quindi possibile utilizzarne una per comandare il sistema utilizzando il software di gestione da remoto, e l'altra per lo scambio dati con un PLC, DB, HMI, etc..

FH Series Sensor Controllers


Item	CPU	No. of cameras	Output	Model
 Box-type controllers	High-speed Controllers (4 core)	2	NPN/PNP	FH-3050
		4	NPN/PNP	FH-3050-10
		8	NPN/PNP	FH-3050-20
	Standard Controllers (2 core)	2	NPN/PNP	FH-1050
		4	NPN/PNP	FH-1050-10
		8	NPN/PNP	FH-1050-20

Figura 15. Modelli unità centrale di elaborazione CPU FH.

L'utilizzo di sistemi *Multi processor* con interfacce ad alta velocità, permette quindi un aumento della potenza di elaborazione, le quali risorse possono esser gestite anche a basso livello dall'utente. L'utilizzo di più processori può esser sfruttato per l'esecuzione di più operazioni, o elaborazioni dell'immagine in parallelo, possiamo inoltre destinare una o più risorse per dei compiti affini al controllo di visione artificiale, come lo scambio dati con un DB, il salvataggio delle immagini, la gestione di altre attività. Il sistema FH si dimostra aperto sotto questo punto di vista, visto che è comunque una piattaforma *PC-based*.

¹² Per un approfondimento si rimanda all'Appendice B. FH-3050 vision system Instruction sheet.

2.1.2 Le telecamere

Troviamo a catalogo diversi modelli di telecamere, che variano per tipologia di sensore in termini di colore o *monochrome*¹³, tipologia di sensore CCD o CMOS, modelli con a bordo illuminazione ed ottica, e di diverse geometrie¹⁴. Nelle immagini a seguito, troviamo l'estratto del data sheet delle telecamere Omron FH con sensore CMOS.





Item	Descriptions	Color / Monochrome	Image read time	Model	
	High-speed CMOS Cameras (Lens required) For FH Sensor Controllers only	12 million pixels (Up to four cameras can be connected to one Controller. Up to eight cameras other than 12 million-pixel cameras can be connected to a FH-3050-20 or a FH-1050-20.)	Color	25.7 ms *	FH-SC12
			Monochrome		FH-SM12
	High-speed CMOS Cameras (Lens required) For FH Sensor Controllers only	4 million pixels	Color	8.5 ms *	FH-SC04
			Monochrome		FH-SM04
	High-speed CMOS Cameras (Lens required) For FH Sensor Controllers only	2 million pixels	Color	4.6 ms *	FH-SC02
			Monochrome		FH-SM02
	High-speed CMOS Cameras (Lens required) For FH Sensor Controllers only	300,000 pixels	Color	3.3 ms *	FH-SC
			Monochrome		FH-SM

Figura 16. Modelli telecamere con sensore CMOS

In termini di risoluzione, troviamo quattro modelli, dallo standard VGA 640x480 pixel, al modello da 12 Mega pixel 4084x3072. Dando un'occhiata alle dimensioni dei pixel, notiamo come il sensore a risoluzione standard offra di sicuro il range dinamico più elevato. È vero anche che grazie ai meccanismi di lettura ottimizzati per estendere il range dinamico in maniera autoadattativa¹⁵, questa informazione perde via via nel tempo importanza nella discriminante in un sensore (almeno in questo settore di applicazione).

In termini invece di velocità, ai due estremi troviamo un tempo di acquisizione (a cui va sommato il tempo dello *shutter speed*) che varia tra 3.3ms per una telecamera con sensore VGA, ai 25.7ms per una telecamera con sensore con risoluzione 12 Megapixel. A parte quest'ultima, tutte le telecamere montano ottiche C-MOUNT, ed il modello va scelto in base alla dimensione del sensore della telecamera. Vengono in appendice riportati i grafici per la scelta del modello dell'ottica, si noti come la dimensione dei sensori delle telecamere vari da 1/3 a 1 pollice rispettivamente per i modelli VGA e 12 Megapixel.

Per ultimo, può essere interessante sapere che è possibile utilizzare le telecamere con velocità standard, e doppia (che è stata riportata nell'analisi). Nel caso di funzionamento ad alta velocità, ogni telecamera utilizzerà due slot di ingresso telecamera nel sistema di visione.

¹³ Anche se la mole di informazioni in gioco varia di un fattore 3, il tempo di acquisizione di una telecamera con sensore *monochrome* rispetto al modello a colori della stessa risoluzione non cambia.

¹⁴ Per approfondimenti si rimanda all'Appendice D. Vision platform FH data sheet

¹⁵ Si veda anche il paragrafo 2.2.1. Tecnica di lettura *Multi-slope*.

High-speed CMOS cameras

Model	FH-SM	FH-SC	FH-SM02	FH-SC02	FH-SM04	FH-SC04	FH-SM12	FH-SC12
Image elements	CMOS image elements (1/3-inch equivalent)		CMOS image elements (2/3-inch equivalent)		CMOS image elements (1-inch equivalent)		CMOS image elements (1.76-inch equivalent)	
Color/Monochrome	Monochrome	Color	Monochrome	Color	Monochrome	Color	Monochrome	Color
Effective pixels	640 (H) × 480 (V)		2040 (H) × 1088 (V)		2040 (H) × 2048 (V)		4084 (H) × 3072 (V)	
Imaging area H x V (opposing corner)	4.8 × 3.6 (6.0 mm)		11.26 × 5.98 (12.76 mm)		11.26 × 11.26 (15.93 mm)		22.5 × 16.9 (28.14 mm)	
Pixel size	7.4 (μm) × 7.4 (μm)		5.5 (μm) × 5.5 (μm)		5.5 (μm) × 5.5 (μm)		5.5 (μm) × 5.5 (μm)	
Shutter function	Electronic shutter; Shutter speeds can be set from 20 μs to 100 ms.		Electronic shutter; Shutter speeds can be set from 25 μs to 100 ms.				Electronic shutter; Shutter speeds can be set from 60 μs to 100 ms.	
Partial function	1 to 480 lines	2 to 480 lines	1 to 1088 lines	2 to 1088 lines	1 to 2048 lines	2 to 2048 lines	4 to 3072 lines (4-line increments)	
Frame rate (image read time)	308 fps (3.3 ms)		219 fps (4.6 ms) *		118 fps (8.5 ms) *		38.9 fps (25.7 ms) *	
Lens mounting	C mount						M42 mount	
Field of vision, installation distance	Selecting a lens according to the field of vision and installation distance							
Ambient temperature range	Operating: 0 to 40 °C, Storage: -25 to 65 °C (with no icing or condensation)							
Ambient humidity range	Operating and storage: 35% to 85% (with no condensation)							
Weight	Approx.105 g		Approx.110 g				Approx.320 g	
Accessories	Instruction manual							

* Frame rate in high speed mode when the camera is connected using two camera cables.

Figura 17. Caratteristiche geometriche ed elettriche dei sensori.

2.1.3 L'ambiente di sviluppo

Il sistema di visione offre un'ambiente per lo sviluppo delle applicazioni di visione artificiale che segue la filosofia degli algoritmi facenti parte del settore di sviluppo Think and See®¹⁶, e mirati ad una semplice comprensione e programmazione di strumenti sofisticati, ed alla flessibilità di un sistema aperto. Vi è la possibilità di gestire diverse classi utente, con le relative restrizioni e permessi. È possibile memorizzare e personalizzare fino a 8 layout di visualizzazione, al fine di poter visualizzare diversi strumenti con diverse modalità e mole di informazioni diverse. Nelle figure a seguito, vengono riportate a titolo di esempio la schermata principale, da cui si ha accesso a tutte le funzionalità del sistema, ed una schermata "operatore" personalizzata, nella quale troviamo solo la visualizzazione di ciò che avviene in macchina, ma non abbiamo la possibilità di interagire col sistema.

I primi elementi che troviamo in alto a sinistra sono i classici pulsanti di sistema (*File, Function, Tool, Window*), dai quali si accede alle diverse funzionalità di sistema, quali impostazioni hardware, avvio del sistema, connessioni, layout, etc. In alto a sinistra troviamo poi il risultato del controllo di visione ("OK"). Accanto a questo alcune informazioni in merito al formato, o "ricetta"¹⁷, tempo di elaborazione, etc..

¹⁶ Per un approfondimento si rimanda all'Appendice A.Technology overview Pattern matching Algorithm Shape search III.

¹⁷ Volgarmente così chiamata dai professionisti del settore.

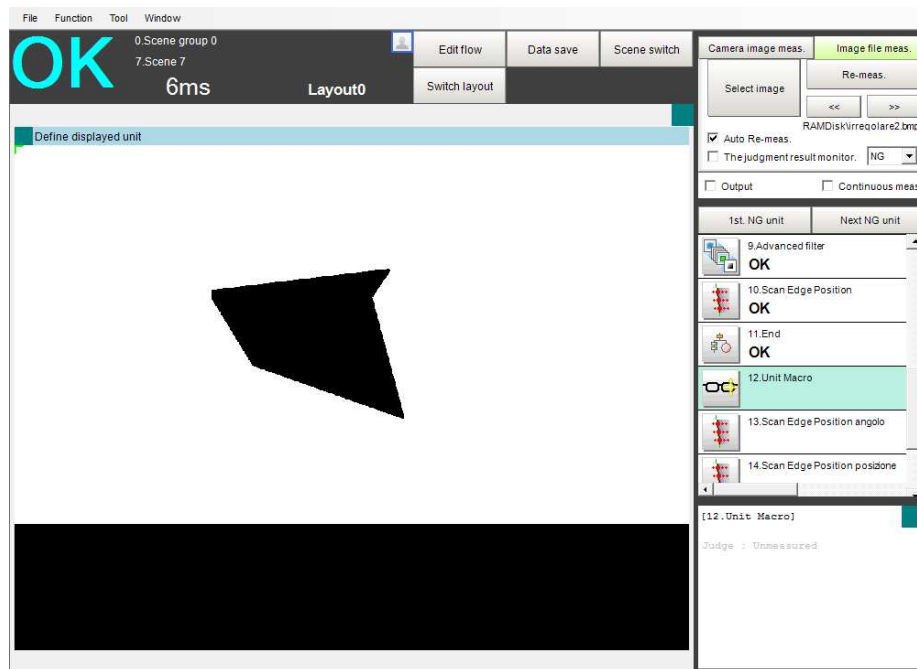


Figura 18. Interfaccia utente di sviluppo del sistema di visione FH-3050

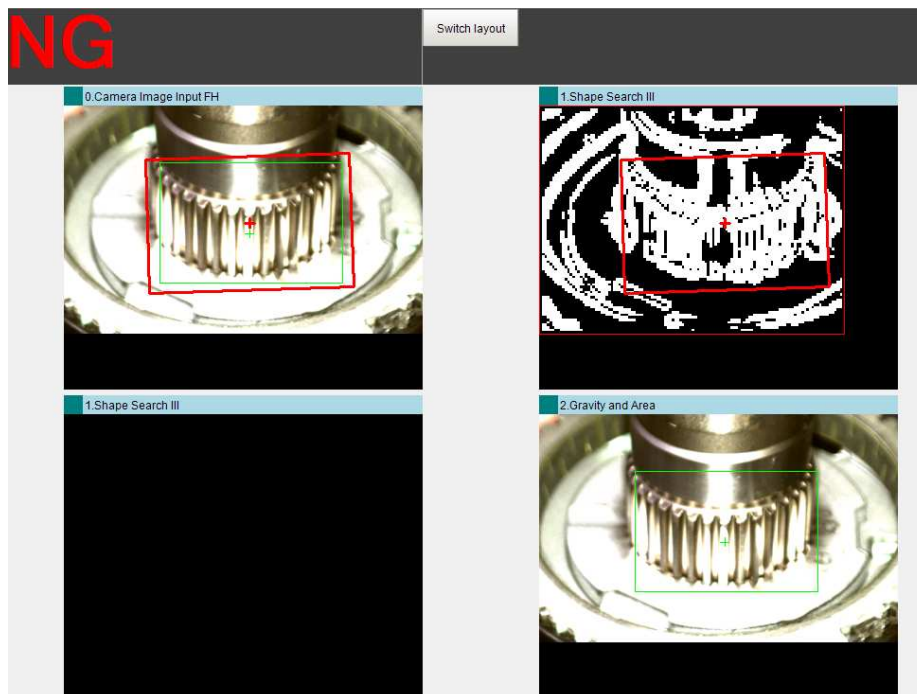


Figura 19. Interfaccia operatore implementata su sistema di visione FH-3050.

Sulla destra, troviamo invece dei tasti per interagire con delle funzionalità del controllo visivo, tra i quali quelli relativi agli algoritmi facenti parte della “ricetta”. È possibile programmare e formulare l’insieme di operazioni che deve eseguire il sistema, accedendo alla sezione “*Edit Flow*”. Vi troveremo all’interno la libreria di algoritmi di elaborazione presenti all’interno dell’ambiente di sviluppo. Da questa sezione sarà possibile comporre il controllo, e sviluppare il flusso di operazioni che dovrà svolgere in sequenza il sistema di visione, secondo la logica assegnata. In figura 20 viene riportata l’interfaccia di sviluppo e programmazione del controllo di visione “*Edit Flow*”:

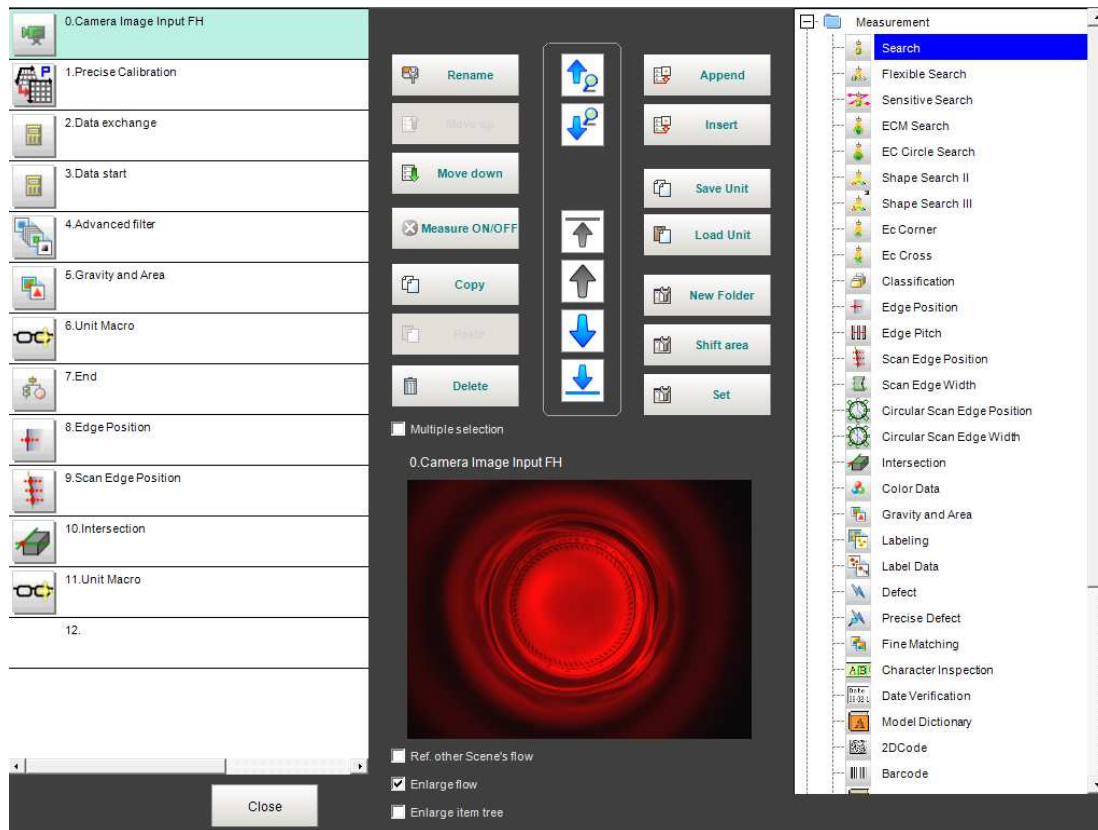


Figura 20. Interfaccia “Edit flow”.

2.2 Gli algoritmi di elaborazione utilizzati

All'interno del sistema di visione Omron FH troviamo una libreria composta da un centinaio di algoritmi di elaborazione delle immagini, nonché di compensazione, filtraggio, e supporto ai controlli, misure e operazioni affini come scambio dati, *logging*, etc¹⁸. L'ambiente di sviluppo dell'applicazione di visione, utilizza interfacce grafiche di impostazione, nelle quali troviamo degli elementi ricorsivi, che facilitano la programmazione. Per ogni algoritmo si ha la possibilità di visualizzare in tempo reale “l'effetto”, il funzionamento, al fine di ottenere un'agevole impostazione non solo dei parametri di controllo, ma anche delle eventuali soglie di giudizio. Fuorché non si tratti di algoritmi di supporto al controllo, in sede d'impostazione troviamo quindi una prima schermata ove è visualizzata l'immagine con una finestra di navigazione. Infine, per ogni algoritmo, troviamo l'ultima finestra di impostazione che è comune a tutti, dove vengono riportate le opzioni di:

- *Calibration*: se l'algoritmo lavora su un'immagine compensata¹⁹, o se fornisce delle coordinate in uscita, è possibile ottenere i risultati secondo il sistema di riferimento del sistema di visione (pixel e/o immagine non compensata), o secondo il sistema di riferimento ottenuto in seguito a calibrazione del sistema.
- *Reflect to overall judgment*: è necessario fare due distinzioni in termini di “giudizio” del controllo. Il controllo in toto sull'immagine, compresa l'elaborazione ed un eventuale scambio di dati, è composta da una serie di algoritmi organizzati secondo una logica dettata

¹⁸ Per approfondimenti si rimanda all'Appendice D. Vision platform FH data sheet.

¹⁹ Al paragrafo 2.2.2. Precise Calibration, verrà fornito un accenno su un meccanismo di calibrazione.

dallo sviluppo dell'applicazione. Otterremo quindi un giudizio “*globale*” del controllo, ma ogni algoritmo fornirà un suo giudizio su ciò che compete (il giudizio “*locale*”). Il giudizio globale sarà la moltiplicazione logica dei singoli risultati.

Includendo o escludendo questa voce, decidiamo se l'algoritmo in questione deve o non deve partecipare al giudizio globale. Ciò non significa che l'algoritmo viene escluso dal controllo, ma che il suo giudizio non compete al fine del giudizio globale del controllo in toto.

Verranno di seguito riportati gli aspetti salienti degli algoritmi utilizzati nello sviluppo del progetto.

2.2.1 Camera image Input FH

In seguito all'acquisizione di un impulso di *trigger*²⁰, il sistema di visione lancia in elaborazione in sequenza gli algoritmi che troviamo organizzati all'interno del flusso di lavoro secondo la logica conferita. Il primo passo di ogni flusso di processo di controllo, inizia con questo algoritmo: l'acquisizione dell'immagine da parte della/e telecamera/e. Verrà trattato in questa sede solo il caso di acquisizione da parte di una sola telecamera. All'interno, troviamo i parametri di acquisizione e calibrazione divisi in due sezioni. Per quanto concerne la calibrazione, è stata adottata per lo sviluppo di tale lavoro la modalità di calibrazione con compensazione della distorsione, si rimanda quindi al manuale per questa prima modalità di calibrazione²¹. Per quanto invece riguarda i parametri di acquisizione, troviamo:

- *Shutter speed*: ovvero la velocità dell'otturatore. Il sensore verrà esposto alla radiazione per un tempo pari allo *shutter speed*. Per quanto riguarda le telecamere adottate, i valori contemplati per questo parametro sono riportati in Tabella 2.

Tabella 2. Valori ammissibili di “*shutter speed*”.

Item	Setting value [Factory default]
Shutter speed	For stand-alone cameras 20 to 100000 [μs] [2000]
	For FZ-SF/SP/S2M/SC2M/S5M/SC5M 20 to 100000 [μs] [8333]
	For intelligent compact cameras 35 to 4000 [μs] [1000]

- *Gain*: ovvero il parametro di guadagno. È il fattore moltiplicativo (ideale) tra il numero di fotoni ed il numero di cariche generate all'interno del singolo fotodiodo del sensore. Solitamente, per non introdurre ulteriore rumore, viene utilizzato il guadagno solo laddove vige la necessità di mantenere “chiuso” lo shutter speed e non vi è una condizione di illuminazione tale da poter captare al meglio i particolari desiderati (in via del tutto generale, ma si possono manifestare diverse casistiche).
- *Mirror image*: l'immagine può esser acquisita già ruotata dell'angolo desiderato, a passi di 90°. Tale operazione è possibile grazie ad una acquisizione hardware da parte della telecamera. La lettura della matrice del sensore, può avvenire secondo la direzione desiderata considerato come “uscita” spaziale del sensore uno dei quattro angoli.
- *Multi slope*: è la capacità autoadattativa del range dinamico del sensore. In seguito ad una eccessiva intensità, cambia il fattore moltiplicativo di generazione delle cariche per effetto

²⁰ A meno che il sistema non lavori in acquisizione continua.

²¹ Pag. 34. Z341-E1-01. Vision Sensor FH/FZ5 Series Vision System Processing Item Function Reference Manual

dell'interazione fotone – reticolo. Lo stesso vale per una scarsa intensità di radiazione ricevuta. Viene di seguito riportato l'estratto del manuale che descrive la funzione:

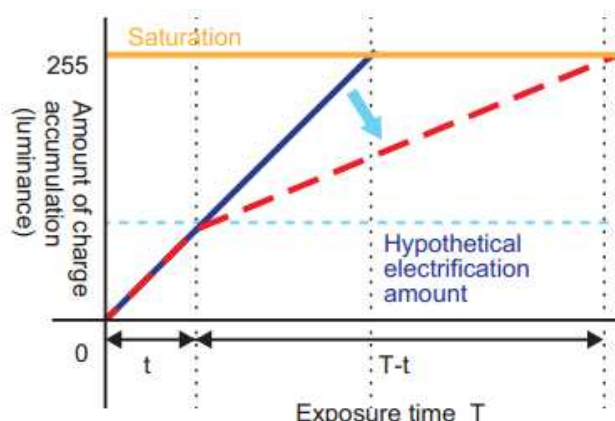


Figura 21. Meccanismo di ottimizzazione del range dinamico *multi slope*.

- *Number of lines to be read*: è possibile acquisire l'informazione relativa a settori del sensore, piuttosto che l'intera risoluzione dello stesso. Nell'ottica di velocizzare i tempi di acquisizione da parte della telecamera (a cui va sommato il tempo dello *shutter speed*), laddove è possibile utilizzare solo parte del sensore per l'acquisizione, vengono così acquisite meno righe. È qui in questa sede possibile scegliere il numero di righe da acquisire, in cui il valore minimo va da 1 a 2 per telecamere *monochrome* o a *colori*, ed il numero massimo dipende dalla risoluzione del sensore, ad esempio per un sensore VGA è 480 (ris. 640X480).
- *Electronic Flash Setting*: vengono qui definiti i parametri dell'impulso da fornire all'eventuale *strobe* (illuminatore) collegato alla telecamera (durata, delay, polarità).

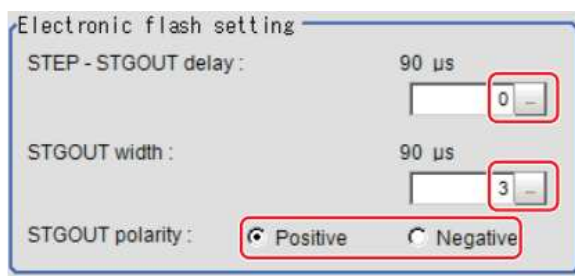


Figura 22. Impostazione dei parametri di impulso per lo *strobe*.

La regolazione di tali parametri avviene in concomitanza alla regolazione del diaframma e della messa a fuoco (ed eventuale regolazione del polarizzatore sull'ottica) mediante l'utilizzo delle ghiera sulle ottiche.

2.2.2 Precise Calibration

In seguito all'acquisizione dell'immagine da parte della telecamera, è buona norma compensare ove possibile, gli effetti della distorsione delle ottiche.

Inoltre, dove per ragioni di spazio vige la necessità di installare la telecamera con l'asse ottico non perpendicolare rispetto alla superficie da analizzare (si riporta per approfondimento alla regola di Scheimpflug [11]), otteniamo la cosiddetta "distorsione trapezoidale". In figura 23 troviamo

l'effetto della combinazione degli effetti citati. Al fine di compensare tali distorsioni, l'algoritmo "Precise Calibration", implementa uno strumento di ausilio semplice ed efficace.

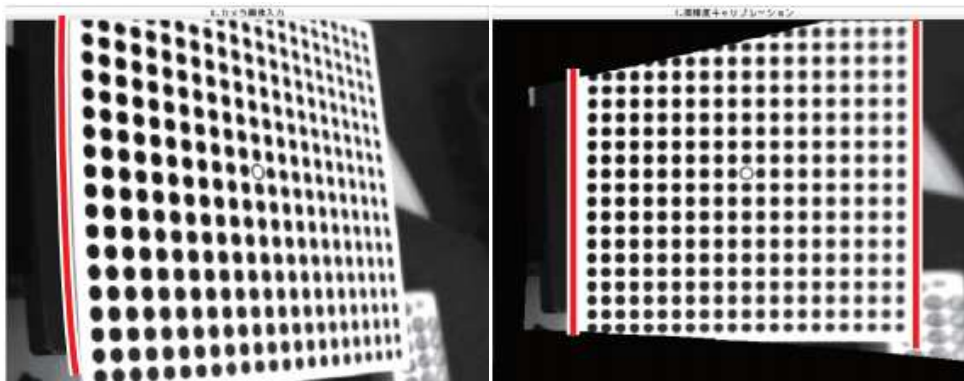


Figura 23. Immagine acquisita ed immagine elaborata dall'algoritmo *Precise Calibration*.

Viene utilizzato un pattern di calibrazione, composto da una matrice quadrata di cerchi delle stesse dimensioni, con dimensioni della matrice dispari, come visibile in figura 23. Sottoponendo il sistema all'acquisizione di tale pattern, esso rileva automaticamente i punti del pattern, ed effettua una compensazione software degli effetti citati. L'effetto è visibile in figura 23 sull'immagine di destra. Inoltre, l'algoritmo fornisce già in sede di acquisizione del pattern, e relativa generazione dei parametri di calibrazione, l'errore massimo riscontrabile secondo la direzione x ed y. Una volta inserita inoltre, la distanza tra ogni cerchio del pattern, gli algoritmi eseguiti in seguito a questo forniranno risultati ed eseguiranno elaborazioni grafiche sfruttando il sistema di riferimento scelto.

Calibration status		
Calibration parameters are generated		
Precision error	X max. (mm)	0.0110
	Y max. (mm)	0.0100

Figura 24. Errori residui di calibrazione.

2.2.3 Advanced Filter

In ambito elaborazione di immagini, i filtri sono degli strumenti molto utilizzati, al fine di ottimizzare il processo di elaborazione successivo. Oltre ai filtri che agiscono sulle caratteristiche cromatiche dell'informazione dei singoli pixel, all'interno di questo *macro* algoritmo troviamo una libreria di filtri completa. In figura 25 viene riportata la lista, da un estratto della schermata visibile sul sistema di visione.

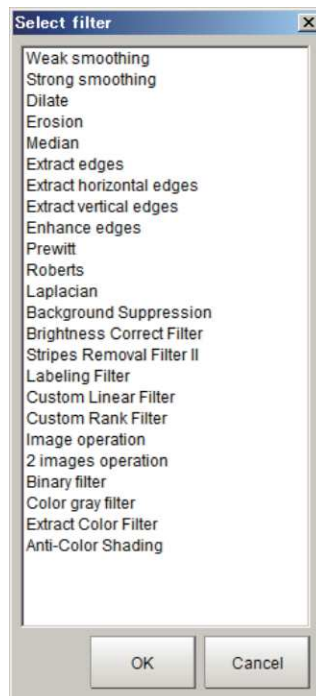


Figura 25. Interfaccia di scelta del tipo di filtro.

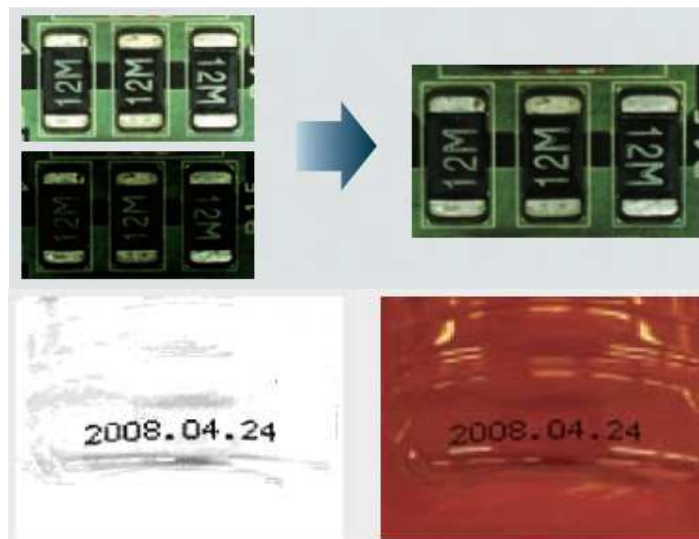


Figura 26. Applicazione di filtri su campioni reali.

Una volta impostata la regione dove il filtro deve avere effetto, è possibile agire sui parametri di filtraggio, caratteristici per ogni filtro nello specifico. Tra i parametri comuni di regolazione troviamo la “maschera”. L’operazione di filtraggio di per se agisce non sulla informazione locale del pixel, bensì delle informazioni relative ai pixel adiacenti. In quest’ottica è possibile parlare di maschera 3x3,5x5,7x7,etc..

2.2.4 Gravity and Area

È questo l'erede del più antico algoritmo implementato sui sistemi di visione artificiale in ambito automazione industriale. Definita la regione di azione dell'algoritmo, esso effettuerà una misura quantitativa di pixel adiacenti tra loro della(e) gamma(e) cromatica selezionata. Quindi in seguito alla definizione della regione, è necessario scegliere il colore dei pixel che si desidera siano contati. Il colore è selezionabile sull'immagine stessa, ed è possibile poi raffinare l'estrazione servendosi della tavola dei colori come visibile in figura 27. Oppure inserendo direttamente i valori secondo il sistema di riferimento dello spazio colore desiderato (Spazio RGB o HSV). In termini di elaborazione, l'immagine *in pasto* all'algoritmo verrà quindi filtrata estraendo i pixel del(i) colore(i) prescelto(i), effettuerà una binarizzazione al fine di rendere snello il conteggio dei pixel.

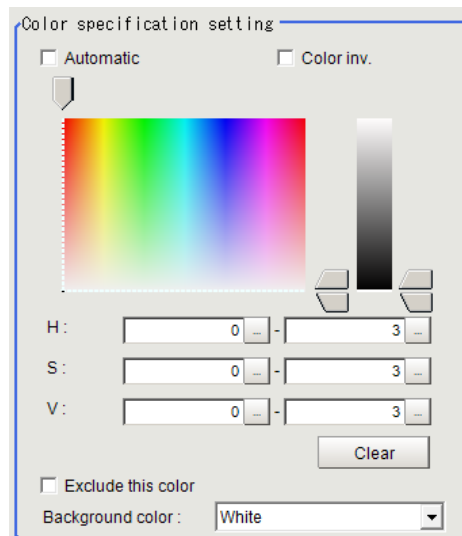


Figura 27. Interfaccia di scelta delle gamme cromatiche.

Fornirà quindi l'informazione riguardante il numero di pixel contati, per il quale potrà esser contemplata una regola di giudizio, e dal quale possiamo avere una informazione diretta dell'area della "macchia" rilevata. Otterremo inoltre il "centro di gravità", ovvero il baricentro, della stessa macchia.

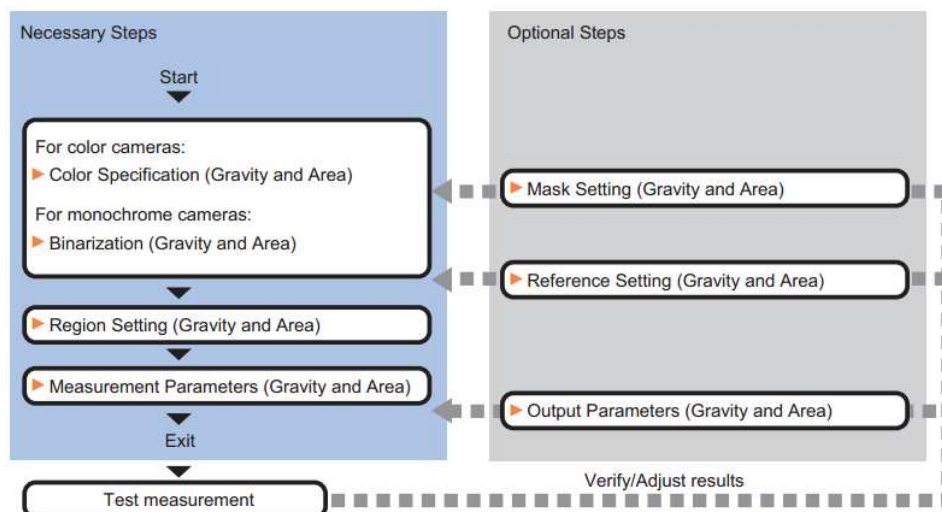


Figura 28. Flusso di impostazione e di azione dell'algoritmo *Gravity and Area*.

2.2.5 Edge Position

Questo è il principe degli algoritmi di visione artificiale di misura. Viene utilizzato laddove è necessario rilevare la coordinata x,y di un punto, che si trova in prossimità di una discontinuità, in termini di valori relativi alla matrice di immagine. All'interno della sua regione di ricerca (rettangolo verde in Fig. 29), e muovendosi lungo la direzione assegnata, viene rilevata la coordinata x,y del punto ove la derivata prima è massima (riferendosi alla matrice dell'immagine). In figura 29 viene riportato un particolare dell'azione dell'algoritmo:

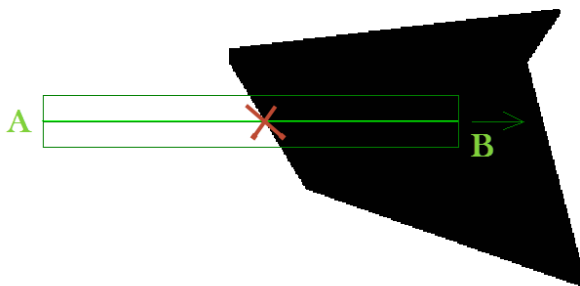


Figura 29. Regione di elaborazione dell'algoritmo *Edge Position*.

Dove è stata marcata con una x rossa il punto rilevato dall'algoritmo. In termini di parametri di giudizio, è possibile agire su posizione x ed y massima e minima. È possibile scegliere il colore, o la gamma cromatica, in prossimità del quale dev'esser riconosciuto l'"edge".

Ed è altresì possibile selezionare la modalità di funzionamento tra le due seguenti: color in e color out, rispettivamente per ottenere il punto dove la derivata è positiva o negativa (in un sistema binario composto dal colore(i) scelto(i), e dal complementare dello spazio delle gamme cromatiche²²).

2.2.6 Scan Edge Position

Componendo ed utilizzando più di un *edge position*, all'interno di una regione è possibile ottenere n punti per quanti n *edge position* vengono scelti. Questo algoritmo funziona così, ed è quindi in grado di fornire non solo il valore di una coordinata x, y , ma un set di tre coordinate, rispettivamente per la coppia x,y minima, media e massima tra i punti rilevati. Oltre al set di coordinate, l'algoritmo fornisce il valore dell'angolo della retta interpolata tra i punti rilevati dagli *edge position*. Ed è in fine in grado di fornire informazioni relative al numero di *edge position* che non rilevano dei punti di discontinuità muovendosi lungo la direzione di scansione²³ (*lost point*). In figura 30 viene riportato l'estratto dell'esempio di applicazione dal manuale:

²² Si faccia riferimento al manuale Z341-E1-01. Vision Sensor FH/FZ5 Series Vision System Processing Item Function Reference Manual

²³ Si faccia riferimento al manuale Z341-E1-01. Pag. 203. Vision Sensor FH/FZ5 Series Vision System Processing Item Function Reference Manual

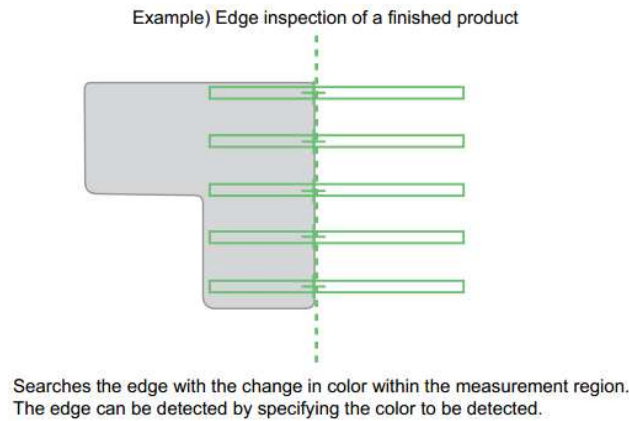


Figura 30. Esempio di applicazione dell'algoritmo *Scan Edge Position*.

2.2.7 Intersection

Questo algoritmo, servendosi dei risultati di due *scan edge position*, costruisce le rette interpolate dai punti rilevati, e, mettendo a sistema le rette, fornisce automaticamente il punto di intersezione. In sede di impostazione dell'algoritmo verranno quindi definite le regioni dei due *scan edge position*, con le relative caratteristiche di ricerca. Come risultati, l'algoritmo fornisce i valori degli angoli delle rette rilevate, punti di intersezione degli *scan edge position*, etc.²⁴.

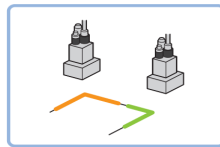


Figura 31. Applicazione dell'algoritmo *Intersection*.

2.2.8 Color data

Una volta definita la regione di misura, è eseguita una misurazione locale (in termini di pixel) del colore. Fornendo quindi una media spaziale dei due seguenti parametri, sui quali è possibile intervenire in termini di parametri di giudizio del controllo: *difference* e *deviation*. Il primo si riferisce alla differenza tra la collezione dei colori (intesi come singole armoniche) all'interno della regione di misura, e la collezione dei valori delle armoniche sui controlli successivi alla registrazione del *master*. *Deviation* invece si riferisce alla deviazione massima in termine di lunghezza d'onda, tra le armoniche rilevate all'interno della regione di misura.

È possibile quindi agire su questi due parametri per il giudizio del controllo, ma è possibile ottenere numerose informazioni tra le quali i valori secondo lo spazio colori desiderato²⁵.

²⁴ Si faccia riferimento al manuale Z341-E1-01. Pag. 253. Vision Sensor FH/FZ5 Series Vision System Processing Item Function Reference Manual

²⁵ Si faccia riferimento al manuale Z341-E1-01. Pag. 266. Vision Sensor FH/FZ5 Series Vision System Processing Item Function Reference Manual.

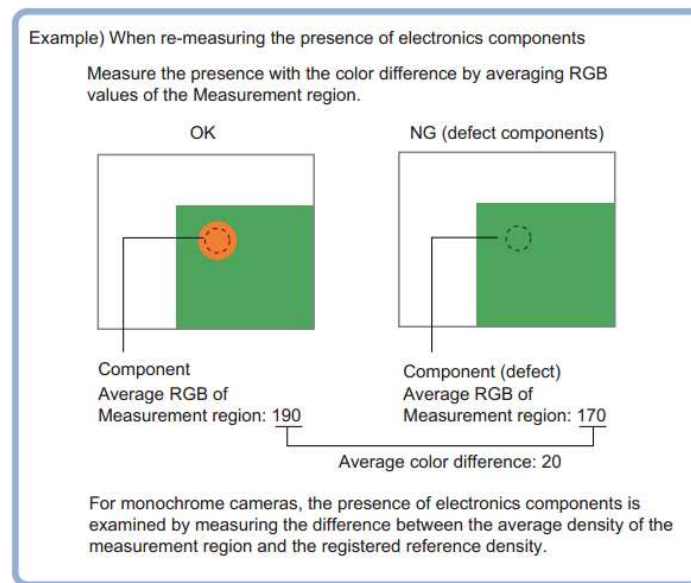


Figura 32. Applicazione dell'algorithmo *Color Data*.

2.2.9 Macro e Calculation

Gli algoritmi all'interno del flusso verranno eseguiti in sequenza secondo la logica, e l'intelligenza conferita la controllo in sé. Il conferimento di un grado di adattabilità del controllo, avviene mediante l'implementazione di meccanismi di *feedback* interni al controllo stesso, e tra i vari passi del diagramma di flusso. Piuttosto che un flusso unidirezionale di processo (in termini di controllo ed elaborazione dell'immagine), il controllo comincia ad assumere adattabilità sulla base di condizioni che possono verificarsi (il classico *If..Then..Else..*). Possono esser richieste delle operazioni cicliche – i classici *For..While..etc...* Strutture che si servono di variabili interne di retroazione fornite dagli algoritmi stessi.

Un esempio classico, se non altro quasi ideale²⁶, potrebbe esser l'implementazione di un meccanismo di adattabilità alle condizioni di illuminazione. Per realizzarlo, basterebbe effettuare una misura di luminosità su una zona fissa dell'inquadratura (o anche una media spaziale dell'intera inquadratura), e dall'informazione azionare un meccanismo di retroazione tale da regolare lo *shutter speed* della telecamera. Bene, servendoci dello strumento di compilazione "Macro", possiamo compilare in linguaggio strutturato la logica che vogliamo conferire al nostro controllo di visione. Viene inserito all'interno della libreria come algoritmo di supporto, al fine di poter ottenere anche una visualizzazione dei risultati dell'algorithmo stesso²⁷.

²⁶ Le variabili in gioco son ben più di quelle trattate in questo elaborato.

²⁷ Si faccia riferimento al manuale Z341-E1-01, Pag. 509. Vision Sensor FH/FZ5 Series Vision System Processing Item Function Reference Manual.

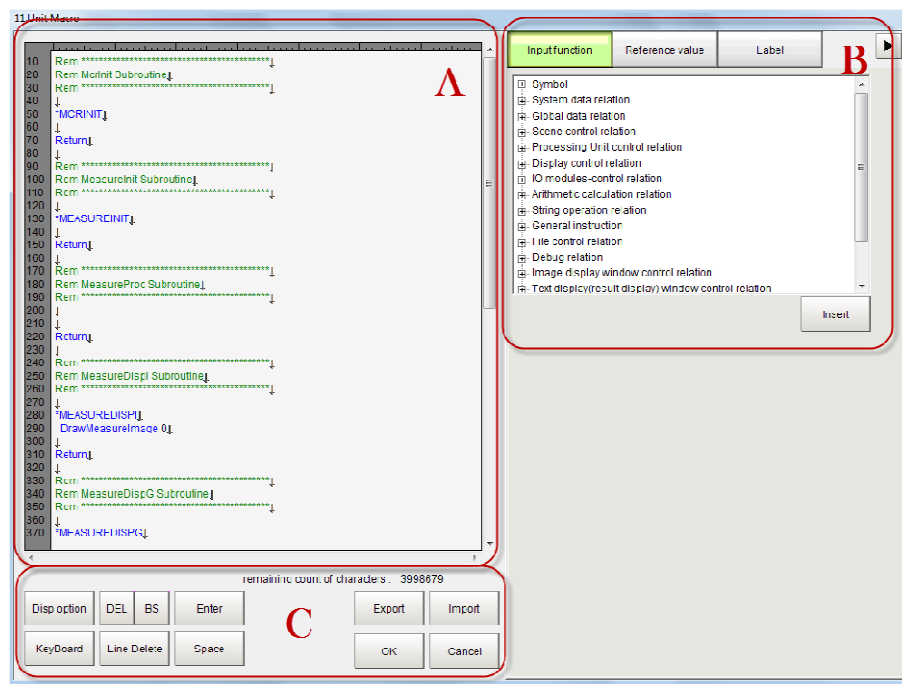


Figura 33. Interfaccia di programmazione *Macro*.

All'interno della "Macro" troveremo tre sezioni, visibili in figa 33, "A" il compilatore vero e proprio, "B" manuale e set di istruzioni disponibili, "C" pulsanti utili alla compilazione.

Per non dilungare troppo in questa trattazione, si rimanda il lettore alla lettura del manuale, e si sappia che le uniche operazioni a livello di sistema non contemplabili, sono le manipolazioni delle informazioni sui valori dei singoli pixel. Ovvero, in termini di misura e controllo sulle immagini, è possibile servirsi dei risultati degli altri algoritmi già presenti²⁸.

Per ultimo, l'algoritmo di supporto "Calculation", è un contenitore di 8 valori. È possibile accedere ad esso ed è possibile prelevare e memorizzare valori o risultati. All'interno di ognuno degli otto campi disponibili, è possibile definire inoltre una qualsiasi forma. Il risultato di ogni calcolo di ogni formula implementata, potrà altresì esser soggetto a giudizio secondo i parametri assegnati.

2.3 Il setup di acquisizione, misura e test

Da specifiche di progetto, si ha la necessità di analizzare delle forme di dimensioni massime pari a 200x200mm. La precisione richiesta è di $\pm 0.5\text{mm}$ sui lati dei poligoni. Queste informazioni sono già sufficienti per poter effettuare una stima della risoluzione della telecamera. A causa del rumore come margine di sicurezza si dimensiona la risoluzione del sensore tenendo in conto che la precisione richiesta sia interpretato almeno da 3 pixel. È questa una regola non scritta, ma che viene utilizzata per esperienza e senza considerare l'effetto (in tal caso benefico) di illuminazione e compensazione software. Quindi, avendo a disposizione la precisione richiesta, non dobbiamo far altro che impostare la seguente proporzione per definire il quantitativo in millimetri a cui corrisponde un pixel:

$$\text{numero pixel riga/colonna} : \text{lato inquadrato in mm} = 1 : \text{risoluzione pixel}$$

Si è voluto riportare la proporzione poiché essa rimane valida anche per fare il calcolo inverso. Ovvero a come ottenere la risoluzione pixel data la risoluzione del sensore ed il lato inquadrato.

²⁸ Per la creazione di algoritmi, esiste invece un ambiente di sviluppo di applicativi FH-AP1 – Application Producer.

Una volta ottenuto da questa, nel nostro caso, il numero di pixel riga/colonna richiesti al sensore è possibile scegliere il modello di telecamera adatta. Nell'ottica di trovare il corretto setup in termini di efficienza/prezzo, è corretto sottolineare che solitamente è possibile anche scegliere un numero maggiore a 1 di telecamere. Sull'acquisizione composita potranno così essere effettuate le elaborazioni desiderate. Nel caso in esame piuttosto come lato inquadrato non è stato scelto 200 mm come da lato massimo da specifica, bensì 300. Questo perché il poligono potrebbe presentarsi orientato in qualsiasi direzione. Il numero di pixel necessari risulta quindi $600 * 3 = 1800$. È stata pertanto scelta una telecamera con risoluzione 2040 x 2048 (Omron FH-S04). Al fine di non aggiungere ulteriori distorsioni è stata utilizzato un sistema di illuminazione di tipo "dome", i cui effetti sono stati esplicitati al paragrafo 1.4. In figura 33A viene mostrato il setup del sistema di misura:

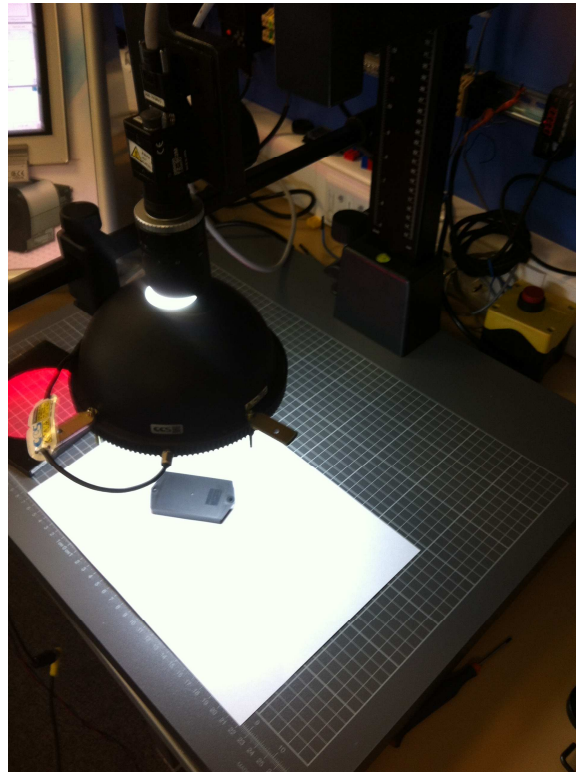


Figura 33A. Setup del sistema di misura.

Una eventuale e plausibile alternativa è il sistema di illuminazione "backlight". In tal caso, l'oggetto viene interposto tra la sorgente di illuminazione e la telecamera stessa. È così estratta la sagoma dell'oggetto. Un esempio si può vedere in figura 33B. È stata simulata una retro-illuminazione poggiando l'oggetto sull'illuminatore collimato.

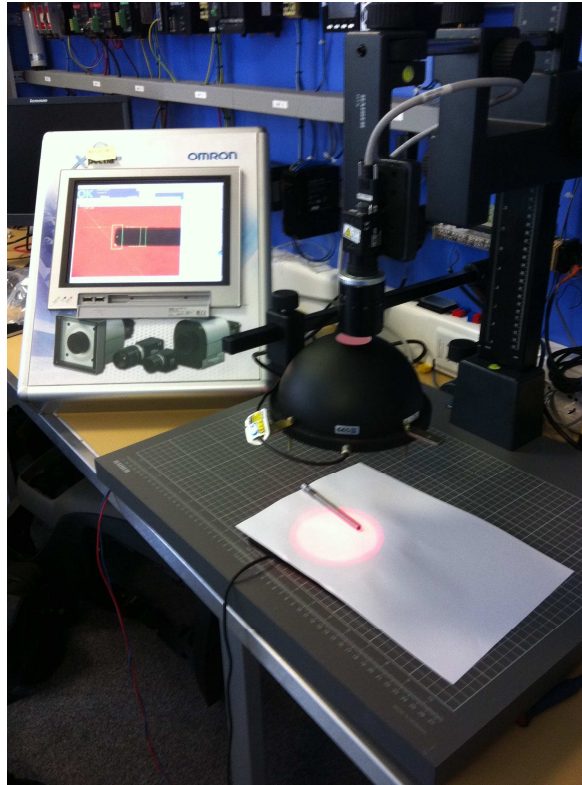


Figura 33B. Setup del sistema con retro illuminazione.

Per le specifiche in oggetto non è ammissibile scegliere un sistema di illuminazione di questa tipologia, visto che lo stesso si dovrebbe integrare con i nastri trasportatori. Si dovrebbe pertanto utilizzare un nastro che trasparente alla sorgente di radiazione, e sullo stesso di depositerebbero altresì altri scarti di lavorazione che andrebbero a inficiare sulla misura. L'ottica utilizzata è stata scelta servendosi del diagramma relativo alla telecamera scelta. Per motivi economici, si è preferito optare per il modello a bassa distorsione, e non una ottica telecentrica (ideale per la massima riduzione dell'effetto "cuscino"). La scelta viene altresì giustificata dall'adozione dell'algoritmo di calibrazione presente sul sistema di visione, come visto al paragrafo 2.2.2.

SEZIONE II

3. Elaborazione delle immagini

Le piattaforme di visione artificiale industriali offrono ad oggi diverse librerie di algoritmi atti al controllo ed alla elaborazione delle immagini acquisite. Tali strumenti molto utili per lo sviluppo di applicazioni, possiedono già i requisiti necessari e sufficienti per rendere le librerie complete sotto il profilo della elaborazione delle immagini. Le dinamiche di mercato e la tendenza dello sviluppo di nuovi macchinari industriali, dipingono uno scenario dove la visione artificiale è chiamata a fornire un feedback più reattivo – in perfetta linea con la teoria della controreazione. Vengono richiesti sistemi non più in grado di svolgere semplici compiti come i controlli qualità, con le dovute e relative difficoltà dei casi. I sistemi di visione di domani non saranno in grado quindi solo di controllare, ma di interpretare, elaborare, collezionare informazioni e processare dei risultati con una mole di contenuti sempre più crescente. Verranno in questo capitolo mostrati due esempi lampanti di sviluppo di un algoritmo per il rilevamento automatico di vertici di forme geometriche ignote. Verranno altresì mostrati pregi e difetti dei due metodi, contestualizzandoli a due applicazioni reali. La novità, oltre l'originalità del secondo metodo, risiede nel tentativo di un approccio ingegneristico per lo sviluppo e l'analisi dei risultati ottenuti con gli strumenti creati.

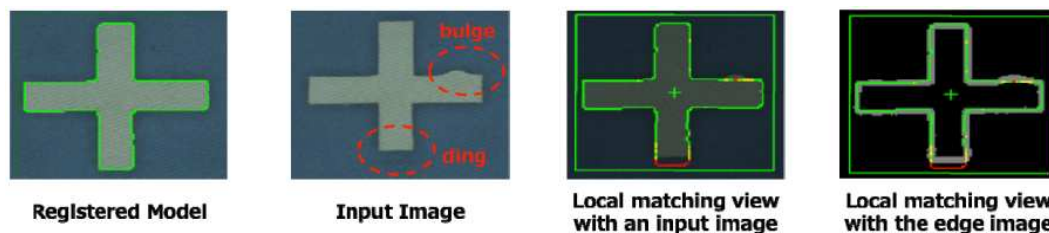


Figura 35. Esempio di visualizzazione dei risultati dell'algoritmo di pattern matching Shape search III.

“La chiave per un corretto (brillante) sviluppo di un programma di visione artificiale, risiede nella capacità di saper analizzare e scomporre il processo mentale e visivo, gli istinti, le impressioni, le percezioni, e nutrirsi di fantasia. Poi rimane solo da riportarlo su un terminale. “ [G2]

3.1 Il metodo “dei raggi” – settore applicativo

Dal punto di vista del rilevamento automatico dei vertici di poligoni ignoti, sono stati sviluppati due metodi che si prestano nel loro insieme alla risoluzione della totalità delle problematiche riportate in questo studio. L'incipit allo sviluppo nasce quindi da un requisito reale di mercato, e diviene spunto per lo sviluppo e la possibile contestualizzazione nel campo del controllo automatico dei processi della micro e nano elettronica. Il primo metodo di elaborazione che verrà presentato, battezzato come *“metodo dei raggi”*, viene suggerito per la risoluzione della problematica sotto definite e restrittive specifiche. Si vedrà che, a causa della natura del metodo, esso è efficace per poligoni che non presentano concavità al loro interno, ovvero delle forme semplici. Il secondo invece si dimostrerà universalmente efficace nel campo della sua applicazione, nonostante presenti dei costi computazionali maggiori. Non è altro che il compromesso che si deve rispettare: la precisione e la potenza di elaborazione a discapito del tempo di elaborazione, e della complessità nello sviluppo del metodo.

Le forme derivanti da sfridi protagoniste di questo progetto, possono presentarsi sotto forma di poligoni composti da un numero di lati non superiore a 4, senza presentare convessità. Tuttavia questo rappresenta solo il primo livello di un progetto più ampio, che verte sull'implementazione di tali sistemi i campi affini, che già presentano delle evidenti criticità. Passando poi al progetto visionario che trova invece spazio in applicazioni a lungo termine, il metodo dei raggi risulta quasi totalmente inefficace, se non immaginato come ausilio al secondo metodo denominato come il metodo “*ad inseguimento*”.

3.2 Il metodo “dei raggi”- sviluppo dell’algoritmo di elaborazione

Al fine di poter rilevare i vertici di un poligono - senza conoscere a priori le sue caratteristiche - il metodo “dei raggi” è il frutto di un’analisi attenta del funzionamento degli algoritmi presenti all’interno della libreria di sviluppo del programma applicativo del sistema di visione. Da esso deriva un metodo semplice, dal basso costo computazionale, e da un carattere di assoluta convergenza all’interno del campo di applicazione.

L’idea nasce dall’utilizzo degli algoritmi di rilevamento dei bordi, denominati all’interno della libreria come “*scan edge*” e “*scan edge position*”, al fine di rilevare dei punti interessanti della caratteristica esterna del poligono. Il primo viene utilizzato per rilevare punti del contorno dei poligoni, mentre il secondo viene utilizzato per rilevare gli angoli relativi, rispetto al sistema di riferimento, normali alla direzione di incidenza degli *scan* di rilevamento dei punti del contorno. Tali algoritmi risultano i più indicati vista la funzionalità, la precisione e la velocità di elaborazione. Utilizzando i risultati forniti da tali algoritmi vengono così calcolati i vertici dei poligoni, esaminando la deviazione dell’angolo delle retta interpolate in uscita dallo *scan edge position* i ed $i + 1$.

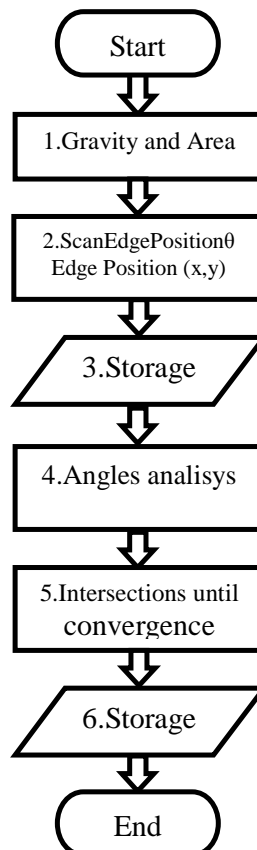


Figura 36. Diagramma di flusso del metodo “dei raggi”.

La semplicità del metodo, risiede nell'impostazione delle regioni d'ispezione da parte degli algoritmi di *scan*. Esse risiedono all'interno di una circonferenza che ingloba l'intero poligono. Dato il metodo di azione, la denominazione trova un *matching* perfetto con l'impostazione delle regioni di ispezione, come si vedrà graficamente in un esempio a seguire. In figura 36, è possibile vedere il diagramma di flusso dell'algoritmo di elaborazione:

Una volta acquisita l'immagine, ed una volta filtrata, verranno rilevate le coordinate del baricentro del poligono. Tale coordinata x,y , sarà il punto di arrivo per ogni *scan edge i*, mentre il punto dello *scan i* di partenza, risiederà su una circonferenza esterna al contorno del poligono. Le coordinate del baricentro, sono altresì di ausilio per compensare la posizione relativa del poligono, rispetto al punto centrale dello spazio di elaborazione dell'immagine. Una volta rilevati i punti e gli angoli, a seguito di una analisi dei dati elaborati, verranno utilizzati n algoritmi di “*Intersection*” per ogni angolo n rilevato. Tali algoritmi, permettono il raffinamento dell'analisi macro metrica effettuata con i risultati dei primi algoritmi. Di seguito verranno descritte le procedure adottate per l'impostazione automatica degli algoritmi, e per ogni passo verrà descritta la strategia di azione, ed i passi salienti del codice. In figura 37 viene riportato l'equivalente del precedente diagramma implementato con l'ambiente di sviluppo del sistema di visione FH ver.5.20:

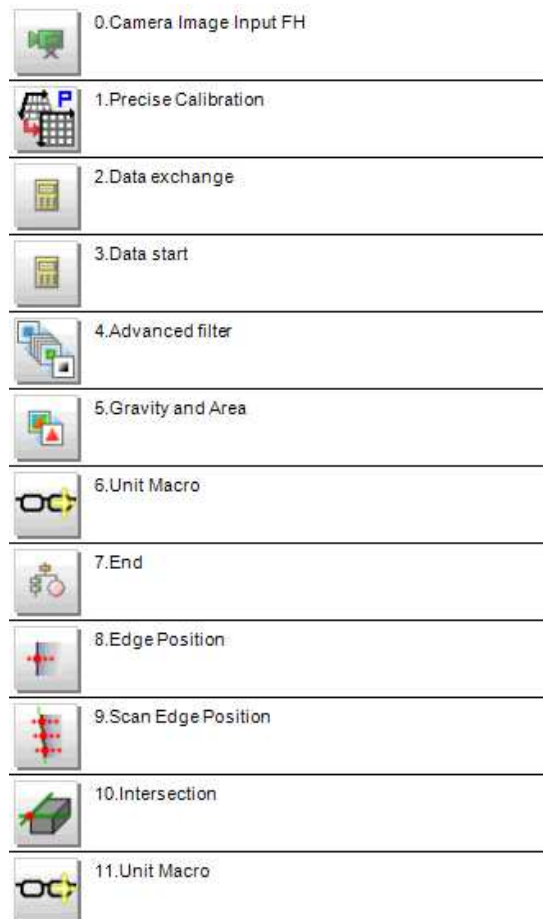


Figura 37. Diagramma di flusso implementato all'interno del software applicativo FH ver.5.20

3.2.1 “Start”

Per un utilizzo agevole e per un inserzione rapida dei parametri dello strumento utilizzato, è stata realizzata una piccola interfaccia grafica, dove troviamo dei campi per l’inserimento dei dati, la conferma o la cancellazione, nonché di tasti per il lancio di finestre di impostazione grafica personalizzata per gli algoritmi di partenza. In figura 38 è possibile osservare l’interfaccia, mentre il significato dei tasti sarà chiaro a breve. È possibile lanciare l’interfaccia di impostazione mediante l’utilizzo del tasto *Custom dialog tool* presente nell’interfaccia principale del software applicativo.

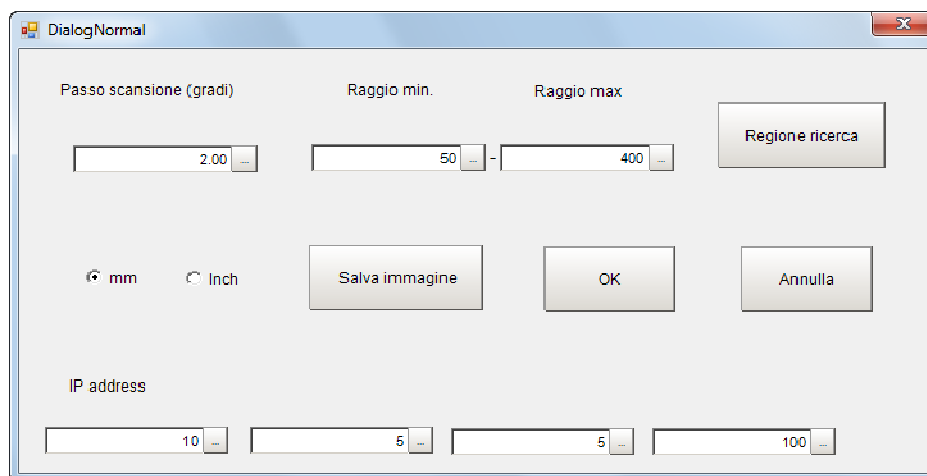


Figura 38. Interfaccia per l’impostazione dei dati di partenza dell’algoritmo dei raggi.

In figura 38 sono presenti i campi per l’impostazione dei parametri iniziali del metodo “dei raggi”. È possibile individuare le seguenti sezioni:

- *Passo scansione (gradi)*: all’interno di questo campo viene inserito il passo di scansione degli algoritmi di *scan*. Il punto di partenza dell’*i*-esimo *scan* risiederà su una circonferenza di opportune dimensioni, e saranno distanziati in angolo secondo il valore qui inserito.
- *Raggio min. e Raggio max.*: il cerchio citato avrà un raggio esterno pari al valore inserito in Raggio max. Benché il punto di arrivo dell’*i*-esimo *scan* sarà definito come il baricentro del poligono, per rendere l’applicazione più flessibile è stata prevista l’inserzione anche del Raggio min. Il punto di arrivo dell’*i*-esimo *scan* non sarà più in questo caso il baricentro del poligono, ma risiederà anch’esso su una circonferenza concentrica a quella esterna.
- *Regione Ricerca*: serve a “lanciare” un’interfaccia per la regione di ispezione del controllo. Tale interfaccia verrà presentata in Fig.43, interfaccia di selezione della regione di ispezione.
- *mm .. inch*: selezione dell’unità di misura secondo i due sistemi di riferimento implementati. I calcoli ed i risultati del metodo verranno forniti secondo il sistema qui selezionato.
- *Salva immagine*: salva l’immagine corrente all’interno dell’hard disk del sistema di visione, ed invia la stessa in forma originale ed elaborata su un percorso di rete definito internamente.
- *IP address*: seleziona l’indirizzo IP del dispositivo destinazione collegato in rete attraverso protocollo TCP/IP.

La fase di *start* del metodo non si ferma a queste impostazioni di partenza, vengono effettuate delle elaborazioni preliminari dell’immagine al fine di poter trovare il perfetto *matching* tra le potenzialità del metodo e le specifiche di progetto.

Per un'agevole lettura del testo, si faccia di seguito riferimento alle figure 36 e 37. In seguito all'evento di trigger, che comanda l'acquisizione dell'immagine da parte della telecamera, il programma applicativo del sistema di visione svolgerà in sequenza le operazioni degli algoritmi presenti nel flusso delle operazioni, seguendo la logica assegnata. Come si evince dalla figura 37, una volta acquisita l'immagine da parte della telecamera – passo 0. *Camera Image Input FH*, viene inserito l'algoritmo *1.Precise calibration*, al fine di attenuare l'effetto di distorsione dovuto alle ottiche. L'effetto di tale algoritmo è stato descritto al paragrafo 2.2.2.. Al passo 2 e 3 troviamo due algoritmi *Calculation*, denominati come “2.Data exchange” e “3.Data start”. Tali algoritmi, tra la categoria degli algoritmi di supporto, fungono da contenitori di variabili all'interno del programma, e sono accessibili sia dall'esterno mediante le interfacce di comunicazione, sia tra applicativi o parti diverse del programma. All'interno di ognuno di essi troviamo 8 campi, disponibili per la memorizzazione, lo scambio ed il calcolo in loco diretto. In figura 39 viene riportato il layout dell'algoritmo 2.Data Exchange.

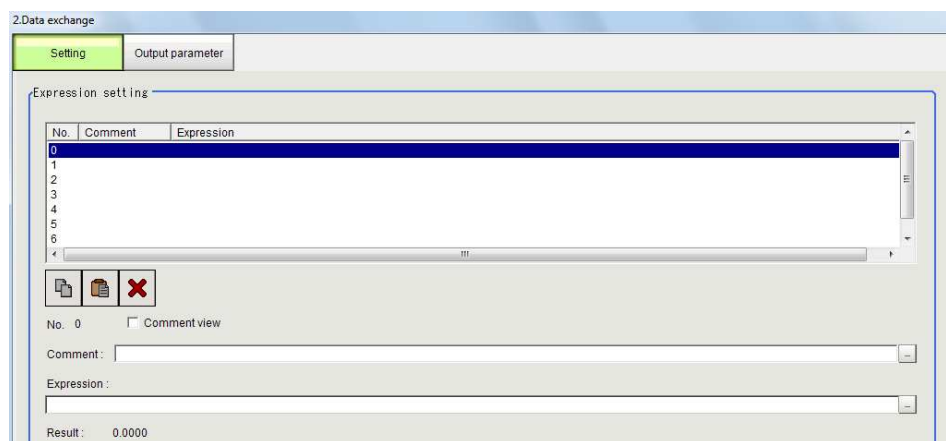


Figura 39. Layout dell'algoritmo *2.Data Exchange*.

All'interno di esso sono state definite le variabili di scambio con la schermata creata visibile in Fig.38, mentre all'interno del *Calculation* successivo sono stati inseriti alcuni valori utili anche in fase di *debug* del programma, come numero di vertici rilevati, somma totale degli angoli rilevati, etc.. Quest'ultimo valore è risultato molto utile nell'ottica di classificare i poligoni ignoti in base al numero di lati. Esiste infatti una correlazione stretta tra numero di lati e somma di angoli interni di un poligono.

È possibile accedere ad ogni campo del contenitore *Calculation - 2.Data Exchange*, utilizzando l'istruzione *UNITDATA*, seguita dal numero della posizione dell'algoritmo all'interno del flusso (in questo caso 2 o 3), e dal numero del campo. Ogni campo avrà un numero progressivo corrispondente, secondo quanto segue nella Tabella 3:

Tabella 3. External Reference Tables – Calculation.

External Reference Tables (Calculation)			
No.	Data name	Set/Get	Data range
5 to 12	Expression result of Expression 0 - Expression result of Expression 7	Set/Get	-999999999.9999 to 999999999.9999

Per accedere quindi in lettura ad un campo, ad esempio al primo dato presente all'interno di *2.Data exchange* in lettura, il comando corrispondente sarà:

UNITDATA 2 5 (1.1)

O alternativamente, in sede di compilazione delle Macro ²⁹, e volendo assegnare tale valore alla variabile A, è possibile utilizzare la seguente istruzione:

A& = UnitData (2,5) (1.2)

In fase di scrittura invece, basta invece aggiungere in coda alle precedenti il valore che si desidera assegnare o memorizzare nella variabile. Supponendo di volere scrivere il valore 7 come passo di scansione – corrispondente ad una distanza di 7° tra l'*i*-esimo *scan* - l'istruzione sarà:

UNITDATA 3 5 7 (1.3)

E questa volta, in sede di compilazione delle Macro, l'istruzione sarà :

SetUnitData 3 , 5 , 7 (1.4)

Ultimo passo della fase di “Start” dell’algoritmo è il filtraggio dell’immagine. Questo è un passo molto delicato, poiché al fine di eseguire delle misure con il grado di precisione richiesto, risulta rischioso distorcere ulteriormente l’immagine originale, ulteriormente aggravata dalle distorsioni già trattate. È stato quindi scelto un filtraggio di estrazione del colore dello sfondo. I contorni dei poligoni non vengono così alterati, ed otteniamo una immagine depurata dal mondo esterno che circonda i punti di interesse. L’elaborazione a seguire (anche a seguito di una ulteriore binarizzazione da parte dell’algoritmo 5.Gravity and Area), soffrirà meno dei residui eventuali di rumore, sarà snella nell’analisi matriciale visto lo scarso contenuto informativo (un bit per pixel). Nelle figure 40,41 e 42, viene riportato a titolo esemplificativo l’effetto del filtro dell’estrazione di alcuni particolari dell’immagine³⁰:



Figura 40. Effetto del filtraggio. Immagine originale.

²⁹ Al passo 6 del flusso di elaborazione troviamo la “Macro” – cuore e motore dell’elaborazione. Al suo interno troveremo numerose istruzioni simili ogni volta che si desidera accedere a delle variabili. Gli indirizzi e le denominazioni di tutte le variabili del programma e del sistema vengono rappresentate nel manuale utente

³⁰ L’esempio riportato in questa figura e nelle successive, è da ritenersi a titolo esemplificativo, non proviene quindi da un setup reale di misura e controllo di visione artificiale.

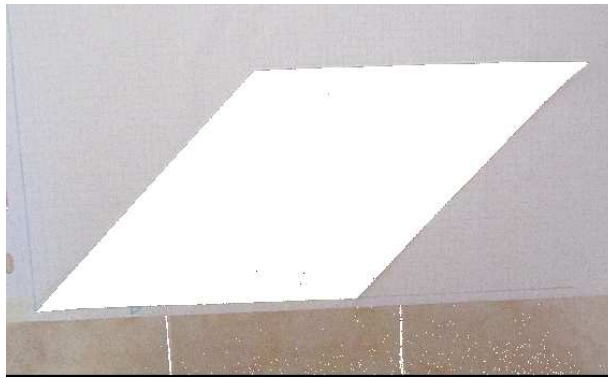


Figura 41. Effetto del filtraggio. Immagine filtrata.

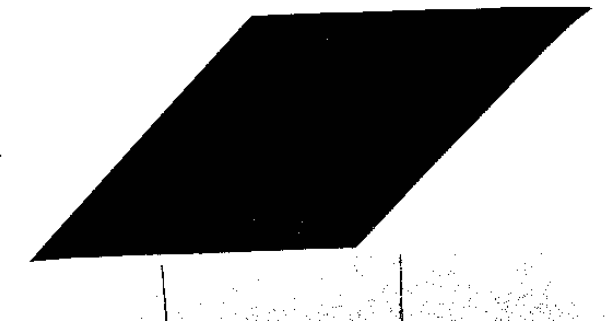


Figura 42. Effetto del filtraggio. Immagine estratta.

3.2.2 “Gravity and Area”

Punto di partenza a questo passo del programma è l’immagine proveniente dal filtraggio. L’algoritmo *Gravity And Area*, sarà capace di rilevare il baricentro del poligono all’interno dell’area di ricerca. Questa è definibile da utente accendendo all’algoritmo, oppure attraverso lo strumento di interfaccia sviluppato - *Custom Dialog Tool* (presente nella schermata principale del programma del sistema di visione)³¹. Entrambe le interfacce si presentano come mostrato in figura 43.

Ed in questa è possibile definire la regione di ispezione dell’algoritmo. Può essere utile rendere disponibile l’accesso all’impostazione di tale regione anche ad un livello operatore, nell’ottica di gestire un bassissimo livello di nuova impostazione di nuovi formati. Una volta definita la regione, alla sezione “Measurement” dell’algoritmo, vengono mostrati in tempo reale, in particolare baricentro in termini di posizione x ed y (contrassegnato visivamente con un cursore), e numero di pixel del poligono.

Quest’ultima informazione, è direttamente gestibile per calcolare l’area del poligono. Volendo implementare tale calcolo all’interno della Macro al passo 6, dovremmo prima prelevare il numero di pixel rilevati da parte dell’algoritmo *Gravity And Area*, e poi moltiplicarlo per un fattore moltiplicativo che dipende dal sistema di riferimento. Nel lavoro svolto, sono state implementate due costanti di proporzionalità per ottenere direttamente in millimetri ed in pollici la misura dell’area.

³¹ Si faccia riferimento al manuale Z341-E1-01. Pag. 277. Vision Sensor FH/FZ5 Series Vision System Processing Item Function Reference Manual

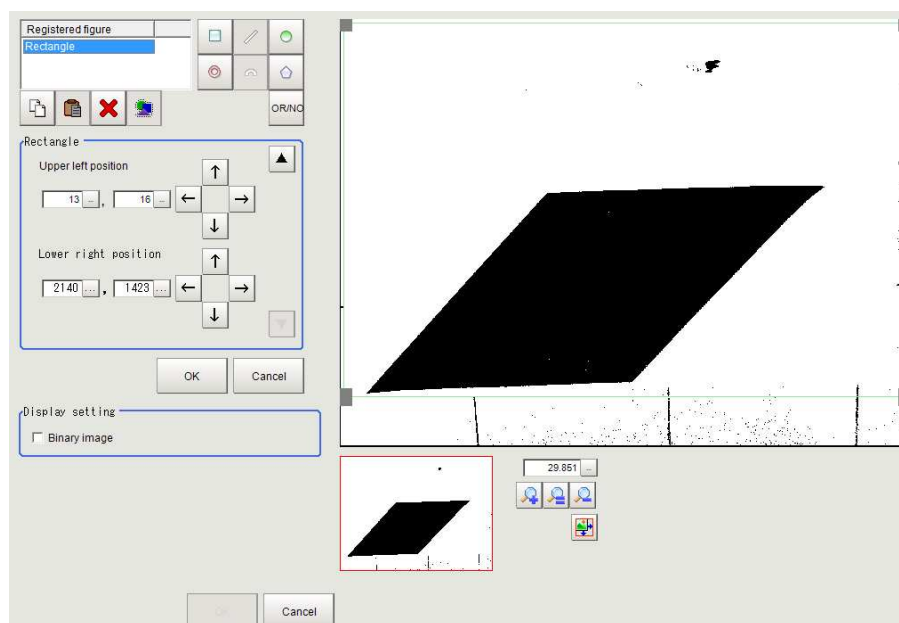


Figura 43. Interfaccia di selezione della regione di ispezione dell' algoritmo *Gravity and Area*.

Detto K_{conv} il fattore di conversione corrispondente (selezionabile anche da *Custom Dialog Tool*), il calcolo da effettuare sarà $N^{\circ} \text{pixel} * K_{conv}$. Per ottenere i risultati dall'algoritmo, viene di seguito riportato un estratto della tabella delle variabili disponibili:

Tabella 4. External Reference Tables – Gravity and Area.

No.	Data name	Set/Get	Data range
0	Judge	Get only	0: No judgement (unmeasured) 1: Judgement result OK -1: Judgement result NG
5	Area	Get only	0 to 999999999.9999
6	Gravity X	Get only	-99999.9999 to 99999.9999
7	Gravity Y	Get only	-99999.9999 to 99999.9999

E da questa, per assegnare il valore dell'area alla variabile AREA, utilizzeremo le seguenti istruzioni:

$$A\& = \text{UnitData}(5,5) \quad \text{'prelievo "Area" 5 da alg.pos.5} \quad (1.5)$$

$$\text{AREA}\& = A\& * K\text{conv}\& \quad \text{'Kconv dipende da mm o inch} \quad (1.6)$$

Rilevare il baricentro ha un duplice scopo all'interno dell'algoritmo sviluppato. Il primo, ed in particolare per questa procedura, è rilevare il punto di arrivo per l' i -esimo *scan*. Il secondo, è quello di rendere possibile una compensazione della posizione del poligono. Il poligono potrebbe infatti trovarsi ai bordi dell'immagine acquisita, non consentendo l'accesso da parte dell' i -esimo *scan* nelle parti prossime ai bordi. L'immagine verrà quindi letteralmente traslata di una quantità che dipende dalla posizione del baricentro stesso. Questo aspetto sarà chiaro analizzando il meccanismo di scansione dei punti nel prossimo paragrafo.

3.2.3 “Scan Edge Position, Edge position”

In questo paragrafo viene illustrato il metodo dell'impostazione degli algoritmi utilizzati per il rilevamento dei punti di bordo e degli angoli delle rette interpolanti i punti di incidenza. Prendiamo in esame dapprima l'effetto e l'efficacia degli algoritmi utilizzati. L'algoritmo “Edge Position” è capace di rilevare³², all'interno della sua regione di ricerca, e muovendosi lungo la direzione assegnata, la coordinata x,y del punto ove la derivata prima è massima (riferendosi alla matrice dell'immagine). In figura 44 viene riportato un particolare dell'azione dell'algoritmo:

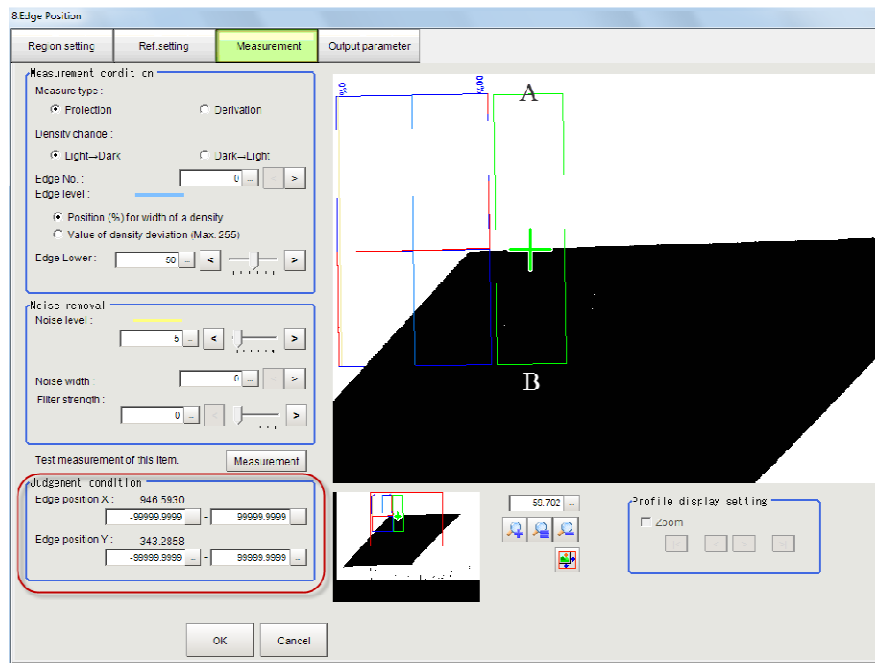


Figura 44. Particolare di impostazione dell'algoritmo “edge position”.

Facendo riferimento alla figura 44, l'algoritmo - muovendosi dal punto A al punto B, all'interno della regione di ricerca delimitata dal rettangolo in verde - rileva la posizione del punto in cui la derivata è massima (come visibile dal grafico a lato della regione verde). Il cursore verde in figura, indica la posizione del bordo rilevato, ed in basso a sinistra vengono riportati i risultati in tempo reale dell'algoritmo (nell'esempio riportato non è stata effettuata calibrazione).

Per immaginare come funziona questo strumento, e la validità dello stesso nel metodo, si può immaginare di muoversi verso la parete del poligono all'interno di quella regione, e non appena si individua un cambiamento, una parete, si è rilevati la posizione della parete stessa [12].

L'algoritmo *scan edge position*, è uno strumento composto da un numero k di *edge position*, ed ha l'effetto di uno *scan* multiplo all'interno di una regione assegnata. Il vantaggio di tale strumento, è che già fornisce automaticamente l'angolo della retta interpolata tra i punti rilevati dai k *edge position*.

In figura 45 troviamo il particolare dell'impostazione e del risultato in tempo reale dello strumento.

³² Si faccia riferimento al manuale Z341-E1-01, Pag. 184 e 203. Vision Sensor FH/FZ5 Series Vision System Processing Item Function Reference Manual.

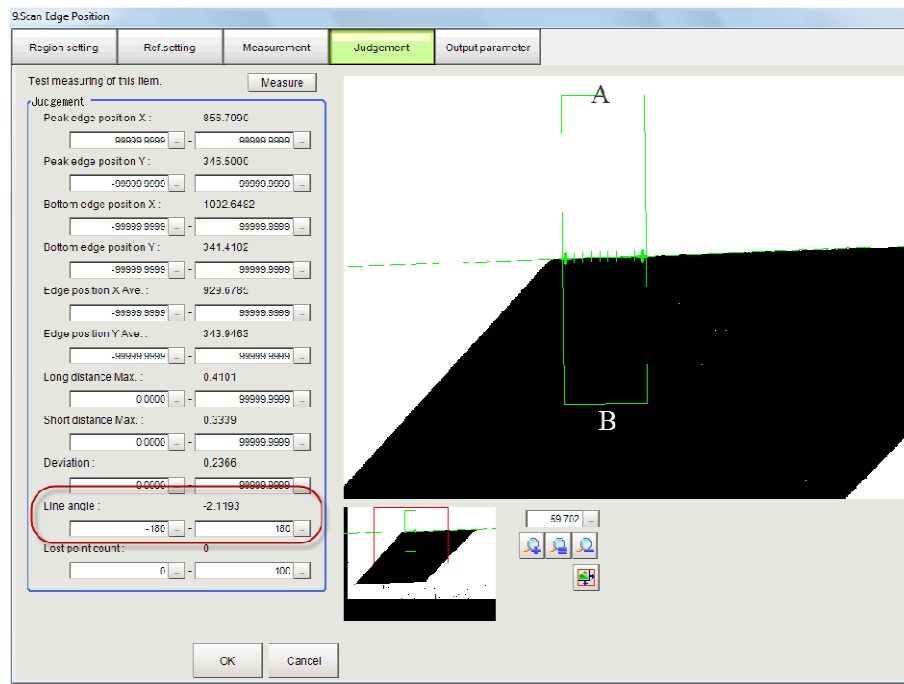


Figura 45. Particolare di impostazione dell’algoritmo “scan edge position”.

Sfruttando la stessa filosofia di azione, è possibile osservare in figura 45 che è stato scelto un numero k di *scan* pari a 10. Il risultato è visibile osservando i cursori verdi relativi ai punti rilevati. La linea verde tratteggiata indica invece la retta interpolata. In basso a sinistra è presente il risultato relativo all’angolo tra la retta rilevata e la retta giacente sull’asse x.

Le regioni di ricerca degli algoritmi descritti verranno impostate secondo il criterio presente nel diagramma in figura, ed il meccanismo di impostazione verrà illustrato al paragrafo 3.3.

3.2.4 Meccanismo di trasformazione degli angoli

È doveroso spendere una nota rispetto al sistema di riferimento degli angoli implementato all’interno degli algoritmi fin adesso descritti (nel software applicativo del sistema di visione FH ver.5.20). Come si evince dalla figura 45, l’angolo rilevato è negativo (-2,12 gradi). Nonostante è possibile notare che l’angolo è positivo (in realtà è 2,12 gradi per il sistema di riferimento da noi adottato), rispetto al sistema di riferimento normalmente adottato.

Prendendo come riferimento il noto cerchio goniometrico, muovendoci in senso antiorario, e partendo dal punto ove $\cos \theta = 1$ e $\sin \theta = 0$, individuiamo angoli crescenti descritti all’interno dell’intervallo $0 \rightarrow \pi \rightarrow 2\pi$ rad.; in gradi da $0 \rightarrow 90^\circ \rightarrow 180^\circ \rightarrow ..$ e via dicendo per individuare gli angoli relativi ai punti ove il coseno ed il seno forniscono i loro valori massimi e minimi. Il sistema di riferimento dell’applicativo potrebbe invece risultare illogico, poiché fornisce e mostra le seguenti corrispondenze con il nostro sistema di riferimento:

$0^\circ \rightarrow 90^\circ$: l’algoritmo fornisce angoli che vanno da 0(-) a -90°.

$90^\circ \rightarrow 180^\circ$: l’algoritmo fornisce angoli che vanno da 0 a 90°.

Al fine della trasformazione e la traduzione di tali dati fondamentali nel nostro sistema di riferimento, è stato implementato il seguente sotto programma. Esso preleva l’informazione angolo dall’algoritmo *scan* alla posizione A& (12 è il valore dell’angolo rilevato: dalle tabelle dei dati

forniti dallo *scan*³³, assegnandola alla variabile ANGLE& e lo trasforma secondo la regola mostrata nel set di istruzioni:

```

ANGLE& = UnitData (A&,12)

If ANGLE& < 0 Then
ANGLE& = ANGLE& * (-1)
Else
ANGLE& = 180 - ANGLE&
EndIf

```

Questo piccolo sotto programma fornirà quindi un angolo compatibile con il nostro sistema di riferimento. Ultima nota a riguardo: effettuando misure di angoli e muovendoci all'interno dell'intero cerchio goniometrico, otterremo delle informazioni che meritano una considerazione per lo sviluppo di quanto seguirà. Immaginiamo infatti di rilevare angoli sulle figure fin qui presentate. Sulla superficie superiore dei poligoni, rileveremo valori di angoli compresi all'interno del range $0^\circ \div 180^\circ$. Sulla superficie inferiore del poligono, otterremo angoli all'interno dello stesso range. Questo fattore è fondamentale ma non desiderato, visto che nell'analisi della deviazione degli angoli dovremmo agire diversamente in base all'emisfero di misura dell'algoritmo.

3.2.5 Storage

I dati forniti dagli algoritmi *scan*, verranno dapprima memorizzati all'interno di array definiti all'interno dell'algoritmo Macro³⁴. La definizione sarà data dalle istruzioni Dim NOME_Array_tipo_dati (numero elementi) - Ad esempio, per definire l'array dove verranno memorizzate le coordinate dei punti rilevati dagli *scan*, viene utilizzata la seguente istruzione:

```
Dim POSIZIONI& (PASSOSCANSIONE&, 1)
```

Verrà così definito un array a due dimensioni, dove il numero di "righe" sarà dato dal numero di elementi selezionata nella schermata di supporto³⁵. La definizione dell'array per gli angoli e delle altre variabili seguono le regole fin qui mostrate. Per potere accedere in lettura e scrittura delle variabili dell'array, basta utilizzare le regole di base di programmazione, prestando molta attenzione agli indici. Come vedremo nel paragrafo successivo, l'accesso agli array citati, sarà automatizzato da cicli FOR.

3.3 Macro: Angle analysys & Intersection until convergence

Siam giunti adesso al cuore del metodo di risoluzione dell'applicazione. L'idea di partenza, è quella di "avvolgere" il poligono, ed analizzarlo come con dei sensori di distanza disposti su una circonferenza, e che puntano al centro della stessa. La figura 46 è sicuramente di ausilio per chiarire il significato:

³³ Si faccia riferimento al manuale Z341-E1-01.. Vision Sensor FH/FZ5 Series Vision System Processing Item Function Reference Manual.

³⁴ Verrà approfondito al paragrafo 3.3

³⁵ Si veda il paragrafo 3.2.1. "Start"

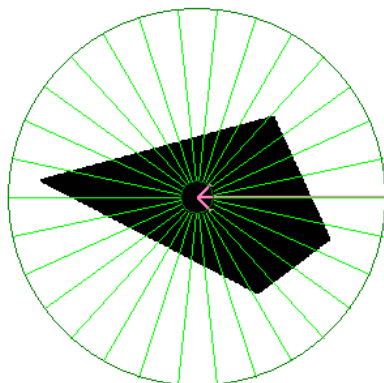


Figura 46. Impostazione delle regioni di scansione.

Ogni raggio della circonferenza – e da qui la denominazione del metodo – rappresenta un algoritmo di misura scan. Per ognuno di essi, avremo quindi come risultato la posizione del bordo del poligono rilevato. Avremo così una quantità di punti rilevati, pari al passo di scansione inserito. Nell'esempio in figura 46 il passo di scansione è pari a 12° , notiamo infatti che sono presenti 30 algoritmi scan (raggi). Le posizioni dei punti di bordo verranno quindi rilevati dagli *edge position*, mentre gli *scan edge position* seguiranno le stesse regole di impostazione, fornendo però gli angoli delle rette interpolate ai punti di rilevamento del multi scan. Per impostare le regioni di ispezione degli scan, viene utilizzata l'istruzione *SetUnitFigure*. Mediante l'utilizzo di tale istruzione imposteremo la regione di ispezione dell'*i*-esimo scan, procederemo alla misura, il prelievo dei risultati, e la memorizzazione negli array corrispondenti. Vengono di seguito riportati i passi salienti del codice, per l'impostazione degli *edge position*. Per quanto concerne gli *scan edge position* verranno rispettate le stesse regole di selezione e impostazione, facendo però attenzione alle voci relative a quanto riportato per singolo algoritmo.

Per descrivere al meglio i passi di codice, vengono altresì riportate in tabella 5 e 6 le voci di interesse, relative all'algoritmo scan edge e scan edge position.

Tabella 5. External Reference Tables – *Edge Position*.

External Reference Tables (Edge Position)			
No.	Data name	Set/Get	Data range
0	Judge	Get only	0: No judgement (unmeasured) 1: Judgement result OK -1: Judgement result NG
5	Edge position X	Get only	0 to 99999.9999
6	Edge position Y	Get only	0 to 99999.9999


```

FIGDATA&(2) = XOFFSET& + RAGGIOMAX& * Cos ( 0.01745329 * TH&)
FIGDATA&(4) = XOFFSET&
FIGDATA&(3) = YOFFSET& + RAGGIOMAX& * Sin ( 0.01745329 * TH&)
FIGDATA&(5) = YOFFSET&

```

```

SetUnitFigure SE&,0,FIGDATA&()
SetUnitFigure S&,0,FIGDATA&()

```

```

MeasureProc S&
VERTICI&(TH&+180,0) = UnitData(S&,5)
VERTICI&(TH&+180,1) = UnitData(S&,6)

```

```

MeasureProc SE&
ANGLE& = UnitData (SE&, 12)
ANGLE& = ANGLE NORM(ANGLE&)
ANGOLI&(TH&+180) = ANGLE&

```

```

Next TH&

```

Analizzando I passi del codice, possiamo osservare che punti di partenza ed arrivo degli scan, risiedono all'interno dell'array FIGDATA&() nelle posizioni 2,3,4,5, rispettivamente per Xstart, Xstop, Ystart, Ystop. Tali coordinate saranno ottenute mediante procedimento descritto, ed in particolare possiamo notare come i punti vengano assegnati sfruttando la semplice relazione:

$$\cos^2 \theta + \sin^2 \theta = 1 \quad (1.7)$$

Essendo stata effettuata quindi una trasformazione polare del sistema di riferimento delle coordinate, poiché è risultato più semplice dal punto di vista dello sviluppo, l'applicabilità, ed il riutilizzo. Il fattore moltiplicativo per l'angolo all'interno delle parentesi dell'argomento delle funzioni trigonometriche, serve solo ad una normalizzazione rispetto al calcolo del software di sviluppo. Si può notare come ci siano due cicli FOR e non solo uno. Come detto, il sistema non è in grado di interpretare autonomamente i risultati secondo le nostre esigenze. La differenza nei cicli FOR, sta nell'impostazione delle coordinate x e y a seconda che esse si trovino sull'emisfero superiore o inferiore rispetto al poligono (angoli compresi tra 0 e 180° per l'emisfero superiore, e da 180° a 360° per l'emisfero inferiore). Benchè la scelta di compilazione possa risultare più laboriosa che non includere all'interno di un unico ciclo una condizione di selezione, i risultati dimostrano che computazionalmente tale metodo risulta più "snello".

Dalla figura 46. notiamo già delle intrinseche limitazioni del metodo. Applicando infatti il metodo descritto, si corre il rischio di richiedere una misura in una regione che oltrepassa i limiti dello spazio di elaborazione. Per rendere l'idea si faccia riferimento alla figura 47.

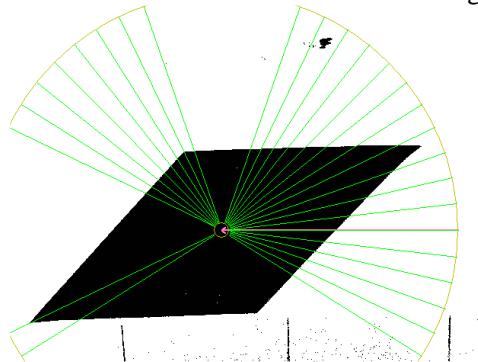


Figura 47. Regioni di impostazione oltre limite.

In questo caso, gli algoritmi non permettono la misura fuori dallo spazio. Si vede infatti che i “sensori” realmente impostati sono quelli mostrati con i raggi verdi. Per ovviare a tale limitazione sono state adottate tre tecniche, e verranno presentate al paragrafo 3.3.1. Adattamento delle regioni dei sensori. Presupposto il superamento di tale limitazione, una volta collezionati i punti e gli angoli relativi ai risultati dei sensori, comincia l’analisi dei risultati ottenuti, al fine del rilevamento vero e proprio degli angoli del poligono. Vengono quindi effettuati ciclicamente dei controlli tra l’angolo i e l’angolo $i+1$ al fine di rilevare deviazioni dell’angolo. Vengono quindi “caricati” l’angolo i ed il successivo, viene effettuato un confronto, e se tale confronto supera la deviazione standard minima definita da specifiche, vengono memorizzati i punti di interesse rilevati. Una volta ottenute le “zone” della regione di ricerca, dove abbiamo di certo un vertice del poligono, vengono utilizzati gli algoritmi “*Intersection*” per un rilevamento preciso del vertice. Il meccanismo fin qui descritto, non è sufficiente infatti ad un rilevamento preciso, a causa della modalità di azione, e della aleatorietà delle forme. L’algoritmo *Intersection* si basa anch’esso su dei multi *scan edge position*. In particolare, in esso vanno definite due regioni della stessa tipologia delle regioni di ricerca dell’algoritmo *scan edge position*. Automaticamente lo strumento è in grado di mettere a sistema le due rette interpolate sui punti trovati dai due *scan edge*, e fornisce la posizione del punto ove le due rette si intersecano, intercettando i vertici del poligono. Ad esso, verranno quindi assegnate delle regioni di ricerca sulla base dei risultati trovati al passo precedente. Per far ciò con estrema efficacia, le due regioni di ispezione verranno posizionate ortogonalmente rispetto ai lati del poligono (servendosi anche qui dei risultati degli *scan edge* per gli angoli, e degli *edge* per le coordinate), agli estremi dei punti rilevati dagli *edge position*. La figura 48 mostra un estratto dell’applicazione dell’algoritmo *Intersection*.

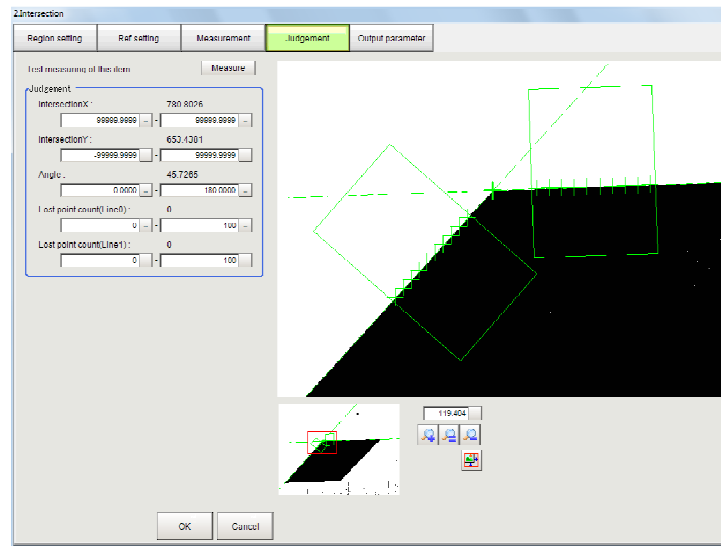


Figura 48. Algoritmo *Intersection*.

Quindi, ipotizzando di aver memorizzato 360 punti, la scansione delle deviazioni avverrà con l’utilizzo dei seguenti passi di codice:

For TH& = 0 To 180 Step PASSO&

A&= ANGOLI&(TH&)

B&= ANGOLI&(TH&+PASSO&)

If B& > (A& + DEVMIN&) OR B& < (A& - DEVMIN&) Then

NUMBERINTERSECTION& = NUMBERINTERSECTION& +1

If ANGLE& < 90 Then

ANGLE& = ANGOLI&(TH&-PASSO&)
 ANGOLO# = 0.01745329 * (ANGLE& +90)

X0&= VERTICI (TH- PASSO&,0)+ DISTANZASCANSIONE& *Cos(ANGOLO#)
 Y0&= VERTICI (TH- PASSO&,1) - DISTANZASCANSIONE& *Sin(ANGOLO#)
 X1&= VERTICI (TH- PASSO&,0) - (DISTANZASCANSIONE&*Cos(ANGOLO#)
 Y1&= VERTICI (TH- PASSO&,1)+ DISTANZASCANSIONE& *Sin(ANGOLO#)

FIGDATA2&(2) = X0&
 FIGDATA2&(4) = X1&
 FIGDATA2&(3) = Y0&
 FIGDATA2&(5) = Y1&

SetUnitFigure INT&, 0, FIGDATA2&()

ANGLE& = ANGOLI&(TH&+PASSO&)
 ANGOLO# = 0.01745329 * (ANGLE& +90)

X0&= VERTICI (TH+ PASSO&,0)+ DISTANZASCANSIONE& *Cos(ANGOLO#)
 Y0&= VERTICI (TH+ PASSO&,1) - DISTANZASCANSIONE& *Sin(ANGOLO#)
 X1&= VERTICI (TH+ PASSO&,0) - (DISTANZASCANSIONE&*Cos(ANGOLO#)
 Y1&= VERTICI (TH+ PASSO&,1)+ DISTANZASCANSIONE& *Sin(ANGOLO#)

FIGDATA&(2) = X0&
 FIGDATA&(4) = X1&
 FIGDATA&(3) = Y0&
 FIGDATA&(5) = Y1&

SetUnitFigure INT&, 2, FIGDATA&()

EndIf

EndIf

Next TH&

Utilizzando la procedura riportata, il punto di partenza dello *scan i* –esimo, farà riferimento alla coordinata rilevata dallo *scan i - passo*, con “passo” non necessariamente uguale al valore di scansione. Per questo motivo è stato volutamente lasciato distaccato dal passo di scansione degli angoli. In questo modo, è possibile raffinare ulteriormente la ricerca ed il rilevamento dei vertici. Prelevando quindi tali coordinate, per costruire le coordinate da fornire all’*Intersection* può esser d’ausilio la visione della figura 49. Si vede infatti, che per costruire la coordinata x del punto di partenza dello *scan edge position* dell’*Intersection*, bisogna tener conto che l’angolo non è quello rilevato e memorizzato in ANGOLI&(), bensì lo stesso a cui si somma un angolo di 90°. DISTANZASCANSIONE&, non è altro che la distanza tra x,y ed x0,y0. Per semplicità, non è stato riportato il caso in cui l’angolo *i* memorizzato. Infatti, nel caso in cui l’angolo fosse maggiore di 90°, l’assunzione:

ANGOLO# = 0.01745329 * (ANGLE& +90)

cambiarebbe sottraendo questa volta -90 . È stata risparmiata altresì la scrittura del caso in cui l'analisi verta sull'emisfero inferiore dell'immagine. In tal caso infatti, punto di partenza ed arrivo dovrebbero essere invertiti. Quindi X_0 ed Y_0 da invertire rispettivamente con X_1 ed Y_1 e viceversa. È infatti possibile notare che il ciclo FOR di partenza, scansioni gli angoli memorizzati nell'intervallo di scansione $0 \div 180^\circ$. Il codice completo conterà quindi di due cicli FOR con le varianti dette, ove all'interno troveremo il caso in cui l'angolo memorizzato è maggiore o minore di 90° , al fine di sommare o sottrarre un angolo retto agli angoli di impostazione dei punti di partenza ed arrivo degli *scan edge*.

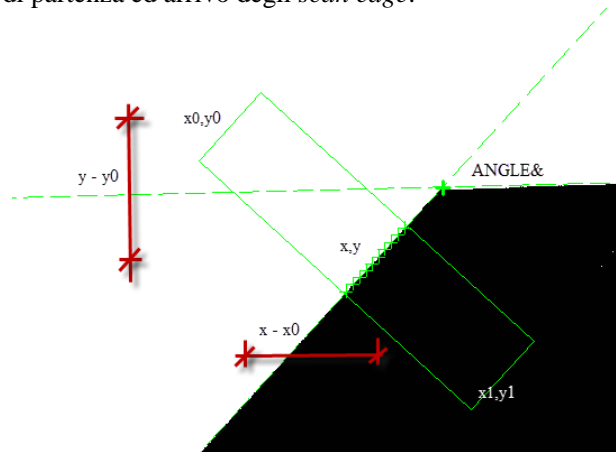


Figura 49. Impostazione regione dello scan edge position 0 o 1 dell'Intersection
Caso 1: angolo memorizzato $< 90^\circ$.

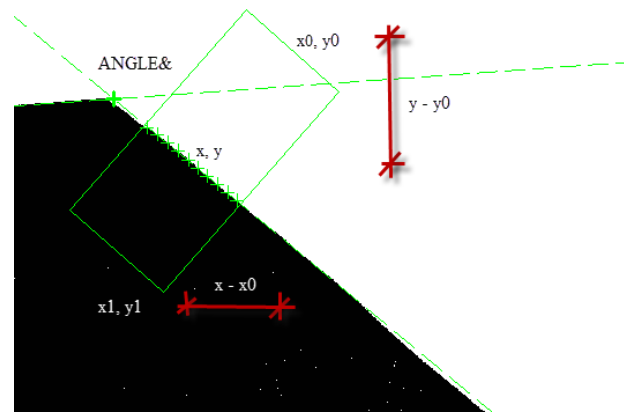


Figura 50. Impostazione regione dello scan edge position 0 o 1 dell'Intersection
Caso 2: angolo memorizzato $> 90^\circ$.

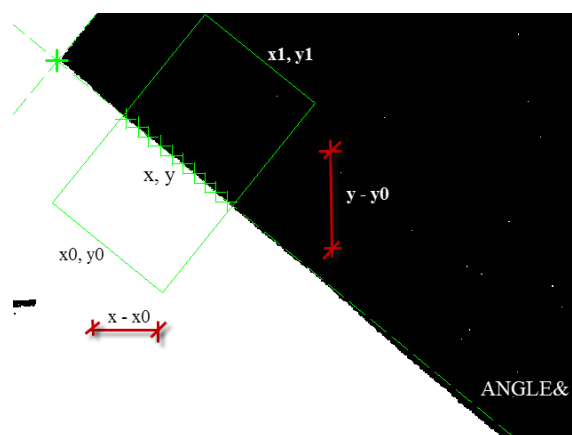


Figura 51. Impostazione regione dello scan edge position 0 o 1 dell'Intersection
Caso 3: angolo memorizzato $> 90^\circ$ - emisfero inferiore della regione di misura.

Giunti a questo punto, non rimane che eseguire un *Intersection* per ogni vertice rilevato, e memorizzare i risultati all'interno di un array, al fine di poter scambiare i dati con l'esterno, e di fornire i dati di partenza verso un ulteriore passo di processo. Al fine di eseguire l'algoritmo *Intersection*, basta inserire l'istruzione MeasureProc INT&, ogni volta che si è impostati le regioni di ricerca. Per memorizzare i risultati basta invece prelevare i dati utilizzando le istruzioni:

```
VERTICEXPOLIGONO& (I&, 0) = UnitData (INT&, 1)
VERTICEYPOLIGONO& (I&, 1) = UnitData (INT&, 2)
```

Ove i numeri all'interno delle parentesi sono stati ottenuti dalla tabella 7, mentre INT& è la posizione all'interno del flusso di elaborazione dell'*Intersection*.

Tabella 7. External reference tables – *Intersection*.

External Reference Tables (Intersection)			
No.	Data name	Set/Get	Data range
0	Judge	Get only	0: No judgement (unmeasured) 1: Judgement result OK -1: Judgement result NG
1	Intersection coordinate X	Get only	-99999.9999 to 99999.9999
2	Intersection coordinate Y	Get only	-99999.9999 to 99999.9999
3	Angle	Get only	0 to 180

Una volta rilevati i vertici, per visualizzare graficamente i risultati, alla sezione "*MeasureDispG*" della Macro, basta inserire un ciclo FOR per la cattura delle coordinate dei vertici rilevati all'interno degli array:

```
VERTICEXPOLIGONO& (numeroelementi&, 0)
VERTICEYPOLIGONO& (numeroelementi&, 1)
```

E visualizzarli con l'istruzione:

```
DrawCursor VERTICEXPOLIGONO& (I&, 0), VERTICEYPOLIGONO& (I&, 1)
```

Viene infine visualizzato in figura 52 il risultato del rilevamento vertici di un poligono che somiglia ad un parallelogramma.

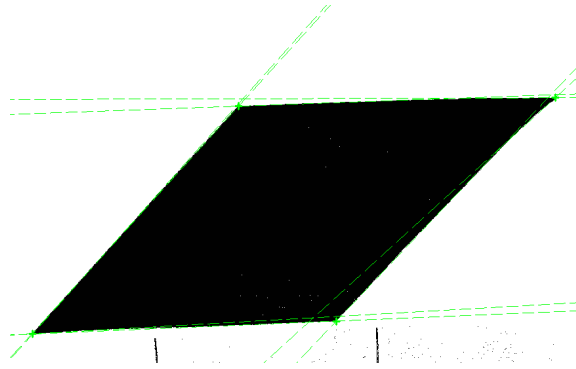


Figura 52. Visualizzazione dei risultati del metodo dei raggi.

Quest'ultima immagine è già occasione di spunto per gli sviluppi futuri, e successivi a questa applicazione. Come si vede, le rette interpolanti i punti ove è stato effettuato lo *scan edge position* all'interno di *Intersection*, non combaciano con i lati del poligono stesso. Questo fattore fortifica ulteriormente la potenza dello strumento sviluppato.

3.3.1 Adattamento delle regioni dei sensori.

Come visto in figura 77, qualora il poligono non dovesse rientrare all'interno della regione di ispezione, o per meglio dire, nel momento in cui la posizione del baricentro, sommata al valore di RAGGIOMAX& sia maggiore della risoluzione lungo x, il metodo manifesta già un intrinseco limite di applicazione. È ovvio immaginare che la soluzione debba necessariamente passare da una soluzione software piuttosto che hardware, più dispendiosa e meno flessibile.

Per ovviare a questo problema, sono state individuate tre strade:

- Modulazione dinamica del raggio di scansione: piuttosto che utilizzare un raggio unico di scansione, qualora la posizione del baricentro sommata al valore di RAGGIOMAX& sia maggiore della risoluzione lungo y (o x a seconda della risoluzione del sensore – *worst case*) basta utilizzare un raggio pari al valore

Min risoluzione Y – baricentro ; risoluzione Y + baricentro

Tale valore dovrà essere contemplato mediante l'utilizzo di un controllo (If..Then..) subito dopo la costruzione della *i*-esima coordinata di partenza. Tale soluzione è molto costosa in termini computazionali, ma garantisce convergenza assoluta del metodo (per forme senza parti concave)

- Punti di partenza degli *scan* sul frame dell'immagine, piuttosto che sulla circonferenza che ingloba il poligono: piuttosto che utilizzare le coordinate polari, i punti di partenza degli *scan* vengono precedentemente mappati dalla inizializzazione della Macro. In questo caso, non ci si potrà più servire di cicli annidati per la costruzione delle coordinate, o per lo meno in toto. Tale metodo si è dimostrato altresì efficace nella risoluzione della problematica citata. In termini di costi computazionali differisce dal precedente poiché in questo caso il tempo di costruzione è costante, mentre per il precedente il tempo di costruzione delle coordinate dipende dalle dimensioni del poligono rispetto all'area inquadrata.
- Costruzione software di un'immagine virtuale: una volta rilevato il baricentro del poligono, per poter costruire le coordinate secondo il metodo descritto nel precedente paragrafo, è

sufficiente creare una immagine riempiendo gli spazi bianchi – nel senso letterario del termine. Si fa in modo che il poligono rilevato dall’algoritmo Gravity and Area venga spostato (considerando il baricentro), in un nuovo punto x,y , ove x ed y sono maggiori della posizione del baricentro + RAGGIOMAX&. Anche questo metodo di risoluzione conferisce completa convergenza al metodo, ed in termini computazionali offre un’ottima alternativa ai precedenti. Per lo sviluppo di tale progetto, è stata pertanto utilizzata questa modalità di compensazione, dimostratasi più veloce in termini di sviluppo, ed efficace secondo specifiche.

4. Il metodo “ad inseguimento”

Il metodo presentato nel precedente capitolo ben si presta alla soluzione rilevamento dei vertici di un poligono. Ma come già analizzato, nel caso in cui si presentasse un poligono che mostri delle concavità al suo interno, il metodo perde efficacia, e non garantisce la convergenza. La figura 53 mostra i punti rilevati dall’algoritmo *edge position* nel caso in cui il poligono presenti delle concavità.

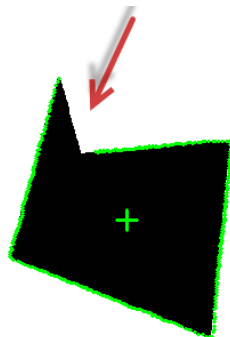


Figura 53. Particolare del funzionamento del metodo dei raggi.

Come si vede in figura 53, i punti in verde rappresentano i punti rilevati dagli n *edge position* (nell’esempio in figura è stato utilizzato un passo da 1°), mentre la parte indicata dalla freccia non è stata effettivamente scansionata.

Da qui lo spunto verso il pionieristico lavoro svolto, che sarà mostrato in questo capitolo. Il metodo ad inseguimento mostra numerose analogie con il mondo della conversione AD, con strumenti di elaborazione grafica, ed è stato sviluppato nell’ottica di ottenere un metodo estremamente potente e flessibile. Obiettivo principale è la convergenza assoluta del metodo a prescindere dalle forme che si potrebbe presentare. Tale sviluppo ha permesso quindi l’applicazione su delle problematiche non inerenti alla specifica di sviluppo.

4.1 Settore applicativo

Dal punto di vista del rilevamento automatico dei vertici di poligoni ignoti, sono stati sviluppati due metodi che si prestano nel loro insieme alla risoluzione della totalità delle problematiche riportate in questo studio. L’incipit allo sviluppo nasce quindi da un requisito reale di mercato, e diviene spunto per lo sviluppo e la possibile contestualizzazione nel campo dell’analisi di immagini nel campo della micro e nano elettronica. Gli algoritmi presenti in libreria e già sviluppati, peccano dell’aspetto di intelligenza sviluppato in questo lavoro, poiché son capaci al massimo di memorizzare in una matrice le coordinate dei punti di contorno (ricerca, riconoscimento, controllo della forma).

Viene di seguito riportata un'immagine relativa all'idea che ha scatenato la possibilità di utilizzo di quanto sviluppato per future applicazioni.

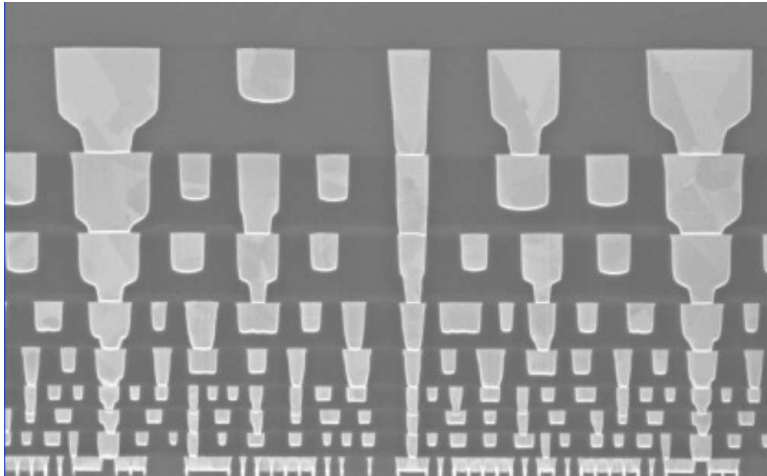


Figura 54. Particolare di un dispositivo con 8 livelli di "metal".

È l'immagine ottenuta mediante scansione SEM, relativa ad un dispositivo Intel.

Nell'ottica dell'analisi di difetti, ma soprattutto al fine di rilevare le criticità nei processi di fabbricazione nel mondo della micro e nano elettronica, vengono quindi effettuati dei rilievi fotografici sui dispositivi, e da essi vengono analizzati graficamente e puntualmente le zone ove si verifica un discostamento tra *ciò che ci aspetteremmo*, e *cosa invece accade*. È il classico "controllo qualità" che troviamo nel campo delle applicazioni di visione artificiale in automazione industriale. Mentre tale controllo viene ad oggi effettuato, analizzato e riportato, da una figura altamente specializzata, con l'ausilio di tale strumento si intravede la possibilità di ottenere un feedback specializzato, automatico, preciso, stabile, da parte del sistema di visione.

Questa visione non racchiude intrinsecamente in sé la volontà o il desiderio di sostituire la macchina all'uomo, ma di fornire un ausilio ed un supporto allo sviluppo ed ai processi nella sua più generica accezione [13].

4.2 Sviluppo dell'algoritmo di elaborazione

Immaginando di esser ciechi e di muoverci nello spazio di elaborazione delle immagini, non rimane che il senso del tatto per rilevare le caratteristiche esterne di una forma. Ed a questo si ispira la costruzione dell'algoritmo.

In figura 55 troviamo il diagramma di flusso del metodo "ad inseguimento".

Servendoci di uno *scan* di partenza (fittizio, che può esser posizionato a piacere, o non posizionato, vedremo più avanti le varianti del metodo), una volta rilevato il primo punto del lato di una qualsiasi forma, l'algoritmo si muove nello spazio di elaborazione servendosi del "tatto" di numerosi *edge* e *scan edge position*. Essi verranno quindi chiamati a rilevare punti, muovendosi lungo una direzione assegnata (nel caso dei poligoni la direzione sarà circolare, in modo da richiudere su se stesso un percorso ad anello). Per rimanere agganciati alla forma, vengono analizzate le sequenze dei punti rilevati, e nel caso in cui si vada "fuori strada", esiste un meccanismo tale per cui si riparte dall'ultimo punto rilevato e si cambia direzione di aggancio. Utilizzando i *lost point* degli *scan edge position*, possiamo intanto rilevare l'assenza del rilevamento di uno o più punti, e da qui l'informazione che si è perso l'aggancio. Al fine di riagganciare la forma, viene utilizzata una misura del colore centrata nella prima regione che ha fornito dei *lost point*. Se il colore di questa regione sarà quello del poligono, significa che il sensore si trova

all'interno del poligono stesso. Se il colore rilevato sarà quello dello sfondo, l'aggancio dei lati del poligono si è perso poiché si è fuori dal poligono. Nel caso in cui il colore rilevato sia del poligono o dello sfondo, al passo successivo in prima battuta verrà impostata la regione del primo *scan edge* sulla regione dell'ultimo *scan* che ha fornito risultati utili. Mentre per quanto riguarda le regioni degli *scan edge* dal secondo all'ottavo, verranno impostate con la seguente modalità:

- Colore poligono: partendo dalla coordinata dell'ultimo rilevamento utile, per il secondo *scan edge* viene "piegata" la zona di rilevamento in senso antiorario, e per i successivi si prosegue con l'inseguimento.
- Colore sfondo: partendo dalla coordinata dell'ultimo rilevamento utile, per il secondo *scan edge* viene "piegata" la zona di rilevamento in senso orario, e per i successivi si prosegue con l'inseguimento.

L'insieme dei processi sarà gestito mediante l'utilizzo di alcuni FLAG, che trovano corrispondenza con le deviazioni del diagramma di flusso (*end point? Lost point? Deviation?*), ed a cui si sommano altri per gestire la convergenza del processo.

Rispetto al metodo dei raggi, questo non presenta una unidirezionalità nel flusso di elaborazione. Mentre nel primo caso, vengono prima rilevati i punti, e poi in successione viene svolta l'analisi della deviazione degli angoli sui dati trovati, tale metodo risulta molto più dinamico, a spese – ovviamente – di un maggiore costo computazionale. Le regioni di ispezione dipenderanno quindi dai risultati ottenuti localmente, per tale motivo il metodo trova analogia con la modalità di funzionamento dei convertitori ADC ad approssimazioni successive.

Il rilevamento *macro* e *micro metrico*, trova spunto dal metodo precedente, infatti, ogni qualvolta verranno trovate delle deviazioni negli angoli rilevati, verrà utilizzato un algoritmo *Intersection* per il rilevamento preciso del vertice.

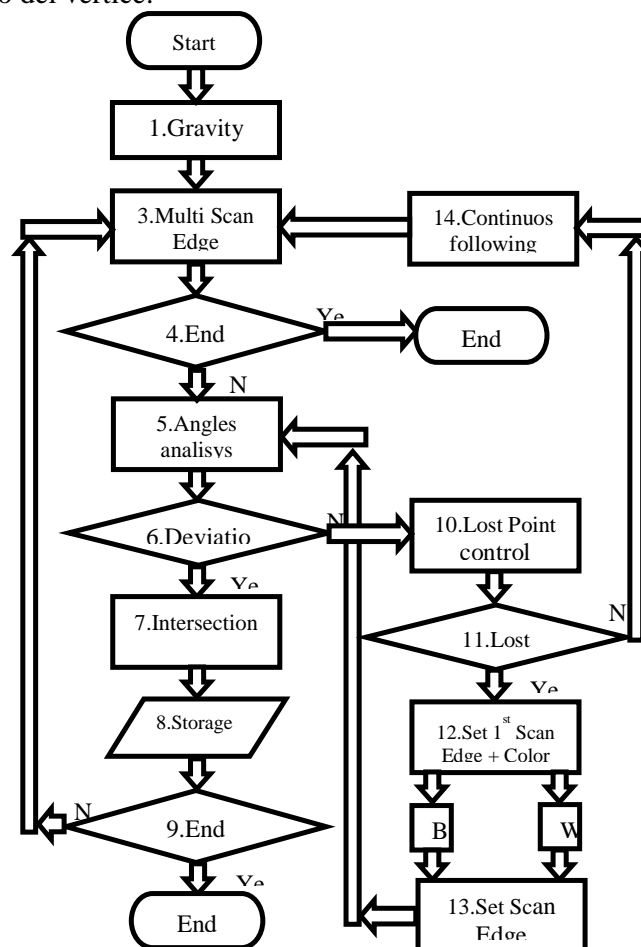


Figura 55. Diagramma di flusso del metodo ad inseguimento

Di seguito verranno descritte le procedure adottate per l'impostazione automatica degli algoritmi, e per ogni passo verrà descritta la strategia di azione, ed i passi salienti del codice. In figura 55 viene riportato l'equivalente del precedente diagramma implementato con l'ambiente di sviluppo del sistema di visione FH ver.5.20.

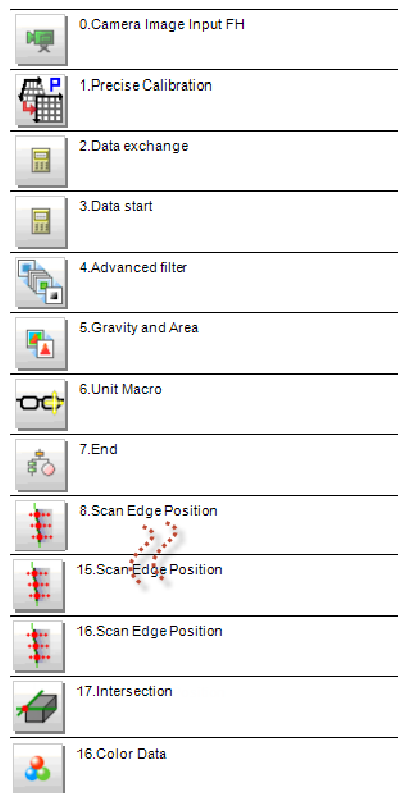


Figura 56. Diagramma di flusso implementato all'interno del software applicativo FH ver.5.20 per il metodo ad inseguimento.

Per semplicità, sono stati tagliati gli algoritmi dalla posizione 9 alla 14, poiché sono tutti degli *scan edge position*. Possiamo notare già in questa sede che il diagramma implementato sul sistema di visione è (sembra!), identico a quello del metodo dei raggi. Quel che cambia è il contenuto della (o delle) Macro.

Nei paragrafi successivi, verranno analizzati passo passo gli algoritmi adottati, e la filosofia dello sviluppo della Macro principale.

Come si evince dal diagramma, tutta l'"intelligenza" del metodo è stata implementata all'interno dell'algoritmo di supporto Macro.

4.2.1 "Start"

Alla stessa stregua del precedente metodo, è stata realizzata una piccola interfaccia grafica, dove troviamo dei campi per l'inserimento dei dati, la conferma o la cancellazione, nonché di tasti per il lancio di finestre di impostazione grafica personalizzata per gli algoritmi di partenza.

Per quanto invece concerne la preelaborazione dell'immagine, si faccia riferimento al paragrafo 3.2.1, poiché in questo caso è stata adottata altresì una calibrazione ed un prefiltraggio dell'immagine.

In fase di start riscontriamo in questo caso due modalità di funzionamento, nel caso in cui si voglia lavorare con dei poligoni, o nel caso in cui si vogliano fare delle auto-analisi sugli spaccati SEM o TEM. In questo capitolo, verrà presentato il metodo per l'applicazione originaria (rilevamento

vertici poligoni), mentre nel prossimo capitolo verranno affrontate le varianti per l'analisi automatica di spaccati SEM o TEM.

Il meccanismo di partenza in prima battuta ripercorre il metodo dei raggi. Infatti, viene utilizzato un algoritmo *Gravity And Area* per il rilevamento del baricentro del poligono. Ottenuta la coordinata x,y, viene dapprima utilizzato un algoritmo *scan edge position* per rilevare un punto di partenza e l'angolo di partenza di incidenza. All'interno della Macro troveremo:

```
BARICENTROX& = UnitData (Gravity&, 6)
BARICENTROY& = UnitData (Gravity&, 7)
FIGDATA&(2) = 0
FIGDATA&(3) = 0
FIGDATA&(4) = BARICENTROX&
FIGDATA&(5) = BARICENTROY&
SetUnitFigure SC&, 0, FIGDATA()
```

```
MeasureProc SC&
START_X& = UnitData (SC&, 5)
START_Y& = UnitData (SC&, 6)
START_ANGLE& = UnitData (SC&, 12)
```

Analizzando il codice, troviamo alle prime due righe il prelievo della coordinata del baricentro del poligono dall'algoritmo *Gravity And Area*, punto di partenza e di arrivo della regione di scansione saranno rispettivamente il punto $[0,0]$ ³⁶, ed il baricentro del poligono. Con l'istruzione *SetUnitFigure* impostiamo la regione di ricerca dello *scan edge position* alla posizione SC&, ed una volta lanciato, vengono memorizzati i dati relativi al punto x ed y medi ed all'angolo della retta interpolata (si ricorda che lo *scan edge position* è composto da multipli *edge position* all'interno della regione designata). È stato utilizzato il punto medio dello *scan edge position* poiché a questo passo, la coordinata serve solo come punto di "aggancio" al poligono, e su di essa non vengono elaborati risultati utili allo scopo finale. Non avremmo potuto utilizzare il punto massimo o minimo, poiché tali coordinate potrebbero non essere coerenti.

Una volta acquisito il punto di partenza, passiamo adesso al nucleo dell'elaborazione: dall'impostazione dinamica delle regioni di ispezione, all'analisi dinamica dei risultati.

4.2.2 Multi scan edge position

Entrando nel vivo dell'elaborazione, notiamo come in questo caso siano stati utilizzati 8 algoritmi *scan edge position* contemporaneamente. Nel caso del metodo dei raggi, veniva utilizzato uno solo di essi per ogni punto, mentre adesso, al fine di poter "inseguire" la forma, ci serviamo di più strumenti. La stessa logica è stata implementata utilizzando uno solo di essi, ma i risultati hanno dimostrato che in termini computazionali, ed efficienza del metodo, questa strada risulta la migliore. Il numero ottimo – 8 - deriva da simulazioni su un vasto campionario di poligoni. Non è possibile infatti in questo caso ottenere una relazione che dia il numero ottimale. Di conseguenza, è stato quindi ulteriormente analizzato il tempo di elaborazione del metodo.

Le regioni degli 8 *scan edge position*, verranno impostate diversamente in base ai risultati locali dei controlli: deviazione angolo (e misura contestuale dell'*Intersection*), *lost point*, *continuos following*³⁷. Nel primo e nel terzo caso, le regioni di ricerca degli *scan*, si muoveranno secondo la direzione assegnata. Quindi, partendo dalla coordinata del punto rilevato dallo *scan n - 1*, dal primo all'ultimo *scan edge* verranno incrementati i valori delle coordinate x,y secondo la direzione di rilevamento dei vertici. Viene di seguito riportato il passo del codice implementato nel caso in cui

³⁶ Nel sistema di visione artificiale il punto in alto a sinistra è appunto 0,0.

³⁷ Si faccia riferimento al diagramma di flusso.

non siano stati rilevati dei *lost point*, ma nel caso del *continuos following* o *Intersection*, alla prima iterazione:

```
*MCRINIT
```

```
Dim FIGDATA&(30)
```

```
DISTANZASCANSIONE& = 40
```

```
DISTANZAEDGE& =40
```

```
Return
```

```
*MEASUREPROC
```

```
ANGLE& = START_ANGLE&
```

```
For INDEX& = 8 To 15 Step 1
```

```
  If ANGLE& <= 90 Then
```

```
    ANGOLO# = 0.01745329 * ( ANGLE&+90 )
```

```
    ANGOLO2# = 0.01745329 * ( ANGLE&)
```

```
    X0&= START_X& - (DISTANZASCANSIONE& *Cos(ANGOLO# ) )
```

```
    Y0&= START_Y& - (DISTANZASCANSIONE& *Sin(ANGOLO# ) )
```

```
    X1&= START_X& + (DISTANZASCANSIONE&*Cos(ANGOLO#) )
```

```
    Y1&= START_Y& + (DISTANZASCANSIONE& *Sin(ANGOLO#) )
```

```
    FIGDATA&(2) = X0& + ((INDEX& -8) *DISTANZAEDGE& * Cos (ANGOLO2#))
```

```
    FIGDATA&(4) = X1& + ((INDEX& -8) * DISTANZAEDGE& * Cos (ANGOLO2#))
```

```
    FIGDATA&(3) = Y0& - ((INDEX& -8) * DISTANZAEDGE& * Sin (ANGOLO2#))
```

```
    FIGDATA&(5) = Y1& - ((INDEX& -8) *DISTANZAEDGE& * Sin (ANGOLO2#))
```

```
  EndIf
```

```
  If ANGLE& > 90 Then
```

```
    ANGOLO# = 0.01745329 * ( ANGLE& -90)
```

```
    ANGOLO2# = 0.01745329 * ( ANGLE&)
```

```
    X0&= START_X& + (DISTANZASCANSIONE& *Cos(ANGOLO# ) )
```

```
    Y0&= START_Y& - (DISTANZASCANSIONE& *Sin(ANGOLO# ) )
```

```
    X1&= START_X& - (DISTANZASCANSIONE&*Cos(ANGOLO#) )
```

```
    Y1&= START_Y& + (DISTANZASCANSIONE& *Sin(ANGOLO#) )
```

```
    FIGDATA&(2) = X0& - ((INDEX& -8) *DISTANZAEDGE& * Cos (ANGOLO2#))
```

```
    FIGDATA&(4) = X1& - ((INDEX& -8) * DISTANZAEDGE& * Cos (ANGOLO2#))
```

```
    FIGDATA&(3) = Y0& + ((INDEX& -8) * DISTANZAEDGE& * Sin (ANGOLO2#))
```

```
    FIGDATA&(5) = Y1& + ((INDEX& -8) *DISTANZAEDGE& * Sin (ANGOLO2#))
```

```
  EndIf
```

```
    SetUnitFigure INDEX& , 0, FIGDATA&()
```

```
Next INDEX&
```

Prima di passare all'analisi del codice, è utile mostrare l'effetto dello stesso mediante la visione delle seguenti immagini, relative ai risultati parziali degli 8 *scan edge position*, più quello di partenza. Per ogni visualizzazione delle seguenti immagini, è stato modificato il layout dell'interfaccia operatore, al fine di visualizzare i risultati parziali degli algoritmi in basso a destra. Le immagini a seguire saranno di fondamentale supporto per la comprensione del funzionamento dell'algoritmo in toto, si farà infatti riferimento a queste anche nei paragrafi successivi.

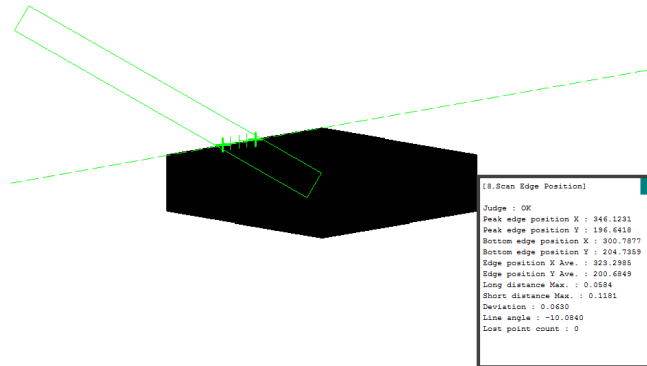


Figura 57. *Scan Edge Position* di partenza.

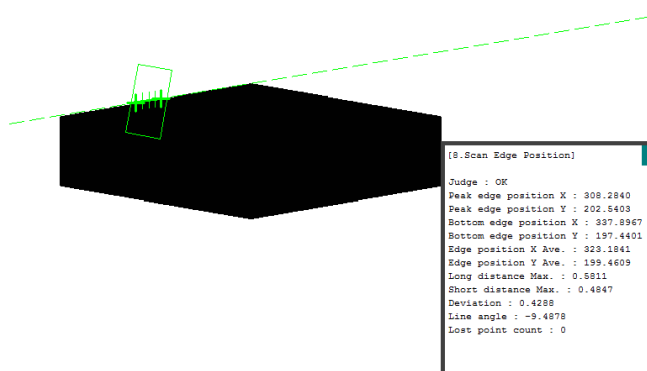


Figura 58. *Scan Edge Position* numero 1.

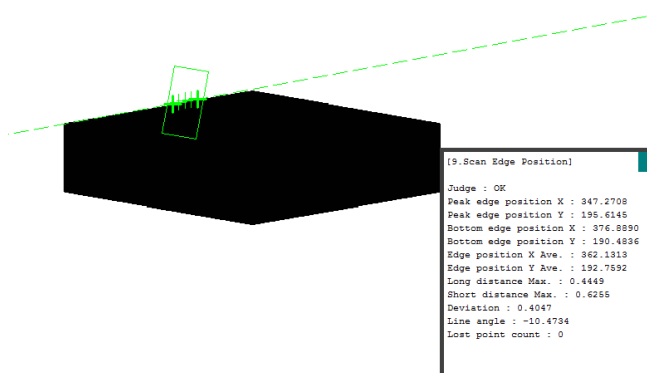
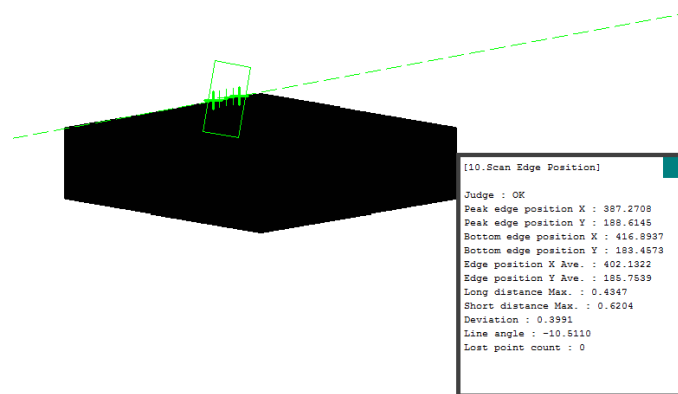
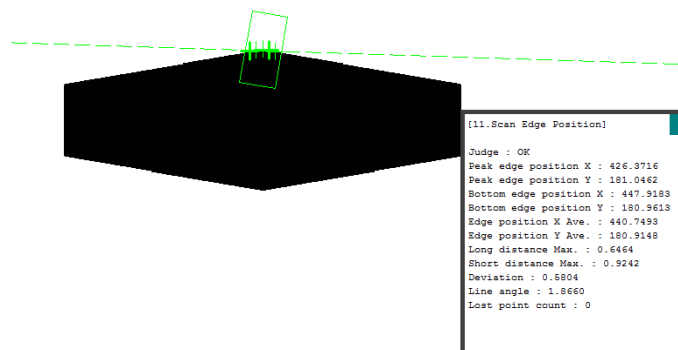
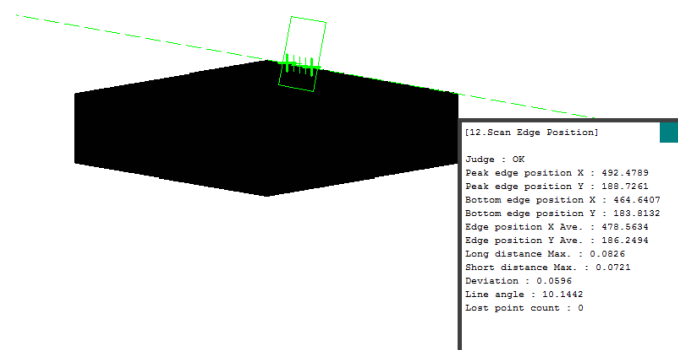
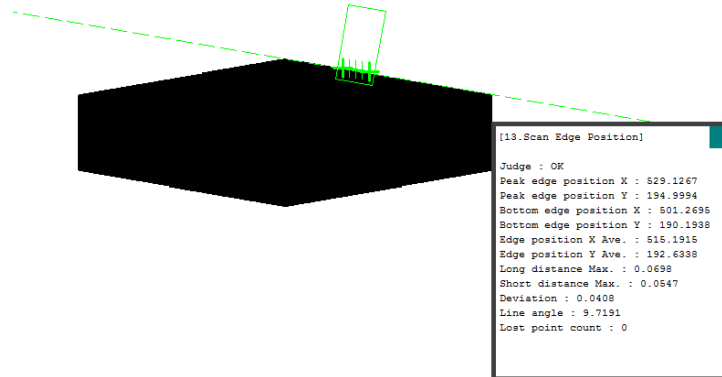
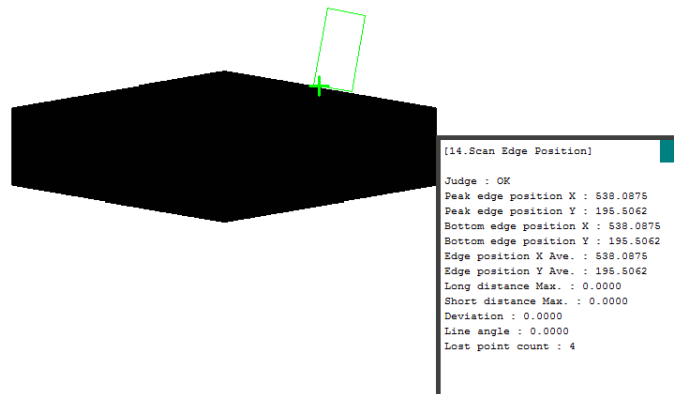
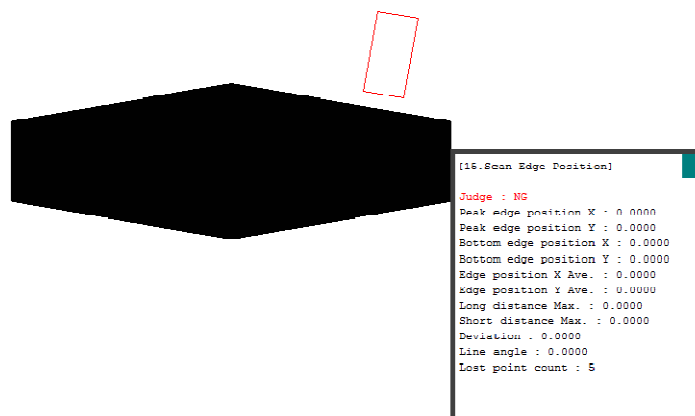


Figura 59. *Scan Edge Position* numero 2.

Figura 60. *Scan Edge Position* numero 3.Figura 61. *Scan Edge Position* numero 4.Figura 62. *Scan Edge Position* numero 5.

Figura 63. *Scan Edge Position* numero 6.Figura 64. *Scan Edge Position* numero 7.Figura 65. *Scan Edge Position* numero 8.

Analizziamo adesso il codice con l'ausilio delle figure. Nella prima parte, alla inizializzazione della *Macro*, troviamo la definizione dell'array delle coordinate della regione x FIGDATA&(), e

due parametri: DISTANZASCANSIONE& e DISTANZAEDGE&. Il primo parametro definisce la distanza tra il punto di partenza della regione di *scan edge position* ed il punto sulla superficie del poligono. Il secondo, definisce il passo – la distanza - di scansione tra gli 8 *scan edge position*. Nell'esempio riportato sono stati definiti uguali e pari a 40 pixel. In merito a questi parametri, basti pensare che il primo definisce quanto vicino, o lontano, ci teniamo alla superficie del poligono. Da esso dipende la capacità del metodo nell'agganciarsi anche in caso di angoli piccoli (dove per piccoli si intende minori di 15°).

DISTANZAEDGE& definisce invece il passo di scansione, e da esso dipende la convergenza del metodo parametrizzato sul valore del lato minimo che può presentare il poligono. Per la convergenza, è importante che esso sia minore di 3 volte il lato minimo. Una volta prelevata la coordinata dello *scan edge position* di partenza, si costruiscono dapprima le coordinate della prima regione (X0&, Y0&, X1&, Y1&), ed in secondo luogo le coordinate delle n regioni a seguire, coordinate dal ciclo FOR. Si noti come siano stati utilizzati due angoli diversi per il calcolo della coordinata del punto di partenza della prima regione, e per gli 8 *scan*. Questo perché partendo dalla coordinata dallo *scan* di partenza, si deve dapprima costruire un punto di partenza della prima regione che giaccia sulla normale alla superficie del lato del poligono. Infatti in questo caso è stato utilizzato ANGLE& + 90 o ANGLE& - 90 a seconda che l'angolo della retta interpolante i punti del lato del poligono sia maggiore o minore di 90°. Mentre per quanto concerne gli 8 *scan edge position* in sequenza, le coordinate dei punti di partenza giacciono sulla direzione del movimento della scansione dei vertici. In questo caso è stato utilizzato l'angolo rilevato dal primo *scan edge position*, nell'ottica appunto di muoversi rasenti alla parete del lato del poligono. È possibile notare nelle figure come i *scan edge position* si muovono letteralmente lungo il lato del poligono, ed è possibile notare nello scorrere delle immagini, come vengano rilevati angoli diversi, ma soprattutto, come vengano rilevati i *lost point*.

In figura 66 possiamo vedere i punti rilevati dagli 8 *scan edge position*. Si noti come già in sede della prima iterazione, analizzando i risultati venga generata una esecuzione di un primo *Intersection*.

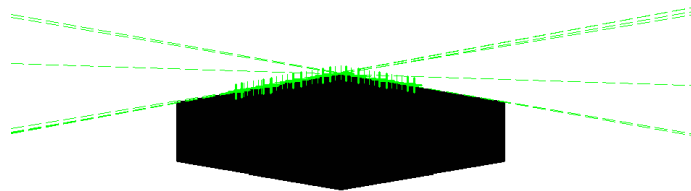


Figura 66. Risultato dell'esecuzione dei primi 8 *scan edge position*.

4.3 Macro: Angle analysis, Intersection, lost point, o continuous following

Una volta acquisiti i risultati relativi ai primi 8 *scan edge position* effettuati, vengono prelevati e memorizzati coordinate ed angoli. A partire da questi si effettua dapprima un controllo di deviazione dell'angolo rilevato, alla stessa stregua di quanto visto per il metodo dei raggi. Non riporteremo quindi di seguito lo stesso passo di codice. Nell'analisi degli angoli degli 8 *scan edge position*, vengono contemplate le seguenti casistiche nell'analisi degli angoli: deviazione angolo, *lost point*, *nulla*. Nel caso venga rilevata una deviazione dell'angolo, utilizzando in questo caso come parametro di confronto DEVMIN&³⁸, verrà impostata la regione dello *scan edge* 1

³⁸ Si veda anche paragrafo 3.3. Macro & Angle analysis.

dell'*Intersection* in prossimità della regione dello *scan -1* . Mentre lo *scan 2* dell'*Intersection* verrà impostato nella regione in prossimità dello *scan* dove si è verificata la deviazione dell'angolo, seguendo le stesse regole di costruzione viste fin adesso. Questo non necessariamente equivale ad aggiungere un ulteriore passo di codice, basta gestire prudentemente ed in maniera ingegnosa i parametri ricorsivi.

Qualora non venisse rilevata una deviazione nell'angolo, si passa dunque all'analisi dei *lost point*. Come si vede in figura 65, l'algoritmo *scan edge position* fornisce informazioni anche in merito al numero di *scan* NG³⁹. L'analisi viene svolta utilizzando un ciclo FOR per la ricerca, la verifica e la memorizzazione. Viene di seguito riportato il passo del codice corrispondente:

```
LOSTMAX& = 2
```

```
For I& = 8 To 15 Step 1
```

```
  LOST& (I&-8) = UnitData (I&, 13)
```

```
  Next I&
```

```
  Rem FINE MEMORIZZAZIONE LOST
```

```
  For I& = 6 to 0 Step 1
```

```
    A& = LOST&(I&)
```

```
    B& = LOST&(I&+1)
```

```
  If B& > LOSTMAX& AND A&<LOSTMAX& Then
```

```
    SetUnitData DE&, 7 , I&+8
```

```
    'DE&-> data exchange
```

```
  Exit FOR
```

```
EndIf
```

Si è volutamente riportato solo questo passo di codice per mostrare una sottigliezza nel rilevamento, o più nella diagnostica dei *lost point*. Come si evince dal codice, mentre vengono memorizzati i *lost* progressivamente, nel caso della diagnostica essi vengono analizzati in verso opposto nella scansione dell'array (For I&= 6 To 0 e non For I& = 0 to 6). A questo si aggiunge il doppio controllo sullo *scan* I& ed il successivo. Servendosi di questo procedimento e percorrendo a ritroso gli *scan*, l'algoritmo con il *lost point* viene "segnalato" (SetUnitData DE&, 7 , I&+10) solo se il precedente ha avuto esito positivo. Così facendo, al passo successivo, verrà impostata la regione dello *scan 1* dell'*Intersection* in prossimità della regione dell'ultimo *scan* che ha avuto esito positivo, e la regione dello *scan 2* dell'*Intersection* "ripiegata" sul *lost point*.

Nel caso in cui non vengano rilevati dei vertici o dei *lost point*, l'iterazione successiva passa dal blocco di "*continuous following*". Utilizzando la stessa metodologia nell'impostazione delle regioni di ricerca come nel caso della prima iterazione, la regione dello *scan 1* dell'iterazione successiva sarà posta in prossimità della regione dello *scan 8* dell'iterazione precedente. Questa casistica corrisponde alla situazione in cui gli 8 *scan edge position* non ricoprono la scansione dell'intero lato. La necessità in questo caso è dunque: proseguire.

4.3.1 Color Data

Nel caso in cui vengano rilevati dei *lost point*, per raffinare la ricerca, ed al fine di garantire il riaggancio della figura, viene effettuata una misura di colore⁴⁰ in prossimità del punto finale della

³⁹ Si faccia riferimento al meccanismo di funzionamento *scan*.

⁴⁰ Si veda anche paragrafo 2.2.8. Color Data

regione dello *scan edge position* che ha fornito esito positivo nel rilevamento dei *lost point*. Viene di seguito riportato il passo del codice:

```
GetUnitFigure I&+8, 0, FIGDATA&()

FIGDATACOLOR&(2) = FIGDATA&(3)
FIGDATACOLOR&(3) = FIGDATA&(5)
FIGDATACOLOR&(4) = 14
FIGDATACOLOR&(5) = 14

SetUnitFigure CD&, 0, FIGDATACOLOR&()      ‘CD& posizione 16 Color
MeasureProc CD&
COLOR& = UnitData (CD&, AV)41
If COLOR& < VALUETRESHOLD& Then
FLAGINTO& = 1
SetUnitData DE&, 8, 1
Else
SetUnitData DE&, 8, 0
FLAGINTO& = 0
EndIf
```

Una volta ottenuti gli estremi delle coordinate di interesse dalla regione che ha fornito *lost point*:

```
GetUnitFigure I&+8, 0, FIGDATA&();
```

Viene impostato come centro del cerchio di misura del colore, il punto X1&,Y1& dello *scan*:

```
FIGDATACOLOR&(2) = FIGDATA&(3)
FIGDATACOLOR&(3) = FIGDATA&(5)
```

E come raggio x ed y nell’esempio è stato riportato il valore 14 (pixel):

```
FIGDATACOLOR&(4) = 14
FIGDATACOLOR&(5) = 14
```

Viene quindi “lanciato” l’algoritmo *Color Data*:

```
MeasureProc CD&
COLOR& = UnitData (CD&, AV);
```

Ed una volta prelevato il valore del parametro “*Valore*” fornito dal *Color Data*, viene impostato il FLAG di interno o esterno poligono (FLAGINTO& = ..):

```
If COLOR& < VALUETRESHOLD& Then ...
```

Nelle figure successive, troviamo i due esempi che possiamo riscontrare a questo passo di elaborazione. Si noti come nei dati riportati in basso a destra, forniti dall’algoritmo *Color Data*, nei due casi il *Valore* ha due valori nettamente diversi: 1 Vs 255, come per le componenti RGB.

Per una parametrizzazione ottima del raggio della regione di ricerca del *Color Data*, si deve tener in conto il massimo di deviazione ed il passo degli *scan*.

⁴¹ “6” è la media della componente verde del colore. È stata scelta tale componente poiché ben si presta ad una misura di luminosità. Si faccia riferimento ai manuale utente del sistema di visione.

Al fine di ottenere una misura di colore stabile che conferisca assoluta convergenza, si deve rispettare la seguente relazione:

$$RMAX < DISTANZASCANSIONE * Sin (ANGMIN) \quad (1.9)$$

Relazione facilmente ottenibile impostando delle opportune condizioni iniziali. Tale condizione è da considerarsi ampiamente conservativa, infatti da simulazioni si è notato che basti già la metà del valore riportato, vista la modalità di funzionamento dell'algorithmo di Color Data, in termini di media spaziale.

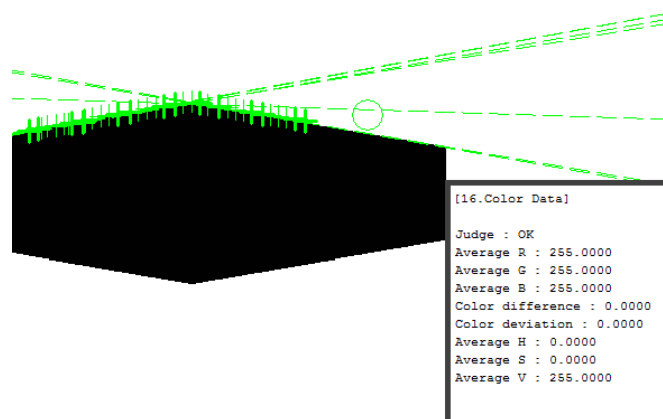


Figura 67. Particolare della misura di colore - sfondo.

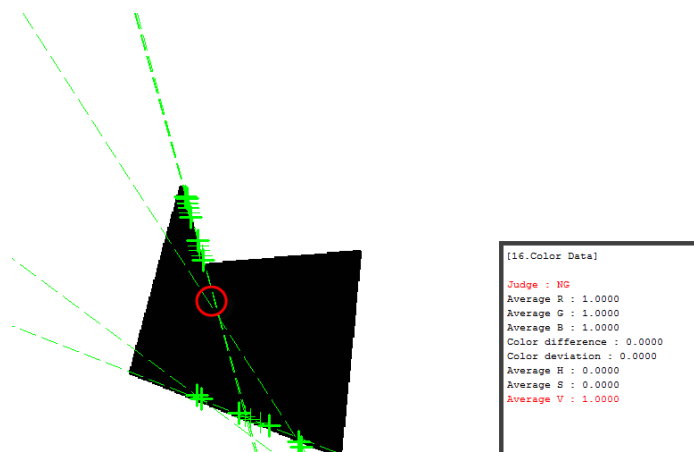


Figura 68. Particolare della misura di colore - poligono.

Prima di passare all'iterazione successiva, la regione dello scan 1 verrà impostata in prossimità dell'ultimo scan OK, mentre la seconda regione verrà impostata ad una distanza (in baricentro, quindi x,y) pari a $DISTANZAEDGE$, e con un angolo pari a:

- $ANGLE - ANGROT$ nel caso in cui il Valore del Color Data sia $COLOR < VALUETRESHOLD$;
- $ANGLE + ANGROT$ nel caso in cui il Valore del Color Data sia $COLOR < VALUETRESHOLD$;

Queste condizioni equivalgono ad impostare una regione dello scan 2 ruotata in senso orario o antiorario rispetto a quella dello scan 1. Dal 2° all'8°, l'impostazione degli scan avverrà seguendo il meccanismo già citato.

4.3.2 “Continuous following”

Questo è il passo che conferisce ciclicità all'esecuzione dell'algoritmo principale. *Continuous following* si riferisce alla situazione in cui non vengono rilevati nuovi vertici, né dei lost point. Immaginiamo quindi, che nel muoverci radenti al lato del poligono, “tastandolo” mediante l'utilizzo degli scan [12], dobbiamo quindi continuare a procedere nella ricerca. Servendoci quindi dei dati relativi all'ultima regione impostata sull'ottavo scan, la regione dello *scan 1* dell'iterazione successiva sarà posta in prossimità della regione dello *scan 8* dell'iterazione precedente. Viene racchiuso in questa modalità, l'intrinseca impostazione dell'angolo della regione, nonché della direzione di avanzamento di scansione derivante da esso. Una volta acquisiti i risultati dell'iterazione successiva il processo di elaborazione procede ciclicamente passando attraverso l'analisi della deviazione degli angoli.

Al fine di ottimizzare questo passo nella mappa del metodo (si faccia riferimento al diagramma di flusso in figura 55), si è valutata l'ipotesi di interporlo in parallelo rispetto all'analisi degli angoli. Affermazione contemplata solo nell'utilizzo di sistemi *Multi processor*. Infatti, se il passo di scansione è molto minore le lato minimo del poligono, la maggior parte delle iterazioni passerà per il passo del *Continuous Following*, che si trova alla quattordicesima operazione nel diagramma di flusso. “*Shuntando*” quindi il *Continuous Following*, si è riusciti ad ottenere un guadagno in termini della velocità di esecuzione del 60% per quanto riguarda i campioni esaminati. Questa scelta di ottimizzazione, si noti, trae ispirazione dal mondo dell'elettronica digitale, ed in particolare dalla struttura del sommatore *Linear carry select*. Il tempo di esecuzione quindi, segue lo stesso andamento lineare, a parte un coefficiente iniziale, che dipende dal tempo di esecuzione degli *scan edge*. Da esso si ottiene quindi anche un ottimo rispetto al numero di *scan edge* utili all'algoritmo, che è prossimo a 6 [14].

Viene di seguito riportato un esempio di rilevamento vertici, utilizzando il metodo “Ad inseguimento”:

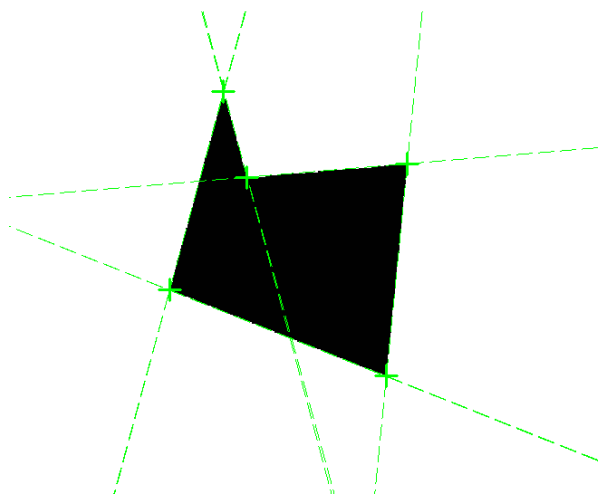


Figura 69. Visualizzazione dei risultati del metodo ad inseguimento.

4.4 Il metodo ad inseguimento - analisi di immagini nel mondo della micro e nano elettronica

In questo paragrafo verrà mostrato lo pseudo visionario approccio nell'analisi automatica di immagini provenienti da acquisizioni da dispositivi e spaccati del mondo della micro e nano elettronica. Analizzando i risultati ottenuti con l'algoritmo sviluppato, ci accorgiamo che esso

interpreta uno strumento capace di ricostruire “intelligentemente” le forme di una qualsiasi forma bidimensionali, adottando opportune condizioni iniziali.

Dando quindi un'occhiata alla figura 54, immaginiamo quindi di dover analizzare dove si trovano le differenze rispetto al layout progettato, compito che possiamo affidare agli algoritmi di “*pattern matching*”. Con l'ausilio del metodo ad inseguimento, il sistema di visione potrà altresì compilare automaticamente un report, sia testuale che grafico, dei parametri che si desidera controllare rispetto al processo. Da questi risultati, possiamo ottenere anche informazioni dettagliate riguardo ad una mancata deposizione di uno o più strati, effetti indesiderati di *etching*[15], allineamento maschere, etc.

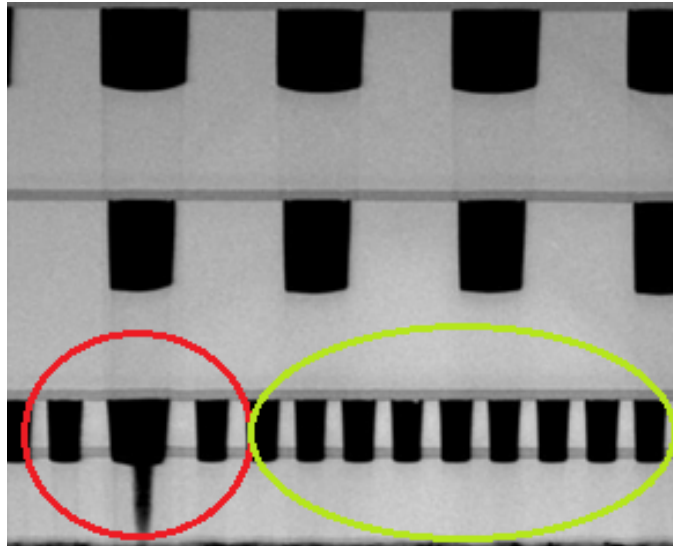


Figura 70. Particolare di *etching* eccessivo.

Immaginiamo di applicare adesso il metodo “Ad Inseguimento” all'immagine relativa al dispositivo mostrato in Fig.70. Analizzando l'immagine notiamo che siamo in presenza di un *etching* eccessivo in prossimità della zona indicata dal cerchio rosso. *Ciò che vorremmo ottenere* dai processi di fabbricazione è quel che invece si verifica nell'area cerchiata in verde. Applicando la ricerca automatica dei vertici, possiamo intanto risalire ai vertici dei “poligoni” rappresentati da ossidi, deposizioni, etc. In tal modo, rispetto al modello sviluppato a layout per la progettazione dei dispositivi, il sistema potrebbe già essere in grado di calcolare le differenze tra le posizioni dei vertici rilevati ed attesi. La precisione nella misura di scostamento dipenderà ovviamente anche dalla risoluzione del sensore della telecamera adottata per l'acquisizione dell'immagine, e dalle eventuali condizioni di rumore aggiuntive.

4.5 Misure e controlli sui lati dei poligoni rilevati

Una volta ottenuti i vertici del poligono, è possibile adesso procedere con le misure richieste da specifica. I vertici rilevati, saranno i nuovi punti di riferimento per i controlli di cui parleremo in questo paragrafo. Per cominciare, per verificare che ogni lato sia all'interno di una fascia di 2mm, possiamo utilizzare in prima battuta un algoritmo *Gravity and Area*, per vedere quanta area ricade all'interno della suddetta fascia. I parametri di giudizio da dare in pasto all'algoritmo saranno quelli da specifica. Mentre le regioni di impostazione verrà automaticamente assegnata una volta rilevati i vertici. Infatti, immaginando che la regione di ricerca sia rettangolare, i due vertici di un lato saranno la coppia di vertici rilevati, e da questi costruiremo due punti sulle normali al lato citato distanti 2 mm dal lato. Per ottenere questo risultato, basta implementare le procedure già viste nello

sviluppo di questo elaborato. Quindi una volta prelevati i vertici, memorizzati all'interno di un array, occorrerà estrapolare la coppia di punti, e le coordinate del *FIGDATA*&() verranno impostate come descritto.

Al fine di rilevare l'area del poligono, be, abbiamo già risposto al primo passo di processo di uno qualsiasi dei metodi sviluppati. Infatti, l'algoritmo *Gravity and Area* fornisce già l'informazione in merito al pixel visto sotto forma di macchia. Non rimane che implementare il fattore correttivo che dipende dal sistema di riferimento utilizzato.

Per quanto infine riguarda tutte le altre misure di linearità, etc. in ultima battuta, utilizzando il meccanismo detto per il primo punto in questo paragrafo, basta utilizzare degli ulteriori *scan edge position*, che puntano questa volta sull'intero lato trovato. Per il resto, basta utilizzare di volta in volta l'impostazione della regione di ispezione, che dipenderà dai vertici rilevati. Il lavoro a monte quindi, richiede un tempi di sviluppo pari a 20 minuti per ogni nuovo controllo implementato.

4.6 Applicazione al mondo della micro e nano elettronica

In figura 71 troviamo uno spaccato di una memoria realizzata con tecnologia STI, ed è possibile notare che si verificano degli effetti di *Loading e Micro-Loading* in seguito ad un processo di etching al plasma [16].

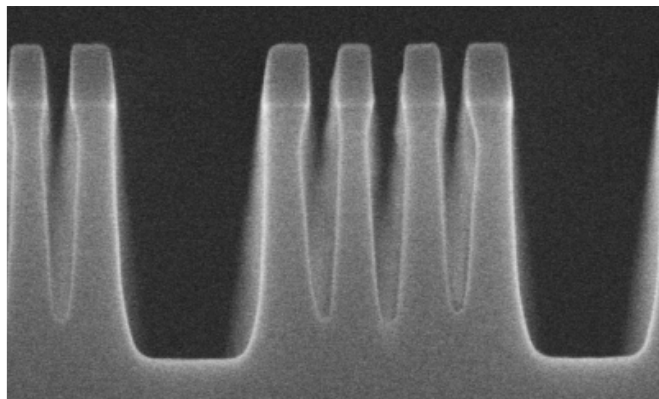


Figura 71. Analisi immagine *STI*.

Volendo analizzare la struttura riportata in figura 71, potremmo applicare il metodo ad inseguimento per ottenere le posizioni relative dei vertici (calibrate secondo l'unità di misura scelta). Un semplice controllo sul numero dei vertici rilevato sarebbe già la prima informazione facilmente ottenibile, per discriminare che non vi sono tutti gli elementi presenti. Impostando poi un calcolo per ottenere le differenze relative ai vari vertici rispetto al layout. Da queste, inserendo delle opportune soglie di giudizio, potremo in automatico ottenere informazioni relative ad un'inefficienza in un passo di processo, una deposizione fuori specifica, etc. Ma vogliamo spingerci verso controlli più complessi.

Per quanto invece riguarda ad esempio l'elemento 2 e 4 della figura 72, possiamo notare che a causa della carica intrappolata, probabilmente la direzione di bombardamento da parte del plasma è stata deviata. Per questo otteniamo una "rientranza" più pronunciata all'interno.

In tal caso, il primo esempio di controllo che si potrebbe applicare è quello dello spessore massimo e minimo in prossimità dei *tranch*. Le regioni dei controlli di *width* potranno essere quindi automaticamente impostate grazie all'ausilio dei vertici rilevati dal metodo ad inseguimento. Impostando le opportune soglie dettate dal progetto, tali algoritmi saranno quindi in grado di segnalare qual è il rapporto tra il fattore di *etching* verticale ed orizzontale. Collezionando poi i vari

risultati relativi ai vari elementi, potrà ulteriormente esser in grado di compilare statistiche, archiviare, visualizzare risultati nella modalità desiderata.

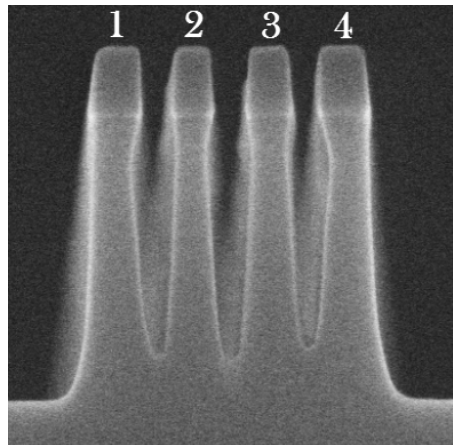


Figura 72. Particolare - analisi immagine *STI*.

5. Conclusioni e sviluppi futuri

All'epoca della scrittura di questo elaborato, siamo a circa trent'anni della prima apparizione di un sistema di visione artificiale nel settore dell'automazione industriale. Ad oggi siamo già in possesso di piattaforme hardware e software in grado di assolvere ai compiti svolti dal magnifico organo "occhio" umano. Dando uno sguardo al prossimo futuro, oltre alla crescita in termini prestazionali dei sistemi, assisteremo allo sviluppo di piattaforme dotate di gradi di intelligenza superiore. Stiamo già assistendo all'integrazione di intelligenza artificiale su piattaforme hardware industriali. I sistemi di elaborazione centrale saranno sempre più svincolati da compiti che verranno assolti già dai sistemi di visione artificiale. Verranno così distribuite le capacità di elaborazione a livello di sistema.

In termini di settore applicativo, ad oggi i sistemi di visione artificiale vengono impiegati in quasi tutti i settori di produzione industriale. È praticamente impensabile che non ci sia campo di applicazione per sensori e sistemi di visione.

L'applicazione del lavoro mostrato in questo elaborato, trova ad oggi impiego per la soluzione alle specifiche dell'applicazione presentata. Una semplice modifica di parametri o un ulteriore sviluppo dei metodi presentati ed elaborati, potrebbe sicuramente esser d'ausilio allo sviluppo di altre applicazioni. Nell'ottica di rendere specifico il contesto di applicazione, il secondo metodo elaborato, vista la tendenza e la necessità dell'automatizzazione dei processi, ben si presenta per la risoluzione di misure automatiche su qualsiasi tipologia di immagine acquisita. Un'ulteriore fattore di "robustezza", viene conferito dal fatto che è stato sviluppato su una piattaforma industriale. Si presuppone possa essere impiegato nel settore della micro e nano elettronica come accennato. Si sta' infatti valutando la possibilità di utilizzo combinata ai microscopi elettronici. Basti pensare all'integrazione del riconoscimento facciale su telecamere e fotocamere di utilizzo commerciale, e nel riconoscimento accessi.

Bibliografia

- [1] VON BERTALANFFY L., *Teoria generale dei sistemi. Fondamenti, sviluppi, applicazioni*, ILI, 1968.
- [2] Cochran W.G. (1977) *Sampling Techniques*, J. Wiley.
- [3] CCS Inc. – 2014 Catalog & Application notice.
- [4] Opto Engineering – 2014 Catalog & Application notice.
- [5] Prof. Federico Zaraga. Slides of the courses of optoelectronic systems and digital imaging. url: <http://risorse.dei.polimi.it/sensorlab/Education.html>.
- [6] Paul M. Hubel Foveon Inc. USA. Foveon technology and the changing landscape of digital cameras. In *Thirteenth Color Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications*, 2005.
- [7] Antonio Longoni, Federico Zaraga, Giacomo Langfelder, and Luca Bombelli. The transverse field detector (tfd): A novel color-sensitive cmos device. *IEEE Electron Device Letters*, 2008.
- [8] 996-2013 MVTec Software GmbH. All rights reserved. All specifications are subject to change without notice. HALCON is a registered trademark of MVTec Software GmbH. 06/2013
- [9] Smart camera IVC-3D 100, Classe IIIb / 3B laser Operation Manual. 2013 Sick- GmbH
- [10] EtherCAT Product Guide, EtherCAT Technology Group (ETG).
- [11] Scheimpflug, Theodor. 1904. Improved Method and Apparatus for the Systematic Alteration or Distortion of Plane Pictures and Images by Means of Lenses and Mirrors for Photography and for other purposes. GB Patent No. 1196. Filed 16 January 1904, and issued 12 May 1904.
- [12] Edwin Abbott, *Flatlandia: storia fantastica a più dimensioni*, traduzione di Masolino d'Amico, collana gli Adelphi, Adelphi, 2003
- [13] L'irresistibile spirito dell'imprenditore. Kazuma Tateishi.
- [14] Rabaey – circuiti integrati digitali. L'ottica del progettista.
- [15] S. M. Sze, K.K. Ng., “Physics of semiconductor devices” third edition, Wiley. 2007
- [16] J. D. Plummer, “Silicon VLSI technology. Fundamentals, practice and modeling”, Prentice Hall 2000

**Appendice A. Technology overview Pattern Matching Algorithm
Shape search III**

Appendice B. FH-3050 vision system Instruction sheet

Appendice C. FH-SC04 camera instruction sheet

Appendice D. Vision platform FH data sheet