

Alma Mater Studiorum Università di Bologna

Scuola di Ingegneria e Architettura

Corso di laurea magistrale in Ingegneria Informatica

SISTEMA DI VISIONE PER L'ISPEZIONE DI BIELLE

Progetto in “Computer Vision and Image Processing M”

github.com/bobcorn/connecting-rods-inspector

Docente:

Prof. Luigi Di Stefano

Studente:

Marco Rossini

Anno Accademico 2019/2020

Indice

Abstract	3
1. Prima parte	4
1.1. Segmentazione dell'immagine	4
1.2. Analisi dei blob	5
1.2.1. Isolamento delle bielle	5
1.2.2. Analisi dei fori e classificazione	5
1.2.3. Posizione ed orientamento	6
1.2.4. Dimensioni	7
2. Seconda parte	8
2.1. Gestione dei distrattori	8
2.2. Separazione dei punti di contatto	10
2.3. Gestione della limatura di ferro	13
Conclusioni	14

Abstract

Il presente report illustra i ragionamenti e le scelte adottate per la realizzazione di un sistema di visione in grado di eseguire l'analisi su due tipologie di bielle per motociclette, allo scopo di guidare un robot che possa smistarle in funzione della loro tipologia e della loro dimensione.

1. Prima parte

1.1. Segmentazione dell'immagine

I requisiti del progetto richiedono che le bielle siano analizzate nelle loro caratteristiche distintive, come dimensioni, orientamento e numero di fori. Per assolvere questo compito, si è scelto di adottare un approccio basato sull'analisi dei blob. Il primo passo è stato pertanto costituito dal separare gli oggetti di interesse dallo sfondo effettuando una **binarizzazione** dell'immagine, agevolata dal metodo di illuminazione scelto per gli oggetti, avvenuto secondo la tecnica del *backlighting*. Nonostante gli indubbi vantaggi del metodo di illuminazione adottato per lo scopo richiesto, esso non ha potuto però garantire l'erogazione di un'illuminazione globalmente uniforme. Per assicurare robustezza verso scenari di illuminazione variabile, si è pertanto scelto di non effettuare la binarizzazione dell'immagine adottando una soglia globale, ma calcolando una nuova soglia ottimale *ad hoc* per ogni specifica immagine in ingresso, mediante l'**algoritmo di Otsu**.

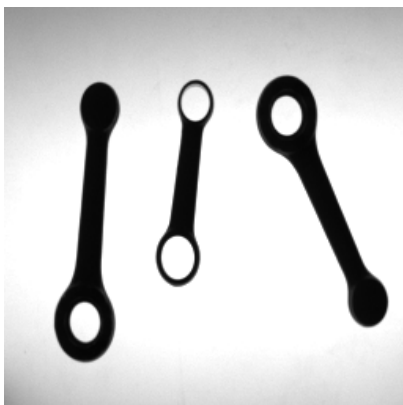


Fig. 1.1a – Immagine originale.

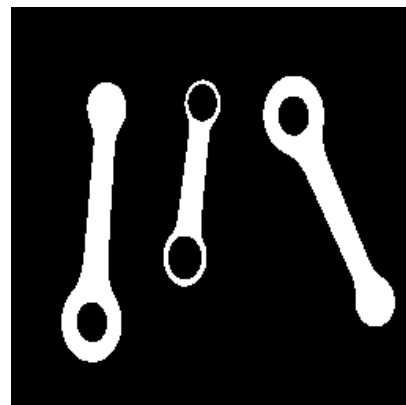


Fig. 1.1b – Immagine binarizzata.

1.2. Analisi dei blob

Ottenuti gli oggetti in primo piano in questo modo, si è reso necessario valutare gli approcci che permettessero di analizzarli più dettagliatamente.

1.2.1. Isolamento delle bielle

Al fine di poter analizzare singolarmente ogni singola biella ed estrarne le informazioni relative, si è scelto di effettuare un **labeling delle componenti connesse** sull'immagine binarizzata. L'operazione effettuata ha permesso di distinguere ogni diversa biella come un oggetto a sé stante, consentendo di iterare su ogni specifica componente connessa rilevata, permettendone un'analisi specifica.

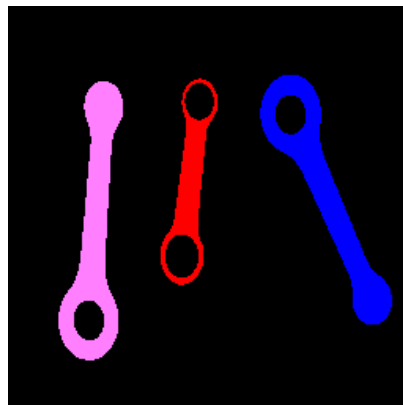


Fig. 1.2.1 – Immagine binarizzata a seguito del labeling.

1.2.2. Analisi dei fori e classificazione

La prima informazione richiesta per ogni biella è la sua **tipologia** (A o B), che risulta determinata dal numero di fori in essa presenti (uno o due). Per assolvere questo compito sono stati valutati alcuni approcci percorribili, tra cui: il calcolo del numero di Eulero, l'analisi della gerarchia di contorni contenuti nella componente, o l'esecuzione di un secondo labeling delle componenti connesse sulla componente stessa (invertita). Poiché tra le informazioni richieste sono presenti anche il **diametro** e la **posizione del centro** dei fori stessi, si è ritenuto più elegante adottare l'ultimo approccio basato sull'esecuzione di un **secondo passo di labeling**, in quanto esso restituisce direttamente anche la posizione del centro della componente-foro e la sua area (dalla quale è possibile ricavare il diametro, sfruttando la mutua

relazione). Grazie al secondo passo di labeling, è stato pertanto possibile classificare la tipologia di ogni biella (contando semplicemente il numero delle componenti-foro rilevate al loro interno), ottenere la posizione dei loro centri e calcolare il loro diametro.

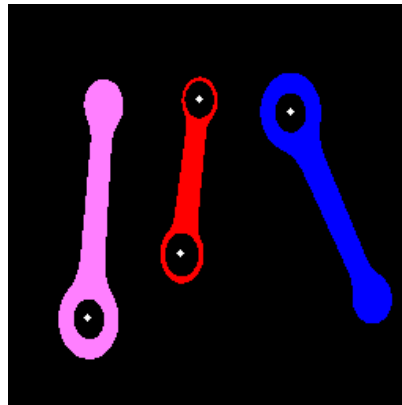


Fig. 1.2.2 – Immagine binarizzata con identificazione dei fori.

1.2.3. Posizione ed orientamento

Un ulteriore requisito è il calcolo della **posizione** e dell'**orientamento** di ogni biella. Per ottenere l'orientamento, si è scelto di calcolare i **momenti centrali** della componente-biella, da cui è stato possibile ricavare l'asse con inerzia minima (asse maggiore), e l'asse con inerzia massima (asse minore). L'angolo di orientamento è stato quindi calcolato come l'angolo compreso tra l'asse orizzontale e l'asse maggiore. Per ottenere la posizione, siccome non risulta specificato se essa sia intesa come il **centroide** dell'oggetto oppure come il suo **rettangolo di chiusura minimo orientato (MER)**, si è scelto di fornire entrambe le informazioni. Il centroide è stato restituito direttamente dalla funzione di labeling, per cui è risultato immediatamente disponibile. Per calcolare il MER, si sono invece calcolati i quattro punti, giacenti sul contorno della componente-biella, posti alla massima distanza (positiva e negativa) dall'asse maggiore e minore. Calcolati questi quattro punti massimali, corrispondenti ai punti di contatto tra i lati del MER e la componente-biella, è stato possibile ricavare le equazioni delle quattro rette relative ai lati stessi, sfruttando, per il calcolo di ogni retta, il corrispondente punto massimale e l'orientamento della retta stessa. Intersecando le quattro equazioni così individuate, è stato possibile ottenere i quattro punti corrispondenti ai vertici

del MER. La posizione di ogni biella è stata quindi fornita come la coppia di informazioni data dal suo centroide e dai vertici del suo MER.

1.2.4. Dimensioni

Per soddisfare il requisito di calcolo della **lunghezza** e **larghezza** della biella, è stato possibile sfruttare il MER orientato ottenuto al passo precedente. La lunghezza è stata infatti fornita come la dimensione del lato del MER parallelo all'asse maggiore, mentre la larghezza come la dimensione del lato del MER parallelo all'asse minore. Per il calcolo della **larghezza in corrispondenza del baricentro**, si è scelto invece di calcolare i due punti, giacenti sul contorno della componente-biella, aventi minima distanza dall'asse minore (ovvero, appartenenti all'asse minore) e, contemporaneamente, massima distanza dall'asse maggiore. Individuati questi due punti, è stato quindi possibile ottenere la larghezza in corrispondenza del baricentro calcolando la distanza tra i punti stessi.

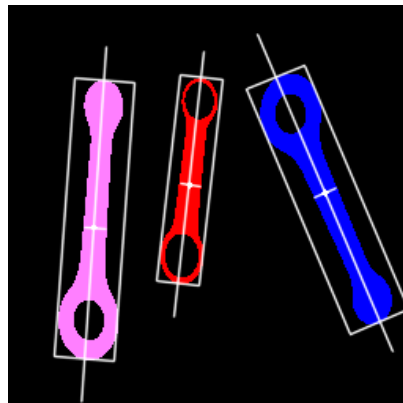


Fig. 1.2.4 – Immagine binarizzata con identificazione di posizione, orientamento e dimensioni delle bielle.

2. Seconda parte

2.1. Gestione dei distrattori

Tra i requisiti aggiuntivi, si richiede di gestire la presenza di **oggetti di disturbo** (come viti e rondelle), che non devono essere analizzati dal sistema. Per soddisfare questo requisito si è scelto di sfruttare, per ogni oggetto, la coppia di informazioni data dal numero di fori in esso rilevati, ed il suo descrittore geometrico di **allungamento**, definito come il rapporto tra la sua lunghezza e la sua larghezza. In questo modo, è stato possibile ignorare le viti (o eventuali molteplici viti adiacenti) ignorando semplicemente ogni oggetto privo di fori, così come eventuali molteplici rondelle (in numero superiore a due) tra loro adiacenti, ignorando ogni oggetto con un numero di fori superiore a due¹. Gestite in questo modo le casistiche relative ad una vite, a più viti adiacenti, e a tre (o più) rondelle adiacenti, sono rimaste da scongiurare le casistiche relative ad una singola rondella, o due rondelle adiacenti. Calcolando il descrittore geometrico di allungamento per ogni oggetto presente nella scena, ottenuto con facilità dai valori di lunghezza e larghezza forniti dalla funzione di calcolo del MER realizzata precedentemente, è stato possibile osservare che: le bielle presentano un valore sempre strettamente superiore a due, mentre le rondelle un valore sempre prossimo a 1 (per via della loro forma circolare). Calcolando questo descrittore per ogni oggetto sfuggito al filtraggio basato sul numero di fori, è stato possibile ignorare le casistiche relative ad una rondella, o due rondelle adiacenti, semplicemente imponendo di ignorare ogni oggetto con soglia di allungamento inferiore o uguale a 2. Tale intervallo include infatti la casistica relativa alla presenza di una singola rondella (con rapporto di allungamento prossimo a 1), e la casistica relativa a due rondelle adiacenti (con rapporto di allungamento minore o uguale a 2). Si è scelto di non adottare una strategia di filtraggio basata unicamente sul descrittore di allungamento, ma di sfruttare anche l'informazione relativa al numero di fori, al fine di garantire maggiore

¹ si noti che questo filtraggio non ha escluso eventuali bielle in situazione di contatto, poiché esso avviene solo successivamente al passo di separazione delle bielle, come illustrato nel paragrafo successivo.

robustezza al sistema, poiché le viti (avendo forma molto variabile), potrebbero in certi casi presentare un allungamento simile a quello delle bielle. Ciò non risulta invece possibile per le rondelle, per via della loro forma circolare (per definizione).

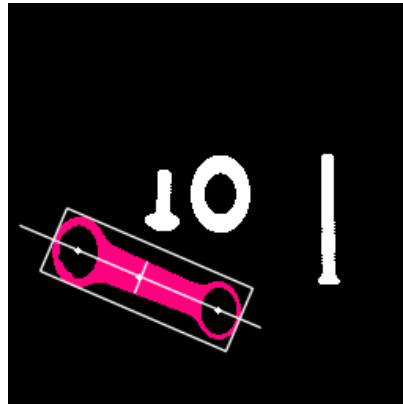


Fig. 2.1 – Immagine binarizzata con identificazione ed esclusione dei distrattori.

2.2. Separazione dei punti di contatto

Un ulteriore requisito è costituito dalla gestione di **bielle in condizione di contatto**, la cui separazione si rende necessaria ai fini di una corretta classificazione. Il soddisfacimento di questo requisito risulta sfidante, poiché la tecnica di illuminazione adottata, pur favorendo ottimamente il processo di binarizzazione, abbatte però i livelli di intensità dei pixel posizionati in prossimità dei punti di contatto: gli oggetti appaiono come se fossero “fusi” tra di loro, e non risulta applicabile una separazione dei contatti basata sul riconoscimento dei contorni. Per soddisfare il requisito, si è pertanto scelto di adottare un approccio alternativo, basato sul calcolo dell'**inviluppo convesso** delle componenti-bielle, e sull'analisi dei **difetti di convessità** (entrambi ottenuti mediante funzioni apposite, messe a disposizione dalla libreria OpenCV). Ottenute le zone di non convessità dall'inviluppo convesso, insieme ai relativi punti giacenti sul contorno della componente-biella a distanza massima dall'inviluppo convesso (anch'essi restituiti dall'apposita funzione OpenCV) per ogni zona di non convessità, sarebbe idealmente possibile **congiungere con una linea** ciascuno di questi punti con il suo vicino più vicino, al fine di poter separare correttamente le bielle. In realtà, ciò non risulta praticabile, poiché i punti di difetto rilevati in questo modo risultano in numero maggiore rispetto a quelli necessari (ossia, quelli in prossimità di un reale contatto), a causa di alcune non convessità “spurie” presenti sul contorno delle componenti-bielle. Al fine di poter applicare con successo questa strategia, si è reso pertanto necessario effettuare un **filtraggio preliminare al tracciamento delle linee**, in modo tale da escludere i punti non corrispondenti a reali situazioni di contatto. Tra le possibili scelte di filtraggio percorribili, si è osservato come, analizzando una piccola finestra di pixel centrata su ogni punto di non convessità, per i punti “spuri” si rilevi una quantità di pixel bianchi e neri quasi equa, mentre per i punti di reale contatto si rilevi una quantità di pixel bianchi molto superiore rispetto a quelli neri. Ciò è intuitivamente dovuto al fatto che una finestra centrata sui punti spuri catturi per metà lo sfondo, e per metà il contributo di una singola biella, mentre una finestra centrata sui punti di reale contatto catturi una porzione di sfondo (minore), e un contributo (maggiore) non più dovuto ad una singola biella, ma a due bielle ravvicinate. In

virtù di questa considerazione, si è applicato un filtraggio all'insieme totale dei punti di difetto rilevati imponendo una **soglia minima sul rapporto tra pixel bianchi e neri** (si è scelto un valore pari a 1.5). Ottenuti i punti filtrati, si è reso necessario accoppiarli in funzione della loro distanza, affinché ogni punto venisse congiunto con il suo vicino più vicino, ma evitando di ricongiungere punti già precedentemente congiunti. Per risolvere questo **problema di assegnamento**, si è scelto di utilizzare l'**algoritmo Ungherese** (o algoritmo di Kuhn–Munkres), assumendo come costo la distanza euclidea tra tutte le possibili coppie di punti, e minimizzando il costo totale. Le coppie assegnate in questo modo sono state quindi tutte congiunte tra di loro. Il tracciamento delle linee avvenuto secondo questa strategia, effettuato sull'immagine binarizzata precedentemente al passo di labeling delle componenti connesse, ha consentito di **separare con successo gli oggetti**, che risultavano prima identificati come connessi. È stato tuttavia rilevato, in una delle due immagini di test in cui sono presenti bielle in contatto, un caso sfavorevole in cui il tracciamento della linea di separazione ha causato l'**apertura del foro** di una biella, a causa dell'estrema sottigliezza dello spessore del relativo anello. Ciò ha provocato un'**erronea classificazione** della biella (da tipo B a tipo A) per il mancato conteggio del foro non più visibile. Per correggere questo inconveniente, è possibile valutare l'esecuzione di una **dilatazione** dell'oggetto (prima o dopo la sua separazione) al fine di scongiurare l'apertura del foro, qualora risulti tollerabile la conseguente sovrastima delle dimensioni calcolate (la strategia è stata verificata essere funzionante, ma non è stata implementata al fine di non alterare le dimensioni rilevate per l'oggetto). Tale sovrastima, tuttavia, potrebbe eventualmente essere calcolata e tenuta in considerazione come costante, al fine di rielaborare successivamente le misurazioni ottenute, sottostimandole nuovamente della stessa quantità. In alternativa, si potrebbe progettare una strategia di **tracciamento della linea più "intelligente"**, non più basata su un segmento che congiunga la coppia di punti, ma su un percorso tracciato in maniera incrementale che verifichi, pixel dopo pixel, che i pixel delle due componenti restino sempre in contatto con il percorso stesso (in caso contrario, esso effettuerà delle "curve").



Fig. 2.2.1a – Immagine originale con bielle in situazione di contatto.

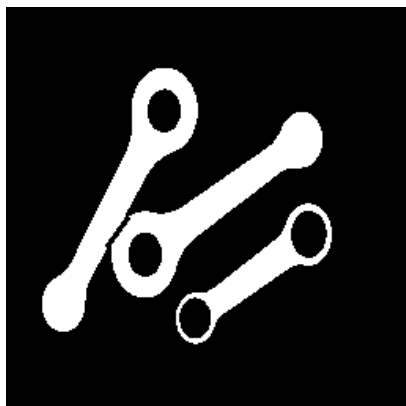


Fig. 2.2.1b – Immagine binarizzata con separazione del contatto.

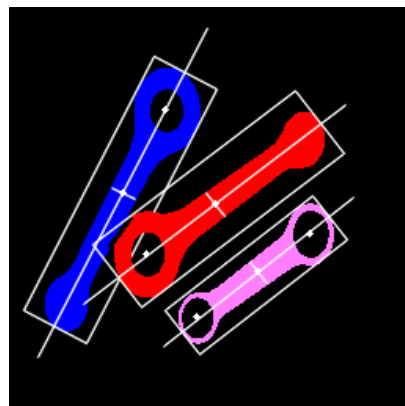


Fig. 2.2.1c – Immagine binarizzata con identificazioni corrette.



Fig. 2.2.2a – Immagine originale con bielle in situazione di contatto.

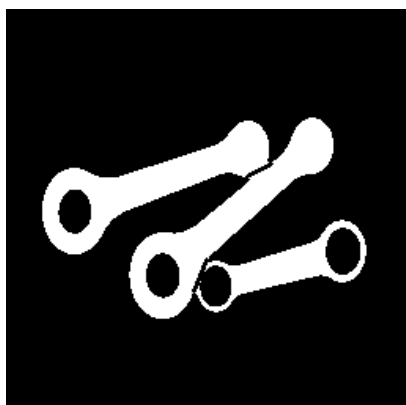


Fig. 2.2.2b – Immagine binarizzata con separazione del contatto e apertura indesiderata di un foro.

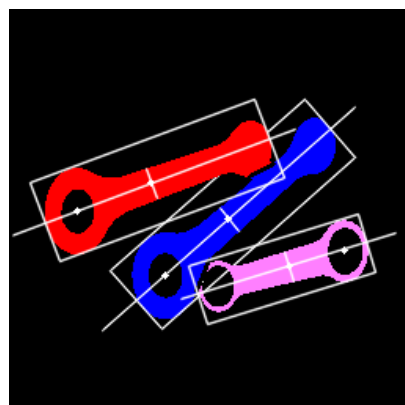


Fig. 2.2.2c – Immagine binarizzata con un'identificazione non corretta.

2.3. Gestione della limatura di ferro

L'ultimo requisito è costituito dalla gestione della **limatura di ferro** presente in alcune immagini, la quale è responsabile di causare una distinzione meno netta tra lo sfondo e gli oggetti di interesse. Per soddisfare questo requisito, si è osservato come la limatura, poiché molto fine, introduca sul primo piano delle piccole zone scure della dimensione di pochi pixel, paragonabili alla varietà di rumore di tipo **salt and pepper**. Si è pertanto scelto di trattare questo disturbo allo stesso modo, applicando quattro iterazioni di un **filtro mediano** con un kernel 3x3 prima del passo di binarizzazione. Il filtraggio applicato ha permesso di eliminare completamente la limatura, risolvendo il disturbo **senza alterare la morfologia delle bielle**. Altri approcci percorribili potevano prevedere l'utilizzo di operatori morfologici, l'utilizzo di un filtro bilaterale, o la gestione della limatura in fase di analisi degli oggetti rilevati, filtrando sulla base di una dimensione minima (pertanto, non intervenendo sull'immagine originale). Si è tuttavia scelto di adottare l'approccio mediante filtro mediano poiché, oltre a risultare semplice e rapido nel rimuovere ogni componente di limatura, ha permesso di correggere senza svantaggi anche alcune irregolarità presenti sui contorni relativi ai fori delle bielle, che in situazioni sfavorevoli avrebbero potuto causare potenziali errori di riconoscimento.



Fig. 2.3a – Immagine originale con presenza di limatura di ferro.



Fig. 2.3b – Immagine binarizzata senza filtraggio preliminare.



Fig. 2.3c – Immagine binarizzata con filtraggio preliminare mediante un filtro mediano.

Conclusioni

Complessivamente, i risultati ottenuti sulle immagini di test a disposizione sono risultati in linea con gli obiettivi. I requisiti relativi alla **prima parte** possono ritenersi soddisfatti con successo, e non sono emerse criticità rilevanti che dovrebbero far presupporre errori di analisi anche su immagini nuove. I requisiti relativi alla **seconda parte** presuppongono invece situazioni più imprevedibili, per cui il margine di errore potrebbe risultare più significativo. In particolare, la **limatura di ferro** dovrebbe risultare gestita correttamente anche su immagini nuove, a meno che in certe situazioni sfavorevoli essa non si raduni in agglomerati eccessivamente densi ed estesi. Anche la rimozione dei **distrattori** dovrebbe risultare robusta, a meno che non vengano introdotti oggetti non contemplati dalle immagini di test analizzate, nel qual caso potrà rendersi necessario adattare compatibilmente i valori di filtraggio basati sul descrittore geometrico di allungamento. Qualora si introducesse un oggetto con allungamento simile a quello delle bielle, potrà invece rendersi necessario ripensare la strategia di filtraggio, prendendo in esame caratteristiche differenti. Relativamente alla gestione dei **punti di contatto**, le immagini di test a disposizione (poiché in numero molto limitato) non permettono di prevederne il livello di robustezza anche su immagini nuove. É tuttavia ammissibile che la strategia implementata possa incorrere nel rilevamento di **falsi positivi** che sfuggano al filtraggio, causando il collegamento di punti “spuri” che potrebbero separare in maniera erranea le bielle. Al fine di scongiurare tale eventualità, potrebbe rendersi necessario ripensare la strategia di filtraggio dei punti di difetto, sostituendola con una basata su caratteristiche differenti, oppure approssimando il contorno. Infine, l’aspetto legato alle **performance** non è stato preso in esame, in quanto non richiesto, pertanto il software presenta sicuramente ampi margini di ottimizzazione. Nonostante ciò, esso risulta comunque piuttosto rapido, con un tempo di elaborazione² medio per ogni immagine di circa 0,03 secondi, in grado di consentire l’analisi di circa **33 immagini al secondo**. In virtù delle performance

² le performance riportate sono state calcolate utilizzando un processore Intel Core i5 Dual-Core 2,7 GHz.

rilevate, assumendo di apportare possibili ottimizzazioni, risulta ragionevole contemplare un suo eventuale utilizzo anche in scenari real-time.