

Computer Vision and Image Processing M

Relazione progetto: "Visual Inspection of Motorcycle Connecting Rods"

Marco Bertolazzi (mat. 0000884790) - marco.bertolazzi3@studio.unibo.it

1 - Introduzione

Il progetto si prefigge l'obiettivo di identificare e descrivere le bielle sia in condizioni agevoli che in condizioni in cui il processo di estrazione delle informazioni è ostacolato da elementi distrattori e da limatura di ferro.

L'elaborazione è basata sulla libreria OpenCV e si divide in quattro passi principali:

- miglioramento dell'immagine e binarizzazione;
- estrazione delle componenti connesse;
- classificazione dei pezzi meccanici;
- descrizione delle bielle.

In alternativa al classico calcolo dei momenti, è stata utilizzata la Principal Component Analysis (PCA) come tecnica per determinare alcune proprietà delle bielle quali il baricentro, l'orientazione, la lunghezza, la larghezza e la larghezza al baricentro.

La funzione della libreria OpenCV **PCACompute2** [1] implementa la PCA calcolando le medie, gli autovettori e gli autovalori a partire dal vettore delle coppie di coordinate (x, y) della componente connessa.

Questa scelta facilita l'estrazione delle suddette proprietà. Ad esempio, è sufficiente un cambio di sistema di riferimento (solidale alla biella e con un asse orientato secondo l'asse maggiore) seguito dalla ricerca di massimi e minimi tra le nuove coordinate della biella per ottenere la lunghezza e le due larghezze richieste.

Di seguito sono descritti con maggior dettaglio i passi di cui sopra.

2 - Miglioramento e binarizzazione dell'immagine

Nel codice questo task è implementato definendo la funzione `enhance_and_binarize(image)`.

Per miglioramento si intende:

- filtraggio dell'immagine con un filtro di media (funzione OpenCV **boxFilter**, [2]) o con un filtro mediano (funzione OpenCV **medianFilter**, [2]) e/o
- applicazione di operatori morfologici (funzione OpenCV **morphologyEx**, [2]).

Queste operazioni sono eseguite rispettivamente prima e dopo l'operazione di sogliatura secondo il noto algoritmo di Otsu (funzione OpenCV **threshold**, [3]).

La scelta degli strumenti da applicare è stata condizionata dai seguenti obiettivi:

- qualità della binarizzazione, misurata come spessore delle parti più sottili delle bielle nell'immagine binaria ;
- eliminazione della limatura di ferro.

Come vantaggio comune, l'applicazione di uno dei due filtri (di dimensione minima (3x3)) permette di risolvere a monte "errori" di binarizzazione in alcune immagini (es. tesi47.bmp, tesi48.bmp, tesi49.bmp), che andrebbero altrimenti risolti con una operazione morfologica di closing.

Il filtro di media permette di ottenere una migliore circolarità finale dei fori ma ha lo svantaggio di attenuare l'intensità nelle parti in cui la biella è più sottile e conseguentemente il suo spessore nell'immagine binarizzata.

Il filtro mediano invece tende a preservare meglio la polarità e quindi mantenere uno spessore più marcato, ma comporta un peggioramento nella circolarità dei fori nell'immagine binarizzata.

Invece per quanto riguarda l'eliminazione della limatura di ferro, tutti gli strumenti testati (filtro di media, filtro mediano e operazione di opening) sono risultati efficaci solamente sulla limatura di ferro più sottile.

Per ragioni di robustezza dell'intero programma, indipendentemente dall'applicazione di un filtro prima della binarizzazione, la funzione prevede una operazione di closing finale dato che la presenza di buchi spuri a valle di questa fase è fatale per il corretto funzionamento del programma.

3 - Estrazione delle componenti connesse

Una volta ottenuta l'immagine binaria si procede con l'estrazione delle componenti connesse (funzione OpenCV **connectedComponents**, [3]).

Di ogni componente connessa se ne calcola l'area e la si confronta con la soglia POWDER_DETECTION_THRESHOLD. Se l'area è inferiore alla soglia, allora la componente connessa è classificata come limatura di ferro residua e viene ignorata, altrimenti si tratta di un componente meccanico e si crea la maschera per le successive operazioni.

Data la semplicità di queste operazioni, non è stata definita alcuna funzione dedicata.

4 - Classificazione dei pezzi meccanici

Nel codice il task è implementato nella funzione `basic_blob_classifier(blob_mask)`. Per poter classificare un oggetto a partire dalla sua maschera binaria la funzione conta il suo numero di fori e calcola la sua elongazione.

Il numero di fori permette di fare una prima classificazione. Infatti si può dire che:

- zero fori: l'oggetto è un elemento distrattore di tipo "bullone";
- un foro: l'oggetto potrebbe essere un elemento distrattore di tipo "rondella" oppure una biella di tipo A;
- due fori: l'oggetto è una biella di tipo B.

Per risolvere l'incertezza del caso in cui l'oggetto abbia solo un foro si calcola l'elongazione dell'oggetto: se il valore è inferiore alla soglia `ELONGATEDNESS_RATIO` allora l'oggetto viene classificato come elemento distrattore "rondella", altrimenti come biella di tipo A.

Il numero di fori si ottiene contando quante componenti connesse ha la maschera (sfondo e oggetto esclusi).

L'elongazione invece è calcolata in modo approssimato ponendo sotto radice quadrata il rapporto tra gli autovalori provenienti dalla PCA.

Per motivi di ottimizzazione, i valori statistici del blob e le sue componenti connesse sono restituiti assieme al tag di classificazione in quanto verranno utilizzati anche dalla funzione di descrizione della biella.

5 - Descrizione della biella

Nel programma il task di descrizione è implementato dal metodo `rod_descriptor(blob_mask, blob_mask_connected_components, stats)`.

Da ogni biella, le informazioni estratte e memorizzate in una struttura a dizionario sono:

- **tipo**: ottenuto contando il numero di fori;
- **posizione**, comunemente definita dal baricentro: coincide con i valori attesi delle due distribuzioni;
- **fori**, per ogni foro è stato trovato:
 - **centro**: corrispondente al baricentro del foro
 - **diametro**: calcolato a partire dall'area del foro ($d = 2 \cdot \sqrt{A/\pi}$)
- **orientazione**: definita come arcotangente del rapporto tra le componenti dell'autovettore associato all'autovalore maggiore, definita tra $[-\pi/2, +\pi/2]$ e con crescita in senso orario.

Viene poi effettuato un cambio di sistema di riferimento in modo che il nuovo s.d.r. sia solidale all'oggetto (traslazione dell'origine nel baricentro) e orientato in modo che la prima coordinata sia nella direzione dell'asse maggiore della biella e la seconda coordinata sia ortogonale alla prima (rotazione degli assi).

Infine la funzione procede con il calcolo di:

- **lunghezza**: calcolata come differenza tra il valore massimo e il valore minimo delle coordinate relative all'asse maggiore;
- **larghezza**: calcolata allo stesso modo ma ricercando i valori estremi sull'asse ortogonale, ovvero l'asse minore;
- **larghezza al baricentro**: ottenuta estraendo le coordinate aventi la prima componente uguale a zero e facendo la differenza tra i valori estremi della seconda componente delle coordinate estratte.

6 - Risultati e conclusione

La funzione `show_rods(blob_mask, rod)` mostra a video e sul terminale i risultati.

7 - Riferimenti

[1]https://docs.opencv.org/master/d1/dee/tutorial_introduction_to_pca.html

[2]https://docs.opencv.org/master/d4/d86/group_imgproc_filter.html

[3]https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html

[4]https://docs.opencv.org/master/d3/dc0/group_imgproc_shape.html