Visual inspection of motorcycle connecting rods

Image Processing and Computer Vision

Project work

Alex Costanzino

alex.costanzino@studio.unibo.it

GitHub

Academic year 2020-2021

Abstract

The project work goal is to develop a software system aimed at visual inspection of motorcycle connecting rods. The system is able to analyse the dimensions of two different types of connecting rods to allow a vision-guided robot to pick and sort rods based on their type and dimensions.

In the first task, images contain only connecting rods, which can be of both types and feature significantly diverse dimensions and they have been carefully placed within the inspection area so to appear well separated in images. Furthermore, images have been taken by the backlighting technique to render rods easily distinguishable from background.

In the second task, images may contain other objects that need not to be analysed by the system, rods can have contact points or inspection area may be dirty due to the presence of scattered iron powder.

In every cases, type, orientation and position, length, width and barycentral width of the rods and also position and centers of the holes, should be calculated.

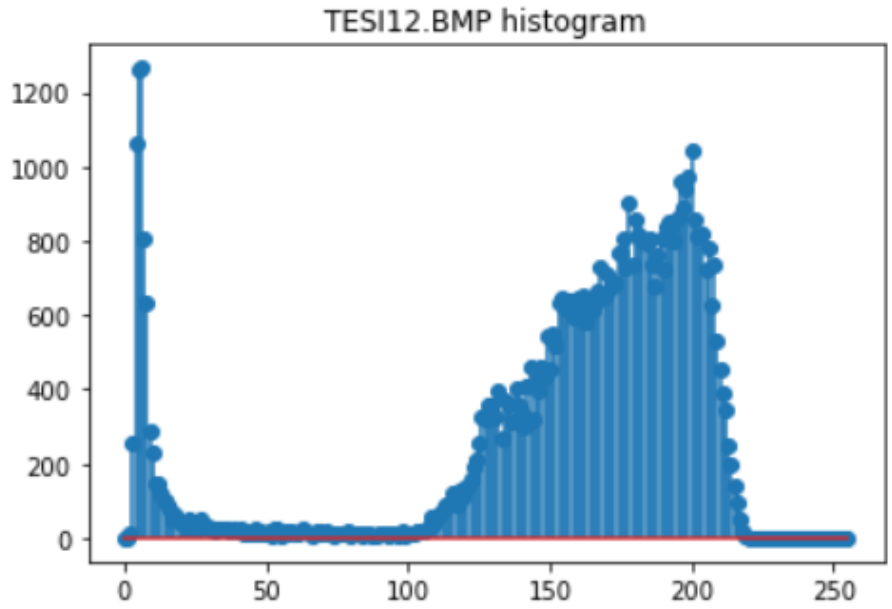
Project work

At beginning, a routine to create a dictionary that contains images from a folder was created. This function also perform a sanity check to ensure that all images are correctly loaded.
Moreover, lists containing the images labels for different task were created, in order to iterate easily over the dictionary.

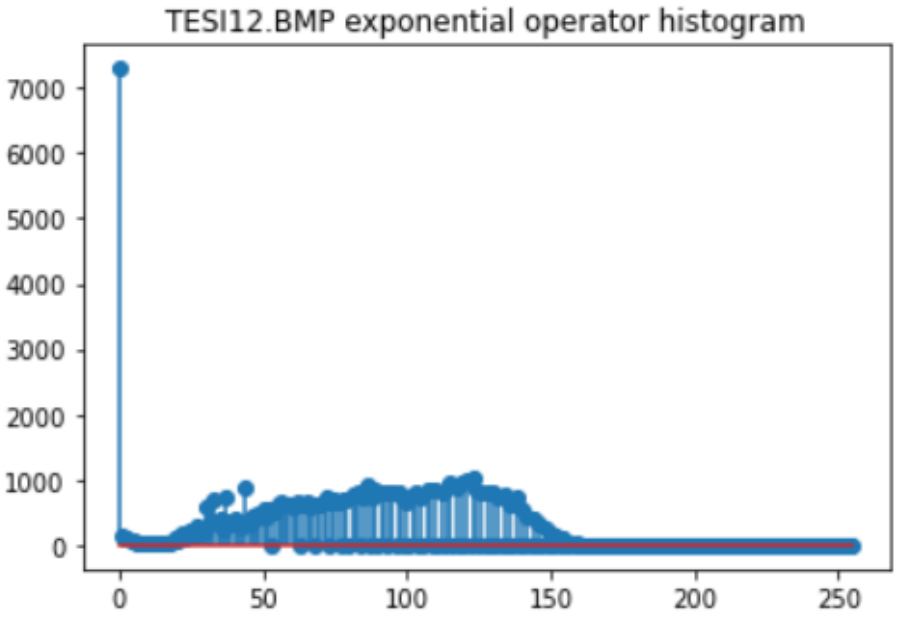
1 First task

As already mentioned, images were taken by the backlighting technique. However, for flexibility reasons the system should not require any change to work properly with lighting sources of different power. So, supposing that the stability over time of the lighting conditions cannot be guaranteed, an automatic threshold selection is required.

Histogram analysis The histogram shows a clearly bimodal distribution in all images, even though the two are not balanced, so a percentile thresholding rather than a mean one was chosen.

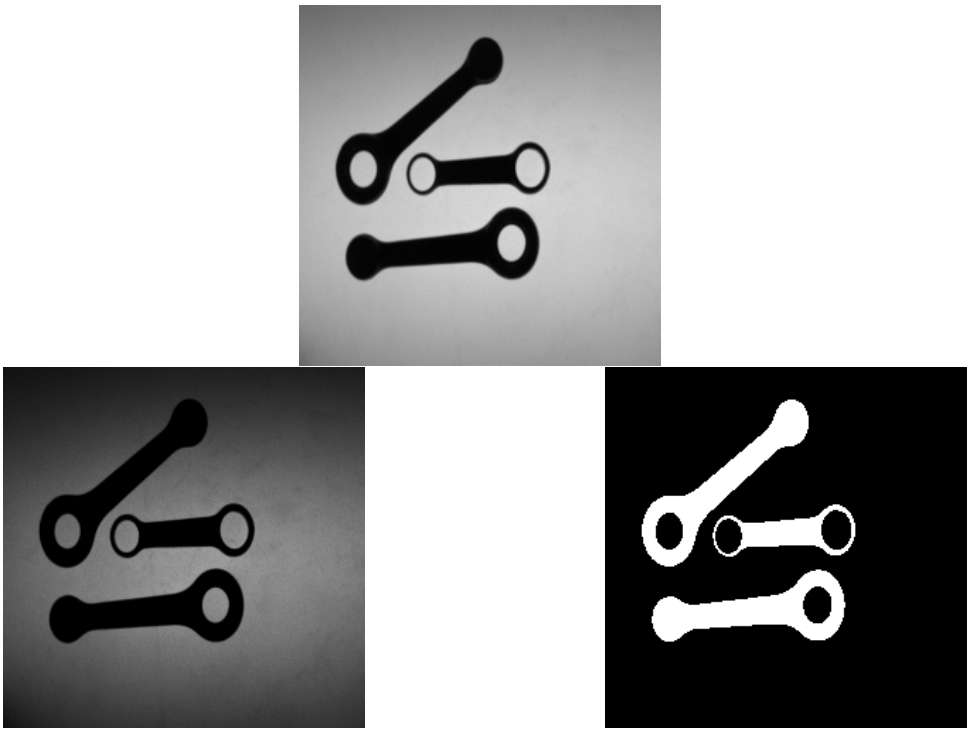


Exponential operator As first instance, the binarization was performed straightforwardly. However, it was not stable across all images. Hence, an exponential operator was applied in order to enhance the thresholding operation.



Binarization After having defined a function to calculate the percentile index, the thresholding was performed – in an inverse fashion – in order to accomplish the successive connected component analysis.

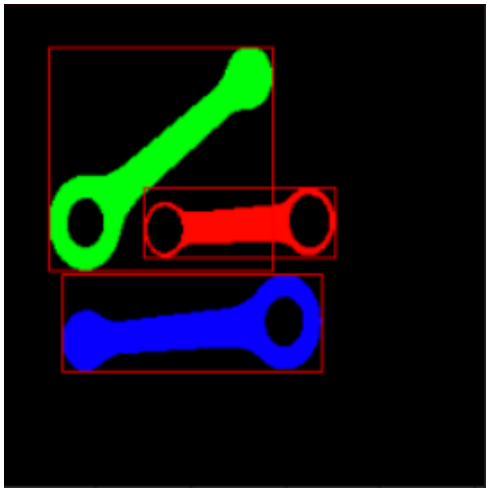
Later, an approach with Otsu threhsolding was also endeavoured.



Connected component analysis To accomplish connected component analysis several different functions have been defined:

- `show_blobs(num_labels,labels,stats)`: is a function that takes the number of connected components plus the background, the labels, the statistics on the components and shows the different connected components with an enclosing rectangle. It separates the images in different channels, one for each label, assigns a new label to show a different colour and then concatenates the channels back;
- `get_blob(labels,label)`: is a function that takes all the labels and the interested label and returns the corresponding mask. The mask is obtained by setting at `255` the pixels of the interest label;
- `distances(vertexes)`: given the vertexes of the oriented MER calculates the lenght of the segments;
- `get_barycenter_width(component,angle,centroid,contours)`: given a connected component, its position and orientation and its contours it returns the width at barycenter. It scans the points of the contour seeking for the one with the shortest distance with respect to the barycenter, by means of the signed distance.

At beginning, a loop over the dictionary that contains the binarized images was performed, using the OpenCV routine `cv2.connectedComponentsWithStats()`, to obtain a dictionary that contains some statistics on each connected components, such as labels, area, centroid, etc...



Then, another loop that iterates the dictionary over each image of the first task, and over each connected component (except for the background) was performed, in order to do:

- Mask extraction using the custom-made function and consequent contour extraction using the the OpenCV routine `cv2.findContours()`;
- A check on the number of found contours, to tell aparts the two different kinds of rods, with consequent analysis of the holes;
- Orientation calculation by means of the OpenCV routine `cv2.fitEllipse()` (the position is already calculated when the contours are extracted);
- Oriented MER extraction with calculation of length, width and width at barycenter.

2 Second Task

2.1 Variant One

Within the first variant, the system should be able to take into account of some other components, called distractors, that should be ignored along the analysis.
Since the distractors were significantly smaller than the rods, a simple filtering by area was done with a custom-made function `filter_by_area()`, that takes as parameters the number of labels, the labels, the statistics and the centroids from the connected component analysis, the minimum area as threshold and returns the relabelled image without the deleted components.

2.2 Variant two

Within the second variant, the system should be able to take into account some touching rods.
An endeavour with morphological operators was attempted (in particular with erosion), with several structuring elements. Nevertheless, the erosion degraded too much the head of the rods, making the analysis fails.

2.3 Variant three

The procedure is basically the same of the precedent tasks, but a median filter is applied before the exponential operator, to accomplish the salt and pepper denoising.

Since some point were still present in the binazired image, an opening operation is applied to remove these spurious points.