MERGING IMAGES USING MATLAB

S.Triaška,- M.Gažo[*]

Institute of Control and Industrial Informatics

Abstract:

The article is focused to describe some options working with images using Matlab. The papers include Matlab function from "Image Processing Toolbox" and they describe the way Matlab interpret the image. The base of our work is to create the program to merge two or more images and create the bigger image with higher resolution. The program for merging images was developed in Matlab, like ".m" file. The possibility to filter color on the image is applied in the program.

1 Introduction

Merging two or more images and photos is one of possible ways how to get high resolution image which includes all the details of the scanned object. The good example of merging images is a panoramic image. Panoramic shot is created from more than one simply image, these two or more images are merging in the divided part of the panorama. Tools for merging images include the program like Adobe Photoshop, its function is "photomerge", PTGui – whose task is only to merge images, and more other programs.

The main target of this article is to show the possible work with images using Matlab. Some basic function of the Image Processing Toolbox which is part of standard Matlab toolboxes are described in the article. The base of our work is to create a program whose task is to merge divided images of one object to one high resolution image. The merged images must correspond to the given element which was scanned.

2 The system description

The problem and task are committed to measure and evaluate the quality of the clutch lining. The program is the part of the procedure whose task is to measure automatically and evaluate the quality of the clutch lining. It is necessary to propose a software solution that creates a graphical model of clutch lining of the scanned element, and filter out unwanted parts, the data element that is irrelevant.

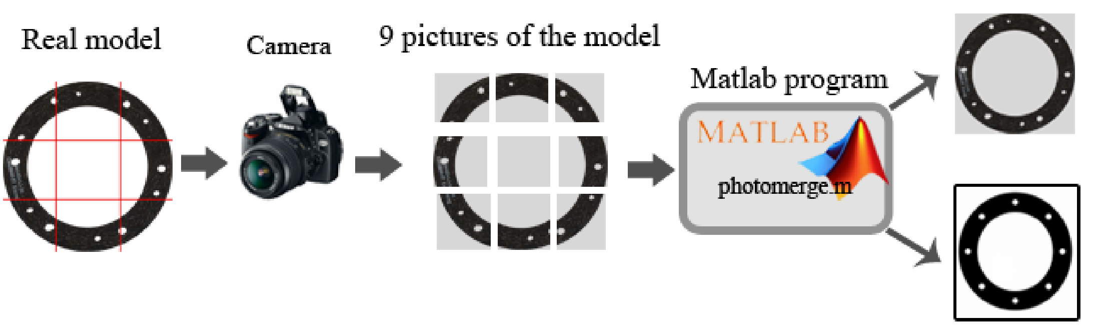

Fig.1 *Simple illustration of the system and the program*

Input to the system are images given from the camera. The camera is operated by program which is separate module of the system. The input image presents a part – segment of scanning element-object.

[*] Institute of Control and Industrial Informatics, Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia; stanislav.triaska@stuba.sk, martin.gazo@stuba.sk

The object consists of 9 separate segments - 9 pictures, which are scanned by the camera. The scanned pictures are in black and white color, and we used green illumination to increase the sensitivity of the camera. The program solution needs to merge these separate images to one image. The simple system illustration and how it works is shown in figure 1.

The system description by solving steps:

- The camera takes a picture of a real model (the picture is smaller than the model)

- The input to the program are the pictures from the camera

- The output of the program is the image created from the merge input images, and binary picture is created from the output image.

3 Matlab photomerge

To make the program for merging images we decided to use Matab version 7.01, because of its toolbox for image processing.

Image processing toolbox

The Image Processing Toolbox is one of standard Matlab toolboxes which supports a wide range of image processing operations including:

- Spatial image transformation
- Morphological operations
- Neighborhood and bloc operations
- Linear filtering and filter design
- Transform
- Image analysis and enhancement
- Image registrations
- Deblurring
- Region of interest operations

Matlab stores most images as two-dimensional arrays (i.e., matrices), in which each element of the matrix corresponds to a single pixel in the displayed image. (Pixel is derived from picture element and usually denotes a single dot on a computer display).

For example, an image composed of 200 rows and 300 columns of different colored dots would be stored in Matlab as a 200-by-300 matrix. Some images, such as truecolor images, require a three-dimensional array, where the first plane in the third dimension represents the red pixel intensities, the second plane represents the green pixel intensities, and the third plane represents the blue pixel intensities. This convention makes working with images in Matlab similar to working with any other type of matrix data, and makes the full power of Matlab available for image processing applications.[1]

An indexed image consists of an array and a colormap matrix. The pixel values in the array are direct indices into a colormap. The colormap matrix is an *m*-by-3 array of class double containing floating-point values in the range [0,1]. Each row of map specifies the red, green, and blue components of a single color. An indexed image uses direct mapping of pixel values to colormap values. A colormap is often stored with an indexed image and is automatically loaded with the image when you use the “imread” function.

Photomerge

The camera takes a picture of the element. Pictures are stored in program folder automatically, every picture in the folder has the same name given by the camera software automatically: "*scan_xy*", the xy- ending of the name picture presents: x -number of actual row and y -number of actual column.

The case of merging images can be realized in two different options:

- **The first option :**

The camera takes pictures one by one, and pictures don't have any over- end points. That means the next picture always starts in the point when the previous pictures stop. This rule is applied to the picture in vertical and horizontal axes of the final matrix picture.

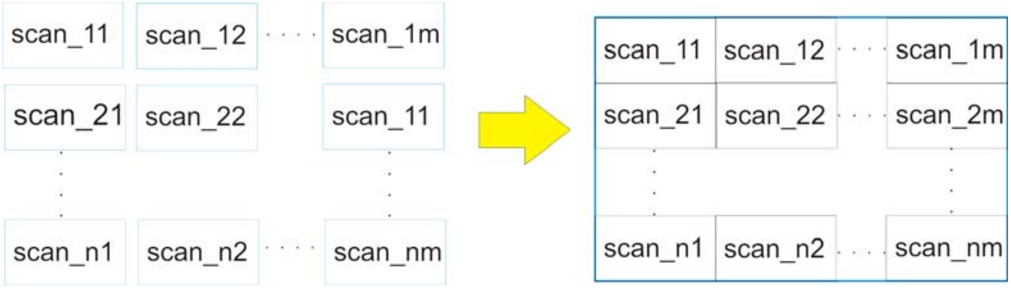

Fig 2. *Merging without over-end points*

We tested this solution of the program in clutch linings.

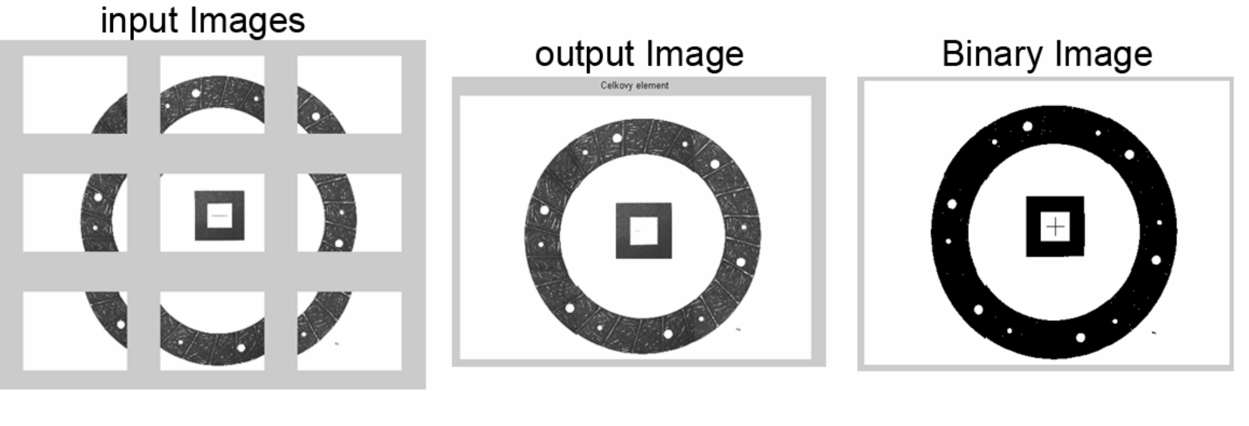

Fig.3 *Merging images of the clutch lining without over-end points*

The algorithms works in a way that in start it creates a new empty image frame. Using "for" cycle we read all grayscale images by the name suffix in a specified part. To read images we used standard Matlab function "imread". The read images we compose to empty image frame and in that way we create the final image.

The size of an input image is the matrix with dimension $img[i] = [n[k]\ m[l]]$,

n[k] – height of the image in pixels

m[l] – weight of the image in pixels

The output image has a dimension:

$$img = \begin{bmatrix} img[1] & \cdots & img[n] \\ \vdots & & \\ img[n] & \cdots & img[n] \end{bmatrix} = \begin{bmatrix} [n[k] \quad m[l]] & \cdots & [n[k] \quad m[l]] \\ \vdots & & \\ [n[k] \quad m[l]] & \cdots & [n[k] \quad m[l]] \end{bmatrix} = \left[\sum_{i=1}^{n} n_i[k] \quad \sum_{i=1}^{n} m_i[l] \right]$$

Binary output image is black and white logical image converted from output image using Matlab function "*im2bw*".

- **The second option :**

The camera takes picture one by one, and the picture always has some part of the over-end with the next picture. This rule is applied to the picture in vertical and horizontal axis of the final matrix picture.

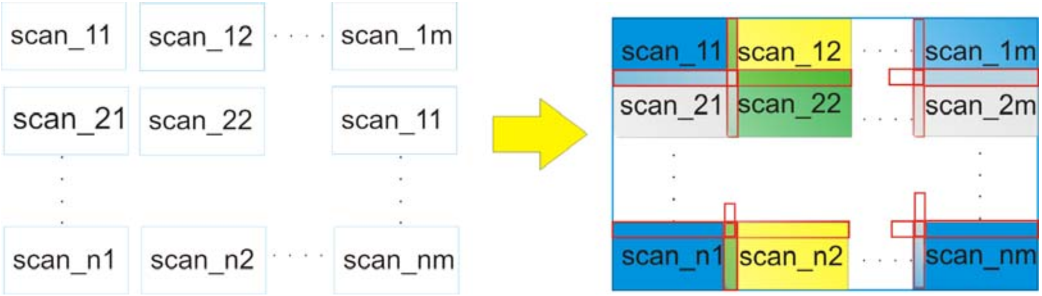

Fig 4. *Merging with over-end points*

The tested solution of this algorithm is shown in figure 5.

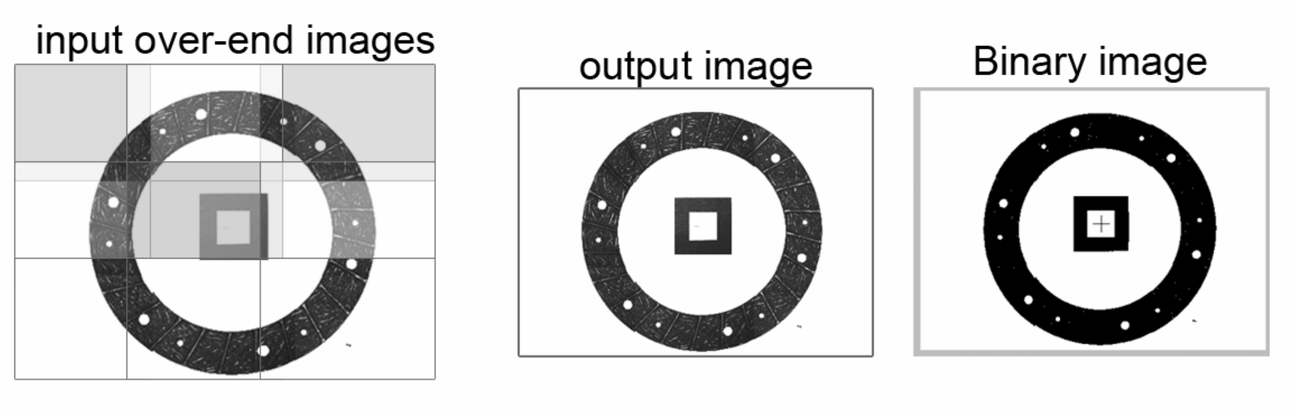

Fig 5. *Merging images of the clutch lining with over-end points*

Input images are grayscale and always have over-end points with the next vertical and horizontal image by the name suffix. The camera makes these over-end points automatically. Algorithm whose task is to compose the final image works by using two "for" cycles. The first cycle compares the row of input image, if images name suffix ends with "1x", these images are in the first row and they don't have over-end points. In this cycle there is another "for" cycle whose task is to compare columns of the input images of their elements of the matrix. If the images suffix ends with the name "x1", this means that the image is the first one in column and that they don't have over-end points with the previous image.
Over end point on input images (the first and the second row) is shown in figure 5.

Filtering the color in the image can be realized by using image colormap matrix or true colors. A truecolor image is an image in which each pixel is specified by three values one each for the red, blue, and green components of the pixel's color. Matlab store truecolor images as an *m*-by-*n*-by-3 data array that defines red, green, and blue color components for each individual pixel. Truecolor images do not use a colormap. The color of each pixel is determined by the combination of the red, green, and blue intensities stored in each color plane at the pixel's location. Graphics file formats store truecolor images as 24-bit images, where the red, green, and blue components are 8 bits each. This yields a potential of 16 million colors.

Conclusion

This paper shows some options and possibilities to merge images using Matlab. The first option of the merging images is without over-end points. This option because of sensitivity and quality of the final picture requires high position of the regulation of the camera. The second option shows the possibility merging images using over-end points. This algorithm is more complicated and more slowl than the first one.

Acknowledgments

Since this work has been supported by the Slovak Research and Development Agency, project reference number is: VMSP-P-0155-09. Therefore the authors of the article would like to give thanks for the support

References

[1] Image Processing Toolbox ^{TM} 7, User's Guide, Matlab, http://www.mathworks.com/help/pdf_doc/images/images_tb.pdf