Practical possibilities for photogrammetric point detecting in open source environment

K. Epresi*, Z. Tóth**
* Óbuda University, Székesfehérvár, Hungary
** Óbuda University, Székesfehérvár, Hungary
epresi.konrad@gmail.com
toth.zoltan@amk.uni-obuda.hu

Abstract- By the spread of UAVs in regular use has increased the importance of automatic recognition of land marks in photogrammetry. The proposed method for automatic recognition is developed under open source software, which results in a time and cost efficient solution.

I. Introduction

Our aim was to investigate the possibilities for automatic recognition of land marks in photogrammetry by using open source software. Development of automatic point detection in many areas would accelerate and make the process more economical. The UAV (Unmanned Aerial Vehicle) records in several areas are part of everyday technology. For example: the important role in the crime scene, where recording the circumstances is quick and accurate.

For our algorithm, we used the Python programming language and its OpenCV library for its image processing.

First of all, based on our preliminary research [10], we created the "perfect" landmark.

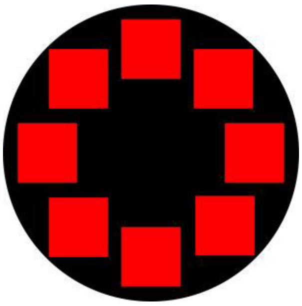

Figure 1. "Perfect" mark

We used the experiences of our previous researches. The point recognition procedure in the entire image area was split into several sections. First, we used the Hough transformation to look for the parts of the image on which we could run our algorithm to give a more accurate result. In each highlighted image part, we segmented the group of pixels which are valuable pixels for search. First, we defined the contour of all shapes (planes) within the circle identified by the Hough transformation using the "Image moment" method and then we calculated the centroid of the shapes. The center of the regression circle fitted to the centroids, regardless of the plane slides shape, which provides the exact position of the cutout image. As the cutout image position is known in the entire image area, the center of the detected point mark can be considered as known throughout the image area as well. The distortion of the circle identified by the Hough transformation (the appearance of the image as an ellipse) gives the opportunity to determine the angle of inclination between object plane and image plane. Based on the literature, we know that the Hough transformation is also suitable for the detection of ellipses [6], but we would like to try other methods during our research as a continuation of the research. Therefore, we hoped to increase the accuracy and we tried out our method.

II. Program Basics

A. OpenCV and Python

The OpenCV (Open Source Computer Vision Library) is an open source software (BSD license) which could be useful with computer vision, image processing, or even machine learning. It has over 2500 optimized algorithms including filtering, contour detection, histogram drawing, Hough transformation, face detection and face recognition, motion classification, tracking of objects, panoramic image creation, and image-based search. Many contributors to the development (since 1999) as: Intel, Willow Garage, followed by Itseez. We can find Python, Matlab and Java programming interfaces available on Windows, Linux, Mac OS, Android and IOS versions.

Figure 2. OpenCV logo [6]

Interpreter language (not separated from the source and object code) are such languages, where the script could only be executed with a Python interpreter. The Python interpreter has been created for multiple types of PC and operating systems. In addition, several libraries were created, one of them is the OpenCV [6]. We chose this programming language because we studied this at university.

B. Hough transformation

First of all, we had to find the easiest recognizable shape and the most applicable transformation. In other word this is the circle and the Hough transformation. Paul Hough, whom the method was named after, designed the algorithm to detect simple, parameter-specific shapes (straight lines, curves) in binary images in 1959, and then patented it in 1961.

In addition to detecting straight lines, other geometry elements are recognized by the process. It is necessary that the complexity of the geometric element carries the number of descriptive parameters needed for storage and computing capacity:

$$r^2 = (x - x_0)^2 + (y - y_0)^2$$

The three parameters are:

- r - the radius of the circle
- (x_0, y_0) - the coordinates of the center of the circle

In practice, there is an edge image on which to find the circle of the known radius, which is formed in the Hough space so that each circle line point corresponds to a circle with potential centers.

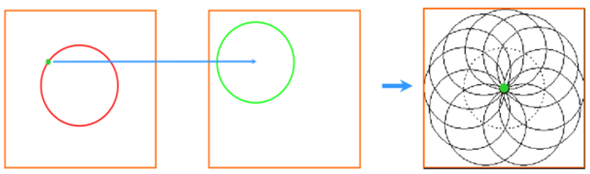

Figure 3. On the left side is the edge image, in the middle the Hough space with a detected circle, on the right there are maximum places in the Hough space → the centers of the detected circles [9].

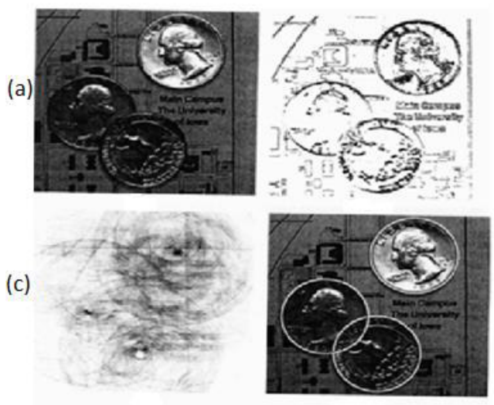

Figure 4. Example [9] (a) original image, (b) edge image, (c) 2D parameter space (searching for circles of a given radius), (d) detected circles in the original image

We used the Hough transformation to cut the useful area. Then we made a smaller image cut from the edge of the image, using the contour detection on this image, where we determined the true point (the center of the circle).

After we use the transformation. This gives us the three parameters of the circle. Use these to make a image cut as shown in the figure. We can do this with the following commands.

cropped = image[A, B]

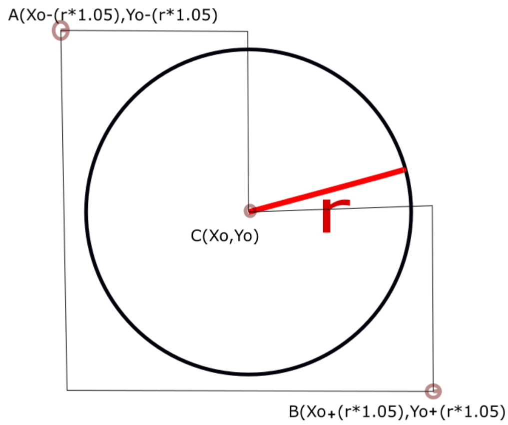

Figure 5. Image cutout using Hough transformation

By the 5% increase in the specified "r" value, we determined the size of the area to crop it to contain the circle line.

C. Thresholding

Image Segmentation is one of the most important basic problem of image processing, which deals with the grouping of pixels with similar properties in homogeneous areas. Many images processing tasks (such as shape recognition, surface descriptions, stereo vision, etc.) originate from such processed segmented images. Our choice of segmentation methods was "Thresholding".

The „Thresholding"

- One of the easiest segmentation methods.
- The best example is to split the portion of the image where we have valuable pixels. We need to modify the pixel values of the image to clearly distinguish the object and its background
- First, we separate the values from each other, and we can also set the pixel intensity if we want. For example, the most common case is to change the valuable pixels to white (255) and the background to black (0), but the threshold value and the pixel value can also be changed arbitrarily.

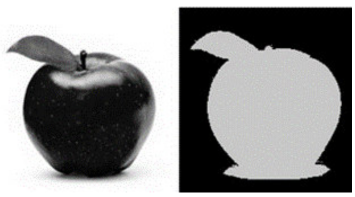

Figure 6. Segmenting the apple

D. Image moment

The "Image Moment" is given to determine the centroid of the shape with the following relation:

$$\{\bar{x}, \bar{y}\} = \{M_{10} \,/\, M_{00}, M_{01} \,/\, M_{00}\}$$

This formula can calculate the centroid of any shape, even in the case of a convex shape.

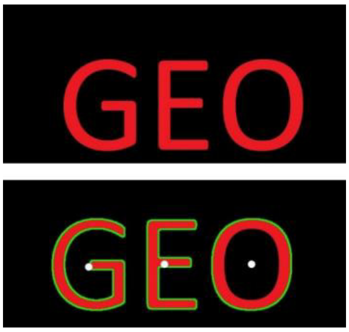

Figure 7. Search for "GEO" inscription centroid

E. Tilt test

The next step was to decide whether the object plane and the image plane deviates from the parallel position by more than 5 °. If this deviation does not exceed 5 °, then the circular regression may still work. If this value is exceeded, then we wont use it. In this case, the method of determining the inclination angle is used to determine the centroid of the mark, which also provides the point. The circle is distorted to an ellipse by varying degrees depending on the inclination of the planes. The semimajor axis of the ellipse is equal to the radius of the circle. The task was to pinpoint ellipse. We use contour detection. We applied the following relationship.

$$inclination = (\cos^{-1}\left(\frac{r}{R}\right))/\frac{\pi}{180}$$

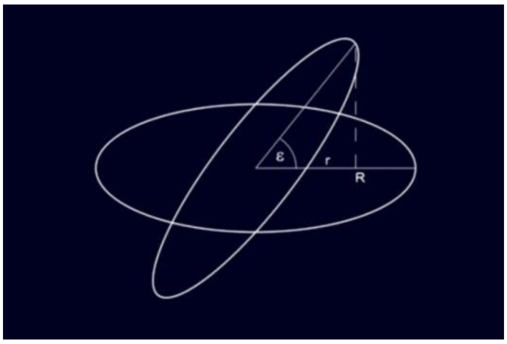

Figure 8. Distortion of the circle

F. Adjustment computation

The regression calculation deals with the alignment of simple shapes contour coordinate points. Only points with final coordinates can be included in the regression calculation. In practice, we use the least squares method.

The circular regression method can calculate the best fitting circle parameters sought from the points measured on the circle line. This requires at least three points along the arc. If we add more to the calculation, we will get the center and radius of the most fitted circle, and even their accuracy parameters.

G. Point detection and Flow chart

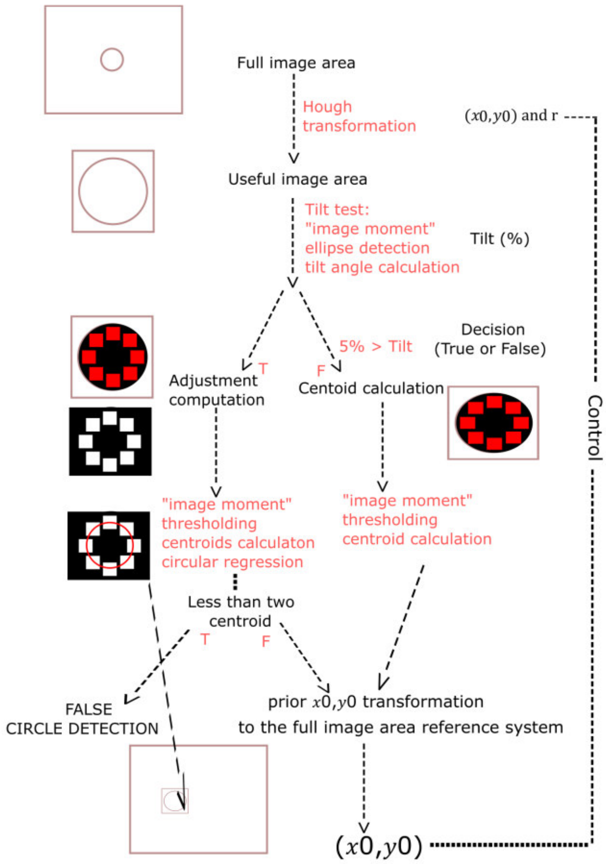

Figure 9. Flow chart

Figure 9 shows the flow chart of the point detection procedure. First, we used the Hough transformation to look for the parts of image on which we could run our algorithm. In order to decide whether or not the slope is over 5 °, the following operations are performed.

- Separate the background and the mark with Binary Thresholding. We created a image that consists of a total of two pixels of intensity. (Black & White Image)
- We inserted an ellipse into the distorted circle with image moment. This is used to determine the angle which between the object plane and image plane is different from the parallel.
- If this angle was greater than 5 ° (the circle was clearly ellipse distorted), the center of the ellipse was considered the center of the mark.
- If the angle value was less than 5 °, then calculate the centroid of the shapes in the circle (triangle, rectangle, etc.). Preparing the program in case any of the shapes in the circle are covered, we have separated and excluded shapes whose area differs by more than 10% from the maximum area shape. If only one or two shapes remain in the circle under the condition, then the signal can not be detected. The program goes to the next mark. In case of 3 or more shapes, a calculated regression circle

has been fitted to the calculated centroids. This will be the exact center of the mark.

In the final step, the defined image coordinates are transcribed in the reference system of the full image area.

The calculation of the difference between the center of the circle defined by the Hough transformation in the full image area and the center of the mark returned from the image offset gives the opportunity for verification.

III. Discussion

The Table 1. Shows our result under laboratory conditions (the center coordinates are 500,500). Unfortunately, we was not able to test field conditions, so we can not report the results of this. All in all, we wanted to create a method that reduces or completely eliminates errors we have detected. In our opinion, we found solutions to these problems, so we could effectively close our research.

Table 1. Results were made under different procedures, under laboratory conditions

Process	**X**	**Y**	**ΔX [pixel]**	**ΔY [pixel]**
Hough transformation	504,600	502,200	-4,600	-2,200
Centroid calculation	500,497	500,491	-0,497	-0,491
Equalizer solution	500,000	499,999	0,000	0,001

According to the promising results under laboratory conditions, the next step is obviously to test the methodology for field conditions, which may be a major project in the near future.

References

[1] M. K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962

[2] J Richard O. Duda and Peter E. Hart (1971): Use of the Hough Transformation to Detect Lines and Curves in Pictures. Artificial Intelligence Center. SRI International.

[3] Duda, R. O. – Hart, E. P. (1972): Use of the Hough Transformation to detect lines and curves in images. Graphics and Image Processing 3, pp. 11-15.

[4] H.K. Yuen, J. Princen, J. Illingworth and J. Kittler (1990): A COMPARATIVE STUDY OF HOUGH TRANSFORM METHODS FOR CIRCLE FINDING, Department of Electronics and Electrical Engineering, University of Surrey, Guildford, GU2 5XH. U.K

[5] Andrew W. Fitzgibbon, R.B.Fisher. A Buyer's Guide to Conic Fitting. Proc.5th British Machine Vision Conference, Birmingham, pp. 513-522, 1995. The technique used for ellipse fitting is the first one described in this summary paper

[6] Intel Corporation (2002): OpenCV Reference Manual

[7] Tóth Zoltán, Barsi Árpád, Analyzing road junctions by geometric transformations, INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY AND REMOTE SENSING (2002-) XXXVI:(I/W3) pp. 1-4. (2005)

[8] Bradski, Gary; Kaehler, Adrian (2008). Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc. p. 6.

[9] Kató Zoltán (2014): Alakzatok és minták detektálása, Digitális Képfeldolgozás.

[10] Epresi Konrád, 2017, Pontjelek automatikus felismerési lehetőségei a fotogrammetriában. Tudományos Diákköri Dolgozat, Székesfehérvár.

[11] K Epresi, B. Gáti, Z. Tóth, Active contours in geoinformatics,In: Orosz Gábor Tamás (szerk.), AIS 2017 - 12th International Symposium on Applied Informatics and Related Areas organized in the frame of Hungarian Science Festival 2017 by Óbuda University: Proceedings, November 9, 2017 Székesfehérvár, Hungary. 204 p. Konferencia helye, ideje: Székesfehérvár, Magyarország, 2017.11.09 Székesfehérvár: Óbudai Egyetem, 2017. pp. 187-190.