AUTOMATIC ANALYSIS THE BRAIDING ANGLE OF THE BRAIDED FABRICS USING IMAGE PROCESSING

Yisheng Liu[1] and Yordan Kyosev[2]

[1]*Zhejiang Sci-Tech University, Hangzhou, The Second Street 298, Xiasha, 310018, China*
[2]*Hochschule Niederrhein University of Applied Sciences, Webschulstrasse 31, D - 41065 Mönchengladbach, Germany*
Lysleo@zstu.edu.cn

Abstract: *The braiding angle is one of the most important parameters of the braided structures, because it influences their mechanical properties. For this reason, any way of automatic inspection of the braiding angle can simplify the quality control of such products. Image analysis is one of the feasible non-contact measurement methods to obtain braiding angle. In this paper is presented a new program for image processing and analysis of the braiding angle. Nine different braided structures are analyzed manually and with the help of the program. The comparison of the results demonstrates that in the most cases the created algorithm produces accurate results but in few situations it analyses different geometrical properties in the image than those, analyzed by humans.*

Keywords: *Braiding angle, automatic detection, Image analysis, Image processing, Python, Hough line detection*

1 INTRODUCTION

Braided products are not so popular like the woven and knitted but are used in large number of applications. In each car are integrated more than 1500 meter of braided products [1]. Large part of the ropes for industrial and marine application are braided and the climbing ropes are as well braided products [2]. In one braided structure there are direct relations between the elasticity, strength, bending stiffness and other parameters and the braiding angle [3, 4]. For this reason its automatical control is important step in the ensuring the quality of the braided products. This paper presents the development of algorithm for automated analysis of the braiding angle based on the Python libraries.

2 APPLICATION OF THE IMAGE PROCESSING

Python is a high-level language with an open source, high-quality, peer-reviewed code, written by an active community of volunteers [5]. Now Python become one of the most popular program languages in the world. For the image analysis and angle identification, several modules such as Numpy, Scipy, Scikit-image and matplotlib were used and all of these can be found on the web.

Scikit-image [6], Image processing in Python, is a collection of algorithms for image processing. For braiding angle identification, it provides a set of functions for image loading, visualization, manipulation, and analysis. The flow chart of the executive program is shown in Figure 1 and it includes several steps from loading image to calculation of the braiding angle. The interface of the executive program is shown in Figure.2.

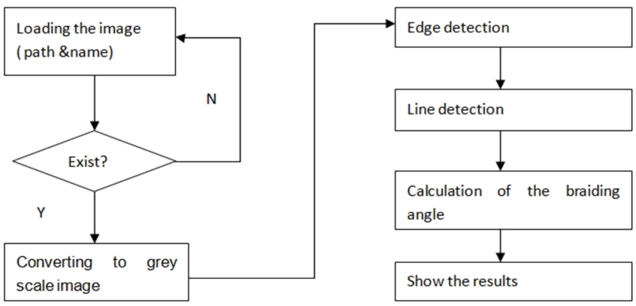


Figure 1 The flow chart of the executive program

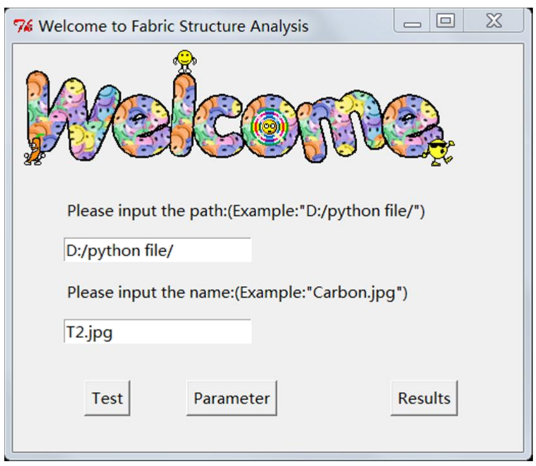


Figure 2 Interface of the executive program

2.1 Loading image

The first step for the analysis is loading image into the executive program. The following code:

```
img2=io.imread(imgpath)
```

opens file with ‘imgpath’, which can be in different format as jpeg, bmp, pngor other and store the information in the variable img2. This variable is now an array, in the current case with size 1077 x 817 x 3 unsigned integers, which means that there are three arrays with the red, green and blue values of each of the pixels.

2.2 Converting to greyscale image

In Scikit-image processing, the angle detection and several image processing functions work only with greyscale images, therefore it is necessary to convert the true color image to a greyscale image. In the example, the true color image is represented as an unsigned integers array, it need to be transferred into floats ones and then to a greyscale image by using the function ‘img_as_float()’ and ‘rgb2gray()’. The result of this conversion is presented in Figure 3 (right).

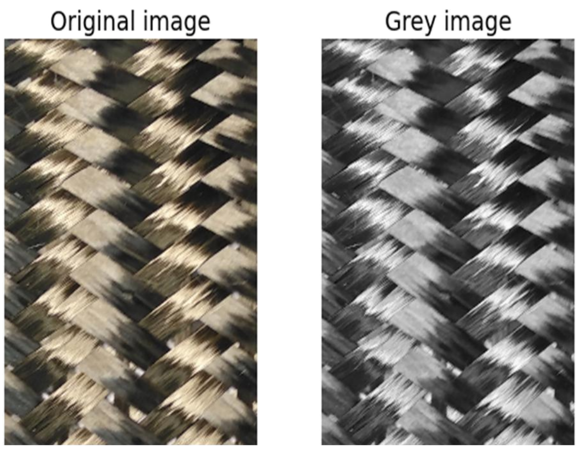


Figure 3 Image of T2.jpg – left) original true color image, right) after conversion to greyscale

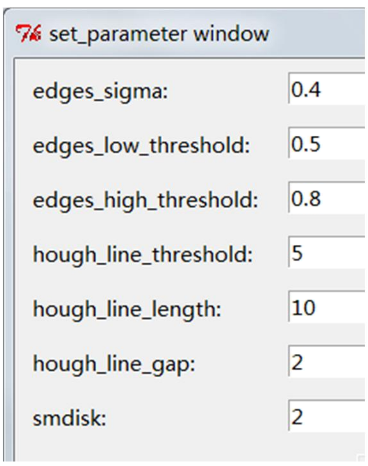


Figure 4 Parameters for the edge detection

2.3 Edge detection

For the image processing several parameters of the algorithms have to be set. As the different samples require different values, a small user interface (Figure 4) was prepared, in order different configurations to be tested quickly. The seven parameters of the image analysis procedure has to be entered, one for morphology, three for edges detection and three for hough line detection. After some experience it was found, that the edge detection works better if the image is „homogenized” – some small bright spots are removed and the small dark cracks are connected. This is done by the function ‘Opening’ of the Skiimage toolbox.

To find edges in the greyscale image, the ‘canny’ function can be used [7]. This function finds places in the image where the intensity changes rapidly. In the image those places could be the edges of the threats within the braid. The most useful edge-detection method is the ‘canny’ method. This method differs from the other edge-detection methods in that it detects strong and weak edges. As visible in Figure 5 (left), the main directions of the yarns can be recognized, The main commands are:

```
import skimage.morphology as sm
from skimage import feature
image = sm.opening(image1,sm.disk(int(smdisk)))
edges = feature.canny(image, sigma=float(pes),
       low_threshold=float(pelt),
       high_threshold=float(pelh))
```

2.4 Hough line detection

To get the braiding angle it is necessary to detect at least one straight line at an edge of a thread. To detect those lines the ‘Hough’ transformation is applied [8]. The Hough transformation creates a dual space where all possible parameters of the to-find point are entered for each point in the image that is located on an edge. After setting the threshold, line length and line gap, The ‘Hough_line’ function will detect the edge line and get a line list, in which the star point and end point of each line is stored as a tuple. As visible in Table 1, the main directions of the edge lines can be recognized. The main command is:

```
import skimage.transform as st
lines = st.probabilistic_hough_line(edges,
        threshold=int(pht),line_length=int
        (phl), line_gap=int(pha))
```

where „edges” is the variable with the result from the application of the canny algorithm over the figure.

3 RESULTS AND DISCUSSION

The braiding angle is the angle between the product axis and the fibers. Assuming that the image was rotate before the analysis so, that the vertical direction corresponds to the y-axis, the braiding angle can be regarded as the angle between the edge line of yarn and y-axis. Since the start point and end point is detected, the angles of all detected lines can be calculated with the equation

$$\alpha = 90° - abs\left(arctan\frac{\Delta y}{\Delta x}\right) \tag{1}$$

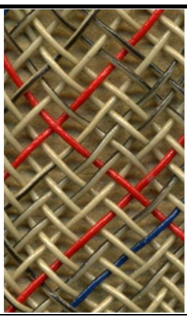
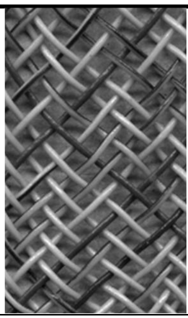
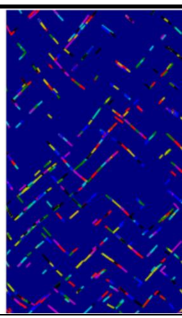
This angle is calculated for each detected line and the histogram of all values is plotted (Table 2). The maximum of the histogram is taken as a braiding angle.

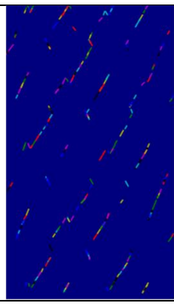
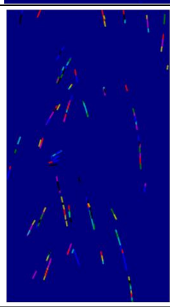
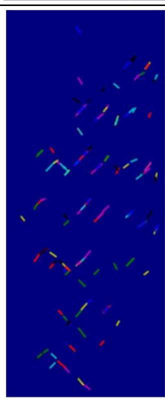
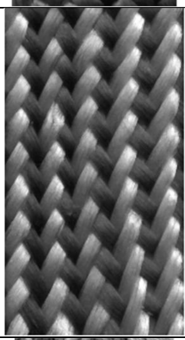
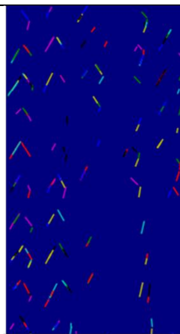
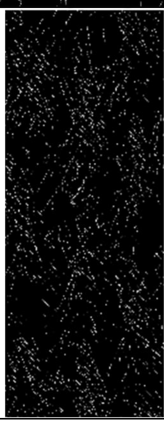
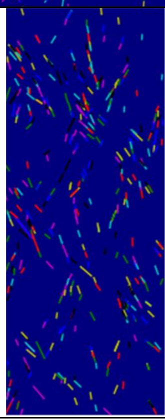
In the current case, nine different samples were selected for testing the algorithm. The samples was selected so, that they cover different application areas and different structures - to have a monofilament braid, ropes and multifilament structures for composites. As demonstrated (Table 2), there are several detected lines from the images, which are not parallel to the fibers. These lines are coming from shadows, different colour appearance or as well the vertical boundaries of image. But the most of the two largest bars shows the actual braiding angle, as the most of the fibers in the braid are detected as lines, oriented at plus or minus the braiding angle degree. The most difficult step for the application of the algorithm was the detection of the parameters of the edge detection algorithms. During the preparation of the manuscript was found that the algorithm is sensible against the parameters in the image detection and these have to be re-identified with several trials and errors for each new sample. The braiding angle was measured as well manually using the software ImageJ on the same images. The result of the five measurements as arithmetic mean value and the variation coefficients are given in the Table 1. Automated detected angle is presented there too, together with the relative error in % between both measurements.

As it can be seen (Table 1 and Figure 5), in the most cases (1-6) the automatic algorithm determine the braiding angle with less than 10% error. The last three samples have errors up to 20%. The histograms and the images of some typical samples (Table 2) can give explanation of these larger errors. The sample 1 in Table 2 is a structure with monofilmanent wires. The edges, extracted from this image, correspond to the yarns and the accuracy at this image is very high – it produced 0.5% error. Sample 3 is a braid produced from slightly twisted yarns. The algorithm detects there a lot of the single filaments which have different angle than the yarn itself. After playing with the edge detection parameters, still was possible to get the main edges of the yarns and have accurate result with 4.7% error, in the same way as for Figure 6 (rope of twisted linen material). In both cases, the yarns are still recognizable as objects and produce better visible edges than the single fibers or filaments. Completely different is the behaviour at the braids for composites. In such case the rovings consist of thousands of parallel filaments, which are recognized as lines. These filaments are not always exactly parallel to the roving axis and the edges of the roving are optically not easy to be detected.

Because of the large number of the single lines of the filaments, there is one very clear maximum in the histogram and all other detected lines remain with significantly lower appearance, almost not visible there. So in this case, the automatically detected angle differs significantly from the manually detected one, because the operator is able to identify the roving main line and the algorithm detect the orientation at the finer – filament level.

Table 1 Tested samples and manual and computed results

No.	Original Image	Grey Scale	Canny Edges	Probalistic Hough	Measured (Mean N=5) [°]	$V=\frac{\sigma}{\bar{x}}$ [%]	Detected [°]	Rel. Error [%]
1					45.1	1.80	44.9	-0.5

2					28.6	1.7	26.5	-7.3
3					19.3	4.6	18.4	-4.7
4					44.4	2.1	44.5	0.2
5					27.5	4.2	26.5	-3.9
6					25.4	3.5	26.6	4.4

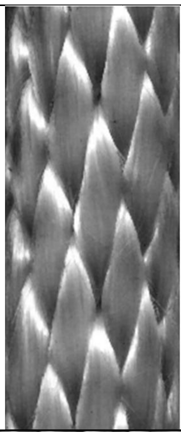
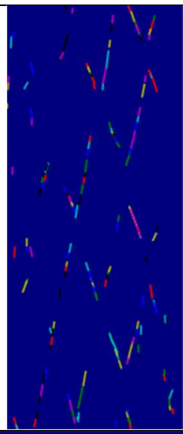
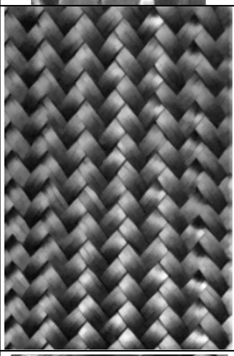
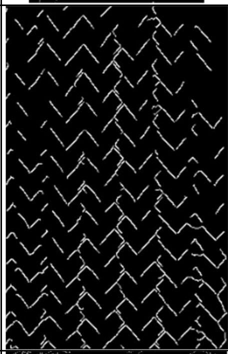
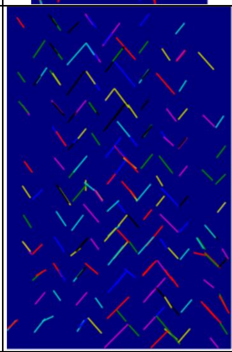
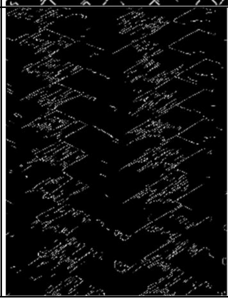
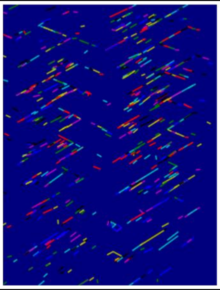
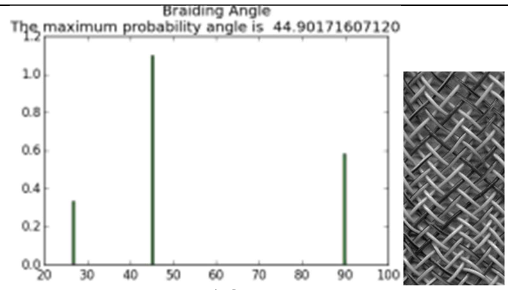
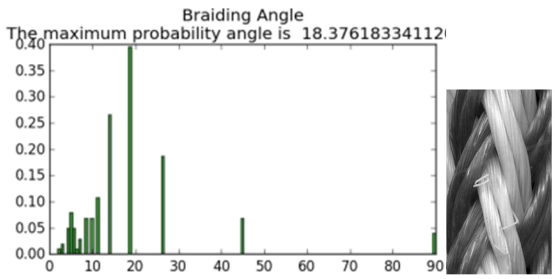
7					15.6	3.9	18.0	15.0
8					40.7	3.8	44.9	10.1
9					56.4	2.1	44.7	-20.7

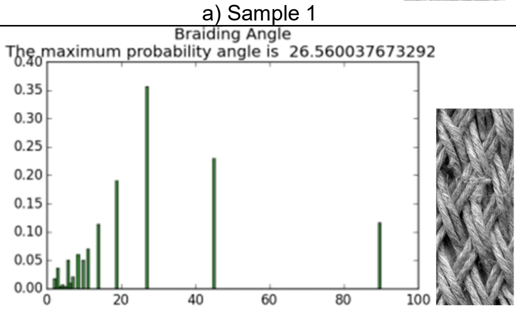
Table 2 Automatically detected braiding angles for some samples



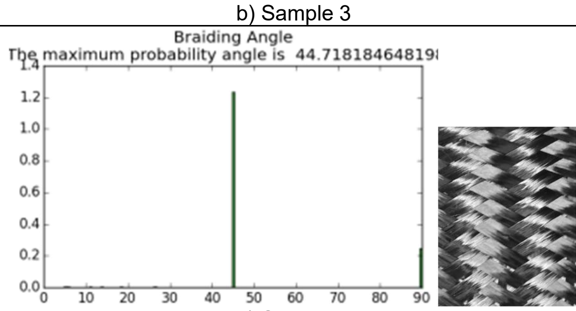
a) Sample 1



b) Sample 3



c) Sample 6



d) Sample 9

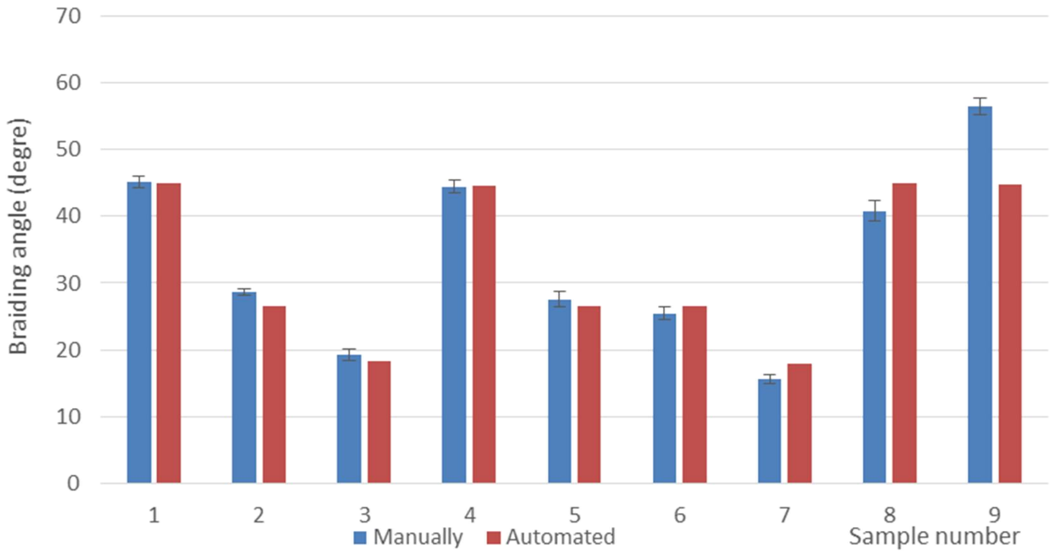


Figure 5 Comparison between the manually and automated estimated braiding angles

4 CONCLUSION

The automated program for image processing based on Python language was developed and applied for automatic detection of the braiding angle of braided products. It can be used for fast quality control of braided product and allows very good accuracy for the braid, where the yarns can be recognized optically. The algorithm can be applied for structures for composites, too, but in this case it detects the fiber orientation and not always the tow orientation. The main task for using such algorithm in the practical application seems to be the detection of the parameters of the image processing functions. Their values depend on the contrast and the brightness of the image.

5 REFERENCES

1. August Herzog, Product Catalogue, Oldenburg, 2013
2. Kyosev Y. (Ed.): Advanced in the braiding technology, Specialized Techniques and Applications, Elsevier, 2016, eBook ISBN: 9780081004265, Hardcover ISBN: 9780081009260
3. Brookstein D.S., Tsiang T.-H.: Load-deformation behavior of composite cylinders with integrally-formed braided and with machined circular holes, Journal of Composite Materials 19(5), 1985, pp. 476-487, https://doi.org/10.1177/002199838501900506
4. Kyosev Y.K.: Braiding technology for textiles: Principles, design and processes, Woodhead Publishing Series in Textiles No. 158, 1st ed, 2014, eBook ISBN: 9780857099211, Hardcover ISBN: 9780857091352, Paperback ISBN: 9780081013298
5. Python, www.python.org (2018)
6. Scikits Toolbox, https://www.scipy.org/scikits.html
7. Canny J.: A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8(6), 1986, pp. 679-698, doi: 10.1109/TPAMI.1986.4767851
8. Hough P.: Method and means for recognizing complex patterns, U.S. Patent 3069654, 18 December 1962