An Application Specific Image Transformation and Stitching Algorithm Using ORB Features

Sebastian Bielmeier
Faculty Electrical Engineering and Media Technology
Deggendorf Institute of Technology
Deggendorf, Germany
Email: sebastian.bielmeier@stud.th-deg.de

***Abstract*—In this work an image transformation and stitching algorithm has been developed to contribute to a solution for a automatic optical inspection problem. In general the task is to transform and stitch four images of the jacket surface of cylindrical parts to get it as just one image. First, the algorithm unrolls the images of the jacket surface onto a plane mathematically. Afterwards, the images are getting overlaid according to previously matched ORB features. Thus, the result is an image of the whole jacket surface of the cylindrical part without applying any mechanical movement. The source code related to this work can be found at https://github.com/BielSeb/ARC2022.**

I. Introduction

A. Scope of application

This work is a contribution to the research and development work of the company Nohau Industrie Elektronik Produktionssysteme GmbH, which is located in Regensburg, Bavaria. Nohau is a manufacturer of special automation machines, among others for testing and packaging of so-called fine fuses. Meant by the term fine fuses are melting fuses that are at minimum design composed by a glass tube, a melting wire and two metal caps. They are standardized in DIN EN 60127 [1] [2]. In detail, it is about the fuses with a diameter of 5 mm up to 6.3 mm and a length of 20 mm up to 32 mm. According to the standard, the fuses have to be labeled with rated voltage, rated current, manufacturer name or sign and a symbol for time-current characteristics. This labeling is usually placed on the jacket surface of the fuse caps. The last remaining task of the fine fuse testing machines is to prove that the fuses are labeled correctly. The designated testing capacity is at least three fuses per second.

B. Mechanical concept

To understand the need for the algorithm in this work, the underlying mechanical concept must be briefly clarified. A study about appropriate sensor technologies and part handling kinematics in project work 1 has shown that the mechanical concept which is shown in figure 1 has the most advantages for implementation in a testing machine [3]. In this prioritized concept are four 90 degrees displaced 2D cameras radially pointing to the fuses. Thus, the whole cap jacket surface can be observed without any mechanical motion. Finally, to judge whether the labeling is correct, it is necessary to check the characters and symbols on the cap jacket surface. This shall be done with some kind of optical character recognition. Since the rotational position of the fuse is random, the goal is to stitch all four images together to get the whole jacket surface in just one image. To approach this task fast enough, a specially designed image transformation and stitching algorithm is described within this work.

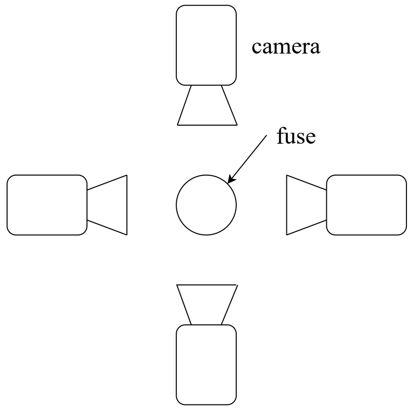

Figure 1. Sketch of the basic mechanical concept.

II. Sample image generation

To develop the algorithm a test setup for generating sample images is needed. Firstly, there is just one 2MP camera in combination with a rotatable fuse clamping device in order to

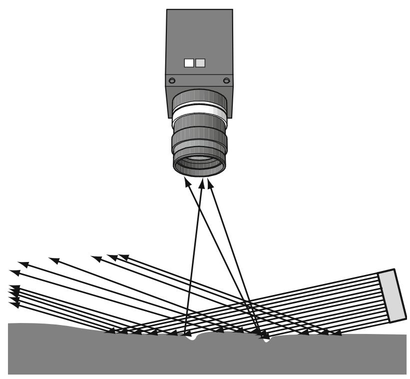

Figure 2. Working principle of dark-field illumination, source: [4].

keep costs low. The illumination of the fuse caps is a very critical part at image generation. Due to the highly reflecting surface and the mechanically stamped symbols and characters a dark-field illumination is necessary to get a sufficient contrast between background and foreground. Figure 2 shows the working principle of dark-field illumination. As practical implementation the use of normal slanted ring lights leads to good results. Figure 3 shows the final test setup to generate the needed sample images. Finally, figure 4 shows a sample

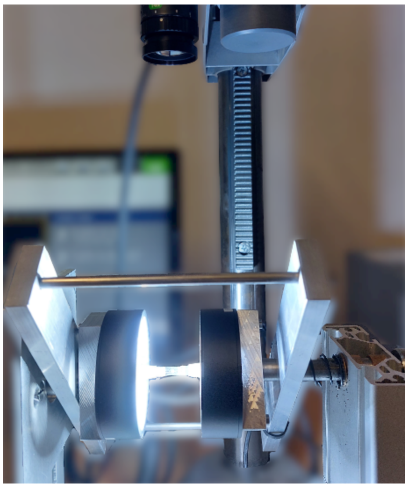

Figure 3. Test setup with rotatable clamping device and ring lights.

image that has been generated with the described test setup. It shows a very good manufactured fuse, but there is a huge difference in quality over all fine fuse manufacturers.

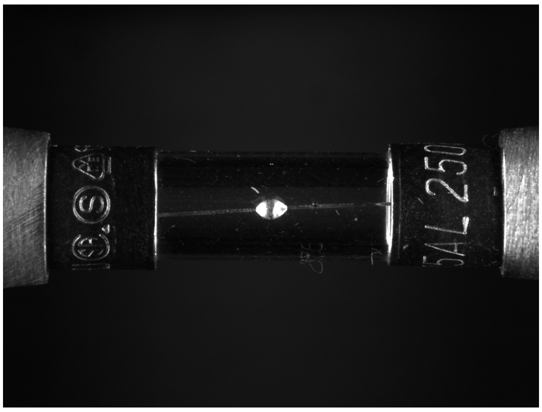

Figure 4. Sample image.

III. Image Projection

The critical part of the chosen concept is the stitching part. It has to work reliably for enabling the desired downstream process steps like reading the characters and symbols and judge the fuse caps. Before stitching is possible, the cap jacket surfaces have to be projected onto a plane. All other steps like cropping the fuse caps are less critical and so far excluded. Figure 5 shows the image processing data flow as it has been planned initially [5].

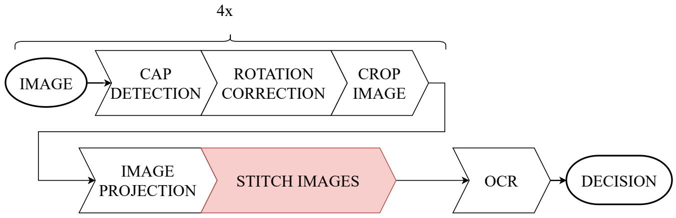

Figure 5. Designated image processing data flow.

A. Camera model

One assumption regarding the camera model has to be made for the image projection that is explained below. In the test setup a standard C-mount lens with a fixed focal length of 35 mm is mounted with some distance rings. This is a very low cost variant and leads to an approximate kind of macro lens behavior. However, for the imagined image transformation a parallel beam path is necessary. This could be achieved by the usage of a telecentric lens, but due to the present dimensions is the error of the current setup negligible. Therefore, a parallel beam path from fuse to camera sensor is assumed.

B. Shifting pixels

The upper assumption allows a comparably simple image projection. As the fuse cap's intersection is a circle, the cap jacket surface is in radial direction distorted along a semicircle in the images. Thus, some kind of unrolling is necessary to get a flat cap jacket surface. This kind of image transformation can be implemented by shifting and interpolating pixels using OpenCV's remap()-function [6], which works as follows. First, in case of gray scale images, one need to create two two-dimensional arrays with the size of the desired target image. One each for row and column coordinate of the source pixel value of the respective target pixel. Afterwards every element in both arrays has to be filled with the respective source pixel coordinates, which also can be floating point values. The function is able to interpolate these.

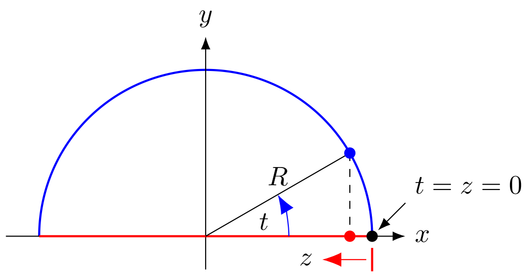

Figure 6. Coordinate transformation to get source positions of pixel values.

To calculate the target image size and the necessary source image pixel values some mathematical basics are required. Figure 6 shows the necessary coordinate transformation for implementing the projection. As already mentioned is the objective to describe the source image coordinate with respect to the target image coordinate. The following approach would also work for other geometry if the initial camera model assumption is adequate. In this special case is the coordinate z along the red line with respect to the length of the blue arc with parameter t wanted. A semicircle can be described by the following parametric equation.

$$\alpha(t) = R \cdot \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix} \quad \text{with } t \in [0; \pi] \text{ and R: radius} \tag{1}$$

To determine the length of a parametric curve, the following relationship applies [7].

$$L(\alpha(t)) = \int_a^b \|\alpha'(t)\| \, dt \tag{2}$$

After some computation the length of the curve will be found as follows.

$$L(\alpha(t)) = R \cdot \int_0^\pi \sqrt{-\sin(t)^2 + \cos(t)^2} \, dt = R \cdot \pi \tag{3}$$

Or as a general expression as follows.

$$L(\alpha(t)) = \int_0^t \|\alpha'(t)\| \, dt = R * t \tag{4}$$

This general relationship of the length of the semicircle L and the parameter t is the first part needed for the transformation. The second part is the introduction of the coordinate z with respect to the parameter t as follows.

$$z(t) = R - R * \cos(t) \tag{5}$$

Finally, the necessary transformation function can be derived from the equations (4) and (5) by formulating the following expression.

$$z(L) = r - r * \cos\left(\frac{L}{R}\right) \tag{6}$$

Figure 7 shows the result of the transformation applied to the right cap in the sample image of figure 4. The raw

Figure 7. Cropped fuse cap before and after transformation.

transformation result shows clearly that the left and right edges are blurred. This is due to the transformation function. The red curve of the plot in figure 8 illustrates the reason for that blur effect. Thus, the image is getting cropped at the left and right

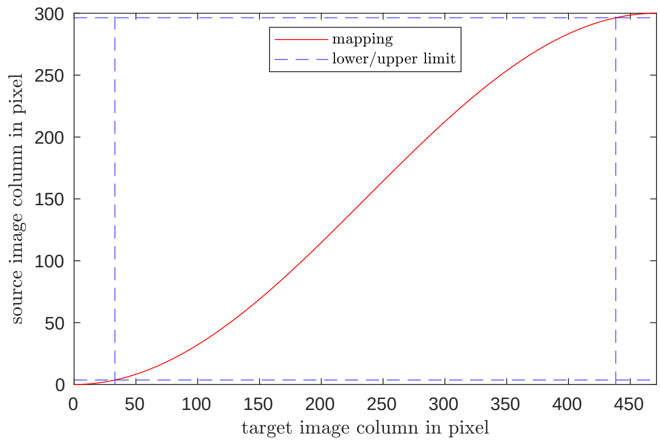

Figure 8. Plotted coordinate transformation function.

edges to ignore the blurry part. The dashed blue lines show that there is almost no loss of information if the first and last 7% are cut away. Finally, the transformation result of the sample image after cropping is shown in figure 9.

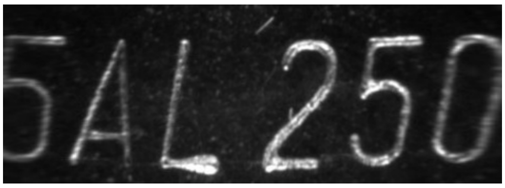

Figure 9. Finished transformation sample.

IV. Image Stitching

The second part of the algorithm is the so-called image stitching. The objective is to overlay all images such that the whole cap jacket surface is pictured in only one image. Image stitching is a very usual application field in image processing, but the algorithms are often computationally extensive. They mostly differ in their number of degrees of freedom (dof). In case the rotation was already corrected before transformation, the necessary number of dof can be reduced to two for this application, namely column and row coordinate. Thus, the task is to get a vector with two elements containing the shift coordinates and apply an appropriate two dimensional transformation.

A. *ORB features*

The approach to determine the desired shift vector is to use feature matching. In this case features are certain key points in images that are described by their location and some information about them and their environment. They are getting detected by special algorithms. In this case Oriented FAST and Rotated BRIEF (ORB) is used for the key point detection and feature extraction. It was published by Rublee et al. in 2011 and implemented in OpenCV with free commercial use licensing [8]. The computed features are then matched

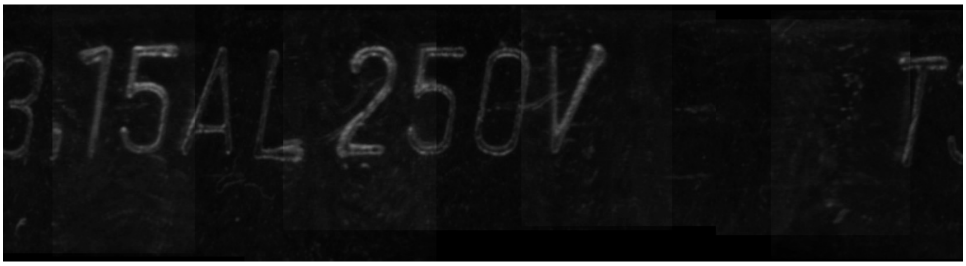

Figure 10. Final stitching result.

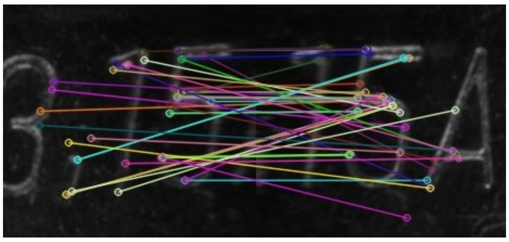

Figure 11. Sample of matched features.

by a so called brute-force matching algorithm, that computes the Hamming distance between all possible combinations and matches the best ones. As the overlapping area is known by the construction, the images are divided centrally into a left and a right part. The right part of the left image has to match the left part of the right image. Figure 11 visualizes a sample of matched features.

B. Shift vector determination

The detected and matched ORB features form the basis for determining the shift vector coordinates. As one can see in figure 11, there are lots of crisscross matches that are clearly wrong, but also a certain share of parallel matches with similar length. Thus, the approach to find the right shift vector is just considering these similar matches. To exclude the outliers a deeper analysis of the matched feature key points coordinates is required. Figure 12 shows a deeper insight to the coordinate difference of the matched key points in four subplots. The lower left plot contains the location difference of the matches calculated using the local coordinates of the respective key point. The upper right plot shows all matches with their difference of coordinates in a scatter plot as blue dots. The last two plots show the distribution of difference of the respective coordinates in a histogram with 25 bins having equally distributed edges. Since there is sufficient quality of the matched features a high density of dots can be found in the upper right scatter plot. This ultimately is the desired shift vector that is necessary to stitch the images. The approach to detect it automatically is to find the highest bin in the histogram and compute a weighted mean with its directly adjacent bins. In figure 12 the result is marked by the green lines and the red circle. Figure 10 shows a sample of the final stitching result applying the described algorithm. The computation time is about 70 ms and therefore fast enough.

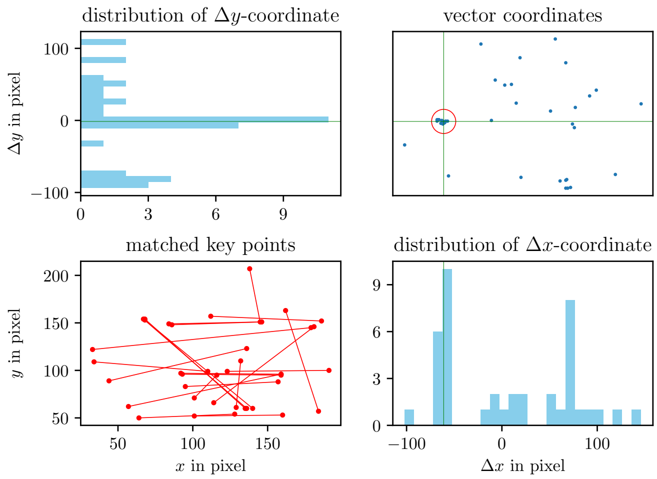

Figure 12. Analysis of matched features sample.

V. Conclusion

The research has shown that it is possible to stitch the images fast enough by the developed algorithm, but the quality of the results is at last highly depending on the quality of the detected features. For the practical usage of the algorithm in a fine fuse testing machine further improvements regarding reliability are necessary, but the results are promising so far.

References

[1] *Geräteschutzsicherungen - Teil 1: Begriffe für die Geräteschutzsicherungen und allgemeine Anforderungen an G-Sicherungseinsätze*, DIN Std. EN 60 127-1.

[2] *Geräteschutzsicherungen - Teil 2: Feinsicherungseinsätze*, DIN Std. EN 60 127-2.

[3] S. Bielmeier, "Conceptual design for automated cap detection of fine-wire fuses," project 1, Deggendorf Institute of Technology, 2021.

[4] C. Demant, B. Streicher-Abel, and A. Springhoff, *Industrielle Bildverarbeitung: wie optische Qualitätskontrolle wirklich funktioniert*, 3rd ed. Springer, 2011.

[5] S. Bielmeier, "Testing an automated fine fuse cap detection concept and developing an application specific image transformation and stitching algorithm," project 2, Deggendorf Institute of Technology, 2022.

[6] "Dokumentation OpenCV." [Online]. Available: https://docs.opencv.org/

[7] Wikipedia contributors, "Arc length — wikipedia, the free encyclopedia." [Online]. Available: https://en.wikipedia.org/w/index.php?title=Arc_length&oldid=1071939246

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2564–2571. [Online]. Available: http://ieeexplore.ieee.org/document/6126544/