ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2019.1.043

SEGMENTATION METHOD OVERVIEW FOR THERMAL IMAGES IN TRAVERSE VEDGE ROLLING PROCESS

Ondrej Bostik✉, Sobeslav Valach, Karel Horak, Jan Klecka

Department of Control and Instrumentation, Brno University of Technology, Czech Republic
bostik@feec.vutbr.cz✉

Abstract

This paper presents an overview of methods usable for segmentation of thermal images in MATLAB computational environment. The goal of this work is the demonstration usage of available methods and evaluate their performance. Part of the work is to present the datasets we create for testing. This paper is part of our ongoing work focused on segmentation of thermal images from the process of traverse wedge rolling.

Keywords: *MATLAB, segmentation, thermal images, dataset, Otsu's segmentation, adaptive thresholding, k-means clustering, active contour.*

Received: 15 April 2019
Accepted: 31 May 2019
Published: 24 June 2019

1 Introduction

Accurate dimension measurement and quality assessment are key factors in the automotive industry. The goal is to make accurate measurement during the process and sort precise parts from the defective parts. The process of traverse vedge rolling is no exception. The challenge in this forming technology is in the high temperatures. Every formed part must cool down before the measurements can be done. If we can take advantage of these high temperatures and make preliminary measurements during the forming process, we can improve the whole process.

The traverse vedge rolling is a preforming process that produces rotationally symmetrical workpieces with unequal mass distribution. During the forming, rotary wedge-shaped tools are moving in the opposite direction and redistribute material along its axis [2]. For further details, see [6, 2, 10].

Thermal images datasets were created as part of our work to make dimension measurement and quality assessment of transverse wedge rolling in the production of automotive parts. The first goal is to observe the production and sort out bad parts. The second goal is to make exact measurements of parts and store them for later use. The third goal is to help workers with prototyping new forming rollers for a better understanding of the process.

The initial stage of the project is to choose a reliable technique for component segmentation. This process was divided into several parts. Firstly we need to create the datasets of thermal images with and without object from forming process. In the second stage, we need human participants to segment regions with an object. And lastly, we need to evaluate and choose from known techniques for the best segmentation.

2 Basic Segmentation Algorithms Overview

In this section, we theoretically introduce used segmentation algorithms. The methods were chosen from several categories by their relevance for this case and from methods available in Matlab computational environment.

Segmentation algorithms can be categorized into these basics categories:

- Threshold methods - simplest methods based on dividing image pixels with respect to their intensity level. The commonly used methods are global and adaptive thresholding.
- Edge-based methods - based on detected edges (rapid changes in image intensity), the segmented areas are formed as regions within these borders.
- Region-based methods - segmentation is driven by certain characteristics, which divide an image into several areas. The main methods are Region growing methods and Region splitting and merging methods.
- Clustering-based methods
- Watershed-based method - methods based on the concept of topological interpretation
- Partial differential equation based methods
- ANN based methods - methods based on using artificial neural networks

More information about the categories can be found in [7].

2.1 Otsu's segmentation algorithm

The first method used in this experiment is Otsu's segmentation method introduced in [9]. This method could be considered as outdated, but the simplicity of the method and it's speed made this method ideal for initial testing and for method comparison. The method proposed by Otsu is based on a search of single segmentation threshold searched not in the image itself, but in the histogram of the image.

The criterion is to maximize the separability of the resultant classes in gray levels. It is done by compute intra-class variance for all possible threshold values t via:

$$\sigma_w^2(t) = \sigma_0^2(t) \sum_{i=0}^{t-1} p(i) + \sigma_1^2(t) \sum_{i=t}^{L-1} p(i) \tag{1}$$

where σ_0 and σ_1 are variances of these two classes, $p(i)$ is value of image histogram in the bin i and L is number of bins in histograms. Note that this method is really quick to evaluate, needs no extra parameters to set, but as a single threshold is computed, this method cannot deal with large variance of the object intensities.

2.2 Adaptive Thresholing Segmentation

The second method considered for use is adaptive thresholding. The method introduced in [3] was selected for the experiment. Instead of a single threshold level for the whole image, the method produces a binary image where decision level is set based on running average of a window around the current pixel.

To speed up the whole process, the presented method uses integral image [5] to sum the value of the surrounding pixels.

The method has several parameters. The main parameter is sensitivity, which determines the threshold for this method. Low values lead to thresholding more pixels as background. The second parameter is to determine, which statistics are used for computing the local threshold at each pixel. The third parameter is the size of the neighborhood used to compute local statistics.

2.3 K-Means Clustering Segmentation

Next method used in this experiment is k-means clustering segmentation. This method is based on splitting pixels into the previously known number of clusters. The method starts with randomly positioned centers. Every pixel is then classified based on the lowest Euclidean distance to the centers. For each class, the new center is computed as a mean position of all pixel in the class. Then the process is repeated until the stable positions are reached.

The main parameter for this method is the number of clusters. Other parameters are the maximum number of iterations and accuracy threshold, which stops the method when either the number iterations are reached or either when all the centers of clusters move less than the threshold.

More explanation of the used method is in paper [1].

2.4 Geodesic Distance-Based Segmentation

Geodesic distance-based segmentation is an interactive algorithm for segmentation of natural images. The process needs user input to provide a hint about the approximate location of regions of interest. In the paper [11], the initial approximations of regions are called Scribbles. Series of Gabor filters are automatically computed based on the "scribbles" to automatically discriminate these regions of interest. Computed Gabor filters produce adaptive weights to obtain a segmented image. The paper [11] provide theoretical background alongside with examples of algorithm's segmentation capabilities.

2.5 Active Contour Segmentation

Active contour segmentation, also known as snakes, is the iterative method of shaping and refining contours to actual object borders. An initial rough estimate of the contour is deformed with internal and external image forces. The contours are evolved until internal and external forces equals. The definition of energies, the evolution process, and final state are explained in [4].

Commonly used parameters for active contour segmentation are the number of iteration, used method (the Chan and Verse region based energy model [4] or edge-based energy model), smooth factor for defining the regularity of the boundaries and contraction bias for defining the tendency of the region grow or shrink.

3 Datasets

During the first stage of our project, we record 7 videos of the above-explained process. Every video was taken with ImagingSource monochrome camera type DMK 33UP500 with IR filter. The resulting video has 30 frames per second with a resolution of 1920x1080px and saved to AVI file with Y800 video codec (8 bits grayscale). Every video was converted to a series of images, which was later segmented with human afford. The resulting annotation is a structure containing the file names and sets of points defining a polygonal region of interest. Every dataset is stored in a folder containing all images and Matlab binary data file with annotation. Every processing was done in MATLAB R2018a with Image Processing Toolbox.

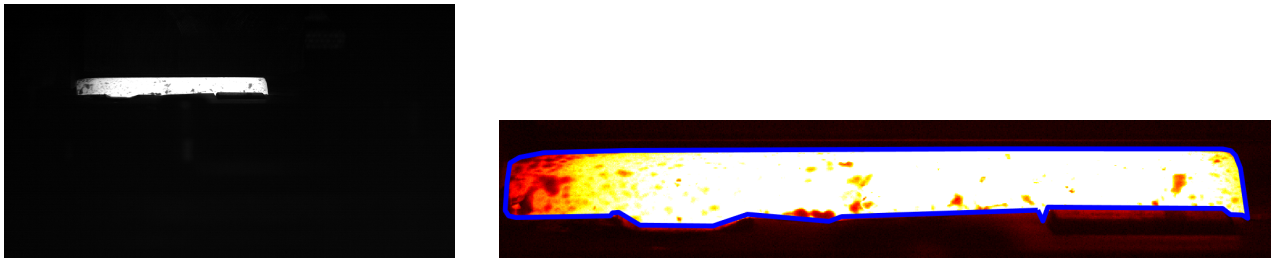
Overview of our datasets is depicted in table 1.

Note that not all resulting images in datasets contain the object. We need to evaluate performance on these images too because in some part of the manufacturing process there is no object in the picture. We need to differentiate between images with object too cold to be recognition out of a thermal image and between image without the object. We are aware of the imbalance between image count with and without object, but this is mainly because of the nature of the previously explained process and we do not want to remove any image from the datasets.

Table 1: Dataset overview

Dataset	Images count	Images with object	Images without object
Dataset A	928	634	294
Dataset B	871	736	135
Dataset C	1183	658	525
Dataset D	1069	448	621
Dataset E	1103	710	393
Dataset F	313	210	103
Dataset G	1100	1009	91
Total	6567	4405	2162

Sample images are shown in Fig. 1 and 2. In the Fig. 1(a), a sample image taken from Dataset A without any processing is shown. The second Fig. 1(b) show detailed view of the image with blue annotated border of target object. This images was colored with pseudo-colors only for visualization purposes.



(a) (b)

Figure 1: Example image from our thermal images dataset A (a) original gray-scale image (b) detail of rolled object colored with pseudo-colors for visualization and a blue border made by annotators

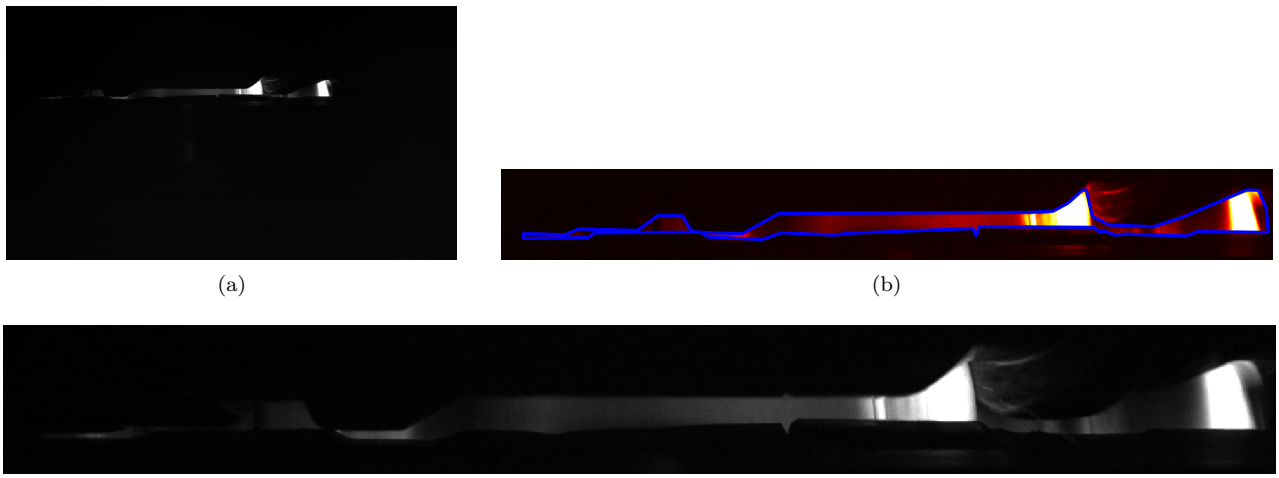
The second example of the single dataset image (Fig. 2) displays one of the challenging images. Original image is shown in Fig. 2(a), the pseudo-colored image with annotated borders is depicted in 2(b). Detailed gray-scale Fig. 2(c) was added to the reader to help better understand the complexity of the segmentation task.

4 Experiments

Before the experiment itself has started, the evaluation framework was prepared. This framework takes all available datasets and compares region created during the annotation with either second region or with a binary image. For every image, we obtain error matrix [**?**] based on classification results. The form of this error matrix with explanation is shown in Fig. 3.

The main parameters we compare during evaluation process are Accuracy, Precision, Recall, and F_1 score.

In binary classification, accuracy (2) can be explained as for how well the classification algorithm identifies the object or foreground. Accuracy is a portion of correctly segmented pixels over the total number of pixels.



(a) (b)

(c)

Figure 2: Example image of one of the challenging images (a) original gray-scale image (b) detail of rolled object colored with pseudo-colors for visualization and a blue border made by annotators (c) enlarged detail of rolled object in gray-scale

		Annotation	
		Object	Foreground
Classification	Object	$TruePositive$ (object segmented as object)	$FalsePositive$ (foreground segmented as object)
	Foreground	$FalseNegative$ (object segmented as foreground)	$TrueNegative$ (foreground segmented as foreground)

Figure 3: Error matrix for classification results evaluation

As the area of the segmented object is the only small portion of the whole image, this number can be very high even the object is not properly segmented.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \tag{2}$$

Precision is the ratio of correctly predicted object pixels to all object pixels classified by the algorithm. Precision can be quantified via:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3}$$

Recall, also known as a probability of detection or sensitivity, is defined in equation (4) as a portion of correctly detected pixels with a presumed object to all pixels with an object from the annotation.

$$Precision = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4}$$

The last parameter we compare is F_1 score. As the datasets is imbalance (the object is on average only at 2% of all pixels), we need to evaluate the F_1 score instead of the normally used accuracy. F_1 score is based on precision and recall, reflects uneven class distribution better than accuracy. F_1 score is the main compared parameter during methods evaluation. F_1 score is computed:

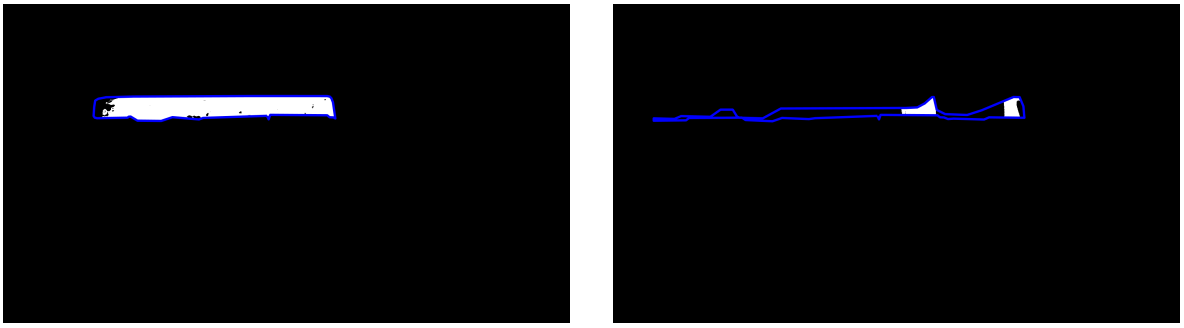
$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

The overall results for all tested methods are in table 2. The results are evaluated as average values obtained from all datasets and all images.

Some methods need initialization from another method. Type of the used method is depicted in the corresponding column.

4.1 Otsu's Segmentation Method

The first experiment with the datasets utilized the Otsu's segmentation method explained above. Illustrative Fig. 4 depicted two cases. Left image 4(a) is almost correctly segmented because the metal is hot and the segmentation is therefore easy. In the right Fig. 4(b) we can observe picture from later part of the process where parts of the metal are not as hot but clearly distinguishable for a human (compare to fig. 2(c)).



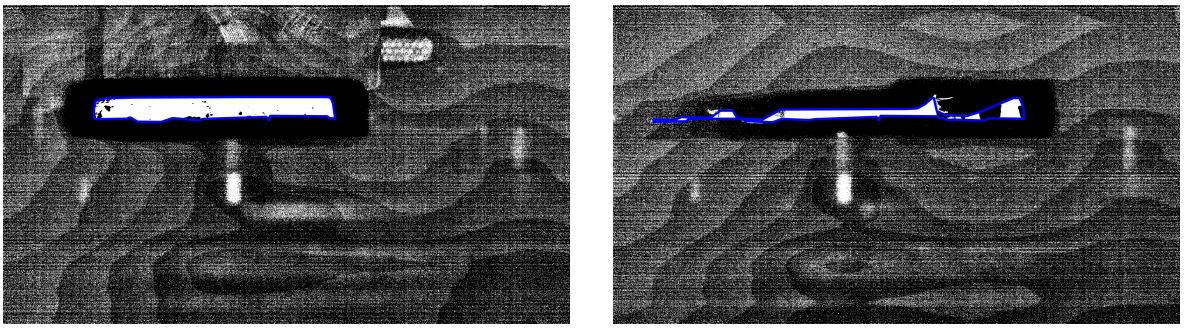
(a) (b)

Figure 4: Segmentation results for Otsu's segmentation method with a blue border from the annotation process (a) result for the image shown in Fig. 1 (b) results for challenging image described in Fig. 2

Otsu's method clearly fails for this kind of pictures. Results are depicted in table 2. As can be seen, the method is quickest of all. As we expected, accuracy is high at 87.72%. But the parameter recall is only 58.18%, which means that in the average image almost half of the image is not correctly labeled. Parameter precision indicated that only every 10th pixel labeled as an object by this method is truly an object. This leads to the important combined parameter F_1 score, which is only 16.11%.

4.2 Adaptive Thresholding Method

The second tested method was the use of adaptive thresholding. We utilized the standard algorithm with neighborhood size set to 1/8 of image dimensions as standard settings for this method, local threshold compute by mean intensity and sensitivity set to 40% to be slightly biased towards background pixels. As depicted in Fig. 5, plain adaptive thresholding works great around the object itself, but elsewhere fails due to noise. In table 2 this situation is shown by decreasing the accuracy and precision, but the increase of recall. F_1 score for adaptive thresholding without any filtering is, therefore, lowest at 9.16%.

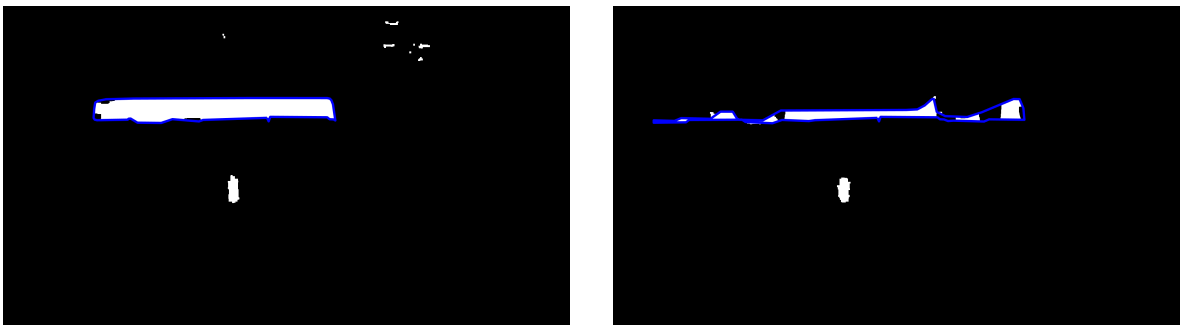


(a) (b)

Figure 5: Segmentation results for adaptive segmentation method with blue border from annotation process (a) results for image shown in Fig. 1 (b) results for challanging image described in Fig. 2

This method was therefore improved by the use of the mathematical morphology [8]. The first added step was morphological opening with a square structural element with a size of 6x6px to reduce noise. The second step was to fill gaps in an object with morphological closing with a square element of 15x15px. Resulting images can be seen in Fig. 6.

In comparison with previous methods, as shown in table 2, adaptive thresholding with morphology is a bit slower, but greatly improve all parameters, except recall which is only slightly better than Otsu's method. The method has a greater F_1 score from all stated methods.



(a) (b)

Figure 6: Segmentation results for adaptive segmentation method augmented with mathematical morphology with blue border from annotation process (a) results for image shown in Fig. 1 (b) results for challanging image described in Fig. 2

4.3 K-Means Clustering Segmentation

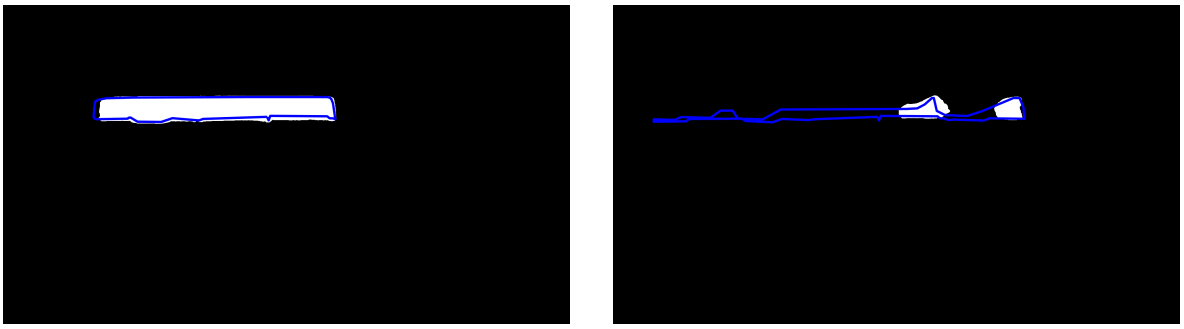
The third method used in this evaluation was k-means clustering segmentation. At the basic level, this method works on a similar principal as Otsu's segmentation method, only with two differences. Firstly, k-means is an iterative method. Secondly, all the computations use pixel values from the whole image, instead of the histogram values. Both these factors slower the method, but doesn't bring any improvements at all. The results in table 2 support that. In comparison, k-means clustering segmentation gains even 2% worse accuracy and F_1 score than simple Otsu's method. And the computational time per image was around 4.8x slower.

The method was run with the number of clusters set to 2, the maximum number of iteration set to 100 and the accuracy threshold set to 0.0001.

4.4 Geodesic Distance Based Segmentation Method

Geodesic distance based segmentation method is an interactive method that needs user input to learn a set of Gabor filters to work with. As we need to all method works automatically, we use Otsu's segmentation algorithm to prepare the initial sample regions of object and background. The object found with Otsu's method was used as object scribble. Then a loose rectangle around this region was selected and uses as background scribble. This background rectangle is defined as a rectangle with a minimal distance of 25 pixels from the object on every side.

Resulting segmentation examples are presented in Fig. 7. This method has a tendency to grow outside from the given area from Otsu's method. This leads to the situation, that slightly more of the object is segmented, but on the other hand, some of the background pixels are also classified as an object.



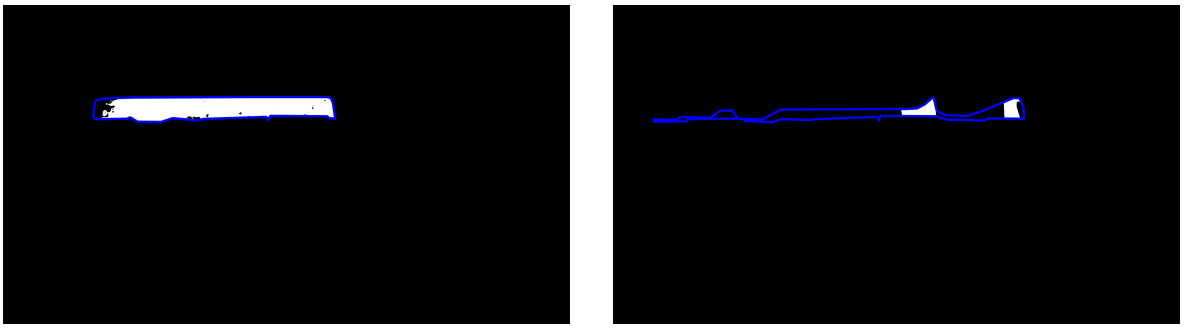
(a) (b)

Figure 7: Segmentation results for geodesic distance based segmentation with blue border from annotation process (a) results for image shown in Fig. 1 (b) results for challanging image described in Fig. 2

The statistics in table 2 clearly indicates that this method has the highest recall of all methods. But the F_1 score is only 20.06%, which is only 5% better than with Otsu's method and far worse than filtered adaptive segmentation. And the time needed to segment one image is 10x higher than with Otsu's method.

4.5 Active Contour Segmentation Method

The last method compared was active contours. This iterative method starts with the region based on the intersection of the previous segmentation Otsu's segmentation for the current image. Then the active contour algorithm refines the object's region based on the energy. For the evolution of active contour uses the Sparse-Field level-set method for implementing active contour evolution [13], Chan-Verse method for segmentation, maximum iteration of 50 iterations, smooth factor and contraction bias set both to 0.



(a) (b)

Figure 8: Segmentation results for Otsu's segmentation with active contour with blue border from annotation process (a) results for image shown in Fig. 1 (b) results for challanging image described in Fig. 2

In table 2, the method has slightly better recall than average, but F_1 score is worse than with Otsu's segmentation method. Considering that this method should refine the region from Otsu's segmentation, this method is not good for this task. The worst part of this method is the computational time, this method is 111x slower than Otsu's method.

Table 2: Summary comparison of all used methods

Method	Initialization method	Time per image	Accuracy	Precision	Recall	F_1 score
Otsu's segmentation method	-	0.14s	87.72%	9.35%	58.18%	16.11%
Adaptive thresholding	-	0.24s	73.61%	4.92%	65.86%	9.16%
Adaptive thresholding with morphology	Adaptive th.	0.29s	98.36%	58.80%	62.72%	60.77%
K-means (2-means)	-	0.67s	85.73%	8.02%	57.84%	14.10%
Geodesic distance-based segmentation	Otsu's th.	1.50s	85.49%	11.28%	90.16%	20.06%
Active contour segmentation	Otsu's th.	15.55s	85.80%	8.80%	67.50%	15.58%

5 Conclusion

This work presents the initial experiments in an ongoing project and describes some segmentation methods usable for segmenting grayscale thermal images. We compare computationally simple methods, namely Otsu's segmentation method for compassion and two variations of adaptive thresholding with some advanced methods like k-means and active contours.

If we evaluate the overall F_1 score, precision, accuracy and time, the clear winner is adaptive thresholding improved with some minor morphological operations. This method evaluated pixel-by-pixel has the F_1 score of 60.77%. But recall is still quite low at 62.72%, that means 37% of an object is not properly located. The method is quick in comparison with other methods. It can be used for initial rough region estimation for further processing and object localization.

The second method is geodesic distance-based segmentation with F_1 score at 20.06% which has also the greatest recall of 90.16%.

The further work will be focused around segmentation with a convolutional neural network. But we need to use another framework, not the Matlab. The reason for not utilize the CNN was that our institution has no

access for Matlab toolbox needed for CNN for research purposes.

Acknowledgement: The completion of this paper was financially supported by Ministry of Industry and Trade (FV10578 - Reseeach and development of new solutions for forming technologies).

References

[1] Arthur, D. and Vassilvitskii, S. 2007. K-Means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA*. SIAM, pp. 1027–1035. DOI: 10.1145/1283383.1283494

[2] Blohm, T., Langner, J., Stonis, M, Behrens, B.-A. 2017. Basic study of incremental forming of serially arranged hybrid parts using cross-wedge rolling. *Procedia Eng.* 207, Jan, pp. 1677–1682. DOI: 10.1016/J.PROENG.2017.10.921

[3] Bradley, D. and Roth, G. 2007. Adaptive Thresholding using the Integral Image. *J. Graph. Tools* 12, pp. 13–21. DOI: 10.1080/2151237X.2007.10129236

[4] Chan, T. F. and Vese, L. A. 2001. Active contours without edges. *IEEE Trans. Image Process* 10, 2, pp. 266–277. DOI: 10.1109/83.902291

[5] Crow, F. C. 1984. Summed-area Tables for Texture Mapping. In *Proc. 11th Annu. Conf. Comput. Graph. Interact. Tech.*. ACM, pp. 207–212. DOI: 10.1145/800031.808600

[6] Fu, X. P. and Dean, T. A. 1993. Past developments, current applications and trends in the cross wedge rolling process. *Int. J. Mach. Tools Manuf.* 33, 3, pp. 367–400. DOI: 10.1016/0890-6955(93)90047-X

[7] Kaur, D. and Kaur, Y. 2014. Various image segmentation techniques: a review. *Int. J. Comput. Sci. Mob. Comput.* 3, 4, pp. 809–814.

[8] Najman, L. and Talbot, H. 2013. *Mathematical Morphology: From Theory to Applications*. John Wiley & Sons.

[9] Otsu, N. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man. Cybern.* 9, 1, pp. 62–66. DOI: 10.1109/TSMC.1979.4310076

[10] Pater, Z., Tomczak, J, and Bulzak, T. 2018. New forming possibilities in cross wedge rolling processes. *Arch. Civ. Mech. Eng.* 18, 1, pp. 149–161. DOI: 10.1016/J.ACME.2017.06.005

[11] Protiere, A. and Sapiro, G. 2007. Interactive Image Segmentation via Adaptive Weighted Distances. *IEEE Trans. Image Process.* 16, 4, pp. 1046–1057. DOI: 10.1109/TIP.2007.891796

[12] Stehman, S. V. 1997. Selecting and interpreting measures of thematic classification accuracy. *Remote Sens. Environ.* 62, 1, pp. 77–89. DOI: 10.1016/S0034-4257(97)00083-7

[13] Whitaker, R. T. 1998. A Level-Set Approach to 3D Reconstruction from Range Data. *Int. J. Comput. Vis.* 29, 3, pp. 203–231. DOI: 10.1023/A:1008036829907