

## RECOGNITION OF HANDWRITTEN LETTERS FROM THE MACEDONIAN ALPHABET USING HAUSDORFF DISTANCE

Ana Atanasovska, Simona Petravić, Kristina Savevska, Sanja Atanasova

*Faculty of Electrical Engineering and Information Technologies,  
"Ss. Cyril and Methodius" University in Skopje,  
Rugjer Bošković bb, P.O. box 574, 1001 Skopje, N.Macedonia  
ksanja@feit.ukim.edu.mk*

**Abstract:** Handwriting recognition is an elementary action performed by human beings due to their traditional five senses. However, computers don't "think" and operate in the same way as humans. In this paper, a simple solution to this problem is provided using the Hausdorff distance. The used data consists of two sets of images: the first set is taken as a reference point used for comparison and consists of handwritten letters from the Macedonian alphabet, and the second set consists of separate letters of the word which is to be recognized and presented to the algorithm. The Hausdorff distance is used to find the closest resemblance between the letter to be recognized and the relevant letter from the reference set.

**Key words:** Hausdorff distance; handwriting recognition; Macedonian alphabet; image processing

## ПРЕПОЗНАВАЊЕ НА РАЧНО НАПИШАНИ ЗБОРОВИ ОД МАКЕДОНСКАТА АЗБУКА СО ПРИМЕНА НА ХАУСДОРФОВО РАСТОЈАНИЕ

**Апстракт:** Проблемот на препознавање ракопис или рачно напишани зборови е многу едноставен за човекот бидејќи тој има повеќе сетила кои учествуваат во процесирањето на таа информација. Овој процес на препознавање ракопис не е толку едноставен за компјутерите поради фактот што тие не „размислуваат“ и не дејствуваат на ист начин како човекот. Во овој труд се приложува едноставно решение на овој проблем со помош на Хаусдорфовото растојание. Се работи со две множества слики: слики од рачно напишани мали букви од македонската азбука кои се земени како референтни и се користат за споредба и второто множество е составено од поединечните букви од зборот кој треба да се препознае. Хаусдорфовото растојание се користи за да се определи со која буква од референтното множество е најслична буквата која се препознава.

**Клучни зборови:** Хаусдорфово растојание; препознавање ракопис; македонска азбука; процесирање слики

### 1. INTRODUCTION

The technological advancement, especially in computer science, provides solutions to many problems in everyday life. Most of these problems pass unnoticed by humans. However, if these solutions are used so that a machine executes a task instead of a person, the time, energy and resources needed can be reduced.

The progress is most evident in personal computers, which are becoming a significant part of the

life of individuals. Despite their continuously improving performance, a computer can only be used as an input-output device for information, which does not provide additional processing of data without human involvement [7]. The solutions to this "shortcoming" are different algorithms written by humans that not only allow processing of specific data, but also perform actions which would take long time if performed by a person. These algorithms are particularly useful when it comes to working with real world data.

In this paper, such an algorithm that recognizes handwritten words from a given image using Hausdorff distance will be presented. Handwriting recognition as an ability of a computer to interpret handwritten text is a topic of research in the field of image processing. The need for investigating methods for handwriting recognition is encountered in the increasing integration of technology into people's everyday lives, and avoiding the traditional "writing by hand".

Using a variety of methods, an image is presented as a sequence of numbers. Each of them corresponds to the light intensity of every pixel in the image. The processing of this image with the aforementioned algorithms may include detecting a specific object, whether it is present in the image, its location, its shape, and so on. This field of computer science not only solves these types of problems, but also makes optimization in terms of time and reliability. In other words, the systems involved in these processes must be adjusted in a way that they will give a solid and faithful result when there is a presence of different obstacles, noise and undefined situations for some desired optimal time.

The Hausdorff is an effective method for determining the similarity/difference between two sets of pixels (images) that gives a reliable result – it claims with certainty that the model is similar to most of the original image, as well as that the image corresponds to most of the model. Namely, the Hausdorff distance between two sets is small when every point in the model is close to some point in the original image and every point in the image is close to some point in the model [7]. This property defines the level of similarity between two sets. This approach naturally handles obstacles such as noise and interference. It is also very useful in applications with overlapping images.

For the purpose of this paper, Python programming language is used, since it contains all the tools needed for image processing and performing the necessary calculations. The models are images of lowercase handwritten letters from the Macedonian alphabet, and based on them the computer should recognize the letters of the new word presented in a form of an image.

This paper is organized as follows: Section 2 covers the mathematical terms and definitions required to develop the problem. We describe our algorithm in details in Section 3. The experimental results are also given in Section 3. Finally, we conclude our work in Section 4.

## 2. MATHEMATICAL TERMS AND DEFINITIONS

### A) Metric and metric space

In order to define the Hausdorff distance it is necessary to introduce the fundamental terms that constitute its definition such as: metric, metric space, complete metric space and compact sets. Detailed overview of this notions can be found in [5].

Let  $X$  be a given set. A **metric** on  $X$  is a real function  $d : X \times X \rightarrow [0, \infty)$ , which satisfies the following conditions:

- 1)  $d(x, y) = 0 \Leftrightarrow x = y$  (identity)
- 2)  $0 < d(x, y) < \infty, \forall x, y \in X, x \neq y$  (positivity)
- 3)  $d(x, y) = d(y, x), \forall x, y \in X$  (symmetry)
- 4)  $d(x, y) \leq d(x, z) + d(z, y), \forall x, y, z \in X$  (triangle inequality).

The ordered pair  $(X, d)$  is called a **metric space**.

We say that a sequence  $\{x_n\}_{n \in \mathbb{N}}$  of points **converges** to a limit  $a$  in a metric space  $(X, d)$  if for every  $\varepsilon > 0$ , there exists  $n_0 \in \mathbb{N}$  such that  $\forall n \geq n_0$ .

$$d(x_n, a) < \varepsilon.$$

A sequence  $\{x_n\}_{n \in \mathbb{N}}$  of points in a metric space  $(X, d)$  is called **Cauchy sequence** if for every  $\varepsilon > 0$ , there exists  $n \in \mathbb{N}$  such that for all  $m, n \geq N$

$$d(x_n, x_m) < \varepsilon.$$

In all metric spaces, convergent sequence is also a Cauchy sequence. A **complete metric space** is a metric space  $(X, d)$  in which every Cauchy sequence is convergent.

Let  $S \subset X$  be a subset of the metric space  $(X, d)$ . A set  $S \subset X$  is called **compact** if every infinite sequence in  $S$  has a subsequence that converges to an element again contained in  $S$ .

### B) Definition of Hausdorff metric space and Hausdorff distance

Let  $(X, d)$  be a complete metric space.

A space  $H(X)$  that contains all the nonempty compact subsets of  $X$  is called **Hausdorff space**. For  $x \in X$  and  $B \in H(X)$ , the **distance from the point  $x$  to the set  $B$**  is defined as

$$d(x, B) = \min\{d(x, y) : y \in B\}.$$

Note that the minimum will always exist since  $B$  is a nonempty compact subset of  $X$ .

Now, let  $A, B \in H(X)$ . **The distance from the set  $A \in H(X)$  to the set  $B \in H(X)$**  is defined as

$$d(A, B) = \max\{d(x, B) : x \in A\}.$$

The function  $d(A, B)$  is also called **directed Hausdorff distance**. In general,  $d(A, B) \neq d(B, A)$ , [3].

The function  $d(A, B)$  ranks each point of  $A$  based on its distance to the nearest point of  $B$ , and then uses the largest ranked point as the distance. In other words, this function identifies the point from  $A$  that has the largest mismatch with the points from  $B$ .

For two given finite point sets  $A = \{a_1, \dots, a_p\}$  and  $B = \{b_1, \dots, b_q\}$  from  $H(X)$ , the **Hausdorff distance** (or Pompeiu-Hausdorff distance) is defined as

$$h(A, B) = \max\{d(A, B), d(B, A)\}.$$

The function  $h(A, B)$  is a metric on the space  $H(X)$  [10]. This function identifies the point of  $A$  that is farthest from any point of  $B$  and vice versa. The Hausdorff distance  $h(A, B)$  measures the degree of mismatching between two sets  $A$  and  $B$ .

Because the Hausdorff distance is a metric, it has to satisfy the conditions for being a *metric* – the distance function is positive everywhere and has the properties of identity, symmetry and the triangle inequality. In context of what is elaborated in this paper – the problem of recognizing handwritten letters from the Macedonian alphabet, the above mentioned conditions mean that a shape is identical only to itself, the order of comparison of two shapes does not matter, and two shapes that are highly dissimilar cannot both be similar to some third shape. This final property, the triangle inequality, is particularly important in pattern matching applications of the Hausdorff distance where several stored model shapes are compared to an unknown shape.

The Hausdorff distance is composed of two asymmetric distances: *the forward distance*, which is the distance from the model to the original image, and *the reverse distance*, which is the distance from the original image to the model. The forward distance is small when each point in the model is close to some point in the image, and the reverse distance is small when each point in the original image is close to some point in the model. In other words, the forward distance indicates when the model looks

like a subset of the image (but not necessarily the other way round) and the reverse distance indicates when the image looks like a subset of the model (but not necessarily the other way round). When both distances are small, then the image and the model look like each other, therefore the Hausdorff distance is also small. The Hausdorff distance can be also called undirected distance, as it incorporates both aforementioned distances. Intuitively, this is a good property to have for model-based recognition: it defines a measure of how much the sets look like each other [7].

### C) Application of the Hausdorff distance

Hausdorff distance between sets of points can be used in problems referring to image comparison and analysis [3], contour fitting, object matching [8], computer vision and operation, pattern recognition such as face recognition [4], and many various fields, where the problems of shape matching and comparison appear. In fact, the Hausdorff distance is used to check if a certain desired object is present in a test image. The smaller the Hausdorff distance between the object and the test image, the higher the probability that the object is present in the image. Furthermore, since the Hausdorff distance is based on proximity between features, altering the model and image features slightly affects the Hausdorff distance.

In this paper, the Hausdorff distance is used as a method of handwriting recognition, i.e. “reading” of handwritten text. This is done by comparing each letter of a word with previously given models of handwritten letters using image processing and calculation over images represented by matrices. The handwritten word which is the subject of recognition is given in the form of an image, which is represented by finite two-dimensional arrays of pixels.

The coordinates of these arrays are integers in a range corresponding to the width and height of the image. For the purpose of the problem, the Hausdorff distance between two two-dimensional arrays needs to be calculated. Therefore, it is assumed that these arrays represent two given sets of points  $A$  and  $B$ . Each point  $a \in A$ , where  $A = \{a_1, \dots, a_n\}$  is represented with Cartesian coordinates as  $(a_x, a_y)$ , and each point  $b \in B$ , where  $B = \{b_1, \dots, b_m\}$  is represented with Cartesian coordinates as  $(b_x, b_y)$ , where  $a_x, a_y, b_x, b_y \in N_0$ , contain the value of the pixel of the corresponding coordinates. For the comparison of the two images, the coordinates of certain

subsets of pixels are used, which in fact identify the shape of the letter represented in the image.

Since the purpose of this work is to read the handwritten word letter by letter, further in the text the letters which need to be recognized will be represented as matrices  $A_i$  of the set  $A$ , where  $i = 1, 2, \dots, n$ , and  $n$  is the number of letters in the word, whereas the reference images of the handwritten alphabet are defined as matrices  $B_j$  of the set  $B$  where  $j = 1, 2, \dots, 31$ . Note that each matrix  $B_j, j = 1, 2, \dots, 31$  represents one letter from the lower case Macedonian alphabet which consists of 31 letters.

### 3. ALGORITHM AND RESULTS

The algorithm used for recognition calculates the Hausdorff distances between the letter to be recognized and all the letters of the Macedonian alphabet. The minimum distance will be used to find the closest matching letter and thus recognize the input.

#### A) Algorithm

The image comparison algorithm consists of several steps:

1. *Image preprocessing of the models of handwritten letters (calculation of the matrices  $B_j, j = 1, 2, \dots, 31$ )*

The first step is to prepare the model images of all thirty-one letters of the Macedonian alphabet. The initial images are RGB images. Calculations over RGB images are practically impossible since they require large computational resources. Since the information that needs to be extracted from the images is the shape of the letters, it is convenient to convert them to grayscale images. Although the RGB channels are reduced, grayscale images still contain numbers in range 0 to 255, which is not suitable for calculations which need to be applied to every single pixel, such as the Hausdorff distance. Therefore, the range of 0 to 255 is normalized to a range of 0 to 1, which is then binarized using the Otsu threshold method [2]. By doing so, pixels receive binary values, 0 or 1, where the pixels denoted with 1 represent the shape of the letter and are of interest to the problem defined in this paper. The binary images are then skeletonized, i.e. reduced to a line with thickness of one pixel, which serves the purpose of removal of unnecessary pixels [11]. As a result of these steps, a thin line which represents the

actual form of the letter is extracted. This image processing is applied to all of the thirty-one images representing the letters in the macedonian alphabet. This is how the set  $B$  of binary matrices  $B_{i,j} = 1, 2, 3, \dots, 31$  is formed.

The choice of the binarization threshold is essential for the optimal generation of shapes of the letters. In some cases, manually adjusting the binarization threshold proved to be of great importance for obtaining better results. It should be kept in mind that real images contain noise caused, for example, from the glare from the camera flash, or brighter and darker parts of the written letters that depend on the pressure applied to the paper while writing. Figure 1 gives examples of the shape of a letter with different thresholds of 0.4, 0.5 and 0.7. Although the human eye can recognize that there is a representation of the letter "κ" even in the first image, a computer would have difficulties making this decision. In computer image processing, in order to accurately interpret the data it is essential for the forms which are connected in the original image to be connected in the processed one as well.



**Fig. 1.** Shape of a letter with different thresholds (left to right: 0.4, 0.5 and 0.7)

2. *Preprocessing of the image of the word to be recognized (calculation of the matrices  $A_i, i = 1, 2, \dots, n$ , where  $n$  is the number of letters in the word)*

The whole word that we want to recognize is represented by only one image. The same preprocessing which is described in the first step is performed on this image. Since a word with length  $n$  is defined as a group of  $n$  letters (letters can be repeated), this image is manually divided into smaller images using a sliding window. In the event of connected letters, the division should be executed more carefully with an appropriate window size, so that the letters can be separated as precisely as possible.

In order to get more accurate Hausdorff distances, it is important that both the models and the letters to be recognized are positioned in approximately the same area within the image. For that reason, redimensioning of the images of the letters is

required. Since the models are positioned in the center of the image, the letters to be recognized must be centered as well. This is done by adding rows and columns of zeros to the sides of the matrices of the letters until the shape of the  $B_j$  matrices is reached. In this case, the models of the letters and the words to be recognized are written in roughly the same size, which is why rescaling is not applied. However, if there is a significant difference in size, rescaling would be a necessary preprocessing step. The redimensioned images of the individual letters represent the matrices  $A_i$ ,  $i = 1, 2, \dots, n$ . The set of all matrices  $A_i$ ,  $i = 1, 2, \dots, n$  is denoted as  $A$ .

### 3. Calculation of the Hausdorff distance between the sets $A$ and $B$

In this step, the Hausdorff distance between each of the matrices  $A_i$ ,  $i = 1, 2, \dots, n$  to each of the matrices  $B_j$ ,  $j = 1, 2, \dots, 31$  is calculated. As stated earlier, pixels with a value of 1 represent the shape of the letter, and those are the pixels that are used to calculate the Hausdorff distance. The Hausdorff distance between each pixel with value 1 of the matrix  $A_i$ ,  $i = 1, 2, \dots, n$  and the corresponding pixels with value 1 of all matrices from the set  $B$  is calculated, and vice versa, calculations are performed for the Hausdorff distance between pixels with value 1 of the matrices from the set  $B$  with the corresponding pixels with value 1 of the matrix  $A_i$ ,  $i = 1, 2, \dots, n$ . According to the definition of the Hausdorff distance stated above, the maximum of these two distances is taken for further calculations:

$$h_j(A_i, B_j) = \max\{d(A_i, B_j), d(B_j, A_i)\},$$

where  $j = 1, 2, \dots, 31$ , and  $i = 1, 2, \dots, n$ , where  $n$  is the number of letters in the word. This way, an array is formed which consists of 31 different Hausdorff distances between the letter of interest and all the letters from the set  $B$ . The most suitable choice for the recognized letter is taken to be the minimum of the array which represents the distance between the matrix  $A_i$  and a matrix  $B_j$ , where  $B_j$  is the matrix of the letter which has the highest match to the letter in  $A_i$ , that is

$$h(A_i, B) = \min\{h_j(A_i, B): j = 1, 2, \dots, 31\},$$

where  $i = 1, 2, \dots, n$ , and  $n$  is the number of letters in the word.

The algorithm is applied on a handwritten word with models which are previously generated from the same handwriting, taking into account the fact that the letters will differ with each writing.

## 4. RESULTS

An example of a handwritten word in the initial RGB format is shown in Figure 2, and the processed binary image with the previously described steps of binarization and skeletonization is shown in Figure 3.

In Figure 4, it is shown how every individual letter from the word is separated in an image suitable for performing step 3 of the algorithm.

In Figure 5, an example of a recognized word is shown, which contains three out of four correct letters.

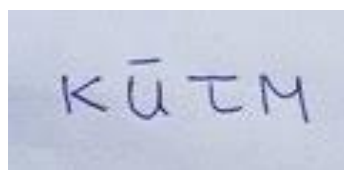


Fig. 2. Handwritten word in the initial RGB format



Fig. 3. Handwritten word after binarization and skeletonization



Fig. 4. Model of the handwritten word



Fig. 5. Model of the recognized word

Another example which is used is the word “математика”. This provides additional insight into the performance of the algorithm on words with repeating letters, considering the multiple instances of the same letter, are slightly different. The different preprocessing steps of this word are shown in Figure 6 and Figure 7.



Fig. 6. Handwritten word after processing

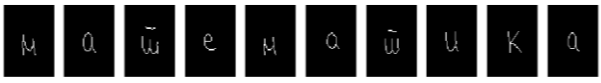


Fig. 7. Model of the handwritten word

It is shown in Figure 8 that even though the multiple instances of the letters “м” and “т” are correctly recognized, that is not the case with the letter “а”. It is correctly recognized only once out of three times and it is recognized as the letter “о” twice. There are seven correctly recognized letters out of ten.



Fig. 8. Model of the recognized word

For the purpose of drawing a better conclusion regarding the accuracy of the algorithm, an additional word is considered. The handwritten word “хаусдорф” through all the the steps of preprocessing and recognition is shown in Figures 9, 10 and 11.



Fig. 9. Handwritten word after processing



Fig. 10. Model of the handwritten word



Fig. 11. Model of the recognized word

Five out of eight letters are recognized correctly. After application of the algorithm on a test set of words with nearly 100 letters in total, the results show accuracy of around 70%.

Due to the algorithm's formulation, the accuracy directly depends on the position and size of the

letters, as well as the similarity of some parts of certain letters. This is reflected in the recognition of the letter “м” as “н” in the word “кпгм”, the recognition of the letter “к” as “х” in the word “математика”, as well as the recognition of “у” as “т” in the word “хаусдорф”. Of course, the solution can be upgraded by adding more models of the letters from the same handwriting and considering additional handwritings from different people. This would make the algorithm more flexible to variations in the letters, such as different inclinations, curvatures, loops and slashes.

The main disadvantage of the direct usage of the Hausdorff distance is that the algorithm is not able to learn different forms of written letters. A more accurate classification could be performed by integrating the Hausdorff distance into a neural network or another machine learning algorithm with a larger dataset [1, 9]. An example of such algorithm is HAVNET (the Hausdorff-Voronoi Network), which contains layers using Voronoi surfaces and Hausdorff distance for the purpose of more natural recognition of letters that resemble the way man thinks. It is crucial that this network is capable of learning, so it enables a more advanced recognition of letters with a much lower degree of similarity, based on the similarity with the other letters [6].

#### 4. CONCLUSION

The Hausdorff distance is a measure of dissimilarity between two sets of points. It is an important metric that is commonly used in many fields. It has many applications in image processing and pattern matching, as well as evaluating the quality of clustering. In this paper, we consider its application in recognizing handwritten letters. In order to simplify the image preprocessing and minimize computational complexity, only lower case letters from the Macedonian alphabet were used. For the same purpose, all letters were written by a single person. The presented algorithm is the simplest and the fastest, even though there are a number of letter recognition techniques that give better accuracy. The simple concept of Hausdorff distance as a measure of resemblance between images can be of great significance if more complex algorithms are applied, but this is not within the scope of this paper. Such algorithms are machine learning techniques combined with Hausdorff distance that give handwriting recognition an accuracy of over 90%.

## REFERENCES

- [1] Andreev, A., Kirov, N.: Word image matching based on Hausdorff distances, *Proc. 10th International Conference on Document Analysis and Recognition*, IEEE, 2009, pp. 396–400.
- [2] Gonzalez R. C., Woods, R. E.: *Digital Image Processing*, second edition, Pearson Education, 2007.
- [3] Huttenlocher, D. P., Klanderman, G. A., Rucklidge, W. A.: Comparing images using the Hausdorff distance, *IEEE Trans. Patt. Anal. Machine Intell.*, Volume 15, Issue 9, pp. 850–863 (1993).
- [4] Jesorsky, O., Kirchberg, K. J., Frischholz, R. W.: Robust face detection using the Hausdorff distance, *Proc. of the Third International Conference on Audio and Video-based Biometric Person Authentication*, Lecture Notes in Computer Science, Springer, 2001, pp. 90–95.
- [5] Kolmogorov, A. N., Fomin, S. V.: *Introductory Real Analysis*, Dover Publication LTD, New York, 1975.
- [6] Rosandich, R. G.: HAVNET: A new neural network architecture for pattern recognition, *Neural Networks*, Volume 10, Issue 1, pp. 139–151 (1997).
- [7] Rucklidge, W.: Efficient visual recognition using the Hausdorff distance, *Lecture Notes in Computer Science*, Vol. 1173, Springer, 1996.
- [8] Sim, D. G., Kwon, O. K., Park, R. H.: Object matching algorithms using robust Hausdorff distance measures, *IEEE Transactions on Image Processing*, Vol. 8, Iss. 3, pp. 425–429 (1999).
- [9] Srisuk, S., Tamsri, M., Fooprateepsiri, R., Sookavatana, P., Suna, K.: A new shape matching measure for nonlinear distorted object recognition, *Proc. VIIth Digital Image Computing: Techniques and Applications*, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10–12 Dec. 2003, Sydney.
- [10] Sternberg, S.: *Dynamical Systems*, Dover Publications, Inc., 2010, pp. 143–145.
- [11] [https://scikit-image.org/docs/dev/auto\\_examples/edges/plot\\_skeleton.html](https://scikit-image.org/docs/dev/auto_examples/edges/plot_skeleton.html)