A k-max Geodesic Distance and its Application in Image Segmentation

Michael Holuša, Eduard Sojka

Department of Computer Science,
FEECS, VŠB - Technical University of Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba, Czech Republic
michael.holusa@vsb.cz eduard.sojka@vsb.cz

Abstract. The geodesic distance is commonly used when solving image processing problems. In noisy images, unfortunately, it often gives unsatisfactory results. In this paper, we propose a new k-max geodesic distance. The length of path is defined as the sum of the k maximum edge weights along the path. The distance is defined as the length of the path that is the shortest one in this sense. With an appropriate choice of the value of k, the influence of noise can be reduced substantially. The positive properties are demonstrated on the problem of seeded image segmentation. The results are compared with the results of geodesic distance and with the results of the random walker segmentation algorithm. The influence of k value is also discussed.

Keywords: Geodesic distance, Shortest path problem, Image segmentation

1 Introduction

Finding the distance between two points is an important task in computer science with many applications in robotics, data clustering, or in image processing [3, 7]. The geodesic distance [10] is a commonly used distance measure in many tasks of image processing [4]. It is defined as the shortest path on the surface that is defined by the image function. In the discrete case, it is the shortest path in the weighted graph that corresponds to the image (the weights of edges reflect the brightness differences between the endpoints of edges).

Many image segmentation methods are based on the geodesic distance. A framework for interactive image segmentation based on the distance computing from the user-provided seeds is presented in [1]. The geodesic distance is also used as an unary term in the graph cut method [9], or as a part of the CRF model that combines the graph cut method and geodesic distance [12]. In [11], the authors improved the geodesic distance segmentation algorithm by incorporating the shape priors. In [5], the author introduced a k-shortest path algorithm that finds k distinct paths between two points. This approach uses the usual geodesic distance determining the k shortest paths between the points. The k-shortest paths method is used, for example, in object tracking [2].

Although the geodesic distance is frequently used, it is also known that it is sensitive to noise in image, which negatively influences the results. This was a motivation to find a distance that ignores the noise if possible and takes into account only the important brightness changes, such as edges. If we assume that the noise is reflected by the low-weighted edges, while the relevant image information is in the edges with higher weights, the sensitivity to noise may decrease by considering only few edges on each path with the highest weights.

In this paper, we propose a new k-max geodesic distance. The length of path is defined as the sum of the k maximum edge weights along the path. The k-max geodesic distance is defined as the length of the path that is the shortest one in this sense. For illustrating the properties of the k-max geodesic distance and for its comparison with the geodesic distance, the seeded segmentation is used. The seeded segmentation uses a priori user-provided seeds scribbled into the particular segment areas. The distance is computed from the seeds to all image points. In the binary image segmentation, an image point is labeled as an object, if the distance from the object seed is lower than the distance from the background seed. Otherwise, the image point is labeled as a background.

We also compare our method with the random walker segmentation method proposed in [6]. This method was chosen since it has a similar approach as the distance-based methods; it determines the probabilities that a random walk from an image point reaches one of the seed points.

The paper is organized as follows. The problems of geodesic distance are presented in Section 2. In Section 3, the k-max geodesic distance is introduced. Section 4 contains the description of the algorithm for computing the k-max geodesic distance. The experimental results and the comparisons are presented in Section 5. Section 6 is a conclusion. We note that in the rest of the paper, we will simply say k-max distance instead of the long name k-max geodesic distance.

2 Geodesic Distance and Its Problems

In this section, we focus on the geodesic distance and its behavior in images. Consider a graph and two nodes in it, denoted by A and B, respectively. Let P be a path connecting A and B; $P = (v_{p_1}, v_{p_2}, v_{p_3}, \ldots, v_{p_n})$, where v_{p_i} are the nodes through which the path is running; $v_{p_1} \equiv A$, $v_{p_n} \equiv B$. The geodesic length of path is the sum of the weights of all its edges, i.e., $l_{\mathrm{g}}(P) = \sum_{i=1}^{n-1} w_{(p_i,p_{i+1})}$. Let $\mathcal{P}_{AB}$ be the set of all existing paths between A and B in the graph. The geodesic distance between A and B is then defined as $d_{\mathrm{g}}(A,B) = \min_{P \in \mathcal{P}_{AB}} l_{\mathrm{g}}(P)$.

In image processing, the weight of edge is often determined by the equation

$$w_{i,j} = 1.0 - e^{-\frac{(b_i - b_j)^2}{2\sigma_w}} + \beta\,, \tag{1}$$

where b_i, b_j are the values of brightness at the v_i and v_j node, respectively; σ_w is a constant. The value of β determines the price for using the edge regardless how big the brightness difference between its endpoints is. It can also be $\beta = 0$, which reflects the fact that the pixels with the same brightness are regarded

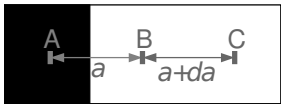


Fig. 1. The synthetic test image (it is shown without noise here) with the points depicted between which the distance is to be measured. We would expect that $d_{\mathrm{g}}(A,B) > d_{\mathrm{g}}(B,C)$ (especially for $da = 0$). Due to noise, the opposite incorrect result is often reported by the geodesic distance (Table 2).

as close to each other; the points may be geometrically distant in the xy plane but they should create one segment. Although this setting does not satisfy the identity of indiscernibles condition of the metric, it can be useful to keep small or even zero distances in the areas with a constant brightness. In the rest of the paper, we consider $\beta = 0$.

We tested the geodesic distance on the synthetic images containing the unit brightness step (Fig. 1) with Gaussian noise added ($\sigma = \frac{1}{3}$) into the image. The geodesic distances $d_{\mathrm{g}}(A,B)$ and $d_{\mathrm{g}}(B,C)$ were measured (see Fig. 1). Since A and B are in different image segments, and B and C are both in the same segment, we expect that $d_{\mathrm{g}}(A,B)$ should always be greater than $d_{\mathrm{g}}(B,C)$. The results in Table 1 show that it is not necessarily true if noise is present. The error rate stated in the table tells how many times the geodesic distance erroneously answered that $d_{\mathrm{g}}(A,B) < d_{\mathrm{g}}(B,C)$. It can be seen that the geodesic distance has a high error rate even in the cases when B is placed directly into the center of AC ($da = 0$).

Table 1. The error rate computed for the geodesic distances from Fig. 1, $\sigma = \frac{1}{3}$ (see text for further explanation). All the values were computed from 10^5 samples.

	$da = 0$	$da = 2$	$da = 5$	$da = 10$
$a = 10$	14.5 %	22.0 %	36.5 %	61.0 %
$a = 20$	18.0 %	25.5 %	37.5 %	60.0 %

Let us now illustrate how this problem influences image segmentation (Fig. 2). Consider an image with two segments: the object and the background. Let O and B denote the object and the background seed, respectively. Let x measure the coordinates of points along the shortest path between O and B; $I(x)$ stands for brightness at x; $d_{\mathrm{g}}(O,x)$ is the distance between the object seed and the point whose coordinate is x. Similarly, $d_{\mathrm{g}}(B,x)$ is the distance from the background seed. The big change of $I(x)$ shows the place where the edge separates the object and the background, which is also the place where the diagrams of the functions $d_{\mathrm{g}}(O,x)$ and $d_{\mathrm{g}}(B,x)$ should intersect if segmentation is done on the basis of distance. Fig. 2 shows how the situation may turn out in the real-life (i.e. noisy) images. The equality of distances may occur at other place than the edge, which

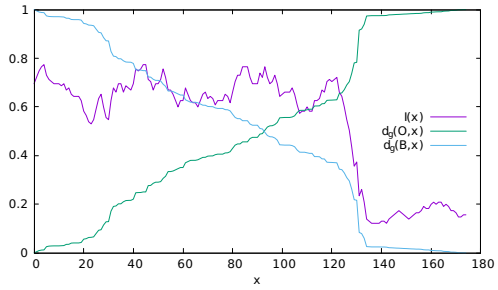


Fig. 2. The diagram of the distances along a path between the object seed and the background seed; x measures the coordinate along the path (for the object seed, we have $x = 0$); $d_{\mathrm{g}}(O, x)$ stands for the distance between the object seed and the point whose coordinate along the path is x. Similarly, $d_{\mathrm{g}}(B, x)$ is the distance from the background seed. $I(x)$ stands for the brightness at x. See text for further explanation.

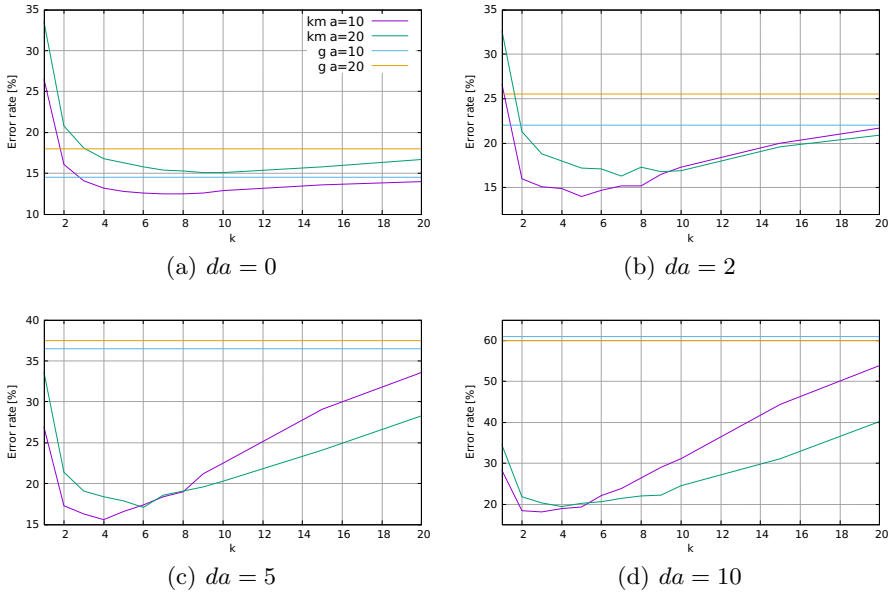
leads to incorrect segmentation. Informally speaking, this problem is caused by summing many small and unimportant values, which may overshadow the values that are important.

3 *k*-max Geodesic Distance

In this section, we introduce the new k-max distance. Let $\sum_{\mathrm{top}_k}(.)$ stand for the sum of the k highest values in a collection of nonnegative real numbers (the edge weights in our case). We define the length of path as a sum of the k highest weights on it, i.e., $l_{k\mathrm{m}}(P) = \sum_{\mathrm{top}_k}(w_{(p_1,p_2)}, w_{(p_2,p_3)}, \ldots, w_{(p_{n-1},p_n)})$, where P is a path. Let $\mathcal{P}_{AB}$ be the set of all paths between A and B. The k-max distance between A and B is then defined as $d_{k\mathrm{m}}(A, B) = \min_{P \in \mathcal{P}_{AB}}\{l_{k\mathrm{m}}(P)\}$.

The behavior of the k-max distance was tested on the same images as in the case of the geodesic distance in the previous section. Firstly, we have measured the distances $d_{k\mathrm{m}}(A, B)$ and $d_{k\mathrm{m}}(B, C)$ between the points in Fig. 1 and evaluated the error rates in the same way as in the previous section. The results for $k \in \langle 1, 20\rangle$ are visualized in Fig. 3. For clarity, we also show the corresponding error rates of the geodesic distance from Table 1, which are visualized as the straight horizontal lines in the diagrams. The results show that, for almost every k, the error rate is much better than it is in the case of geodesic distance. The k-max distance performs worse only for $k = 1$ combined with a low value of da. For the high values of k, the error rate of the new distance approaches (from the bottom) to the error rate of the geodesic distance, which is expected since for $k = \infty$, $d_{\mathrm{g}}(A, B) \equiv d_{k\mathrm{m}}(A, B)$.

We also tested how the k-max distance behaves along the shortest path between the object and background seed, similarly as we did for the geodesic distance in the previous section (Fig. 2). The result is shown in Fig. 4. It can be seen that the k-max distance is much less sensitive to noise and the edge



(a) $da = 0$ (b) $da = 2$

(c) $da = 5$ (d) $da = 10$

Fig. 3. The error rates of the k-max distance for various values of k, a and da (see Fig. 1). For a wide range of k values, the k-max distance reaches much better error rates than the geodesic distance. The explanation of colors presented in (a) is valid also for all remaining images; km stands for the k-max distance, g stands for the geodesic distance.

position is detected correctly. (We explain that the range on the x axis is bigger than in the case of geodesic distance since it happened here that the shortest k-max path is longer in the xy plane than it was for the geodesic distance in Fig. 2. This conforms with the expectation since we have $\beta = 0$ in Eq. (1)).

4 The Algorithm

In this section, we present how the algorithm for computing the k-max distance is constructed. Taking into account the fact that the distances in graph are computed from the shortest to the longest, the algorithm may be regarded as similar to the well-known Dijkstra algorithm. On the other hand, substantial modifications are required so that the distance whose definition was presented in Section 3 could be computed. When computing the geodesic distance, only one value is stored in each node. During the updates, it is changed into the distance that has been found to this node. When computing the k-max distance, a list of vectors containing k maximum weights has to be associated with each node. The following example shows the need for storing such a list.

Consider three nodes in a graph (Fig. 5), denoted by A, B, and C, respectively. Say that we have $k = 2$ and that two paths exist from A to B. Let

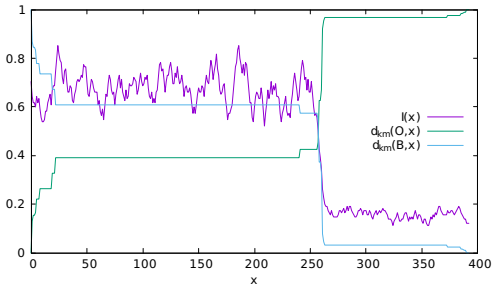


Fig. 4. The diagram of the k-max distances along the path between the object seed and the background seed. In contrast with the geodesic distance (see Fig. 2 for comparison and further explanation), the position of edge is indicated correctly.

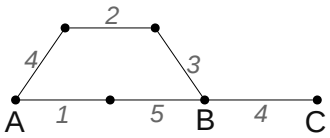


Fig. 5. An example of weighted graph on which the k-max distance algorithm is illustrated.

$\boldsymbol{m}_{B_1} = (4, 3)$ and $\boldsymbol{m}_{B_2} = (5, 1)$, respectively, be the vectors of maximum values (*max vectors* briefly) corresponding to both paths from A to B (Fig. 5). The vectors contain two (since $k = 2$) maximum weights found along the corresponding path (for simplicity, we now suppose that the weights are integer numbers). It follows that the distance between A and B is $d_{\mathrm{km}}(A, B) = 5 + 1 = 6$. Let us now say that B and C are connected with an edge whose cost is $w_{B,C} = 4$. The first mentioned path from A to B leads to the vector $\boldsymbol{m}_{C_1} = (4, 4)$ at C (the value of 3 in $\boldsymbol{m}_{B_1}$ is replaced with the value of 4), which gives the distance $d_{\mathrm{km}}(A, C) = 4 + 4 = 8$. Surprisingly, the second path from A to B with the vector $\boldsymbol{m}_{B_2} = (5, 1)$ that determined the k-max distance $d_{\mathrm{km}}(A, B)$ gives the vector $\boldsymbol{m}_{C_2} = (5, 4)$ that does not determine the distance $d_{\mathrm{km}}(A, C)$ since $5+4 > 4+4$. In spite of the fact that the vector $\boldsymbol{m}_{B_2}$ was decisive for determining the distance between A and B, another vector that was not the best one at B was needed at B for determining the distance $d_{\mathrm{km}}(A, C)$. It was necessary to have at B not only the vector $\boldsymbol{m}_{B_2}$, but also the vector $\boldsymbol{m}_{B_1}$, i.e. a list of vectors is generally needed at each node. The algorithm can now be formulated as follows.

Algorithm The k-Max Distance in Graph
Input: The graph and its node S from which the distances should be computed.
Output: The k-max distances to all the nodes in the graph.

1. For all nodes in the graph, clear the list of the attached max vectors.

2. Attach the max vector $\boldsymbol{m}_{S_1} = (0, 0, \ldots, 0)$ to S and mark this vector as active.
3. **while** an active vector exists in any node in the graph **do**
4. Among all active vectors (attached to all nodes of the graph), find the vector, denoted by $\boldsymbol{m}_T^*$, whose sum of values is minimal. The sum determines the k-max distance for the node, denoted by T, to which this vector is attached.
5. Output the distance of T. Mark the vector $\boldsymbol{m}_T^*$ as inactive; it will not be used in searching in the subsequent passes through the while cycle.
6. For each neighbor of T (let U be such a neighbor, let $w_{T,U}$ be the weight of the edge connecting T and U, and let $\mathcal{M}_U$ be the list of the vectors attached to U) update $\boldsymbol{m}_T^*$ with $w_{T,U}$ (updating is described later). If it is possible that, under certain circumstances in future, the updated vector can lead to a better distance than all other vectors in $\mathcal{M}_U$ (the test is described later), add the updated vector into $\mathcal{M}_U$.

The operation of updating $\boldsymbol{m}_T^*$ with $w_{T,U}$ is carried out as follows. Find the minimum value in $\boldsymbol{m}_T^*$. If the minimum value is greater then $w_{T,U}$, $\boldsymbol{m}_T^*$ remains unchanged. Otherwise, the minimum value in $\boldsymbol{m}_T^*$ is replaced with $w_{T,U}$.

Now we focus on the decision whether a new (updated) max vector should be added into the list of vectors that are associated with the node. Say that we have two max vectors associated with a certain node in the graph (like the B node in Fig. 5). It is possible that the shortest paths to some other nodes (like to the node C in Fig. 5) will run through this node. The key question now is: Are both the vectors important for generating the possible shortest paths to other nodes? Is it possible to say, for example, that some of them will never be used in any shortest path and, therefore, it is not necessary to store it? The answer is given in the following observation: Consider two max vectors $\boldsymbol{m}_1 = (m_{1,1}, m_{1,2}, \ldots, m_{1,k})$ and $\boldsymbol{m}_2 = (m_{2,1}, m_{2,2}, \ldots, m_{2,k})$. Suppose that the values in the vectors are sorted from the biggest to the smallest, i.e. $m_{1,1} \geq m_{1,2} \geq \cdots \geq m_{1,k}$ and similarly also for the second vector. If the inequality $\sum_{j=1}^{q} m_{1,j} \leq \sum_{j=1}^{q} m_{2,j}$ holds for all q, $1 \leq q \leq k$, then the vector $\boldsymbol{m}_2$ will never be used in any shortest path to some other node, i.e. there is no need for storing it.

In order to show this decision method, say that we measure the length of path from A through B to C (see Fig. 5, contrary to Fig. 5, the common subpath from B to C has generally more than one edge). We have two max vectors $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ at B. Say that r values are replaced in $\boldsymbol{m}_1$ and s values are replaced in $\boldsymbol{m}_2$ on the way from B to C. For simplicity, we firstly explore the case $r = s$. We compare the sums of $k - r$ maximum values in both vectors. If $\sum_{j=1}^{k-r} m_{1,j} \leq \sum_{j=1}^{k-r} m_{2,j}$, $\boldsymbol{m}_2$ cannot give a shorter path length to C (for the BC subpath that is being considered). The lengths of the particular whole paths (that are different between A and B) are given by the above mentioned sums plus the sum of the newly included costs. These, however, are the same for both paths since they are the maximum costs on the considered common subpath from B to C. Now we consider the case $r > s$ (the case $r < s$ may be explained by interchanging the

vectors). From the fact that $r > s$, it follows that $r - s$ entries from r smallest entries in $\boldsymbol{m}_2$ were greater than $r - s$ smallest entries that modified $\boldsymbol{m}_1$ on the subpath from B to C. Again, if $\sum_{j=1}^{k-r} m_{1,j} \leq \sum_{j=1}^{k-r} m_{2,j}$, $\boldsymbol{m}_2$ cannot give a shorter path length to C. The lengths of the whole paths are obtained by adding the sum of the weights of s new elements that are the same in both cases (s maximum weights on the common subpath from B to C) and by adding the sum of the remaining $r - s$ elements that is greater for $\boldsymbol{m}_2$. This consideration is valid for all values of r, $0 \leq r \leq k$, which corresponds to various subpaths between B and C and various weights of edges along these subpaths.

5 Experiments

The efficiency of the k-max distance is evaluated on the seeded image segmentation. At the beginning of this section, we show how the choice of the value of k influences the quality of segmentation. Then, we focus on the relationship between the segmentation results and the positions of seeds. After that, we will test the methods on the real-life image segmentation. We will compare the results obtained from the k-max distance, the geodesic distance, and the random walker segmentation. In the experiments, we set the parameters from Eq. (1) to $\sigma = \frac{1}{3}$ and $\beta = 0$ for the distance-based methods. In the case of random walker, we experimentally found σ that achieves the best result for each image. The seeds were defined manually.

Fig. 6. On the influence of the k value on the segmentation result. The input images with the seeds (the first column), the results of seeded segmentation based on the geodesic distance (the second column), and the results of seeded segmentation based on the k-max distance for various values of k (the columns 3-5).

As was demonstrated in Fig. 3, the error rate of the k-max distance varies with the value of k. In the first experiment, we show that the value of k also affects the result of segmentation. In Fig. 6, the results of segmentation are presented for the geodesic distance (column 2) and for the k-max distance for

several k values ($k = 1$, $k = 6$, and $k = 20$; column 3-5). In the first image (row 1), the goal is to segment only the yellow umbrella into which the object seed is scribbled. The segmentation based on the geodesic distance is apparently bad (Fig. 6). In the case of the k-max distance, the segmentation is close to the desired result if low values of k (even $k = 1$) are used. For $k = 20$, the object segment that is detected "overflows" from the true object, and the result becomes similar to the result of the geodesic distance (as was explained before, this behavior is expected). On the other hand, a good segmentation of the gravestone in the second image (row 2) is achieved with a relatively high value of k ($k = 20$); lower values of k lead to an incomplete object. We can conclude that the choice of the k value is not universal, but it depends on image properties.

In the next experiment, we explore how the positions of seeds influence the result of segmentation (Fig. 7). It can be seen that the results of geodesic distance (row 2) are dependent on the seed positions, which is undesirable. The k-max distance provides much better results (row 3), which again, simply speaking, is mainly due to the fact that it does not sum the noise values that are unimportant. In both cases, the results are as expected; the roots of the behavior that can be seen in the figure were explained sufficiently in Sections 2 and 3.

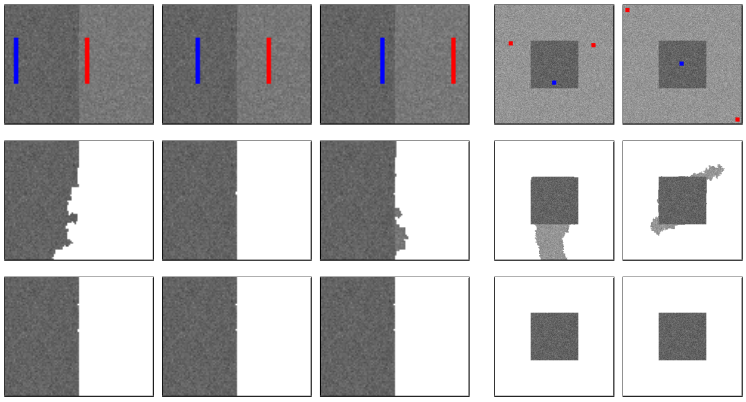


Fig. 7. The sensitivity to the seed positions. The input images with the seeds (first row), the results of segmentation using the geodesic distance (second row), the results of segmentation using the k-max distance, $k = 2$ (third row).

Finally, we test and compare the methods on the images obtained from the Berkeley segmentation dataset [8]. In addition, we compare the distance-based methods with the random walker segmentation algorithm. The results are shown in Fig. 8 and Fig. 9 (in the case of k-max distance, the value of k was chosen to give the visually best segmentation). It can be seen again that the k-max distance outperforms the geodesic distance, and also outperforms the random walker method. Naturally, the results could be improved by adding new seeds.

We did not do so since our goal was to compare the distance measuring methods under the same and difficult circumstances that can be expected in practice.

From the algorithm description in Section 4, it is obvious that computing the k-max distance is more complicated than computing the geodesic distance. This is reflected in a bigger time complexity. Table 2 shows the run times that were needed for computing the segmentations of particular previously shown images; various values of k were considered.

In Section 4, we also explained the need for storing more than one max vector in each graph node, which slows down the algorithm and increases the memory consumption. Table 3 shows the average number of max vectors per a node for a set of selected images and for various values of k. It can be seen from the table that the number of vectors increases with k. Fortunately, according to our experience, good segmentation results are usually achieved with $k < 30$; the number of max vectors remains acceptable for these values of k.

Fig. 8. The results of segmentation based on the geodesic distance, the k-max distance and the random walker technique. The input images with the seeds (first row), the results for the random walker (second row), the results for the geodesic distance (third row), the results for the k-max distance (fourth row).

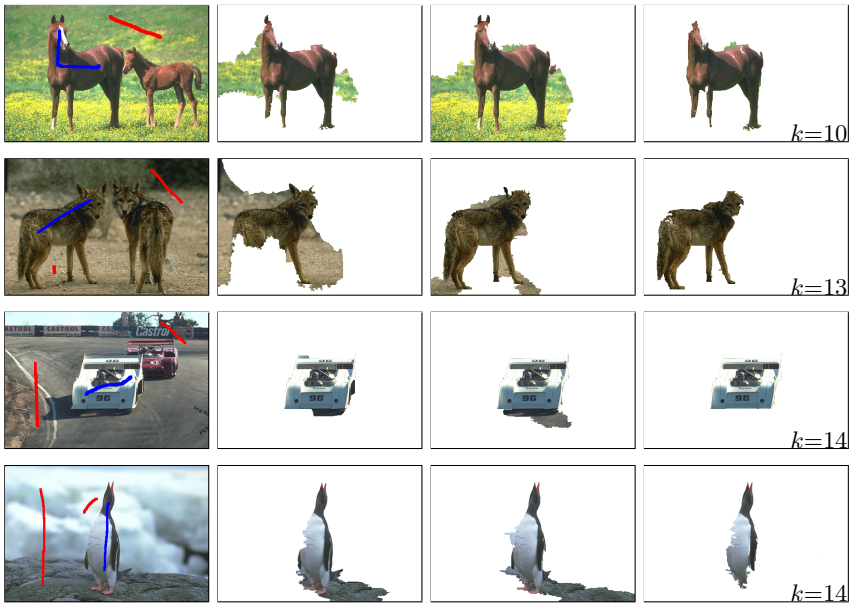


Fig. 9. The results of segmentation based on the geodesic distance, the k-max distance and the random walker technique. The input images with the seeds (first column), the results for the random walker (second column), the results for the geodesic distance (third column), the results for the k-max distance (fourth column).

6 Conclusion

In this paper, we introduced a new distance in graphs, called the k-max geodesic distance. The length of path is defined as the sum of the k maximum edge weights along the path. The distance is defined as the length of the path that is the shortest one in this sense. We also presented the algorithm for computing the new distance. We demonstrated the problems of geodesic distance and we showed that the k-max distance has a chance to overcome them. The new distance measure was tested by using it in seeded image segmentation; the geodesic distance was used for comparison, and also the random walker method, which is a recognized algorithm in the seeded segmentation area. According to the results, the k-max geodesic distance achieves the best results. On the other hand, its computation requires more time and memory than the computation of the geodesic distance, which is caused by the need for storing (at each graph node) a certain number of vectors containing k maximum weights along the possible paths to the node. Therefore, our future goal is to try to further reduce the number of stored vectors by improving the method that decides about keeping or discarding the vectors. This should reduce the time and memory requirements of the algorithm.

Acknowledgements. This work was supported by the grant SP2015/141 of VŠB - TU Ostrava, Faculty of Electrical Engineering and Computer Science.

Table 2. The run times for computing the selected previously shown segmentations using the k-max distance (one 2.5 GHz core computer, image size = 450×300).

image	$k = 1$	$k = 10$	$k = 30$	$k = 50$
penguin (Fig. 9 row 4)	0.29 s	0.48 s	0.97 s	1.90 s
mushroom (Fig. 8 col 2)	0.28 s	0.58 s	1.96 s	5.73 s
horses (Fig. 9 row 1)	0.29 s	0.64 s	2.83 s	15.04 s
totems (Fig. 8 col 3)	0.30 s	0.70 s	3.57 s	17.47 s

Table 3. The average number of max vectors per pixel for selected images (and various values of k).

image	$k = 10$	$k = 20$	$k = 30$	$k = 50$
penguin (Fig. 9 row 4)	1.22	1.46	1.76	2.29
mushroom (Fig. 8 col 2)	1.47	2.34	3.18	6.23
horses (Fig. 9 row 1)	1.63	2.58	4.00	8.65
totems (Fig. 8 col 3)	1.63	3.02	5.19	12.08

References

1. Bai, X., Sapiro, G.: Geodesic matting: A framework for fast interactive image and video segmentation and matting. Int. J. Comput. Vision 82(2), 113–132 (2009)
2. Berclaz, J., Turetken, E., Fleuret, F., Fua, P.: Multiple object tracking using k-shortest paths optimization. IEEE Trans. Pattern Anal. Mach. Intell. (2011)
3. Borgefors, G.: Distance transformations in arbitrary dimensions. Computer Vision, Graphics, and Image Processing 27(3), 321 – 345 (1984)
4. Criminisi, A., Sharp, T., Blake, A.: Geos: Geodesic image segmentation. In: Proceedings of the 10th European Conference on Computer Vision: Part I. pp. 99–112. ECCV '08, Springer-Verlag, Berlin, Heidelberg (2008)
5. Eppstein, D.: Finding the k shortest paths. SIAM J. Comp. 28(2), 652–673 (1998)
6. Grady, L.: Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 28(11), 1768–1783 (2006)
7. Hajdu, A., Kormos, J., Nagy, B., Zrg, Z.: Choosing appropriate distance measurement in digital image segmentation. Annales Univ. Sci. Budapest. Sect. Comp 24, 193–208 (2004)
8. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision. vol. 2, pp. 416–423 (2001)
9. Price, B.L., Morse, B.S., Cohen, S.: Geodesic graph cut for interactive image segmentation. In: CVPR. pp. 3161–3168. IEEE (2010)
10. Toivanen, P.J.: New geodesic distance transforms for gray-scale images. Pattern Recogn. Lett. 17(5), 437–450 (1996)
11. Wang, J., Yagi, Y.: Shape priors extraction and application for geodesic distance transforms in images and videos. Pattern Recogn. Lett. 34(12), 1386–1393 (2013)
12. Zhou, L., Qiao, Y., Yang, J., He, X.: Learning geodesic crf model for image segmentation. In: Image Processing (ICIP), 2012 19th IEEE International Conference on (2012)