Evaluation of Established Line Segment Distance Functions[1]

S. Wirtz and D. Paulus

Institute for Computational Visualistics, University of Koblenz-Landau, Universitätsstr. 1, D-56070 Koblenz
e-mail: wirtzstefan@uni-koblenz.de, paulus@uni-koblenz.de

Abstract—In this paper we present an evaluation of six well established line segment distance functions within the scope of line segment matching. We show analytically, using synthetic data, the properties of the distance functions with respect to rotation, translation, and scaling. The evaluation points out the main characteristics of the distance functions. In addition, we demonstrate the practical relevance of line segment matching and introduce a new distance function.

Keywords: distance function.

DOI: 10.1134/S1054661816020267

1. INTRODUCTION

Evaluating the similarity of two geometric shapes is an important issue in different fields of computer science including computer vision and pattern recognition. We will introduce a new distance function and present an evaluation study of this method and six established ones within the scope of line matching.

The evaluation lines out the characteristics of these distance functions.

The requirements of line matching differ depending on the application domain. In pairwise image matching applications it is useful to be invariant against blurring, scaling, rotation and translation.

In other cases, it is import to be variant against all these attributes. For example, if the aim is to search the best fitting rendering of a taken image using a set of 2D renderings of a 3D object, it is important to notice differences in scaling, rotation and translation.

To choose eligible distance functions concerning the requirements of an application the knowledge of the abilities is indispensable. For that reason, we describe and analyze in this approach established distance functions for lines or line segments: Hausdorff-distance (HD), Trucco-distance (TD), Modified line segment Hausdorff-distance (MHD), Modified perpendicular line segment Hausdorff-distance (MPHD), Midpoint-distance (MD), Closest point-distance (CD) and our new Straight line-distance function (SD)—definitions and references will be given below. These functions are applied on and analyzed with simulated data which allows an exact evaluation. This approach focus on the evaluation of the distance function independently of the matching algorithm—knowing that there exist a lot of sophisticated approaches for line matching.

2. RELATED WORK

We describe approaches dealing with chamfer, line, and segment matching. For these topics an extensive literature exists which we summarize to a limited overview in the following.

Bay et al. [6] present an approach for matching line segments between two uncalibrated wide-baseline images. This approach is able to estimate the fundamental matrix robustly even from line segments only. In that approach no prior knowledge about the scene or camera positions is needed. The authors generate an initial set of line segment correspondences, which is iteratively increased by adding matches consistent with the topological structure of the current ones. Finally, a coplanar grouping stage allows to estimate the fundamental matrix. Schmid and Zisserman [4] describe two algorithmic approaches. These methods require apriori the fundamental matrix of an image pair, or the trifocal tensor of an image triple, and cover the cases of both short and long range motion. Both methods use cross-correlation as matching scores for similarity measurements of line segments, whereby the method for long range motion has to adapt additionally the correlation measure. Werner and Zisserman [5] present a fully automatic approach for reconstructing of buildings from multiple images. They fit geometric models by sweeping and introduce for that purpose sweeping scene planes about a line at infinity, correlation based search for building edges using translating rectangles, and inter image homographies; they search for local gradient maxima along translating line segments.

Witt and Weltin [11] present the approach *Iterative Closest Lines* which extends the algorithm *Iterative*

[1] This paper uses the materials of the report submitted at the 9th Open German-Russian Workshop in Pattern Recognition and Image Understanding, held on Koblenz, December 1–5, 2014 (OGRW-9-2014).

[2] The article is published in the original.

Received September 9, 2015

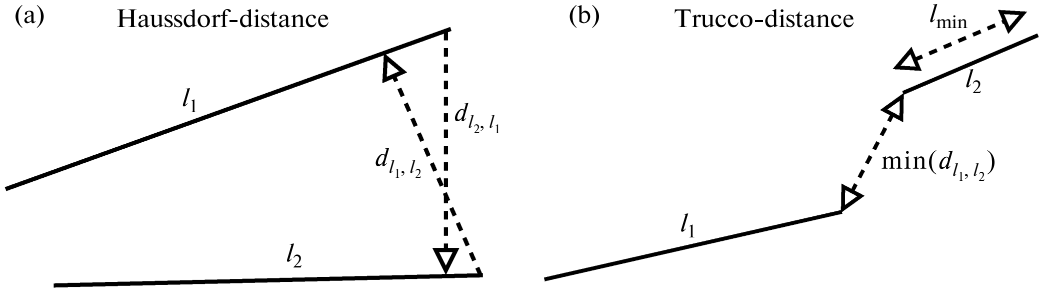

Fig. 1. Visualization of the used quantities of the Haussdorf- and Trucco-distance.

Closest Points (ICP)[7] on line segment matching. The approach works similar to ICP. Line segments are detected, line correspondences to this line segments with minimal Euclidean distances are searched and a 3D rigid transformation is applied on the correspondences.

Liu et al. [2] introduce a study dealing with the object localization problem in images given a single hand-drawn example or a gallery of shapes as the object model. They present an approach for improving the accuracy of chamfer matching while reducing its computational cost. For these purposes, they proposed a method for incorporating the edge orientation in the cost function and solving the matching problem in the orientation augmented space. The novel cost function is smooth and can be computed in sublinear time in the size of the shape template.

Sudarshan [8] addresses the problem of determining the path of a vehicle on a given vector map of roads, based on tracking data such as that obtained from onboard GPS receivers. He presents techniques for modifying existing methods for map-matching based on the idea of segment-wise matching, where a segment is contiguous sequence of track points. Track points and segments are assigned scores that quantify the expected accuracy of matching them to map features. Therefore, the Frechet distance is used, which takes the position along paths into account.

3. APPROACH

We consider in our application that we have two sets of line segments and do not care where they come from (e.g., Canny edge detector or a rendered 3D CAD model). The source of the line segments is unimportant for the examination of the distance function, because we want to evaluate these functions independently of line segment extraction mechanism, which could cause bias.

In this approach, we work with points in pixel coordinates and define a line segment l by $\mathrm{l}_i = p_{1, \mathrm{l}_i} \times p_{2, \mathrm{l}_i}$. The geometric object G is the generalization of the set of line segments L.

A distance function for line segments is described as $\mathrm{d}^{(\mathrm{l})} : \mathrm{l} \times \mathrm{l} \mapsto \mathbb{R}^+$ and measures the dissimilarity of line segments, where a score of zeros means that they are identical.

For the matter of line matching, we match the sets $\mathrm{G}_i = \{g_{1, i}, g_{2, i}, \ldots, g_{N_i, i}\}$ with $i = \{1, 2\}$. Resulting we get sets of matched geometric objects $\mathrm{G}_i^{(m)}$ and sets of unmatched geometric objects $\overline{\mathrm{G}_i^{(m)}}$ with $\overline{\mathrm{G}_i^{(m)}} \cap \mathrm{G}_i^{(m)} = \mathrm{G}_i$. The assignment γ will be optimized to yield γ^* by using a distance function and a penalty function δ_p which punishes unmatched geometric objects.

$$\gamma^* = \underset{\gamma}{\operatorname{argmin}} \sum_{g \in \mathrm{G}_i^{(m)}, g' \in \overline{\mathrm{G}_1^{(m)}}, g'' \in \overline{\mathrm{G}_2^{(m)}}} d(\gamma(g), g) + \delta_{p_1}(g') + \delta_{p_2}(g''). \tag{1}$$

The penalty functions are described as $\delta_{p_j} : \overline{\mathrm{G}_j^{(m)}} \mapsto \mathbb{R}^+, j = \{1, 2\}$ and can be identical with $\delta_{p_1} = \delta_{p_2}$.

3.1. Distance Functions

For the evaluation we use the (i) Hausdorff-distance, (ii) Trucco-distance, (iii) Modified line segment Hausdorff-distance, (iv) Perpendicular-distance, (v) Midpoint-distance, (vi) Closest point-distance, and (vii) our own Straight line-distance function, which all measure the distance of two given line segments l_1 and l_2.

The (i) Hausdorff line segment-distance [9] is illustrated in Fig. 1a and described as following:

$$d_{l_1, l_2}(l_1, l_2) = \max_{\mathbf{p} \in l_1} \min_{\mathbf{q} \in l_2} \|\mathbf{p} - \mathbf{q}\|, \tag{2}$$

$$d_{l_2, l_1}(l_1, l_2) = \max_{\mathbf{p} \in l_2} \min_{\mathbf{q} \in l_1} \|\mathbf{p} - \mathbf{q}\|, \tag{3}$$

$$d_{\text{Hausdorff}}(l_1, l_2) = \max(d_{l_1, l_2}, d_{l_2, l_1}). \tag{4}$$

The (ii) Trucco-distance [10] is illustrated in Fig. 1b and described as following:

$$l_{\min} = \min(\|l_1\|, \|l_2\|), \tag{5}$$

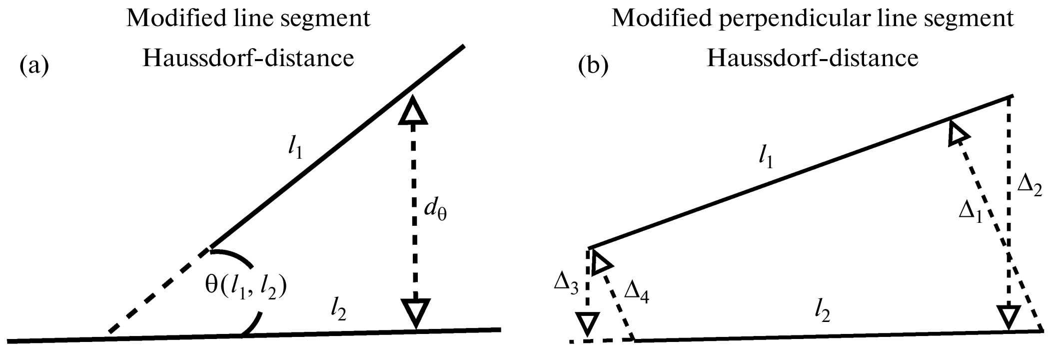

Fig. 2. Visualization of the used quantities of the Modified line segment Hausdorff- and Modified perpendicular line segment Hausdorff-distance with $\Delta_1 = \max(d_{\perp, l_1, 1}, d_{\perp, l_1, 2})$, $\Delta_2 = \max(d_{\perp, l_2, 1}, d_{\perp, l_2, 2})$, $\Delta_3 = \min(d_{\perp, l_1, 1}, d_{\perp, l_1, 2})$, and $\Delta_4 = \min(d_{\perp, l_2, 1}, d_{\perp, l_2, 2})$.

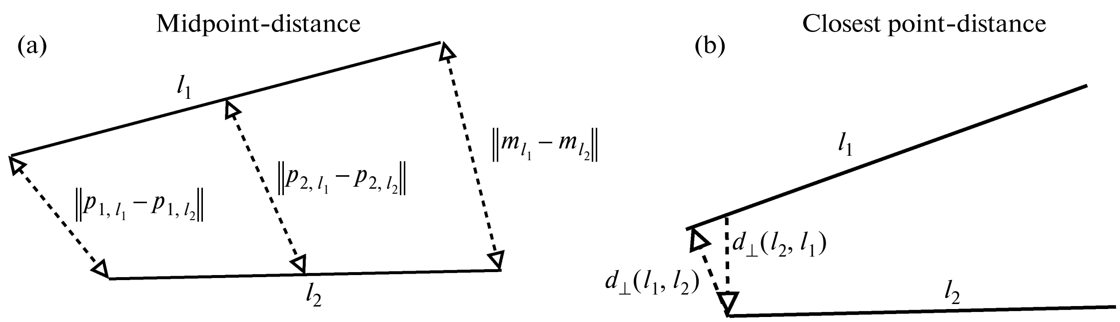

Fig. 3. Visualization of the used quantities of the Midpoint- and closest Point-distance.

$$\begin{aligned}\min(d_{l_1, l_2}) = \min(&\|p_{1, l_1} - p_{1, l_2}\| \|p_{1, l_1} - p_{2, l_2}\|,\\ &\|p_{2, l_1} - p_{1, l_2}\|, \|p_{2, l_1} - p_{2, l_2}\|),\end{aligned} \tag{6}$$

$$d_{\text{Trucco}}(l_1, l_2) = \left(\frac{l_{\min}}{\min(d_{l_1, l_2})}\right)^2. \tag{7}$$

The (iii) Modified line segment Hausdorff-distance [3] is illustrated in Fig. 2a and uses the angle function $\Theta(l_1, l_2) : \mathbb{R}^2 \times \mathbb{R}^2 \mapsto \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ and includes them into

$$d_\Theta(l_1, l_2) = \min(\|l_1\|, \|l_2\|)\sin(\Theta(l_1, l_2)). \tag{8}$$

The (iv) Modified perpendicular line segment Hausdorff-distance [1] is illustrated in Fig. 2b and uses the perpendicular-distance $\mathrm{d}_\perp(l_1, l_2) = s_\perp$ and including them into

$$\begin{aligned}s_{\perp, 1} = \min(&\max(d_{\perp, l_1, 1}, d_{\perp, l_1, 2}),\\ &\max(d_{\perp, l_2, 1}, d_{\perp, l_2, 2})),\end{aligned} \tag{9}$$

$$\begin{aligned}s_{\perp, 2} = \min(&\max(d_{\perp, l_1, 1}, d_{\perp, l_1, 2}),\\ &\min(d_{\perp, l_2, 1}, d_{\perp, l_2, 2})),\end{aligned} \tag{10}$$

$$w_i = \frac{s_{\perp, i}}{(s_{\perp, 1} + s_{\perp, 2})} \text{ with } i = \{1, 2\}, \tag{11}$$

$$d_{\perp\text{mod}}(l_1, l_2) = \frac{1}{2}(w_1 s_{\perp, 1} + w_2 s_{\perp, 2}). \tag{12}$$

The (v) Midpoint-distance is illustrated in Fig. 3a and uses the midpoints of the line segments $m_{l_i} = (p_{2, l_i} - p_{1, l_i})/2$ and includes this into

$$\begin{aligned}d_{\text{midpoimt}}(l_1, l_2) &= \|p_{1, l_1} - p_{1, l_2}\|\\ &+ \|p_{2, l_1} - p_{2, l_2}\| + 3\|m_{l_1} - m_{l_2}\|.\end{aligned} \tag{13}$$

The (vi) Closest point-distance [5] is illustrated in Fig. 3b and described as

$$d_{\text{closesrpoint}}(l_1, l_2) = \min(d_\perp(l_1, l_2), d_\perp(l_2, l_1)). \tag{14}$$

The (vii) *Straight line-distance* uses $\mathrm{d}_1 = \|p_{1, l_1} - p_{1, l_2}\|$, $\mathrm{d}_2 = \|p_{1, l_1} - p_{2, l_2}\|$, $\mathrm{d}_3 = \|p_{2, l_1} - p_{1, l_2}\|$, and $\mathrm{d}_4 = \|p_{2, l_1} - p_{2, l_2}\|$ and includes them into:

$$d_{\text{translation}}(l_1, l_2) = \frac{d_1 + d_2 + d_3 + d_4}{4} - \frac{\|l_1\| + \|l_2\|}{4}, \tag{15}$$

$$d_{ST}(l_1, l_2) = d_{\text{closestpoint}} + 0.25 d_\Theta(l_1, l_2) + d_{\text{translation}}, \tag{16}$$

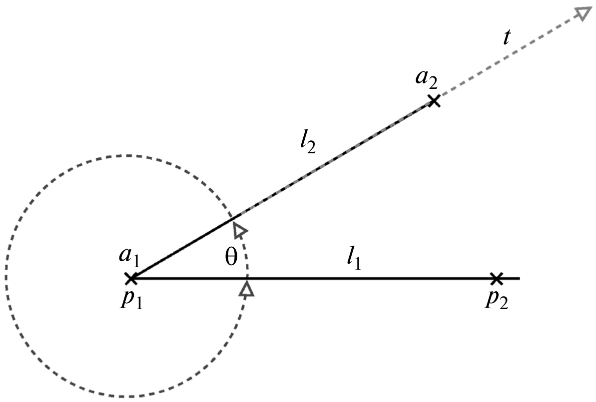

Fig. 4. Process of the evaluation.

$$d_{\text{srtaightLine}}(l_1, l_2) = \min(d_{ST}(l_1, l_2), d_{ST}(l_2, l_1)). \quad (17)$$

3.2. Analytical Evaluation

Distance functions have to distinguish line segments using their parameters which are the position, length and orientation. In general we assume that the metric of the line segments does not influence the measurement of the distance function. Therefore, we choose as measurement unit of centimeters and not pixels, because the number of pixel (resolution) in an image can vary from less than VGA (=640 × 480) up to multiple of Full WD (=1920 × 1080).

Therefore, we generate two line segments l_1 and l_2 of length 100 cm. The line segment l_2 will be rotated around l_1, translated in direction of l_2 and scaled where the point p_1 stays equal. We rotate with $\theta = [0, 2\pi]$, translate with $t = [0 \text{ cm}, 200 \text{ cm}]$ and scale line segment l_2 from [1 cm, 200 cm] (see Fig. 4). In Section 4 the results are presented and discussed.

4. EXPERIMENTS AND RESULTS

The results of the analytical evaluation are shown in the Figs. 5a–5g. Figure 5 clarifies the meaning of translation and rotation for the respective distance function. The Trucco- and the Closest point-distance (Figs. 5b and 5f) are inappropriate to distinguish line segments with different angles. Also, the Trucco-distance is not monotone increasing. Monotonicity is important if the aim is to interpret the score of the distance function as dissimilarity-degree.

The Modified line segment Hausdorff- and the Modified perpendicular line segment Hausdorff-distance (Figs. 5c and 5e) distinguish very well differences of the angles, but are not able to distinguish line segments which are adjoining, what becomes apparent at the angle of 180° where the line segments lie next to each other. The Hausdorff-distance (Fig. 5a) like the Trucco-distance does not increase monotone for the test scenario, but distinguish very well angle- and translation changes. The missing monotony property yields that the distance outputs are not clearly interpretable. The remaining Midpoint- and Straight line-distance (Figs. 5d and 5g) seem quite appropriate to estimate rotation and translation. They are also monotone increasing.

Next to translation and rotation errors, which could be caused by imprecision of the used data or due to segmentation errors in the image, we have to handle

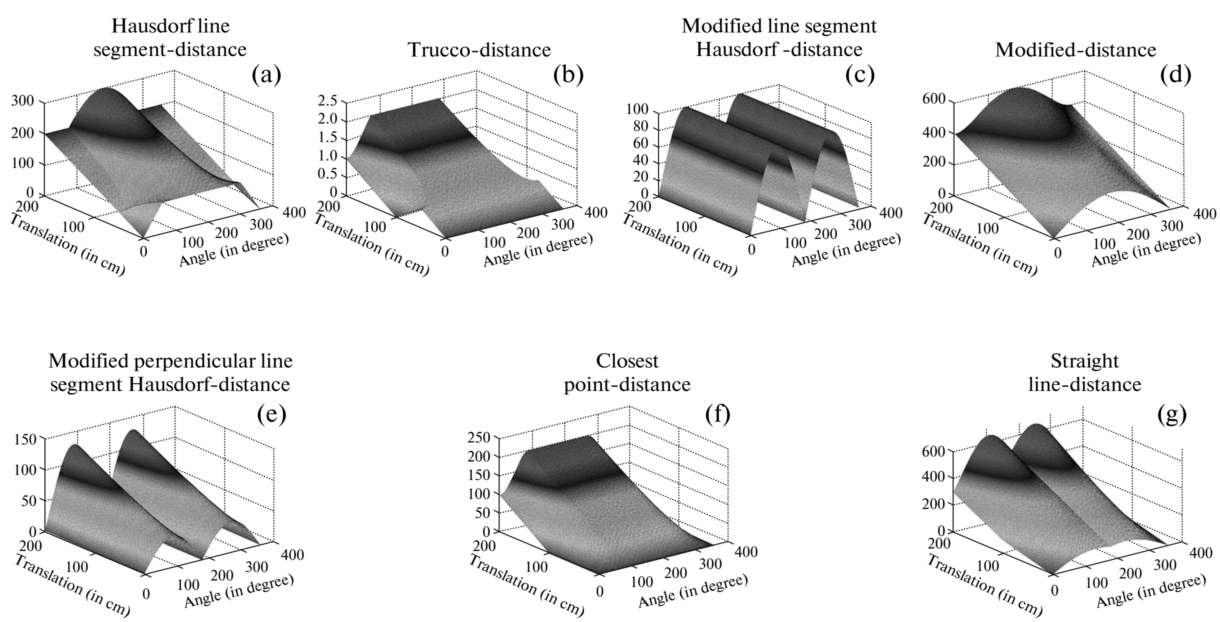

Fig. 5. Distance functions with the angle in degree on the x-axis, the translation in centimeter on the y-axis and the distance on the z-axis.

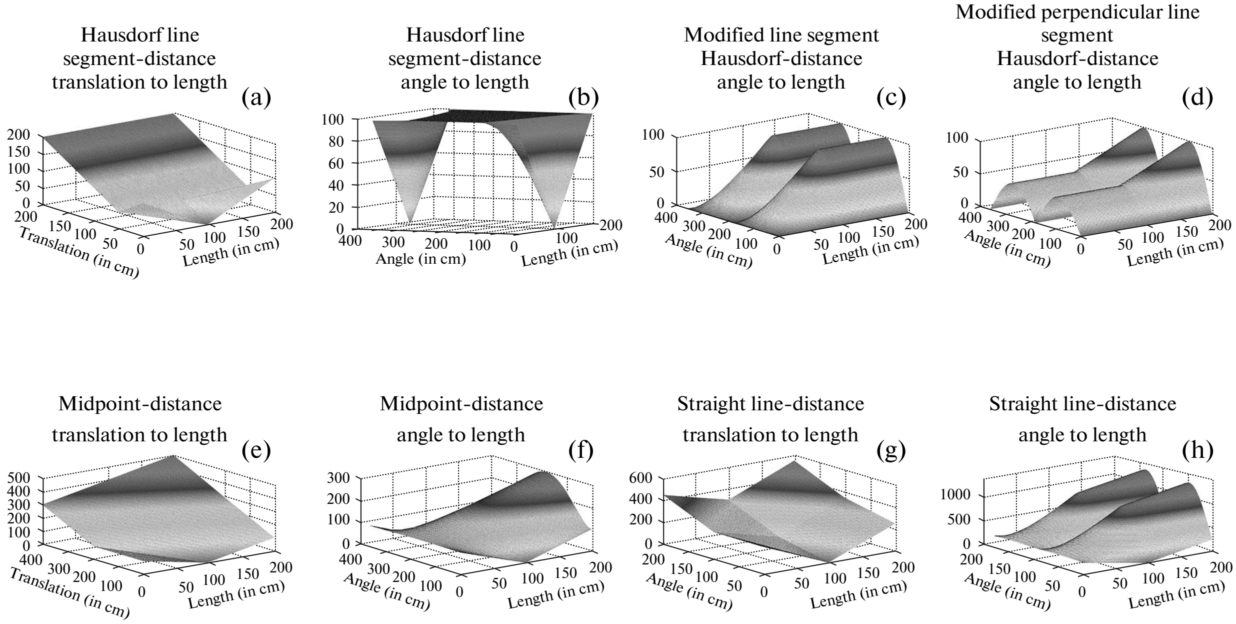

Fig. 6. Distance functions which are affected by the length of the line segments with the length on the x-axis and the angle in degree or the translation in centimeter on the y-axis and the distance on the z-axis. The not plotted value (translation or angle) is set to zero.

incomplete data, which results in differences between the length of the corresponding line segments. Figures 6a–6h show the distance functions which are affected by the length of the line segments.

One distance function which is independent of the line segment length is the closest point-distance. The other one is the Trucco-distance which uses a normalization of the line segment length and is thus also independent of the line segment length. However, the Hausdorff-distance depends on the line length, but from a certain error in translation or angle the length difference is of no consequence (see Figs. 6a and 6b). The Modified line segment Hausdorff-distance uses the shortest line segment length, so that shorter lines segments reduce the error (see Fig. 6c). This behavior could be inappropriate in cases where the length of the line segments covers a broad range.

In extreme case, this could lead to a matching of a very short line segment with a long line segment which is perpendicular to the short line segment. The opposite behavior occurs in the modified perpendicular line segment Hausdorff-distance, there the longest line impacts the score of the distance function (Fig. 6d). Even so, the differences of the line segments lengths are ignored.

We show for the modified Hausdorff distances only the angle plot, because there the length is independent of the translation. The Midpoint-distance and the Straight line-distance are both depending on the difference between the line segment lengths, which is shown in the Figs. 6e–6h. Table gives an overview of the properties of the distance functions.

CONCLUSIONS

We presented six well known distance functions for comparison of line segments and a new one which is variant to scale, translation and rotation. These functions were analytically evaluated by simulated data, so that we were able to stress out the properties of each distance functions concerning angle, translation, and scaling changes. The extracted properties make it now easy choosing an adequate distance function to a specific application.

We present the dependencies of the distance functions to angle-, translation- and scale-/length-differences. The sign $\oplus$ stands for dependent of this quantity, $\odot$ stands for dependent with limitations and $\ominus$ stands for independent of this quantity

	HD	TD	MHD	MPHD	MD	CD	SD
Angle	$\oplus$	$\ominus$	$\oplus$	$\oplus$	$\oplus$	$\ominus$	$\oplus$
Translation	$\oplus$	$\oplus$	$\ominus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
Scale	$\odot$	$\ominus$	$\odot$	$\odot$	$\oplus$	$\ominus$	$\oplus$

REFERENCES

1. Y. X. Supervisor and M. Leung, Proc. 4th LASTED Int. Conf. on Visualization Image and Image Processing (Marbella, 2004), pp. 1–25.
2. M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (San Francisco, 2010), pp. 1696–1703.
3. J. Chen, M. K. Leung, and Y. Gao, Pattern Recogn. **36**, 943 (2003).
4. C. Schmid and A. Zisserman, in Proc. Conf. on Computer Vision and Pattern Recognition CVPR'97 (Washington, 1997), p. 666.
5. T. Werner and A. Zisserman, in Proc. European Conf. on Computer Vision (Springer-Verlag, 2002), Vol. 2.
6. H. Bay, V. Ferrari, and L. J. V. Gool, in Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05) (Washington, 2005), Vol. 1, pp. 329–336.
7. S. Rusinkiewicz and M. Levoy, in Proc. 3rd Int. Conf. on 3D Digital Imaging and Modeling (3DIM) (Quebec, 2001).
8. S. S. Chawathe, in Proc. IEEE Symp. on Intelligent Vehicle (Istanbul, 2007), pp. 1190–1197.
9. H. Alt, B. Behrends, and J. Blomer, Ann. Math. Artificial Intellig. **13**, 251 (1995).
10. E. Trucco and A. Verri, Introductory Techniques for 3-D Computer Vision (Prentice Hall, New York, 1998).
11. J. Witt and U. Weltin, in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IEEE/RSJ, 2013) (Tokyo, 2013).

Stefan Wirtz obtained a diploma in Biomathematics (Dipl.-Math. (FH)) from the University of applied science RheinAhrCampus Remagen in 2008. He is now working for Motec GmbH in the sector of driver assistance development since 2012. Previously, he worked for the Institute of Computational Visualistics in the Working Group Active Vision (Prof. Paulus) at the University of Koblenz-Landau. There he worked as PhD student in the project "Software Techniques for Object Recognition (STOR)" which is funded by the German Research Foundation (DFG). His scientific interests can be associated with the fields of Image Processing, Pattern Recognition and especial the handling of uncertain knowledge.

Dietrich Paulus obtained a Bachelor degree in Computer Science from University of Western Ontario, London, Canada, followed by a diploma (Dipl.-Inf.) in Computer Science and a PhD (Dr.-Ing.) from Friedrich-Alexander University Erlangen-Nuremberg, Germany. He obtained his habilitation in Erlangen in 2001. Since 2001 he is at the institute for computational visualistics at the University Koblenz-Landau, Germany where he became a full professor in 2002. He is head of the Active Vision Group (AGAS). His primary interests are computer vision and robot vision.