Grid Shape Descriptor Using Path Integrals for Measuring Sheet Metal Parts Similarity

Mikko Y. Ylihärsilä[1] and Juha Hirvonen[2]

[1] Seinäjoki University of Applied Sciences, mikko.yliharsila@seamk.fi
[2] Seinäjoki University of Applied Sciences, juha.hirvonen@seamk.fi

Corresponding author: Mikko Ylihärsilä, mikko.yliharsila@seamk.fi

Abstract. In this study, a new kind of grid-based shape descriptor is presented. The new descriptor has properties suitable for searching and retrieving exact duplicates or similar sheet metal parts or other 2D-shapes from the database. The descriptor is defined by integrating CAD-drawing geometries instead of using rasterized images or a limited number of sampled points. The descriptor is very sensitive to the slightest changes in the model. It can be robustly used to quickly find exact duplicates or, vice versa, alarm about small changes in part detail. Search, and retrieval can be done independently for outer and inner geometries or the part as a whole. This allows, e.g., searching parts having similar hole patterns or distinctive cutouts, even if the outer shape is different. Computing the descriptor is fast, and comparison is trivial vector comparison.

Keywords: shape descriptor, shape similarity, sheet metal, CAD
DOI: https://doi.org/10.14733/cadaps.2022.712-721

1 INTRODUCTION

Shape matching and similarity measurement are vital for many applications, and they are well-studied subjects in computer science. Most of the studies focus mostly on 2D images rather than 2D CAD-geometries. However, similar strategies that are used for images can be applied to the parametric curves used in the CAD-geometries. Shape descriptors can be used both for matching shapes and for measuring similarities.

A shape descriptor is a set of numbers that describe a given shape. It should have some essential properties, such as identifiability, noise resistance, invariance to translation, rotation, scale, affine transform and occultation. Furthermore, the descriptor should be compact to store, and computing the difference between the two descriptors should be simple and efficient. [9]

Shape descriptors can be categorized into three main categories: contour-based descriptors, region-based descriptors, and hybrid descriptors. The contours-based descriptors use the contour edges for generating the descriptor e.g., by sampling points and generating height histograms. These descriptors are known to be sensitive to noise in the shape and thus not very accurate. However, their computational complexity is low. Region-based descriptors take into account the boundary and the interior of the shape e.g., creating 2D histogram by assigning pixels into 2D bins. And hybrid descriptors combine two or more techniques. [3]

Currently, the most performant descriptor for the MPEG-7 bulls-eye test [4] is a hybrid descriptor [7], which combines discrete cosine transformation (DCT) [6] and height functions [8] based shape descriptors. DCT captures the shape's spatial information, and the height functions capture the boundary of the shape as sampled points.

In this study, a new grid-based shape descriptor is presented, focusing primarily on sheet metal part geometries. Grid-based descriptors can belong to either contour-based or region-based. Early grid-based descriptors were binary-based matrices [2]. Later, the Blurred Shape Model (BSE) [1] was presented, with a matrix containing gray-level values. Most of the descriptors presented earlier are weak in detecting exact duplicates and small differences or they lack the ability to take into account multiple contours and internal shapes. Comparing e.g., two exact duplicate geometries, but the other one rotated can result in lesser or more significant errors depending on the size of the used descriptor vector or sampling frequency.

The descriptor presented here has some similarities to the Blurred Shape Model, but the descriptor is computed for the precise CAD-geometry instead of the rasterized image. The new descriptor can accurately recognize duplicate geometries and distinguish geometries that have slight differences. Most of the other methods use point sampling with a fixed number of points. The new method uses path integrals instead, which notices every small detail in the geometry, which could otherwise fall between the sample points when using other methods. The new method also allows search queries for the outer and the inner geometries separately or for the whole part.

Chapter 2 describes the exact formulation of the shape descriptor and the concept. In chapter 3, practical aspects are taken into account and covers details about numerical software implementation. In chapter 4, the shape descriptor is tested for sensitiveness to small differences in the geometries and retrieval tests with industrial parts and known benchmark shape database. In chapter 5, conclusions are made about the strong and weak points of the method, and also further improvement ideas are discussed.

2 SMOOTH GRID SHAPE DESCRIPTOR

2.1 Geometry Normalization

The orientation and the size of the geometry are normalized to allow computing comparable shape descriptor matrices. Geometry normalization is done by computing bounding rectangle around the outer geometry of the part. Geometry is rotated by the bounding rectangle's rotation angle $\boldsymbol{\theta}$ and scaled by inverse of bounding rectangle's width $\boldsymbol{w}$ and height $\boldsymbol{h}$, so that the bounding box of the normalized geometry is a unit square with size of $\mathbf{1} \times \mathbf{1}$.

Bounding box can be computed in many different ways. In this study two methods were used, minimum bounding box and bounding box constructed in the angle of principal axis. These methods align bounding boxes in slightly different angles $\boldsymbol{\theta}$. It is beneficial if the bounding boxes are oriented nearly in the same angle for shapes that are thought to be similar.

2.2 Shape Descriptor Matrix

The shape descriptor presented in this study is an $\boldsymbol{n} \times \boldsymbol{n}$ matrix containing positive real numbers. We will call it Smooth Grid Shape Descriptor, later referred to as $\boldsymbol{SGSD}$. $\boldsymbol{SGSD}$ can be considered as a 2D dimensional grid, where the $(\boldsymbol{n}-\mathbf{2}) \times (\boldsymbol{n}-\mathbf{2})$ innermost cell's overall size is unit square. Therefore, each cell has a size of $\frac{\mathbf{1}}{\boldsymbol{n}-\mathbf{2}} \times \frac{\mathbf{1}}{\boldsymbol{n}-\mathbf{2}}$. Scaled geometry will fit inside these innermost cells leaving extra cells outside of the geometry. These extra cells are used to have even weighting everywhere in the geometry. Without the extra cells, the outer geometry would have a smaller impact on the descriptor value.

The value for each element in $\boldsymbol{SGSD}$ (as seen in equation 1) is calculated by integrating each curve in the geometry over the weight function $\boldsymbol{\omega}(\vec{\boldsymbol{p}})$ (as seen in equation 2). $\boldsymbol{n_C}$ is the number of

curves. Each curve is defined as parametric curve $\boldsymbol{r(t)}$, where $\boldsymbol{t}$ is parameter within interval $[\boldsymbol{a}, \boldsymbol{b}]$. Curve's end points can be calculated when function $\boldsymbol{r(t)}$ is evaluated at parameters $\boldsymbol{a}$ and $\boldsymbol{b}$. $\boldsymbol{\acute{r}(t)}$ is the first derivative of the curve.
W is $\boldsymbol{n} \times \boldsymbol{n}$ matrix containing precomputed center positions for each cell in the $\boldsymbol{SGSD}$ grid. These values are used to offset the geometry to the corresponding cell being calculated.

Function $\boldsymbol{\omega(\vec{p})}$ describes the weight of how much the geometry contributes into the cell value in certain position of $\boldsymbol{\vec{p}}$. If the weight function would $\boldsymbol{\omega(\vec{p}) = 1}$, then the integral would result in length of the scaled curve. Parameter $\boldsymbol{\phi}$, which we also refer to as spotsize, defines how far away from the grid cell center point the geometries accumulate the cell value. Each multiple of $\boldsymbol{\phi}$ increases the bottom radius of the weight function by the amount of cell size $\boldsymbol{\frac{1}{n-2}}$**.** The function is set to zero when the point to be evaluated is further from the cell center point than specified by the parameter $\boldsymbol{\phi}$. Otherwise, the function is a cosine function. The cosine function is modified so that the values are always positive values between 0 to 1.

$$\forall(i,j) \in [n \times n]\ SGSD_{i,j} = \sum_{c=1}^{n_c} \int_a^b \omega\left(r_c(t) - W_{i,j}\right) \cdot \|\acute{r_c}(t)\| dt \tag{1}$$

$$\omega(\vec{p}) = \begin{cases} \dfrac{cos\left(\dfrac{\pi \cdot \|\vec{p}\|}{\frac{\phi}{n-2}}\right)+1}{2}, & \|\vec{p}\| < \dfrac{\phi}{n-2} \\ 0, & \|\vec{p}\| \geq \dfrac{\phi}{n-2} \end{cases} \tag{2}$$

Visualization of this function can be seen in figure 1. The leftmost figure represents rectangular geometry, which is integrated over the the weight function $\boldsymbol{\omega(\vec{p})}$**.** The integral is easy to understand as surface area of the projected extrusion. The rightmost figure represents the 8×8 SGSD where weight function is plotted for each cell. In this study spotsize of $\boldsymbol{\phi = 1.5}$ is used as it gives reasonably uniform weight field over the whole grid. Sum of the weight functions is constant on certain cross-sections, similarly as the sum of the three-phase electricity. However, on 2D-plane it doesn't result in perfect constant sum, but very close. Note that in the figure, only the individual weight functions are plotted, not the sum.

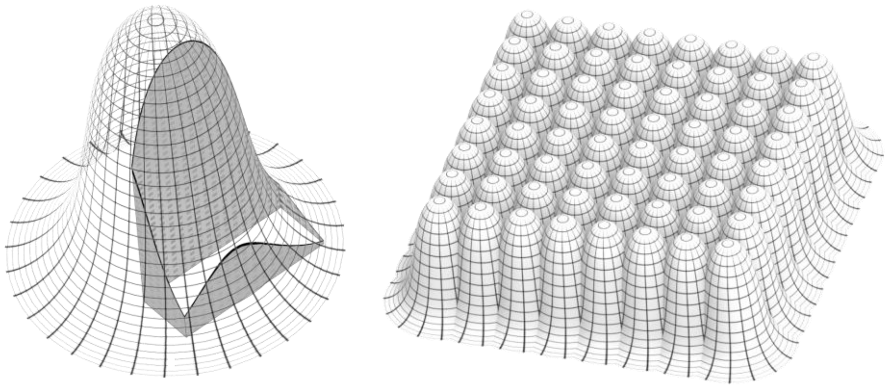


Figure 1: Left – rectangular geometry integrated over $\omega(\vec{p})$. Right - 8×8 SGSD weight functions visualized.

2.3 Measuring Similarity of the Shapes

The length of the shape descriptor is measured using Manhattan distance (as shown in equation 3). Similarly, the difference between the two shape matrices is measured using Manhattan distance (as shown in equation 4). In this study Manhattan distance gave much better results than using

Euclidean distance. Previously also Mahalanobis distance has been used instead of Euclidean distance [5].

$$length(SGSD) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left|SGSD_{ij}\right| \tag{3}$$

$$difference(SGSD_A, SGSD_B) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left|SGSD_{A_{ij}} - SGSD_{B_{ij}}\right| \tag{4}$$

The similarity between the two shape descriptor matrices A and B is defined in equation 5. This equation gives similarity in range 0 to 1. In the equation, the difference is divided by 2 to avoid negative similarities. The difference can be larger than the maximum length of the shape descriptor. This can be illustrated by thinking of a part that contains one hole. If the hole is removed from the part, it will result in difference that is equal to the sum that the hole contributes to the shape matrix. This would yield to inner geometry similarity of 0%. If the hole is moved into a totally separate location in the part, the difference would be twice from the previous and yield to negative -100% similarity. Dividing the difference by 2 will clamp the similarity between 0 and 100%. Now the previous similarities would be 50% and 0%. Also using the maximum length as the divider, the order of the parts becomes insignificant and the condition in equation 6 is fulfilled.

$$similarity(SGSD_A, SGSD_B) = 1 - \frac{difference(SGSD_A, SGSD_B)/_2}{max\left(length(SGSD_A), length(SGSD_B)\right)} \tag{5}$$

$$similarity(SGSD_A, SGSD_B) = similarity(SGSD_B, SGSD_A) \tag{6}$$

Finally, the shape descriptor matrix A is rotated into four possible angles and the orientation with the maximum similarity is chosen. The same is repeated for the mirrored shape descriptor.

3 NUMERICAL IMPLEMENTATION

As there is no closed-form formula for calculating the integral described in the equation 1, an approximation method is used. It is beneficial to keep the original geometry unscaled and take the scaling into account during the integration. This allows to keep the curves in their original parametric presentations, which have known formulas for the curve lengths, at least for the lines and arcs, which are the most typical curve types in sheet metal models.

$\boldsymbol{H(t)}$ is local scaling factor for curve at parameter $\boldsymbol{t}$, where $\boldsymbol{\hat{r}'(t)}$ is unit tangent of curve (as seen in equation 7). Unit tangent is scaled by the width and height of the bounding box and the scaling factor is the norm of the result vector.

$$H(t) = \left\|\left[\frac{1}{w}, \frac{1}{h}\right] \odot \hat{r}'(t)\right\| \tag{7}$$

Equation 1 can now be rewritten into approximation format including the scaling factor $\boldsymbol{H(t)}$ (as seen in equation 8). In approximation formula, the curve is split into $\boldsymbol{n_{C_s}}$ pieces. $\boldsymbol{\Delta s}$ is the curve length of the piece, which can be explicitly calculated for lines and arc, and approximated for more complex curves.

$$\forall(i,j) \in [n \times n]\ \ SGSD_{i,j} \approx \sum_{c=1}^{n_C}\sum_{s=1}^{n_{C_s}} H(t) \cdot \omega\left(r_c(t) - W_{i,j}\right) \cdot \Delta s \tag{8}$$

For the final implementation an $\boldsymbol{8 \times 8}$ matrix was chosen for the shape descriptor with spotsize of $\boldsymbol{\phi = 1.5}$. After some practical testing with 1150 real industrial parts, separate shape descriptors were decided to be used for the outer and the inner geometries. The outer geometry means the outer contour of the part and inner geometry the possible holes and other features in the part. These two descriptors can be stored as one vector containing 128 floating point numbers. Two separate shape descriptors allow searching parts by their outer geometry, inner geometry or by overall geometry.

Sometimes the part family can contain e.g., two parts with same outer geometry, but holes are removed from the second part. This yields to poor overall similarity, but the outer geometry similarity can still be 100%. Sometimes, it can be the opposite: the inner geometry is the driving feature for the part and the outer geometry can vary. Considering for example the three parts in Figure 2, we can compute three different similarities, the best outer geometry similarity, the best inner geometry similarity and the best overall similarity. Parts can be sorted according to the chosen similarity value and the threshold values can be used to filter out e.g., parts whose outer geometry similarity is below 75%.

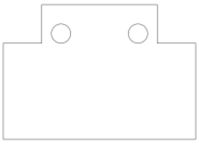
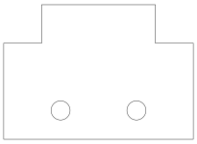
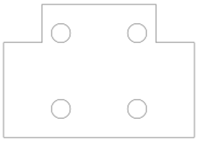
Part	Query Part	Part A	Part B
Image			
Outer similarity	100%	100%	100%
Inner similarity	100%	100%	75%
Overall similarity	100%	82.1%	92.4%

Figure 2: Different similarity values.

Shape descriptor can be visualized as an image by scaling the maximum value of the descriptor to white in the image (as seen in figure 3). This visualization maybe partly misleading as some of the black pixels can still contain information which is truncated when converting the 32-bit floating point values into the 8-bit grayscale image.

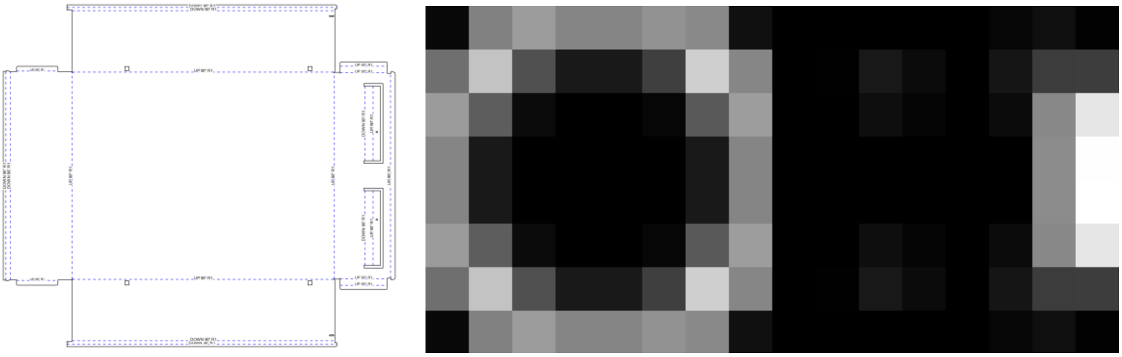


Figure 3: Shape descriptor visualized as image.

4 RESULTS

The practical tests were done with a part database containing 1150 industrial sheet metal parts. Most of the parts were rectangular, containing significant differences in the corner notches. The database contained nearly 100 different part families and many unique parts. Testing was done manually as the dataset was not classified. When querying first 40 unique parts belonging to different part families, the best query matches were every time from the same part family. Approximately 90% of the parts were more or less rectangular looking shapes, but they contained specific details that made them identifiable to some certain part family group.

Unique inner geometry features were queried by drawing a rough sketch of the bounding box of the part and the inner feature to be searched. Figure 4 shows two examples of the correct query results from the database.

Duplicate detection performance was tested with a sample part containing a single hole with a diameter of 5 mm (as seen in figure 5). The part was mirrored and rotated, which should yield identical similarities for both test cases. For both cases, very small errors were measured. The errors can be explained by the numerical noise induced when the geometries were transformed. Then two test cases were done to detect different parts. The hole in the part was moved by 50 µm and 1 µm. The measured errors for these cases were clearly larger, which means that even slight differences in part geometries can be robustly recognized. The scale of the 1µm movement can be illustrated by dividing it by the width of the part, which is 553mm. 1µm is approximately 0.00018% of the part's width. From the logarithmic error scale, it can be seen that e.g., threshold level of 1e-9 could be used to determine exact duplicates. This would leave sufficient margin to prevent either false negative or false positive duplicates.

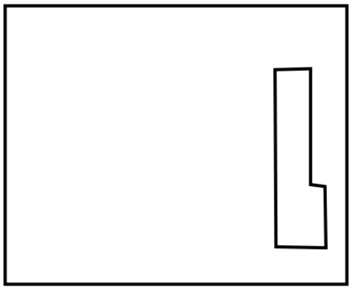
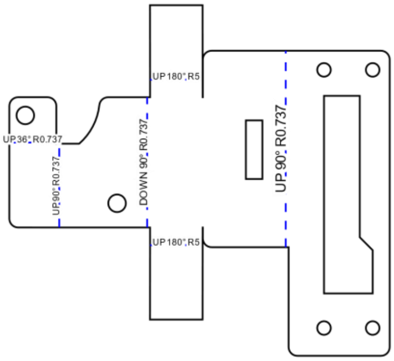
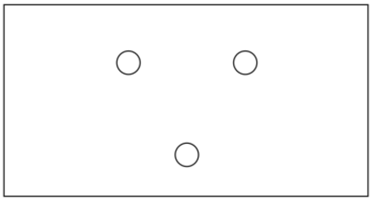
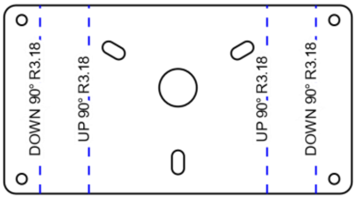
Rough query shape	Best inner geometry match from the database
	 UP 180° R5 UP 36° R0.737 UP 90° R0.737 DOWN 90° R0.737 UP 90° R0.737 UP 180° R5
	 DOWN 90° R3.18 UP 90° R3.18 UP 90° R3.18 DOWN 90° R3.18

Figure 4: Searching by inner features.

MPEG-7 database is a well-known benchmark library for comparing shape descriptors [4]. The library contains 1400 images belonging into 70 different classes. Library images were first converted into vector geometry using a converter and saved as dxf-files.

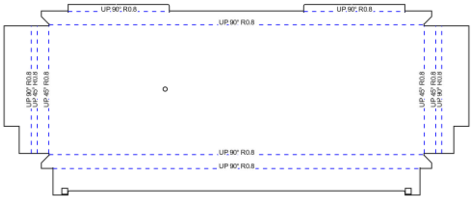
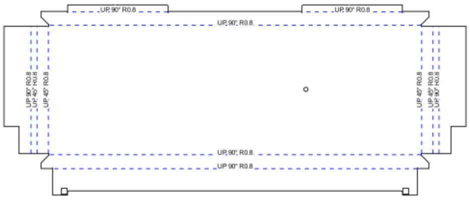
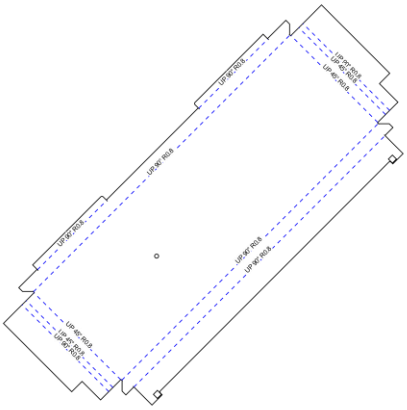
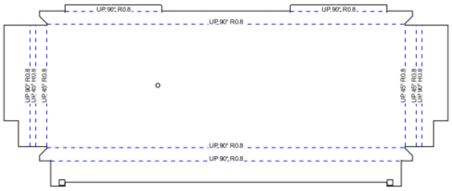
Test case	Image	Error
1 – Original part		0.00 1E-15 1E-13 1E-11 1E-09 0.0000001 0.00001
2 - Mirrored		5.24e-15 1E-15 1E-13 1E-11 1E-09 0.0000001 0.00001
3 - Rotated part		5.86e-15 1E-15 1E-13 1E-11 1E-09 0.0000001 0.00001
4 - Hole moved a) 50 µm b) 1 µm		a) 3.59e-5 1E-15 1E-13 1E-11 1E-09 0.0000001 0.00001 b) 7.19e-7 1E-15 1E-13 1E-11 1E-09 0.0000001 0.00001

Figure 5: Testing duplicate detection robustness.

Two examples of converted images can be seen in figure 6. A bulls-eye test is performed by counting the correct query results from the first 40 results. This number is then divided by 20, which is the maximum number of correct matches. The graph on figure 7 shows the bulls-eye scores for 70 different classes. Average performance was 75.1%, which is similar performance as with Blurred Shape Model (~75% with 8x8 grid) [1]. For comparison, DCT + height functions hybrid [7], has 91% bulls-eye score. Our methods performance in MPEG-7 benchmark was mediocre, but the shapes don’t present typical industrial sheet metal parts and the performance with these shapes was secondary objective.

Computing shape descriptors for MPEG-7 database took 57.6 ms/part on average and the pairwise comparison took 0.235 ms on average using AMD Ryzen 3950X processor. Both computations were executed using a single thread.

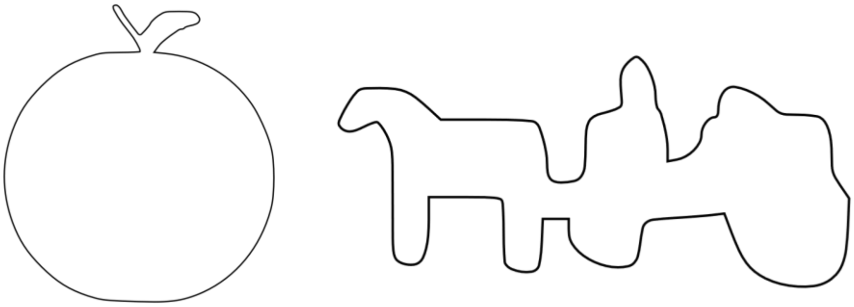


Figure 6: Two examples from MPEG-7 library.

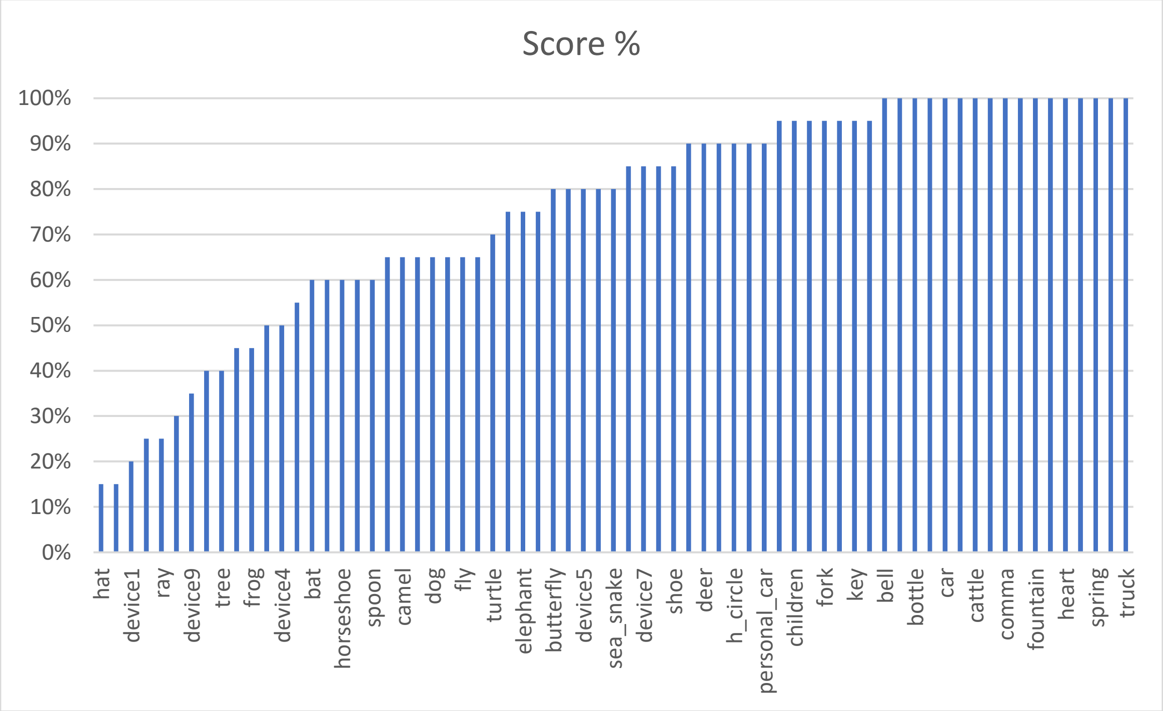


Figure 7: MPEG-7 bulls-eye performance.

For industrial sheet metal parts bounding box was constructed by finding minimum bounding box for the shape. This worked better than principal axis method as usually the bounding box settled into exact same angle for similar parts. Principal axis method was more sensitive to rotate when geometry differed. However, for the MPEG-7 shapes the principal axis method worked significantly better as the shapes had geometries sticking outwards, which resulted radical changes in bounding box angle when using minimal bounding box angle.

5 CONCLUSIONS

The new descriptor's main benefits are the ability to robustly detect duplicates and distinguish almost similar geometries having only a very slight difference. Sheet metal parts belonging to the same part family can have only very slight differences. It is useful to be able to compare them quickly by just comparing the stored shape descriptor values. A new shape descriptor can also be used for searching similar parts according to the outer geometry or the inner geometry separately or for searching for the whole geometry. Searching by inner geometry allows searching for some unique internal shape without knowing the outer geometry details and vice versa.

The presented descriptors' weaknesses are finding shapes that are stretched non-linearly or if the geometry change causes a radical change in the principal axis orientation. Orientation invariance relies solely on the principal axis's orientation and is not the property of the stored shape descriptor.

Performance with generic shapes, e.g., MPEG-7 bulls-eye test, could be improved by combining the presented method with, e.g., height functions [8], which performs better with the single contour geometries having more variety in their shapes in a radial direction. However, for typical sheet metal parts, the benefits of being able to detect small differences and being able to make queries with internal shapes out weighted the property of finding vastly different but for human eyes similar parts.

In future, it would be interesting to improve or experiment an idea of extending the algorithm to integrate the part surface with the similar weight function instead of integrating the curves. This would probably improve the performance with MPEG-7 benchmark and some certain sheet metal parts. However, a combination of both methods would probably work better, as the surface integral is not as sensitive to narrow cuts in the part and vice versa; the curve integral might be too sensitive for noise even if the general shape of the part is very similar.

Mikko Ylihärsilä, http://orcid.org/0000-0002-6540-3527
Juha Hirvonen, http://orcid.org/0000-0001-6704-572X

REFERENCES

[1] Escalera, Sergio; Fornés, Alicia; Pujol, Oriol; Radeva, Petia; Sánchez, Gemma; Lladós, Josep: Blurred Shape Model for binary and grey-level symbol recognition, Pattern Recognition Letters, 30, 2009, 1424-1433, http://doi.org/10.1016/j.patrec.2009.08.001.

[2] Flusser, J: Invariant shape description and measure of object similarity, Proc. 4th International Conference on Image Processing and its Applications, 1992, 139-142.

[3] Kazmi, I. K.; You, L.; Zhang, J. J.: A Survey of 2D and 3D Shape Descriptors, 10th International Conference Computer Graphics, Imaging and Visualization, Macau, 2013, 1-10, http://doi.org/10.1109/CGIV.2013.11.

[4] Latecki, L. J.; Lakamper, R.; Eckhardt T.: Shape descriptors for non-rigid shapes with a single closed contour, Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662), Hilton Head Island, SC, vol.1, 2000, 424-429, http://doi.org/10.1109/CVPR.2000.855850.

[5] Li, Z.; Qu, W.; Cao, J.; Su, Z.; Qi, H.: Geometry-based 2D shape descriptor for retrieval in large database, 2012 IEEE 11th International Conference on Signal Processing, Beijing, 2012, 1096-1101, http://doi.org/10.1109/ICoSP.2012.6491769.

[6] Mannan, A., Babri, H. A.; Saeed, M.: Offline shape recognition using flexible DCT grid, Scientia Iranica, Volume 19, Issue 6, 2012, 1722-1730, ISSN 1026-3098, http://doi.org/10.1016/j.scient.2012.10.031.

[7] Shekar, B. H.; Bharathi, P.: Discrete Cosine Transformation and Height Functions Based Shape Representation and Classification, Procedia Computer Science, Volume 58, 2015, 714-722, ISSN 1877-0509, http://doi.org/10.1016/j.procs.2015.08.092.

[8] Wang, Junwei; Bai, Xiang; You, Xinge; Liu, Wenyu; Latecki, Longin Jan: Shape matching and classification using height functions. Pattern Recognition Letters, 33, 134-143, 2012. http://doi.org/10.1016/j.patrec.2011.09.042.

[9] Yang, Mingqiang; Kpalma, Kidiyo; Ronsin, Joseph: A Survey of Shape Feature Extraction Techniques, Pattern Recognition Techniques, Technology and Applications, Peng-Yeng Yin, IntechOpen, 2008. http://doi.org/10.5772/6237.