Fast Shape Classification Using Kolmogorov-Smirnov Statistics

Alexander Köhler
BTU
Cottbus-Senftenberg
Platz der deutschen
Einheit 1
03046, Cottbus,
Germany
koehlale@b-tu.de

Ashkan Rigi
BTU
Cottbus-Senftenberg
Platz der deutschen
Einheit 1
03046, Cottbus,
Germany
Ashkan.Rigi@b-tu.de

Michael Breuß
BTU
Cottbus-Senftenberg
Platz der deutschen
Einheit 1
03046, Cottbus,
Germany
breuss@b-tu.de

Abstract

The fast classification of shapes is an important problem in shape analysis and of high relevance for many possible applications. In this paper, we consider the use of very fast and easy to compute statistical techniques for assessing shapes, which may for instance be useful for a first similarity search in a shape database. To this end, we construct shape signatures at hand of stochastic sampling of distances between points of interest in a given shape. By employing the Kolmogorov-Smirnov statistics we then propose to formulate the problem of shape classification as a statistical hypothesis test that enables to assess the similarity of the signature distributions. In order to illustrate some important properties of our approach, we explore the use of simple sampling techniques. At hand of experiments conducted with a variety of shapes in two dimensions, we give a discussion of potentially interesting features of the method.

Keywords

Statistical Shape Analysis; Shape Classification; Shape Similarity; Kolmogorov-Smirnov; Hypothesis Testing; Sampling Methods

1 INTRODUCTION

Shape classification is the problem of finding similar shapes among a set of shapes. The corresponding techniques have numerous applications in many fields such as computer vision [1], medical imaging [9, 10] and engineering [20]. Besides of numerous advances in the use of neural networks in the field, there is still a need for approaches that are easy to interpret and give rise to simple and fast computations without the need to deal with intricate neural network architecture, training and data generation issues. In this paper we follow the statistical shape analysis approach, a recent overview may be found in [6]. A more general overview on classic shape analysis methods including some statistical approaches can be found in [3, 4].

Turning to the construction of shape analysis methods in the statistical approach, usually a computational representation of shapes, called a descriptor or signature, is defined that is used for comparison and classification. There is a wide range of possible descriptors which often represent geometric information that can be derived from a given shape [21]. To this end the shape of an object may suitably be described by its boundary. Basically, the descriptors may be classified in two categories: local and global descriptors. Local descriptors are defined for each point of the shapes' boundary and often show the local geometric structure of the shape around the point, whereas global descriptors are defined based on information derived over the entire shape. Examples for some of the most commonly used shape descriptors are centroid distance function, tangent angle, curvature function, area function, triangle-area representation or chord length [19]. In this paper we follow a common terminology in shape analysis by denoting the shape signature as the global representation for comparison, whereas the signature is constructed at hand of local descriptors. Once useful shape signatures have been computed, often a metric is introduced in order to asses quantitatively how close any two given shapes are in terms of the metric distance, see [2] for a recent discussion.

In order to assess shape statistics and employ them for shape comparison, there have also been some efforts to formulate the task as a hypothesis test, see for instance [8, 17]. In this work, we explore the use of the Kolmogorov-Smirnov test in order to assess if two shape signatures are similar. While the use of Kolmogorov-Smirnov statistics appears to have found few applications in image processing, see e.g. [18], it appears that it has not been used for the purpose of shape classification in a similar way as in this paper.

https://www.doi.org/10.24132/CSRN.3201.22

Turning to the construction of statistical shape distribution functions, we build upon the article [15] which is at the same time the most related work to the current paper. It is one of the first works that formulates the shape comparison task in 3D in terms of shape distribution comparison. The comparison of easy-to-compute, statistical shape distribution functions is potentially highly attractive for shape comparison and classification, since this is methodically much simpler than traditional shape matching methods that often rely on pose registration, feature correspondence or model fitting.

In [15], the idea is to represent the signature of an object as a shape distribution sampled stochastically from a shape, measuring in this way global geometric properties of an object. Therefore the authors of [15] have studied several ways to construct shape distributions, including the use of the following descriptors among others: *(i)* the distance between the centroid of a shapes' boundary and randomly selected points of the boundary, as well as *(ii)* the distance between two random points on the boundary. We will also make use of these two basic building blocks in the current work. After choosing the descriptor, [15] calculate the shape signature by stochastic sampling the needed points over an interpolated mesh, followed by a binning procedure of corresponding descriptor values. For shape comparison, in their work dissimilarity measures based on L^p norms have been employed.

Our Contribution. In this paper we adapt the 3D method from [15] to the 2D setting, which allows us to investigate some interesting properties of the proceeding. In doing this, we perform a few technical adaptations, for example we do not interpolate between shape boundary points for stochastic sampling. As our main contribution, we propose how to formulate the shape classification task as a statistical hypothesis test in this setting, making use of the Kolmogorov-Smirnov statistics. This test is efficient to conduct, however, its use appears to be uncommon in shape analysis. It is one of the benefits of the proposed framework that it represents a natural methodical fit to Kolmogorov-Smirnov testing. In order to illustrate some properties of the discussed setting, we consider an adaptive sampling strategy as well as a simple coarsening routine. By adaptive sampling it is possible to emphasize the role of shape-specific points like corners within the statistics, following the classic idea of shape analysis by exploring landmarks, cf. [6]. We exemplify and illustrate our proceeding by several tests with shapes from standard shape datasets.

2 DESCRIPTION OF OUR MODEL

Now we give a detailed account on the used methods for shape classification.

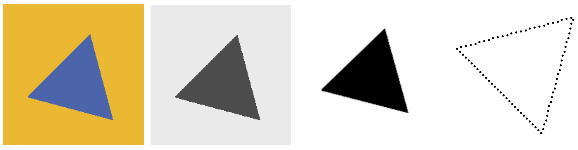

Figure 1: Illustration of the basic process of generating point clouds from color images (left). We take the first color channel (middle left) and transform it into a black and white image (middle right). After that we can obtain a point cloud (right) via the MATLAB routine *bwboundaries*.

2.1 About Shapes

Our shapes are given as a curve in $S \subset \mathbb{R}^2$. Any of these curves is represented as a point cloud consisting of $n \in \mathbb{N}$ points $p_i = (x_i, y_i)$, $i \in \{1, 2, \ldots, n\}$, where x_i and y_i describe the x- and y-coordinates of the i-th point, respectively. Our point clouds are in practice already ordered, so that the points p_i and p_{i+1} are neighbors. We make use of the established neighbourhood relationship within the adaptive sampling scheme.

For illustrating our proceeding, we will consider in this section few shapes taken from the Mendeley 2D geometric shapes dataset [12]. While we discuss the database and its use for experimental validation in more detail later on, let us mention here that typically shape databases contain colour or gray value images of shapes. The Figure 1 illustrates how we obtain from such given images the shapes in terms of point clouds with ordered boundary points.

2.2 Adaptive Boundary Sampling

As indicated in the introduction, for some studies we employ an adaptive sampling of shape boundary points. The aim of the adaptive method we employ is to reduce the number of boundary points in regions that are close to line segments, and to keep salient points of the boundary like at corners or regions with many details. The adaptivity is realised here via the concept of adaptive areas.

The idea of adaptive area comes from [7]. To find out if a point p_i is of interest we calculate the area A_i of the triangle spanned by p_i and its direct neighbours p_{i-1} and p_{i+1}, cf. Figure 2. Then, for an arbitrary triangle the area A_i can be computed via

$$A_i = \frac{1}{2} |\vec{v}_1| |\vec{v}_2| \sin(\gamma) \qquad (1)$$

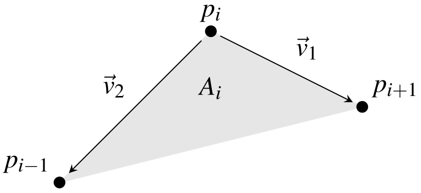

Figure 2: Triangle corresponding to point p_i

where $|\vec{v}|$ denotes the length of the vector $\vec{v}$, and γ the angle enclosed by these vectors. The angle can be computed with

$$\cos(\gamma) = \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1|\,|\vec{v}_2|} \tag{2}$$

with $\vec{v}_1 \cdot \vec{v}_2$ being the scalar product of these two vectors. So the area A_i depends on the angle between the vectors $\vec{v}_1$ and $\vec{v}_2$. If both vectors are parallel the area reaches a minimum. The maximal area is obtained when both vectors are orthogonal to each other. With this in mind we say, that a point p_i is of interest if the area is larger than a user-defined threshold T. If the area is below this threshold we can assume that the point p_i is located at a line-like segment.

To make this approach a bit more robust against scaling we normalize the vectors $\vec{v}_1$ and $\vec{v}_2$. This means we redefine $\vec{v}_k := \vec{v}_k / |\vec{v}_k|$, for $k = 1, 2$. Then the formula for $\cos(\gamma)$ and A_i simplify to

$$\cos(\gamma) = \vec{v}_1 \cdot \vec{v}_2 \quad \text{and} \quad A_i = \frac{1}{2}\sin(\gamma) \tag{3}$$

In the end we neglect all points p_i for which the corresponding triangle has an area smaller that a threshold parameter T, i.e. in case $A_i < T$.

2.3 Shape Descriptors and Signature

In this section we want to describe how to construct our shape signature functions. With these functions we then perform the classification with the Kolmogorov-Smirnov test.

The signature functions we consider are based on shape descriptors. These should be geometrically meaningful as well as fast to compute. Several possible signature functions were presented in [15]. We opt here to adopt the D1 and D2 shape descriptors from the latter work and renaming them into d^1 and d^2, respectively, since both of them are readily interpreted and obtained in low computing time:

$\mathbf{d^1}$ distance For the d^1 shape descriptor we calculate the distance between a sampled point on the shape and a fixed point $p_c = (x_c, y_c)$.

$$d^1(p_i) = \|p_i - p_c\|_2 \quad \text{with} \quad p_c = \frac{1}{n}\sum_{i=1}^{n} p_i \tag{4}$$

with $\|\cdot\|_2$ the Euclidean norm. For the fixed point p_c we, thus, consider here the geometrical centre of a given shape, determined by arithmetic averaging of its boundary points. Thereby n is again the total number of boundary points of the given shape.

$\mathbf{d^2}$ distance For the d^2 shape descriptor we calculate the Euclidean distance between two points p_i and p_j, $i \neq j$, from our shape:

$$d^2(p_i, p_j) = \|p_i - p_j\|_2 \tag{5}$$

For demonstration purposes we focus on the d^1 distance descriptor. The Figure 3 gives an account of the descriptor d^1, when selecting all boundary points of depicted shapes in the order they make up the boundary. At hand of this figure, it is surely easy to perceive how the d^1 descriptor works and that it gives a useful account of a shape. Let us note already here that the descriptor values of triangle and pentagon appear to be very characteristic, while descriptor values of nonagon and circle appear just within a certain narrow band width, which may make these shapes hard to distinguish at hand of d^1.

Having defined our shape descriptors d^1 and d^2, we draw m stochastic samples of them, obtained by random selection of boundary points for the computation of d^1 and d^2 distances, respectively. This means, by taking m random samples of boundary points we can evaluate m times the descriptors d^1 and d^2, respectively, and store the corresponding values as d^k_j with $j = 1, \ldots, m$ and $k = 1, 2$. The corresponding shape signatures are obtained then as the collections $D^1 = (d^1_1, \ldots, d^1_m)$ and $D^2 = (d^2_1, \ldots, d^2_m)$ of these samples. For ease of notation, we consider in the following mainly (i.e. if not stated otherwise) the d^1 distance descriptor, and denote by D_i the signature of the i-th shape obtained by the corresponding D^1 collection of descriptor values.

2.4 Comparing Distributions

Having computed the signatures D_i for each shape, we want to test if they belong to the same distribution.

By default we can not expect that the signatures obtained by the procedure explained in previous section are comparable. Imagine a small and a large triangle. The possible distances from the sample of the small triangle will on average be smaller than the distances from the sample of the larger triangle. To tackle this normalization issue the idea is to scale the different samples D_i to one reference sample D_{ref}. See for example [8] for a thorough discussion of normalization methods. In this work we adopt a *mean-scaling*, which equalizes the means of two given samples. In a slight abuse of notation, we keep D_i for the signature of the i-th shape after normalization in the following.

Now we can test if two signatures D_i and D_j are from the same shape distribution. As indicated, we propose to do this at hand of testing the corresponding hypothesis. There is a large variety of possible hypothesis testing setups in the statistical literature, see for instance, [14] for an overview. One could, for example, test if the medians of the populations from which two or more samples are drawn are equal or not, or one may compare the maximum mean discrepancy of two given samples. However, among the various possibilities, the best fitting test for our circumstances is the Kolmogorov-Smirnov test, since this is designed to evaluate if two

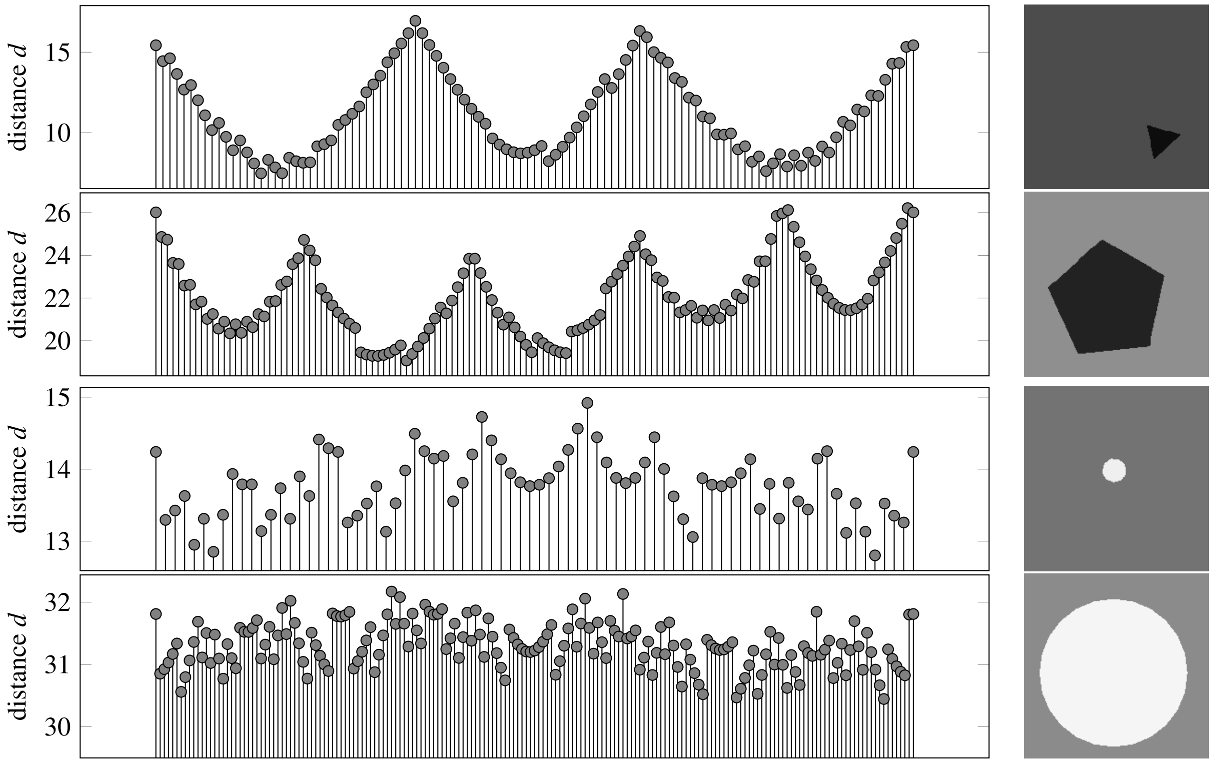

Figure 3: We plotted d^1 distance samples for four geometrical shapes, taking all available boundary points. From top to bottom we see the samples of a triangle, pentagon, nonagon and a circle. Additionally, we show the resource images of the samples on the right side.

samples have the same underlying probability distribution, which means in our case, if they belong to the same shape category.

2.4.1 The Kolmogorov-Smirnov Test

We now recall the setting for the Kolmogorov-Smirnov hypothesis test, cf. [14] for more details.

The Kolmogorov-Smirnov test will give us the value for the hypothesis H which we here cast like

$$H_{i,j} = \begin{cases} 0 & \text{if } F^n_{D_i} = F^m_{D_j} \\ 1 & \text{if } F^n_{D_i} \neq F^m_{D_j} \end{cases} \tag{6}$$

where $F^n_{D_i}$ is the empirical cumulative distribution function for the sample D_i containing n elements. This means H is zero if both samples are drawn from the same distribution and 1 if not. You cannot generally assume that n and m are equal. Even the normalization process does not change this condition.

Thus, in our application, $H_{i,j} = 0$ indicates that the shapes described by the normalized signatures D_i and D_j are of the same category.

Letting for simplicity $D_i = (d_1, \ldots, d_n)$, we can calculate $F^n_{D_i}$ via

$$F^n_{D_i}(x) = \frac{1}{n} \sum_{j=1}^{n} \xi_{(-\infty,x]}(d_j) \tag{7}$$

where $\xi_{(-\infty,x]}(d_j)$ is the indicator function given by

$$\xi_{(-\infty,x]}(d_j) := \begin{cases} 1 & \text{if } d_j \in (-\infty,x] \\ 0 & \text{if } d_j \notin (-\infty,x] \end{cases} \tag{8}$$

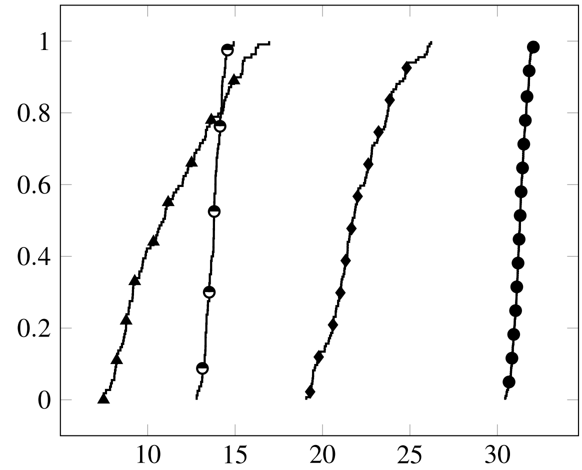

Figure 4: The empirical cumulative distribution function from the samples in Figure 3. The functions from the triangle shape is marked with triangles, the pentagon shape with diamonds, the nonagon with half filled circles and the circle is marked with circles. We can see the different slopes that will make the comparison possible.

Some examples of empirical distribution functions that arise given some shapes and their normalized signatures are presented in Figure 4.

We may then calculate the Kolmogorov-Smirnov statistics in general via

$$S_{n,m} = \sup_x \left| F^n_{D_i}(x) - F^m_{D_j}(x) \right| \tag{9}$$

After [11, eq. (15), section 3.3.1] [16, p. 402, theorem 9.8.3], the hypothesis is accepted if

$$S_{n,m} \leq c(\delta, n, m) = \sqrt{-\ln\left(\frac{\delta}{2}\right)\frac{n+m}{2nm}} \tag{10}$$

where $c(\delta, n, m)$ is the critical value for Kolmogorov-Smirnov test, which depends on the significance level δ and the sample sizes n and m.

Putting all things together, we thus obtain by Kolmogorov-Smirnov test:

$$H_{i,j} = \begin{cases} 0 & \text{for } S_{m,m} \leq \sqrt{-\ln\left(\frac{\delta}{2}\right)\frac{n+m}{2nm}} \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

2.4.2 *Hit Rate*

If we test for the similarity of a signature D_i with a reference signature D_{ref} making use of the Kolmogorov-Smirnov test we get an result $H_{i,\mathrm{ref}} \in \{0, 1\}$. And now we want to quantify the test results when comparing a set of signatures with a reference signature. With N the total number of samples we can define the hit rate in the following manner

$$[0,1] \ni \text{hit rate} = 1 - \frac{1}{N}\sum_{i=1}^{N} H_{i,\mathrm{ref}} \tag{12}$$

This is a natural definition, since if the Kolmogorov-Smirnov test failed we obtain a 1 and if the test succeeded we get a 0. Adding up these results and dividing by the number of samples we compared, we end up with the percentage value of failures. Subtracting this value from 1 finally gives us the value for successful Kolmogorov-Smirnov tests.

We expect that the hit rate is close to 1 if, for example, we compare all triangles with a reference triangle. Conversely, the value should be close to 0 if we compare the triangles with other geometrical shapes.

The hit rates for different shape types and different reference shapes will form a so called hit rate matrix, as seen in discussion of experiments. The rows of this matrix represent the different shape types and the columns represent the reference shape type. The grey value shading of the matrix entries will give a visual account of the hit rate.

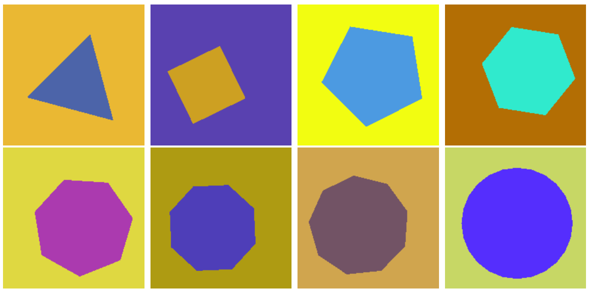

Figure 5: Examples of 8 different shapes one can find in the 2D image shape database [12]. From top left to bottom right we see a triangle, square, pentagon, hexagon, heptagon, octagon, nonagon, and a circle. All images have a resolution of 200 × 200 pixel.

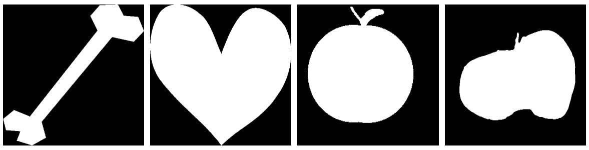

Figure 6: Selection from the MPEG-7 Core Experiment database [13]: pictures from a bone, heart, apple and a shoe.

3 EXPERIMENTS

Before we talk about the experiments we need to consider our database. After that we give a clear account of the general procedure when conducting the experiments and indicate how fast the method works. After that we give a detailed account of the experiments. For this we concentrate first on geometric shapes and proceed after that with a discussion of some additional experiments.

3.1 Database for the Experiments

We use two different databases. The *first database* is a large 2D geometrical shape database [12] containing 90,000 pictures of nine geometrical object. Taken from this we consider images of triangles, squares, pentagons, hexagons, heptagons, octagons, nonagons and circles. Each of these objects is given in total in 10,000 different sizes, rotations and positions.

It should be noted that all images of the triangle pictures the same triangle in different positions, size and rotation. This also applies to the other geometrical shapes. This means, the angles within the shapes do not vary, and they are not transformed in a non-rigid way.

Let us note that for our experiments we ignored the star shapes also contained in the Mendeley database [12] due to some difficulties encountered in the process of point cloud generation. Examples of the shape classes we mentioned used for our experiments can be seen in the Figure 5.

The *second database* we consider is the MPEG-7 Core Experiment database [13]. This database contains binary (black and white) images of variety of objects like apples, bat, Teddy's, birds and many more. We chose the apple, bone, cup, heart and shoe shapes for our experiments, which can be seen in Figure 6.

In the next step we need to generate point clouds out of all the considered pictures of the two databases. In the case of the MPEG-7 Core Experiment database, [5] gives access to the point clouds via github. For the 5 example shapes we chose, the point clouds contain between 102 and 118 points. For each shape there are 20 samples.

For the database of the geometric shapes we needed to build the point clouds by ourselves. The basic steps can be seen in Figure 1. For this process we first need to transform the color images into a binary format. With this in mind, we take the first color channel of a given picture and produce a binary image via the MATLAB function *imbinarize*. Out of this image we can extract a point cloud through the *bwboundaries* MATLAB routine. After that we end up with point clouds with round about 90 up to 700 points for the geometric figures.

The point clouds obtained in the one or other way are already ordered representing the neighbourhood relation. So we do not need to do some further preprocessing.

As one may observe at hand of Figure 7, the considered Mendeley shape database allows to study invariant properties of shape analysis methods that are important for tackling applications. These properties are the invariances with respect to translation, scaling and rotation, which are often considered as the most basic and fundamental properties of useful schemes.

It is quite obvious that the schemes we consider should give by construction invariant results with respect to translations and rotations of a given shape. These invariances are by construction given since the descriptors we study work by considering distances taken from the shape barycenters to boundary points, respectively, distances between points on the shape boundaries. These again are supposed to stay invariant under translation or rotation. The invariance with respect to scaling is addressed by the mean-scaling normalization.

3.2 Procedure and Processing Time

In this section we summarize the general procedure for our experiments, giving a clear account of our method.

1. Generate point clouds for the shapes and define a reference shape.
2. When applying adaptive sampling:
   Compute the adaptive area for every shape using equation (3)
3. Generate descriptor samples for all shapes using the functions (4) or (5).
4. Normalize the samples with respect to the reference shape sample. In the end, all samples should have the same mean value. This gives the shape signatures.
5. Compare the signatures to the reference signature via Kolmogorov-Smirnov test using equations (7), (9) and (11).
6. Compute the hit rate via equation (12).

Let us note that for performing the Kolmogorov-Smirnov test, there exist in several software packages builtin functions like: *kstest2* (MATLAB), *ks.test* (R) or *scipy.stats.ks_2samp* (Python with SciPy).

For all our experiments we used the first shape of a given group as the reference shape. Furthermore, we used the MATLAB builtin function for the Kolmogorov-Smirnov test. By default this test is implemented with a significance level of $\delta = 5\%$.

All calculations were executed with MATLAB R2021a installed on a computer with Intel Xeon W-2145 CPU and 62,4 GiB of memory.

Proof of Fastness

For processing $80,000$ points without adaptive sampling ≈ 50 seconds. That will be $\approx 6.25 \cdot 10^{-4}$ seconds per shape. With adaptive sampling we need ≈ 60 seconds, which will be $\approx 7.5 \cdot 10^{-4}$ seconds per shape. We conjecture that this is very fast and makes the method suitable for potential applications like a first similarity search in a large shape database.

3.3 Experiments and Discussion

We now proceed along the lines of several experiments that allow to point out several properties of our method.

For all the basic experiments discussed first, we followed the steps from the previous section but do not consider adaptive sampling or a coarsening of points yet, which will follow in subsequent experiments.

3.3.1 Basic Experiment with Geometric Figures

In Figure 7 we find the hit rate matrix for the geometrical Mendeley shape dataset. We see the diagonal structure in the hit rate matrix. Let us recall that this dataset contains the eight mentioned shapes in different sizes, rotations and positions. Because of the invariance properties of the method by construction, a strong diagonally dominant entry structure in the hit matrix has also been expected. However, it is surely a very reasonable result with respect to the use of the Kolmogorov-Smirnov test for classification. Let us also note that simple geometries like the triangle, square and heptagon are classified very robustly.

Let us comment on the less pronounced but visible off-diagonal entries here. The more corners are involved

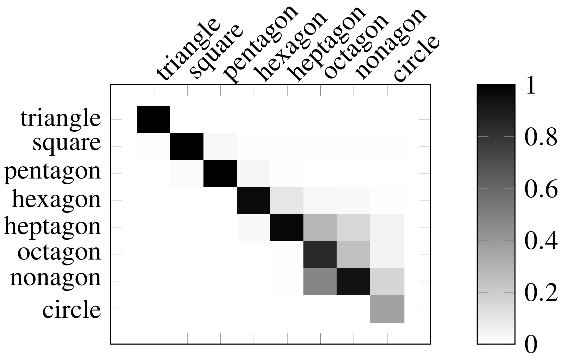

Figure 7: Hit rate matrix for an experiment without adaptive sampling. On the x-axis we wrote the reference shape type and the y-axis we put the different shape types. We see that our basic setup is invariant to rotation, translation and scaling.

in the geometric objects, the more they resemble each other already in the descriptor values, cf. Figure 3 and corresponding discussion. Therefore the basis for performing the Kolmogorov-Smirnov test becomes less discriminative which is reflected in the results.

3.3.2 Basic Experiments with Shapes from MPEG-7

In this experiment, we consider the shapes displayed in Figure 6 supplemented by the cup image as seen in Figure 10. Furthermore, we compare here the two distances d^1 and d^2 introduced in Section 2.3. The results are displayed in terms of the hit rate matrices in Figures 8 and 9.

For these non-geometric objects we may observe that the proceeding appears to be less discriminative in some cases. However, we conjecture that one can mainly notice the properties of the underlying distances used for building the descriptors. Let us note that in [15], the d^2 descriptor gives best results among the tested distances, while by Figure 9 it appears to be less discriminative.

3.3.3 Sampling Experiments

The motivation behind the study of adaptive sampling and a coarsening as we will apply here, can be summarized as follows. By adaptive sampling, salient points of a given shape can be stressed. We compute these points and add them to the list of points from which we sample. By coarsening, we take every second point of the boundary and add it again, which gives a uniform distribution of additional points. This is conceptually in contrast to stressing salient points and shows at the same time the influence of number of points in a shape. The Figure 10 gives an account of resulting points for the cup shape.

The main results of this study are depicted in Figure 11 and Figure 12. We compare the cup using the sampling strategies with the chosen shapes from MPEG-7.

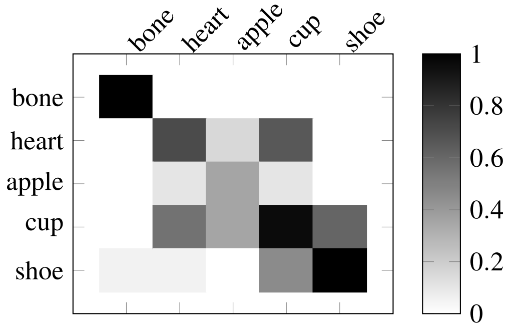

Figure 8: Hit rate matrix for a basic experiment without adaptive sampling. Here we used the d^1 distance function to produce samples with a sample size equal to number of points in the point cloud. The reference shape type is listed on the x-axis and on the y-axis we find the shape type. We notice the diagonal like structure of this plot.

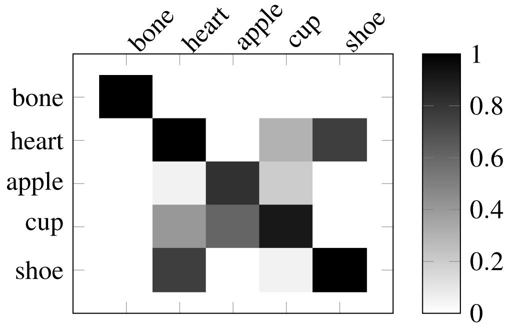

Figure 9: Hit rate matrix for a basic experiment without adaptive sampling, analogously to Figure 8 but using the d^2 distance. We observe more pronounced off-diagonal entries than the hit rate matrix obtained from the d^1 distance.

Exploring just the additional points from adaptive sampling, meaning that salient points are stressed within sampling, we observe that in most cases the accuracy of classification by Kolmogorov-Smirnov test declines. Turning to combination of adaptive and coarse sampling, we observe a similar effect of adaptivity, while

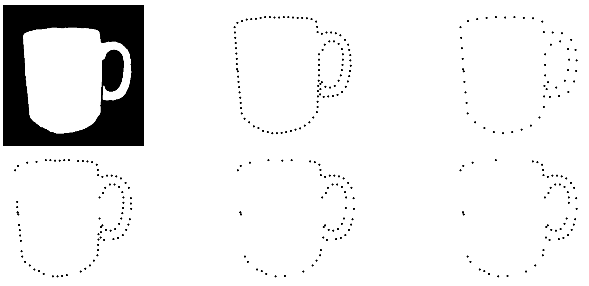

Figure 10: Different variations of the point cloud of the cup shape. **Top Left:** Cup Image. **Top Middle:** Point cloud of this shape. **Top Right:** Every second point. **Bottom Left:** Adaptive area sampling with $T = 0.02$. **Bottom Middle:** $T = 0.04$. **Bottom Right:** $T = 0.06$.

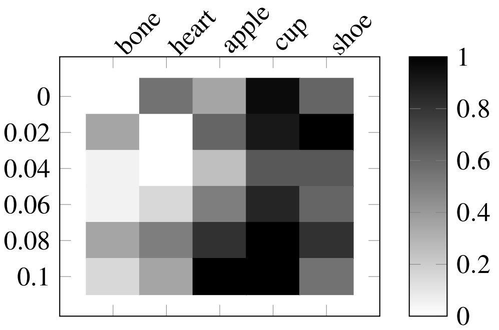

Figure 11: Cup only experiment. Hit rate matrix for the cup, exploring different adaptive sampling parameters. First row coincides with basic experiment.

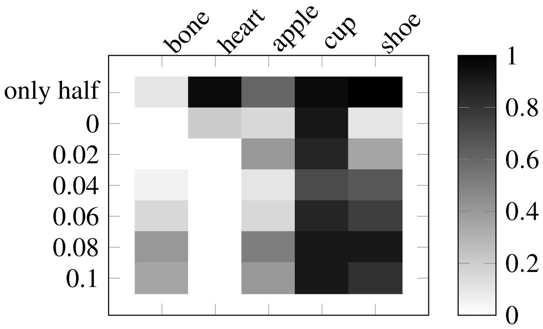

Figure 12: Cup only experiment. Here we combine the adaptive sampling with the coarse shape representation, adding both point sets for sampling to the original shape. Especially by results for choice $T = 0$ (see y-axis) we observe in the second row that adding a coarse shape representation gives favourable results over adding salient points to the sampling set.

adding the coarsened version of the cup gives significant better results than for original cup only.

As a result, one may conjecture that a higher uniform density of shape boundary points is beneficial for the overall strategy. Let us note that this does not imply that offering just the original points several times to the sampling routine is to be favoured. Doing this will just lead to randomness of the complete result so that no useful classification can be performed.

Let us note that we refrain from performing the same study for geometric shapes as in Mendeley dataset. The reason for this may be observed via Figure 13. In geometric shapes with many straight lines, at certain stages (that depend on the way the shape boundaries are determined on the discrete pixel grid) the adaptive routine drops out many boundary points at once. Therefore, we opt to perform the study on sampling on the mentioned subset of MPEG-7.

4 CONCLUSION

In this paper a new method for shape classification is introduced based on Kolmogorov-Smirnov test. To

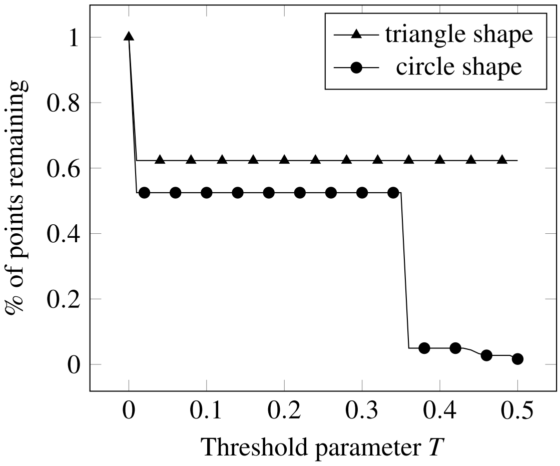

Figure 13: Study of point numbers when applying adaptive sampling, in dependence of area threshold parameter T

do this, we propose an adaptation of the method from Osada and co-authers [15] and showed how to set up a useful framework. One may conclude that the Kolmogorov-Smirnov hypothesis test is well suitable for the task. The whole proceeding is computationally very efficient and may be explored, for instance, for a first quick search for similarities in a shape database.

One of the main construction points is the shape signature. We tested two underlying descriptors, and for future work we may infer that it could be reasonable to combine several descriptors for a joint inference.

By the discussion of the sampling experiments, we may deduce that the complexity of shapes tied to their resolution may be a major factor for the quality of classification results. In future work we aim to explore this point and consider a mathematical investigation of sampling strategies. Furthermore, in future work we will consider other datasets or create our own so that we can transfer our approach to 3D.

5 REFERENCES

[1] J. Arias-Nicolás and F. Calle-Alonso. Poster: A novel content-based image retrieval system based on bayesian logistic regression. *WSCG*, page 19, 2010.

[2] B. S. Bigham and S. Mazaheri. A survey on measurement metrics for shape matching based on similarity, scaling and spatial distance. In *The 7th International Conference on Contemporary Issues in Data Science*, pages 13–23. Springer, 2019.

[3] M. Breuß, A. Bruckstein, and P. Maragos. *Innovations for shape analysis: models and algorithms.* Springer Science & Business Media, 2013.

[4] M. Breuß, A. Bruckstein, P. Maragos, and S. Wuhrer. *Perspectives in Shape Analysis*. Springer, 2016.

[5] A. Carlier, K. Leonard, S. Hahmann, G. Morin, and M. Collins. The 2D shape structure dataset: A user annotated open access database. *Computers & Graphics*, 58:23–30, 2016. Shape Modeling International 2016.

[6] M. Dai, S. Kurtek, E. Klassen, and A. Srivastava. *Statistical Shape Analysis*, pages 1–16. Springer International Publishing, Cham, 2020.

[7] L. H. de Figueiredo. Adaptive sampling of parametric curves. In *Graphics Gems V*, pages 173–178. Elsevier, 1995.

[8] I. L. Dryden and K. V. Mardia. *Statistical shape analysis: with applications in R*, volume 995. John Wiley & Sons, 2016.

[9] H. Hufnagel, X. Pennec, J. Ehrhardt, H. Handels, and N. Ayache. Shape analysis using a point-based statistical shape model built on correspondence probabilities. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 959–967. Springer, 2007.

[10] M. S. Junayed, N. Anjum, A. Noman, and B. Islam. A deep cnn model for skin cancer detection and classification. *WSCG 2021: Full Papers Proceedings*, 2021.

[11] D. E. Knuth. *The Art of Computer Programming, Volume 2 (3^{rd} Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., USA, 1997.

[12] A. E. Korchi. 2D geometric shapes dataset, 2020. Mendeley Data, V1, doi: 10.17632/wzr2yv7r53.1.

[13] L. J. Latecki and R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.

[14] E. L. Lehmann and J. P. Romano. *Testing Statistical Hypotheses*. Springer Texts in Statistics. Springer New York, New York, 2005.

[15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3d models with shape distributions. In *Proceedings international conference on shape modeling and applications*, pages 154–166. IEEE, 2001.

[16] G. R. Shorack and J. A. Wellner. *Empirical Processes with Applications to Statistics*. John Wiley & Sons, Dec. 1986.

[17] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Transactions on pattern analysis and machine intelligence*, 27(4):590–602, 2005.

[18] R. Sun and C. H. Lampert. Ks (conf): a lightweight test if a multiclass classifier operates outside of its specifications. *International Journal of Computer Vision*, 128(4):970–995, 2020.

[19] M. Yang, K. Kpalma, and J. Ronsin. A survey of shape feature extraction techniques. *Pattern recognition*, 15(7):43–90, 2008.

[20] T. Zawadzki, S. Nikiel, and E. Ribeiro. 3-d mesh-classification method based on angular histograms. *WSCG 2013: Poster Proceedings*, 2013.

[21] J. Žunić. Shape descriptors for image analysis. *Zbornik Radova*, (23):5–38, 2012.