GLOBAL SAMPLING OF IMAGE EDGES

Demetrios P. Gerogiannis, Christophoros Nikou, Aristidis Likas

Department of Computer Science and Engineering,
University of Ioannina, 45110 Ioannina, Greece
{dgerogia,cnikou,arly}@cs.uoi.gr

ABSTRACT

An algorithm for sampling image edge points is presented. At first, the image edges are summarized by line segments, which represent the long axis of highly eccentric ellipses. Then, each line segment is partitioned into a number of bins and the point which is closer to the center of the bin is selected. Experiments on widely used databases demonstrate that the proposed method is accurate and provides samples that preserve the coherence of the initial information of the edge map, which is of importance in image retrieval applications.

***Index Terms*—** Sampling of point clouds, image representation, image retrieval.

1. INTRODUCTION

As modern image analysis and computer vision algorithms have become more complex requiring a large number of operations and the the data to be processed are big, a preprocessing step is a necessary task that may assist toward efficient and fast processing. In many cases, that step involves sampling an original image or it edge map (e.g. computation of the vanishing point) in order to keep a fraction of points that describe with fidelity the initial information. More specifically, in image processing, this leads to edge pixel sampling so as to extract the eventually hidden patterns (e.g. object contours) inside an initial observation so that the result is as close as possible to the observation.

The most straightforward approach is to apply random sampling, which assumes that the edge points are observations of a random variable that follows a specific distribution. As soon as we model that distribution, point sampling is augmented to sampling observations from a known distribution. In the simplistic random sampling, it is assumed that the original set follows a uniform distribution. A more advanced, but notoriously time consuming probabilistic model is Monte Carlo sampling [1]. J. Malik independently proposed contour sampling in [2] to apply it to an object retrieval algorithm [3]. Initially, a permutation of the points is computed and a large number of the samples is drawn from that permutation. Then iteratively, the pair of points with the minimum pairwise distance is detected and one of them is kept as a valid sample. This process is iterated until the desired number of samples is reached and it ensures that points from image regions with large density will be part of the final data set.

In [4], the fast marching farthest point sampling method is introduced for the progressive sampling of planar domains and curved manifolds in triangulated point clouds or implicit forms. The basic idea of the algorithm is that each sample is iteratively selected as the middle of the least visited area of the sampling domain. For a comprehensive review of the method, the reader is also referred to [5].

The Fourier transform and other 2D/3D transforms have been applied for describing shape contours for compression purposes. For example, in [6], the idea is to warp a 3D spherical coordinate system onto a 3D surface so as to model each 3D point with a parametric arc equation. However, this method demands an ordering of points and cannot model clouds of points, where more complex structures, like junctions and holes, are present.

A framework for shape retrieval is presented in [7]. It is based on the idea of representing the signature of each object as a shape distribution sampled from a shape function. An example of such a shape function would be the distance between two random points on a surface. The drawback of the algorithm is that the number of initial points has to be relatively small for the method to be fast and efficient. This may lead to a compromise between the number of points of the sample and the information loss. Moreover, the efficiency of the method highly depends on the presence of noise.

This type of edge sampling is a preponderant step before other algorithms are applied. This is the case in [3], where sampling reduces the amount of data for object recognition and in [8], where a shape classification algorithm necessitates a small number of samples to reduce its complexity. Hence, the quality of sampling may affect the final result if the resulting point set does not preserve the coherence of the initial information. Considering also that most of these algorithms demand large complexity in terms of resources (i.e. memory allocation) in order to extract complex features that discriminate better the various data, one may come to the conclusion that sampling may be a very crucial step.

In this work, we propose an algorithm for fast, accurate and coherent sampling of image edge maps. The procedure consists of two steps. At first, the image edges are summarized by a set of line segments, which reduces the initial quantity of points but accurately preserves the underlining information contained in the edge map. Then, based on the ellipse-based representation, a decimation of the ellipses is performed and samples are drawn according to their location on the long ellipse axis.

2. SAMPLING IMAGE EDGE MAPS

The first step of the algorithm is accomplished using the Direct Split and Merge Method (DSaM) [9], where the edges are represented by the long axis of highly eccentric ellipses. The main idea is that a set of points that are collinear, are characterized by a covariance matrix with one very large eigenvalue and one very low (ideally equal to zero) eigenvalue. In any other case, the degree of non linearity of the points may be represented by the minimum eigenvalue. Also, if there is no large gap between collinear points we may assume that they belong to the same line segment. Based on these remarks, the iterative direct split-and-merge algorithm [9] is executed as follows: at first, it considers that all points belong to a single line segment and performs iterative splits, producing a relative large number of highly eccentric ellipses modelling the point cloud. Then, in order to reduce the complexity of the model, iterative merges of adjacent highly eccentric ellipses are performed subject to the constraint that the resulting ellipses are still highly eccentric and the corresponding sets of points assigned to these ellipses are tightly distributed (i.e. there are no large gaps between successive points).

An example of some steps of the execution of the DSaM algorithm on an edge map of a natural image from the database presented in [10] are shown in Fig. 1. The contribution of DSaM is to model the local manifold by fitting line segments. Experiments have shown that the DSaM provides better models compared to other line fitting algorithms.

Assume now that the goal is to sample the set of points that are presented in Fig. 2 and keep only $Q\%$ of them. The black dots represent the original points. This may be considered as the output of the DSaM algorithm [9]. More specifically, these points lie on the long axis of a highly eccentric horizontal ellipse. The axis is shown in red. In order to approximate the local point distribution, a histogram is computed with a number of bins equal to $Q \times L$, where L is the number of initial points in the set to be sampled. Then, we represent each bin by its mean value, which under e.g. Gaussian assumption it is the geometric center of the points in the bin and we select in each bin the point that is closer (in terms of Euclidean distance) to this geometric mean. By repeating the procedure for each line segment produced by the application of the DSaM algorithm we are able to sample the original point cloud.

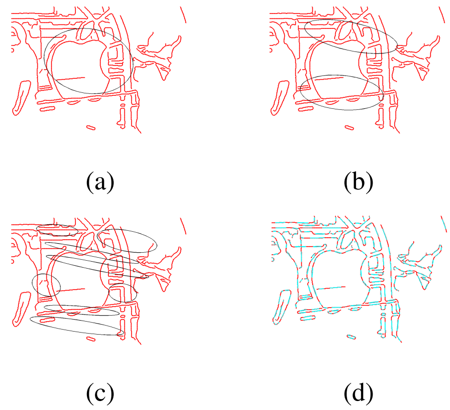


Fig. 1. Steps of the DSaM algorithm [9] on the edgemap of a natural scene image. (a) First step, (b) second step, (c) fourth step, and (d) the final line segments after merging.

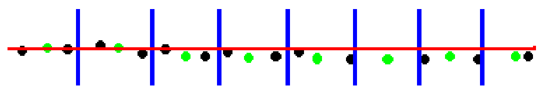


Fig. 2. An example of the sampling process. The black points represent the original set of points, while the red line is the is their *summary* computed by DSaM [9]. The vertical blue lines depict the limits of the histogram bins. The green points are those selected to represent the sampled set because they are closer to the mean value of the bin. The figure is better seen in color.

It may be easily understood that the efficiency of the approach is highly related to the correct determination of a model approximating the local manifold of the point set. The larger the deviation of the model from the local manifold becomes, the less accurate is the sampling method. This is true as the model fails to compute the histograms correctly and therefore to establish accurately the bin centers. Consequently, the selected samples will be less representative of the distribution of the initial edges. For that reason, we relied on the DSaM algorithm which may accurately describe the edge map.

An important issue of the sampling algorithm is the value of the sampling frequency Q, that is how densely should we sample? Moreover, the number of samples should vary locally with respect to the number of image edges present in an image region. To this end, based on the clustering of points to highly eccentric ellipses, we propose to select the number of samples to be equal to δ times the number of points that are present in the mostly populated ellipse, where $\delta \geq 1.0$. This guarantees that, highly concentrated image regions will be more densely sampled but also that sparse regions should always have some representatives as they have already been assigned to an ellipse. This is in contrast to random or even Monte Carlo based sampling where sparse areas may have no representatives in the sampled data set.

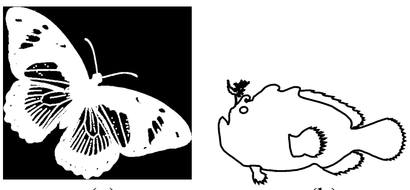
In other words, the sampling rate is computed by

$$Q = \delta \frac{R}{L}, \tag{1}$$

where R is the maximum number of members in the clusters, L is the total number of points in the original set and $\delta \geq 1.0$ is a real positive number. The larger the value of δ is, the more samples we get, and thus the closer to the initial set our sampling result is. In other words, the estimation of the p-value of parameter δ is a compromise between the quality of the result and its complexity.

3. EXPERIMENTAL RESULTS

Two widely used datasets containing shapes were employed in our experiments. The MPEG-7 dataset [11], contains a collection of contours belonging to 70 categories, with each category containing 20 members. The Gatorbait dataset [12], is a collection of 38 shapes of various fishes, belonging to 8 categories. Fig. 3 shows representative images from each dataset. The edges were extracted with the Canny edge detector [13]



(a) (b)

Fig. 3. Some representative images from the datasets used in our examples. (a) A butterfly from the MPEG7 dataset [11], (b) the shape of a fish from Gatorbait dataset [12].

and the coordinates of the edge pixels were used as input to our experiments.

A minimum description length (MDL) approach [1] is adopted to compute the value of δ. We define:

$$\Phi(\delta) = \mathcal{D}(\downarrow(X_{or}, \delta), X_{or}) + \lambda |\downarrow(X_{or}, \delta)| \tag{2}$$

where X_{or} is the original set of points, $\downarrow(X_{or}, \delta)$ is the output of the sampling process applied to set X_{or} with the sampling rate computed by (1), $|\cdot|$ denotes the cardinality of the corresponding set and $\mathcal{D}(P, Q)$ is the Hausdorff distance between the set of points P and Q.

In order to learn parameter δ, we randomly selected 119 images from our dataset. The DSaM sampling method was executed for various values of the parameter δ in the interval $[1.0, 3.0]$ and $\delta = 1.6$ minimized $\Phi(\delta)$, which was used in our experiments (Figure 4).

The efficiency of our method was evaluated by comparing it to widely used methods such as the sampling scheme proposed by Malik [2], Monte Carlo sampling and simple random sampling. In order to quantitatively measure the fidelity

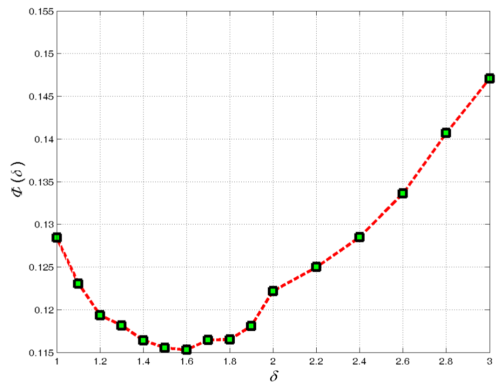


Fig. 4. The value $\delta = 1.6$, which minimizes $\Phi(\delta)$ was used in our experiments.

of the sampled point set to the original one we used the Hausdorff distance between the aforementioned point sets. The rational is that the smaller the $D(X, Y)$ becomes, the closer the sample is to the initial data. This concept may be considered as a try to minimize the distortion-compression rate. In other words, we wish to sample a set of points (compress) by keeping the information loss small (distortion). Moreover, to establish a common baseline, we used the same sampling rate for all of the compared methods, which is the one described in the previous section. In order to avoid any possible bias, we also tested smaller sampling rates for the other methods. the idea was to explore whether they produce better results, in terms of similarity with the original shape using these smaller sampling rates. However, the results proved that by decreasing the sampling rate the results become poorer for the other methods.

The overall results are summarized in Table 1. As it can be observed, our method provides better results in all cases with regard to all of the compared methods. Representative results on Gatorbait [12] dataset are demonstrated in Fig. 5. The reader may observe that our method manages to preserve better the details of the original set, as it produces more uniform results and thus the distribution of the points in the sampled set is closer to the original.

In a second set of experiments we examined the improvement of the result of a shape retrieval algorithm that includes a sampling preprocessing step. This is a very common problem in computer vision and image analysis and much research has been performed in this field. We focused on the pioneering algorithm introduced in [3] and explored the improvement of the detection rate (*Bull's Eye Rate*) by applying various sampling methods including ours.

The overall evaluation is presented in Table 2. It may be seen that the proposed method improves the retrieval percentage . In case of the Gatorbait dataset [12] the improvement of the Bull's Eye Rate is around 2.5% with respect to the second method.

Table 1. hausdorff distance between the and the sampled sets using different sampling methods.

MPEG7 [11] (70 shapes)

Algorithm	**mean**	**std**	**min**	**max**
Proposed method	0.00	0.02	0.00	0.29
Malik [2]	0.00	0.05	0.00	1.10
Monte Carlo	0.03	0.32	0.00	6.21
Random Sampling	0.01	0.19	0.00	3.31

GatorBait100 [12] (38 shapes)

Algorithm	**mean**	**std**	**min**	**max**
Proposed method	0.21	0.04	0.05	0.35
Malik [2]	0.30	0.11	0.11	0.51
Monte Carlo	3.08	3.27	0.61	17.02
Random Sampling	1.09	0.38	0.51	1.88

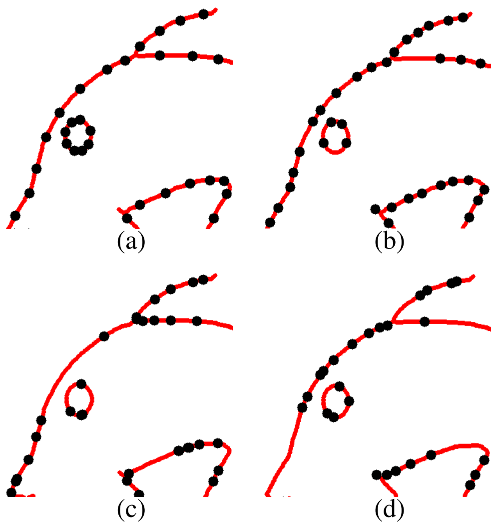


Fig. 5. Representative results of sampling of the Gatorbait dataset [12]. Details of the upper left part of a fish contour. Sampling with (a) the proposed method, (b) the method of Malik [3], (c) Monte Carlo sampling and (d) Random sampling.

In order to measure the similarity between two shapes we adopted the χ^2 distance between shape contexts, as explained in [3]. However, in this problem, a more informative index should be applied to take into consideration the deformation (e.g. registration) energy that is demanded so as to transform one edge map onto the other. Yet, as we wish to investigate the improvement that our method provides in terms of similarity between samples and original signals, we opted not to compute the related parts of the similarity metric in [3]. Moreover, to speed up our experimental computations, we opted not to use a dynamic programming approach to guarantee a one-to-one matching. Instead, we assigned each point from one set to each closest in the other and computed the cost of this assignment in terms of corresponding histogram distances. By repeating that process for all points and summing the related distances, we computed the total distance between two shapes. These remarks, explain the differences in Bull's Eye Rate index computed for the MPEG7 dataset, compared to the one provided in [3].

Since a crucial step of the proposed method is the accurate manifold detection, we compared our sampling method with a widely used method for manifold detection, namely Locally Linear Embedding (LLE) [14]. The reader may observe that LLE does not provide accurate results, since it fails to model the various inner structures and junctions that are present in the experimental data (Table 2).

Table 2. Bull's Eye Rates for the retrieval of sampled sets using different sampling methods.

MPEG7 [11] (70 shapes)

Algorithm	**Bull's Eye Rate**
Proposed method	65.40%
Malik [2]	64.96%
Monte Carlo	50.71%
Random Sampling	57.86%
LLE	53.81%

GatorBait100 [12] (38 shapes)

Algorithm	**Bull's Eye Rate**
Proposed method	96.57%
Malik [2]	93.89%
Monte Carlo	77.69%
Random Sampling	91.11%
LLE	93.98%

4. CONCLUSIONS

A method for sampling image edge points was presented, which is based on an unsupervised fitting of line segments. The experimental results demonstrated that that our method provides a better result regarding the similarity between the initial and the sampled sets. Also, shape retrieval indices were improved by the proposed algorithm. Finally, the method can be easily modified to handle $3D$ points. In that case, instead of computing the $1D$ histogram over the principal axis of each line segment, the analogous $2D$ principal plane should be computed.

5. REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, Dordrecht, The Netherlands, 2006.

[2] Berkeley University, "Matching with shape contexts," `http://www.eecs.berkeley.edu/`

Research/Projects/CS/vision/code/sc_demo/, Accessed on January, 2014.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1–19, 2002.

[4] C. Moenning and N. A. Dodgson, "Fast Marching farthest point sampling," in *EUROGRAPHICS*, 2003.

[5] A. M. Bronstein, M. M., M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*, Springer-Verlag, New York, 2008.

[6] J. J. Chen, C. C. Chiang, and D. W. Lin, "A generalized 3d shape sampling method and file format for storage or indexing," in *International Conference on Image Processing (ICIP)*, 2000, vol. 2, pp. 780–783.

[7] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.

[8] J. Wang, X. Bai, X. You, W. Liu, and J. Latecki, "Shape matching and classication using height functions," *Pattern Recognition Letters*, vol. 33, pp. 134–143, 2012.

[9] D. Gerogiannis, C. Nikou, and A. Likas, "Modeling sets of unordered points using highly eccentric ellipses.," *EURASIP Journal on Advances in Signal Processing*, , no. 11, 2014.

[10] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Object detection by contour segment networks," in *European Conference on Computer Vision*, 2006, pp. 14–28.

[11] Temple University, "College of science and technology, mpeg-7 dataset," http://www.dabi.temple.edu/~shape/MPEG7/dataset.html, Accessed on January, 2014.

[12] University of Florida, "Gatorbait 100," http://www.cise.ufl.edu/~anand/publications.html, Accessed on January, 2014.

[13] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.

[14] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.