



SHAPE DESCRIPTOR FOR OBJECT CLASSIFICATION

Oleksandr Koriukalov, Vasyl Tereshchenko

Taras Shevchenko National University of Kyiv, Kyiv, Glushkova st, 4d, 83000, Ukraine,
karia@bigmir.net, www.univ.kiev.ua

Paper history:

Received 15 March 2018

Received in revised form 13 February 2019

Accepted 15 June 2019

Available online 30 June 2019

Keywords:

object classification;
shape descriptor;
computer vision;
contour smoothing.

Abstract: In this article we propose an efficient modification of a previously published shape descriptor, which is fast and simple to compute. Proposed descriptor was developed for object classification and should be used with classifiers: SVM, KNN, etc. Object classification is a common task of computer vision, which has many applications in different areas: computer intelligence, robotic vision, smart cameras, autonomous driving, etc. Because the properties of objects are largely determined by their geometric features, shape analysis and classification are essential to almost every applied scientific and technological area. Main steps of the proposed algorithm are as follows: to find the object bounds; to smooth the bound contour using extremes based approach (if needed); to find side contour feature for each of N object rotation; to gather common feature vector; to classify object contour, using pre-trained classifier (SVM, KNN). In this work, in addition to method modifications (saving object proportions, rotation invariance, applying KNN classifier), we provide a wide comparison of our algorithm with existing approaches. The described method provided state-of-the-art performance on 100 leaves and Mpeg7 datasets, and showed good results on our own Mushroom dataset separately or together with texture or color based features.

*Copyright © Research Institute for Intelligent Computer Systems, 2019.
All rights reserved.*

1. INTRODUCTION

The shape classification problem can be defined as follows: object shape is given, for example, image segmentation, and assign it to one of the predefined classes. We assume an existing set of labeled objects (training set), the labels for the training set are assigned by a human object. We use binary image (mask) for shape definition.

Today computer vision tasks become more and more popular. Object classification is a common computer vision task, which has many applications in different areas: autonomous driving, geometry, optimization, computer intelligence, robotic vision, cognitive vision, statistics, biological vision, smart cameras and so on. Computer vision algorithms are increasingly used in real life, so there is a growing need for descriptors that are fast to compute, fast to match, memory efficient and exhibit good accuracy. Because the properties of objects are largely determined by their geometric features, shape analysis and classification are essential to almost

every applied scientific and technological area. In this work we introduce efficient modification of a previously published shape descriptor, which is fast and simple to compute [13], and should be used with classifiers (for example SVM, KNN). Also we provide wide comparison of our algorithm with existing approaches. Described method provided state-of-the-art performance on 100 Leaves Plant Species and Mpeg7 datasets, and showed good results on our own Mushrooms dataset.

The paper is organized as follows: Section 2 contains related work. Section 3 contains wide algorithm description. In Section 4 we present experimental results and accuracy comparison, we make conclusions in Section 5.

2. RELATED WORK

Here we will describe existing shape descriptors and algorithms for shape classification. Article contains a wide comparison of proposed algorithm with existing approaches.

For today many approaches to image classification have been developed. Scientists and practitioners have made great efforts in developing advanced classification approaches and techniques for improving classification accuracy. In the majority of image classification algorithms, two main steps can be distinguished: extraction of features for constructing object descriptor and classification of the descriptor by the classifier. Today, many descriptors have been invented to describe the object: Local Binary Pattern (LBP), Scale-invariant feature transform (Sift), Histogram of Oriented Gradients (HOG), Bag of colors and others. In [21] comprehensive survey is presented. Most of the mentioned descriptors take much computational time, and do not work on binary object shape, that is why, there is a need in constructing fast and efficient shape descriptor.

Contour coding algorithms are often applied for representing object shape. The first work in this direction was presented in [15], they attempted to facilitate geometric configuration analysis and manipulation by means of a digital computer. Article [16] presents a new algorithm for converting a binary image into chain codes using its run-length codes, and contains a wide problem definition. The basic idea of conventional chain-coding algorithm is to follow boundary pixels by convolving a 3×3 window with the image and to sequentially generate chain codes. The proposed algorithm has two phases: run-length coding and chain-code generation. In [3] differential chain code histogram is proposed for classification hand-written digits, other chain coding variations are used for representing object contour in [2, 17].

The most fundamental work devoted to shape classification is presented in [18]. It shows the most advanced imaging techniques used for analyzing general biological shapes, such as those of cells, tissues, organs, and organisms; offers techniques that can be used in any computer vision application such as optical character recognition and face recognition; contains chapters on 3D shape characterization, shape analysis and data mining, dynamic shape analysis, and structural shape recognition.

In [7] they focused on mid-level modeling and introduced a new shape representation called Bag of Contour Fragments (BCF) inspired by classical Bag of Words (BoW) model. In BCF, a shape is decomposed into contour fragments each of which is then individually described using a shape context descriptor, and encoded into a shape code. Compact shape representation is built by pooling shape codes from the shape and requires an efficient linear SVM classifier.

An expectation-maximization (EM) [6] approach

is applied to separate a shape database into different shape classes, simultaneously estimating shape contours, that best exemplify each of the different shape classes. They employ the level set function as the shape descriptor. For each shape class they assume that there exists an unknown underlying level set function, whose zero level set describes the contour that best represents the shapes within that shape class. For each example level set function is modeled as a noisy measurement of the appropriate shape class's unknown underlying level set function.

In [4] they combine both contour and skeleton (local and global) information for shape analysis, and apply SVM or Boosting as a classifier. Discrete contour evolution [5] method is used to extract simplified polygons. Five simple shape descriptors: variance, compactness, convexity, elliptic variance, principal axes and pairwise geometric histogram were introduced in [1]. Paper [20] proposes a part-based approach to address a problem of classes that have a large nonlinear variability: bayesian classification is performed within a three-level framework, which consists of models for contour segments, for classes, and for the entire database of training examples.

Article [11] describes plant leaf classification using probabilistic integration of shape, texture and margin features. They introduce 100 leaves plant species dataset for evaluating an algorithm. The texture and margin features use histogram accumulation; shape is represented by normalized description of contour. Two different methods are used to generate separate posterior probability vectors for each feature, using data associated with the k-Nearest Neighbors apparatus. The combined posterior estimates produce the final classification (where missing features could be omitted). In addition, the framework can provide an upper bound on the Bayes Risk of the classification problem and assess the accuracy of the density estimators. Our methods show by 13% better accuracy than shape classification algorithms proposed in this article, giving 65.4% accuracy using KNN, and 75,7% accuracy using SVM classifier.

Paper [10] is devoted to plant leaves classification. They also used 100 leaves plant species dataset for algorithms evaluation. This paper proposes a comparison of supervised plant leaves classification approaches, based on different representations of these leaves. Beginning with the representation of leaves, they described leaves by a fine-scale margin feature histogram, by a Centroid Contour Distance Curve shape signature, or by an interior texture feature histogram. Using 64 element vector for each one, they also tried different combination among these features to optimize results. Our shape classification algorithm

outperforms shape classification algorithms compared with the article [10], giving by 16% percent better accuracy.

Another shape classification algorithm is proposed in [12]. They analyzed the histogram of normalized distance between each two points of the image (algorithm I), the histogram of normalized distances between three points and the normalized angle of the image edge points (algorithm II). The probabilistic neural network (PNN) was implemented to do shape classification, the approach was tested on ten classes of MPEG7 image database. In addition, these algorithms ensure invariance to geometric transformations (e.g., translation, rotation and scaling). The best classification accuracy is 90% for algorithm I, and 92.5% for algorithm II. Our algorithm also ensures invariance to geometric transformations, and strongly outperforms approach proposed in [12], which was the best for MPEG7. While testing on full MPEG7 dataset, which consists totally of 68 classes, we achieve better accuracy – 94.2% than it was achieved in [12] on 10 classes.

3. MAIN ALGORITHM

The basic version of algorithm, described in this article was proposed earlier in [13]. Here we introduce the following modifications of previously described approach: different shape representation, without losing shape proportions; another way of classifying objects for rotation invariance; application of one more classifier - kNN. Modified objects shape classification algorithm takes as input data object shape (for example, image segmentation), and consists of five main steps (Fig. 1):

- Present object contour as a sequence of vertexes.

- Smooth object contour if needed. For some datasets better results are shown while smoothing object contour using our extremes based approach, described in [14].

- Rotate object U times, for each object rotation find side contour feature, concatenate side features in common feature vector.

- Apply Homogeneous Kernel Map [10] to feature vector.

- Predict resulting feature vector, using KNN or SVM classifier.

Now we will describe each step of the algorithm in details. Single channel annotation (ground truth) is used for representing object shape (255 for foreground pixels, 0 for background pixels).

First, we store object contour as a sequence of vertexes (without loss of generality we can call it simple polygon). This is done to optimize image rotation speed, now we can rotate only object

contour using $O(m+n)$ time, we don't need to rotate the whole image, which takes $O(m*n)$ time.

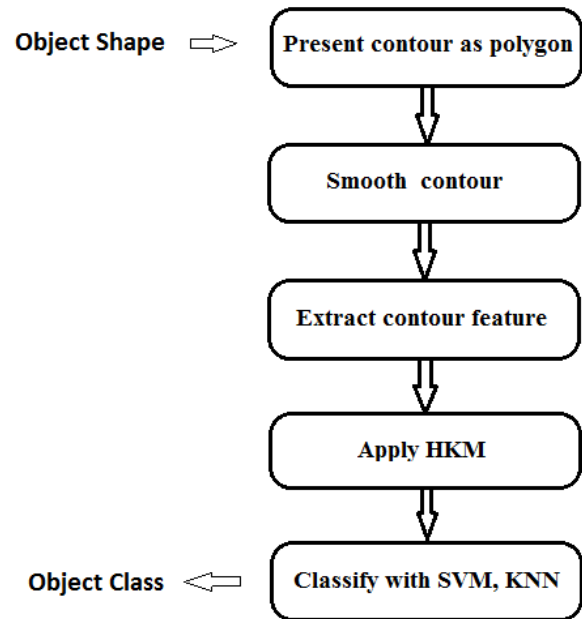


Figure 1 – Main algorithm scheme

In the second step, we smooth the object contour, presented as polygon, with extremes based method (if needed), described in our previous work [13]. In shape classification tasks contour noise may arise as a result of inaccurate human annotation, segmentation algorithm mistakes, or after the image mask jpeg compression, therefore we need a method for smoothing the object contour. Proposed algorithm finds vertexes, where contour convexity changes, using those vertexes obtains local minimums or maximums (there is ability to choose different criteria), which are used in resulting contour.

In the following step, we extract from polygonal shape feature descriptor, describing contour. Feature extracting algorithm has two input parameters, which values are selected empirically depending on the object specificity:

- U – number of sides = number of the image rotations;

- V – image side size = feature vector dimension for each side/rotation;

Resulting feature vector is $U*V$ dimensional. We rotate input image U times, taking the side description of the left object as a side feature vector, then concatenate all side feature vectors to form resulting feature vector. It is easy to see, that such description is more appropriate for objects, whose contours can be presented as a star-shaped polygon [19].

Now, let us tell how we calculate the side feature vector. First, we add to image mask empty

background parts from each side, to avoid image mask corruption while rotating. Rotating object totally U times, for i -th image rotation (original image rotated at an angle of $360*i/U$ degrees) we remove borders by finding left, right, top, and bottom foreground object coordinates, and then proportionally scale resulting ROI to make rows count equal V . In previous work [13] we use $V*V$ image representation losing object proportions. For each mask row, we store to side feature vector the most left object X coordinate, and finally concatenate all side feature vectors to $U*V$ dimensional resulting feature vector. Example of beetle description with $U = 6$, $V = 100$ is shown in Fig. 2.

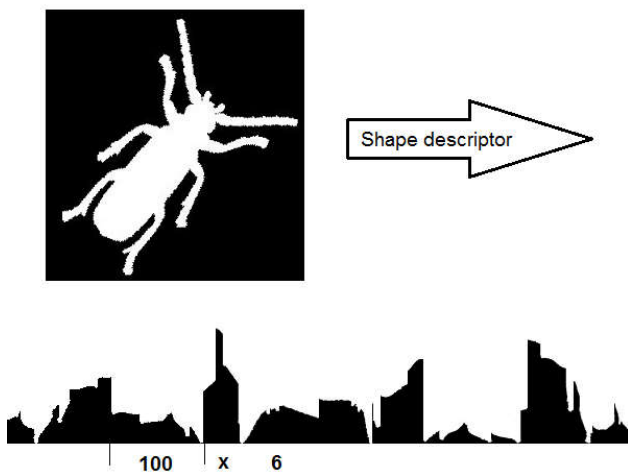


Figure 2 – Contour feature representation

In the fourth step, we normalize feature vector, results of applying different vector normalizations are presented in application part. Then, for emulating nonlinear kernel, homogeneous kernel map is applied to feature vector, making descriptor's dimension $2*Order+1$ times bigger ($Order$ is an input parameter of HKM algorithm).

Finally, we classify resulting feature vector. In earlier work [13] we use only SVM classifier, here KNN classifier was successfully applied too, for MPEG7 dataset it shows better results than SVM. While training multiclass SVM, we use 1 vs 1 winner choosing strategy, that is why we train totally $K*(K-1)/2$ binary classifiers, where K is a number of predefined classes. While building KNN classifier, we can use different data structures for representing training set, to speed-up KNN prediction we use K-D-Tree representation.

In addition, we improve previously proposed approach [13], adding scaling and rotation invariance. For extending training set we can add rotated images, the training time should grow, but prediction time stays the same, it makes proposed approach invariant to rotation. We can change the

direct image rotation by an $i*V$ -positional object descriptor shifting on feature extraction step. Scaling all images to images with the size of the row, which is equal V makes algorithm invariant to scaling.

Theorem 1. The time complexity of feature extraction algorithm is $O((m+n)*U)$, where m = width, n = height of image, U = steps count.

Proof. Smoothing and rotation steps depend on the shape and discontinuities of the object, but in the worst case contour have maximum $m+n$ vertexes. Smoothing contour of $m+n$ vertexes takes $O(m+n)$ time. Rotation contour of $m+n$ vertexes takes $O(m+n)$ time. Computing the object shape feature consists of U steps, each step needs $V < m+n$ iterations that is why the total time of a contour feature calculation part is $O((m+n)*U)$.

Optimal parameters for most datasets are $U = 24$, $V = 100$, so object shape is represented as 2400 dimensional feature vector. Most dataset images count is equal to 1000, maximum class count is equal to 100, that is why, proposed method is real-time, algorithm speed is 30 fps for SVM and 300 fps for KNN classifiers.

The same idea should be applied for more than 2 dimensional spaces, for example, for 3 dimensions we can use spherical angles using square $V*V$ side description for each rotation.

4. EXPERIMENTAL RESULTS

Proposed approach was evaluated on three datasets: MPEG7, 100 Leaves Plant Species, and our own Mushrooms 10 dataset. The dynamic of accuracy depending on algorithm parameters is shown mostly for Mushrooms dataset.

MPEG7 is open image shape dataset, shared among developers for algorithm quality comparison. It consists of 68 classes of different type shapes, 20 images for each class. The better accuracy – 94.2% for MPEG7 is archived using $U = 24$, $V = 100$, 96 rotations dataset augmentation, Homogeneous Kernel Map and KNN classifier (Table 2). Such quality is much better than quality of all existing methods (described in [12]), tested on that dataset. While testing on full MPEG7 dataset consisting of 68 classes, we achieve by 2 percent better accuracy than the best method from [12] achieved on 10 classes (Table 1).

Table 1. Comparison of different approaches.

Study	Accuracy (%)
Mokhtarian et al.	75.44
Belongie et al.	76.51
Mokhtarian & Bober	81.12
Ling & Jacobs	85.4
Felzenszwalb & Schwartz	87.7
PNN Atefeh Goshvarpour	92.5 (10 classes)
Koriukalov	94.2 (68 classes)

100 Leaves Plant Species is also open images dataset, shared for algorithm quality comparison. It consists of 100 classes of different leaf types, 16 images for each class. Our algorithm demonstrates 65.4% accuracy using KNN, and 75.7% accuracy using SVM, exceeding by 13% better results than methods, presented in [10] and [11] (Table 2).

Table 2. Proposed algorithm results.

Algorithms	Accuracy (%)
Naïve Bayes	52.625
Decision Tree	42.125
K-NN $k=3$	60.437
K-NN $k=4$	61.19
Our descr+KNN	65.4
Our descr+SVM	75.7
Prop sha	62.13
Wprop sha	61.88

In [13] we introduced our own Mushrooms dataset. Dataset consists of ten different mushrooms classes, totally 395 segmented images. The results of the improved algorithm application separately or together with texture and color descriptors are presented in Table 3. For Mushrooms dataset $U=12$, $V=100$ is optimal, $L1$ and $L2$ normalizations give the same result. Proposed contour descriptor improves classical feature based approach (Bag of Words for MSDS, Sift, Color) on Mushrooms dataset from 71% to 76%.

Table 3. The results of the improved algorithm application separately or together with texture and color descriptors.

Algorithm parameters	Accuracy (%)
Contour, $u = 1$	33
Contour, $u = 4$	48
Contour, $u = 4, L1$	50
Contour, $u = 12$	54
Contour, $u = 12, L1$	53
Contour, $u = 20$	54
BOW (MSDS+Color+Surf)	71
BOW (MSDS+Color+Surf) + Contour	76

5. CONCLUSION

In this article we introduce an efficient modification of a previously published shape descriptor [13], which is fast and simple to compute. Proposed descriptor was developed for object classification and should be used with classifiers. Described algorithm consists of five main steps: to represent object contour as a polygon; to smooth contour using extremes based approach; to rotate image, find side contour descriptor for each rotation, and append it to common feature vector; to normalize and apply HKM; to classify object contour descriptor, using SVM or KNN. Proposed

approach outperformed existing approaches on 100 Leaves Plant Species and Mpeg7 datasets, and showed good results on our own Mushroom dataset separately or together with color and texture descriptors.

6. REFERENCES

- [1] J. Iivarinen, *2 Shape Coding Techniques*, 1997, [Online]. Available: <http://www.cis.hut.fi/research/IA/paper/publications/bmvc97/node2.html/>
- [2] H. Park, G.R. Martin, A.C. Yu, *Lossless Contour Representation Using Efficient Multiple Grid Chain Coding*, 2005, [Online]. Available: <http://www.ieeexplore.ieee.org/document/7078158/>
- [3] Y. Qian, W. Xichang, Z. Huaying, S. Zhen, L. Jiang, "Recognition method for handwritten digits based on improved chain code histogram feature," *Proceedings of 3rd International Conference on Multimedia Technology (ICMT-13)*, Published by Atlantis Press, November 2013, 438-445.
- [4] X. Bai, W. Liu, Z. Tu, "Integrating contour and skeleton for shape classification," *Proceedings of the 12th International Conference on Computer Vision Workshops*, Kyoto, Japan, September 27 – October 4, 2009, pp. 360-367.
- [5] L. Latecki, R. Lakamper, "Shape similarity measure based on correspondence of visual parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1185-1190, 2000.
- [6] A. Tsai, W.M. Wells, S.K. Warfield, A.S. Willsky, "An EM algorithm for shape classification based on level sets," *Medical Images Analysis*, vol. 9, issue 5, pp. 491-502, 2005.
- [7] X.W. Bin, F.X. Bai, W.L. Longin, J. Latecki, "Bag of contour fragments for robust shape classification," *Journal Pattern Recognition*, vol. 47, issue 6, pp. 2116-2125, 2014.
- [8] Canny Edge Detection, 2009, 09gr820
- [9] A. Vedaldi, *VlFeat Library*, 2013, [Online]. Available: <http://www.vlfeat.org/>
- [10] M.E. Rahmani, A. Amine, M.R. Hamou, "Plant leaves classification," *Proceedings of the First International Conference on Big Data, Small Data, Linked Data and Open Data ALLDATA'2015*, Barcelona, Spain, April 19-24, 2015, pp. 75-80.
- [11] C. Mallah, J. Cope, J. Orwell, "Plant leaf classification using probabilistic integration of shape, texture and margin features," *Proceedings of the IASTED International Conference on Signal Processing, Pattern*

- Recognition and Applications (SPPRA'2013)*, February 12-14, 2013 Innsbruck, Austria, pp. 279-286.
- [12] A. Goshvarpour, H. Ebrahimnezhad, A. Goshvarpour, "Shape classification based on normalized distance and angle histograms using PNN," *International Journal on Information Technology and Computer Science*, no. 9, pp. 65-72, 2013.
- [13] O.V. Koriukalov, V.M. Tereshchenko, "Contour feature for simple objects classification real-time algorithm," *Bulletin of Taras Shevchenko National University of Kyiv, Series Physics & Mathematics*, vol. 1, pp. 143-146, 2016. (in Ukrainian)
- [14] O.V. Koriukalov, V.M. Tereshchenko, "Contour smoothing algorithm based on contour extremes," *Proceedings of the International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing*, Funchal, Madeira, Portugal, July 2-4, 2016, pp. 283-286.
- [15] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. EC-10, issue 2, pp. 260-268, June 1961. [Online]. Available: <http://ieeexplore.ieee.org/document/5219197/>
- [16] S.-D. Kim, J.-H. Lee, J.-K. Kim, "A new chain-coding algorithm for binary images using run-length codes," *Computer Vision, Graphics, and Image Processing*, vol. 41, issue 1, pp. 114-128, January 1988. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0734189X88901211>
- [17] D.G. Bailey, *Chain Coding Streamed Images through Crack Run-Length Encoding*, [Online]. Available: <https://www.cs.uic.edu/~super/SunSuperCVPR05.pdf>
- [18] L. da Fona Costa, R. Marcond Cesar, Jr., *Shape Classification and Analysis: Theory and Practice*, Second Edition, April 1, 2009. [Online]. Available: <https://www.crcpress.com/Shape-Classification-and-Analysis-Theory-and-Practice-Second-Edition/Costa-Jr/p/book/9780849379291>
- [19] Star-shaped polygon, [Online]. Available: https://en.wikipedia.org/wiki/Star-shaped_polygon
- [20] K.B. Sun and B.J. Super, *Classification of Contour Shapes Using Class Segment Sets*, [Online]. Available: <https://www.cs.uic.edu/~super/SunSuperCVPR05.pdf>
- [21] E. Salahat, M. Qasaimeh, *Recent Advances in Features Extraction and Description Algorithms: A Comprehensive Survey*, 19 Mar 2017 arXiv:1703.06376v1 [Online]. Available: <https://arxiv.org/pdf/1703.06376.pdf>



Oleksandr Koriukalov, Taras Shevchenko National University of Kyiv, PhD training – Images Classification SDK, RING Labs Senior CV engineer, computer vision, machine learning, computational geometry.



Vasyl Tereshchenko, Taras Shevchenko National University of Kyiv, Dr. of Sciences, Professor.

Scientific interests: computational geometry, computer graphics, computer vision, parallel algorithms, information systems, integral and differential equations.