Research Journal of Applied Sciences, Engineering and Technology 4(16): 2728-2734, 2012
ISSN: 2040-7467

Submitted: March 23, 2012 Accepted: April 20, 2012 Published: August 15, 2012

A Segmentation and Recognition Method of Rigid Image Targets

[1]Jian Cao, [1]Haisheng Li, [1]Qiang Cai and [2]Shilong Guo
[1]College of Computer and Information Engineering, Beijing Technology and Business University, Beijing, 100048, China
[2]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081, China

Abstract: In this study, we proposed a robust method to segment and recognize the rigid image targets automatically. First, the codebook is constructed by the local features from a certain kind of objects. Then, the target is segmented from the complex background as full as possible based on the prior knowledge which is acquired from the codebook. Moreover, a novel corner feature is proposed. This feature is invariant to translation, rotation, scale change and is shown robust to addition of noise. We present a system to recognize the object with changes in 3D viewpoint using this feature and BP network. The performance on the obtained experimental results demonstrated that the proposed method is more effective than the other three ones.

Keywords: Corner pointer, image segmentation, neural network, rigid target, target recognition, visual word

INTRODUCTION

Image target recognition is one of the most important topics currently in the domain of image processing and pattern recognition and has promising applications to visual surveillance, human machine interaction and medical diagnosis. There are two types of targets in image recognition which are rigid and nonrigid. A rigid target is not easily deformed and it generally has a rigid structure, such as aircraft, vehicles, buildings and other man-made objects. The rigid targets are suitable to be described by the geometric model and distinguished by the methods based on shape features.

The performance of an image target recognition system crucially depends on the following three issues: the segmentation of the interesting targets, the representation of the objects and the adopted classification algorithm. The previous two issues is called feature extraction, which as a key technology of target recognition, has a profound influence on the eventual performance (Gonzalez and Woods, 2007).

In the past few years, local features computed at keypoints have proved to be very successful in applications such as matching and recognition, e.g., SIFT (Lowe, 2004), PCA-SIFT (Ke and Sukthankar, 2004) and GLOH (Mikolajczyk and Schmid, 2005). These approaches tend to perform well when enough local appearance information can be found. Incorporating spatial information about feature distributions often proves helpful, even though surprisingly good results have been reported for simple bag-of-words approaches that neglect feature location altogether. The shape related features, such as k-Adjacent Segments, Geometric Blur (Berg and Malik, 2001) and Shape Context (Belongie *et al.*, 2000), have been studied in the context of pedestrian detection and performed comparably well Leibe *et al.* (2008) and Shilane *et al.* (2004).

The first contribution of this study is the introduction of the codebook of the rigid targets from the distinctive image features. The second is the segmentation of the targets from the complex background based on the prior knowledge which is acquired from the codebook. The third is a novel corner feature is proposed, which is invariant to translation, rotation, scale change and is shown robust to addition of noise. Finally, as our fourth contribution, we improve a state-of-the art recognition system and validate the feature ranking on a challenging target classification task.

In this study, we proposed a robust method to segment and recognize the rigid image targets automatically. A novel corner feature is proposed. This feature is invariant to translation, rotation, scale change and is shown robust to addition of noise. Moreover, we present a system to recognize the object with changes in 3D viewpoint using this feature and BP network. The performance on the obtained experimental results demonstrated that the proposed method is more effective than the other three ones.

KNOWLEDGE DRIVEN TARGET SEGMETATION APPROACH

Segmentation by morphological watersheds: Segmentation subdivides an image into its constituent

Corresponding Author: Jian Cao, College of Computer and Information Engineering, Beijing Technology and Business University, Beijing, 100048, China

regions or targets. The level to which the subdivision is carried depends on the problem being solved. That is, segmentation should stop when the objects of interest in the application have been isolated. Segmentation by watersheds embodies many of the principal concepts and this approach provides a simple framework for incorporating knowledge-based constraints in the segmentation process. We are reminded that humans often aid segmentation and higher-level tasks in every-day vision by using a priori knowledge, one of the most familiar being the use of context (Yu *et al.*, 2011). Thus, the fact that segmentation by watersheds offers a framework that can make effective use of knowledge is a significant advantage of this method.

The concept of watersheds is based on visualizing an image in three dimensions: two spatial coordinate versus gray levels. In such a "topographic" interpretation, the principal objective of segmentation algorithms is to find the watershed lines. Suppose that a hole is punched in each regional minimum and that the entire topography is flooded from below by letting water rise through the holes at a uniform rate. When the rising water in distinct catchment basins is about to merge, a dam is built to prevent the merging. The flooding will eventually reach a stage when only the tops of the dams are visible above the water line. These dam boundaries correspond to the divide lines of the watersheds. Therefore, they are the (continuous) boundaries extracted by a watershed segmentation algorithm.

One of the principal application of watershed segmentation is in the extraction of nearly uniform (bloblike) objects from the background. Regions characterized by small variations in gray levels have small gradient values. Thus, in practice, we often see watershed segmentation applied to the gradient of an image, rather than to the image itself. In this formulation, the regional minima of catchment basins correlate nicely with the small value of the gradient corresponding to the objects of interest.

Direct application of the watershed segmentation algorithm generally leads to over segmentation due to noise and other local irregularities of the gradient. In this case, over segmentation means a large number of segmented regions and can be serious enough to render the result of the algorithm virtually useless. An approach used to control over segmentation is based on the concept of markers. Marker selection can range from simple procedures based on gray-level values and connectivity, as was illustrated recently, to more complex descriptions involving size, shape, location, relative distances, texture content and so on. The point is that using markers brings a priori knowledge to bear on the segmentation problem.

Maker selection by visual words: To select the suitable makers for segmenting the target from the image, we require distinctive parts that are specific to the target class but can also capture the variation across different instances of the target class. Our method for automatically selecting such parts is based on the extraction of interest points from a set of representative images of the target object. A similar method has been used in some literatures (Cao *et al.*, 2011a, b; Leibe *et al.*, 2008).

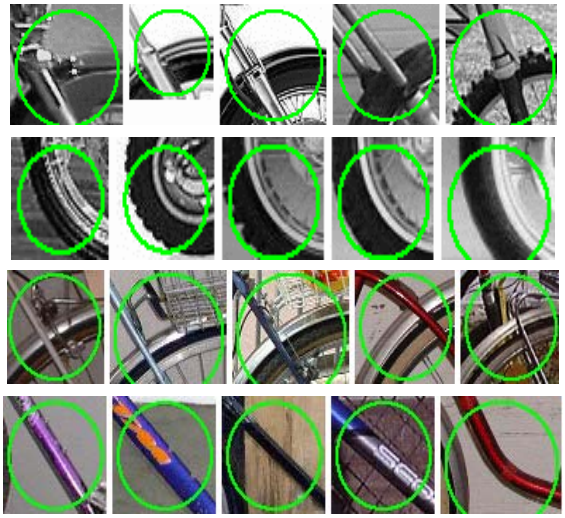

Fig. 1: Example of a some of the "part" clusters formed after grouping simialr patches together

Many different techniques for detecting and describing local image regions have been developed. The cost of extracting these features is minimized by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test. There are four major stages of computation used to generate the set of image target features: scale-space extrema detection, keypoint localization, orientation assignment, keypoint descriptor (Lowe, 2004).

As seen in Fig. 1, several of the patches extracted by this procedure are visually very similar to each other. To facilitate learning, it is important to abstract over these patches by mapping similar patches to the same feature id (and distinct patches to different feature ids). This can be achieved via various clustering algorithms, e.g., *k*-means algorithm, agglomerative hierarchical clustering.

The k-means algorithm used in this study is one of the simplest and most popular clustering methods, which allows it to very large data sets. It pursues a greedy hill-climbing strategy in order to find a partition of the data points that optimizes a squared-error criterion. The algorithm is initialized by randomly choosing *k* seed points for the clusters. In all following iterations, each data point is assigned to the closest cluster center. When all points have been assigned, the cluster centers are recomputed as the means of all associated data points. In

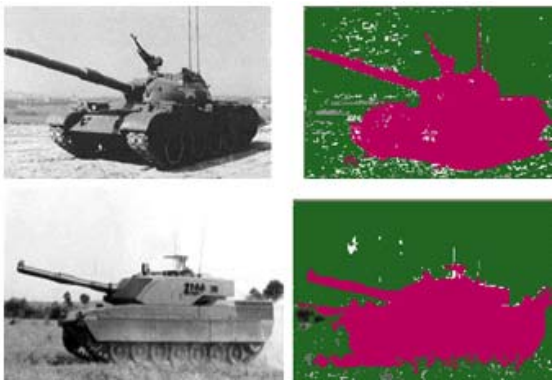

Fig. 2: Result of segmentation driven by knowledge

practice, this process converges to a local optimum within few iterations.

We group visually similar features to create a codebook of prototypical local appearances. The visual words in the codebook, as a priori knowledge of the target, can be used to guide the image segmentation. Following are the major stages of the segmentation method:

- Obtain the local features from the image datasets whose targets have been labeled, such as Caltech, PASCAL, LabelMe, LotusHill and so on.
- Create a codebook of the target class from the visual words using the algorithms proposed in these literatures.
- Detect the targets in the images based on the codebook automatically and with as little human intervention as possible.
- Segment the targets with the morphological watershed which is guided by the priori knowledge.

As can be seen from Fig. 2, we segment the tanks from the background using the method we proposed above.

CORNER FEATURE EXTRACTION ALGORITHM

Corner detection by SUSAN algorithm: One of the most intuitive types of feature point is the corner. Corners are image points that show a strong two-dimensional intensity change and are therefore well distinguished from the neighbor points. As seen in Fig. 3, corners provide a sufficient constraint to compute image displacements reliably and by processing corners the data is reduced by orders of magnitude compared to the original target image. Meanwhile, the corner detection is robustness to illumination changes and insensitive to noise. These advantages clearly show that using the features formed by corners can bring a significant improvement of target recognition and classification.

Fig. 3: Corner and other shape parts of the objects

One of the first approaches to finding corners was to segment the image into regions, extracting the boundaries as a chain code, then identify corners as points where directions changes rapidly. This approach has been largely abandoned as it relied on the previous segmentation step (which is a complex task itself) and is also computationally expensive. Smith and Brady (1997) introduced the SUSAN algorithm for low-level image processing and it can overcome these problems.

This corner detector bases on the "USAN", an acronym standing for "Univalue Segment Assimilating Nucleus" and the SUSAN (Smallest Univalue Segment Assimilationg Nucleus) principle: An image processed to give as output inverted USAN area has edges and two dimensional features strongly enhanced, with the two dimensional features more strongly enhanced than edges.

This approach to corner detection has many differences from the well-known methods, the most obvious being that no image derivatives are used and that no noise reduction is needed. The accuracy and speed of SUSAN algorithm are reasonable. Meanwhile, it can handle all types of junctions. The principle and detailed steps of this corner detector are not introduced here, they can be found easily in the related literature (Cao *et al.*, 2009; Cao *et al.*, 2010).

Optimizing the feature spaces: The corners and their spatial relationship have been observed and analyzed. For an integral part of the whole, there are greater differences between the corners at different locations. A corner carries the important information where there are few corners in a region; in contrast, it carries relatively unimportant information where there are a mass of corners together. For example, among the corners detected from the image of an airplane, the individual corners at the nose or wings are more important than the corners at the body. There are so many corners at the body of the airplane that we can reduce some ones.

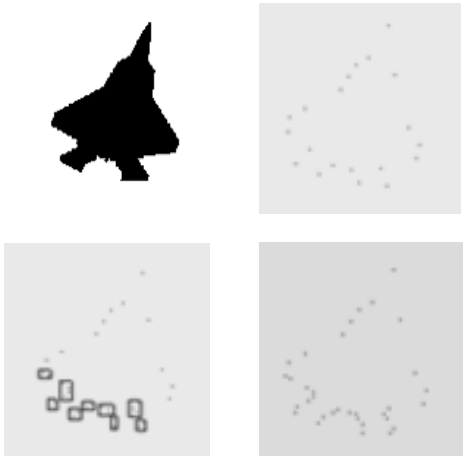

Fig. 4: Optimum design of corner-based feature space. The model of airplane is on the upper left. The result of the corner detection is shown on the upper right. The process of clustering is shown on the lower left. The fourth image is the final version

Several studies of major clustering methods have appeared in the data mining literature. We choose agglomerative hierarchical clustering to reduce redundant information for augmenting the robustness to the perceptual "noise". This is achieved via a bottom-up clustering procedure. Initially, each corner is assigned to a separate cluster. Similar clusters are then successively merged together until all of the corners are in a single cluster or until certain termination conditions are satisfied. In merging clusters, the similarity between two clusters and is measured by the average similarity between their respective corners:

$$sim(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{p_1 \in C_1} \sum_{p_2 \in C_2} sim(p_1, p_2) \qquad (1)$$

where, the similarity between two corners is measured by a simple Euclidean metric. In this study, the termination condition is obtaining a desired number of clusters. Then we use the cluster center to replace all the points in the cluster and the redundant information is reduced.

As can be seen from Fig. 4, the corners are detected from the airplane F22 and then be optimized using our method. The final version is used to generate the pattern vectors.

Corner signature: Three common pattern arrangements used in practice are vectors (for quantitative descriptions), strings and trees (for structural descriptions). As a pattern vector, the signature is a 1-D functional representation of a contour. Regardless of how a signature is generated, however, the basic idea is to reduce the boundary representation to a 1-D function, which presumably is easier to describe than the original 2-D boundary.

In the following we present the implementation details for the feature descriptor which is based on the basic idea of signatures. The feature vector can be expressed in the form after clustering for reduction:

$$X = (x_1, x_2, \ldots, x_n)^T \qquad (2)$$

where each component x_i represents the distance from the i th corner to the centroid and n is the total number of the corners.

The feature vectors generated by the approach just described are invariant to translation, but they do depend on rotation and scaling. Normalization with respect to rotation can be achieved by finding a way to select the same order to generate the feature vector, regardless of the shape's orientation. The approach to do so in this study is to order the components of the vector as the distance from each corner to the centroid and let $x_1 \geq x_2 \geq \ldots x_n$.

Changes in size of a shape result in changes in the amplitude values of the corresponding feature. The way to normalize for this result is to scale all functions so that they always span the same range of values, say, [0, 1]. The main advantage of this method is simplicity, but it has the potentially serious disadvantage that scaling of the entire function depends on only two values: the minimum and maximum. If the shapes are noisy, this dependence can be a source of error from object to object. The basic idea of the approach is to remove dependency on size while preserving the fundamental shape of the waveforms.

BP network design and train: BP network is one of the principal models of neural networks currently in use. In

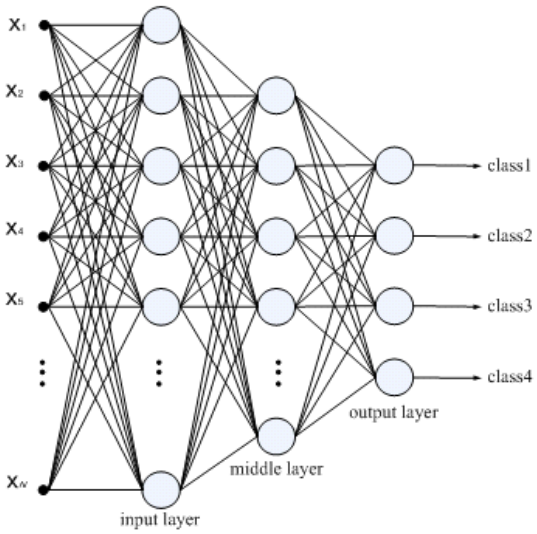

Fig. 5: Three-layer BP network used to recognize the targets

the experiment, we implement the target classification using a three-layer BP network of the form shown in Fig. 5. The number of neuron nodes in the first layer can be chosen corresponding to the dimensionality of the input pattern vectors N. The neurons in the third (output) layer correspond to the number of pattern classes and the number of neurons in the middle layer can be heuristically specified as the average of the number of neurons in the input and output layers. There are no known rules for specifying the number of nodes in the internal layers of a neural network, so this number generally is based either on prior experience or simply chosen arbitrarily and then refined by testing.

After the network structure has been set, activation functions have to be selected for each unit and layer. In this study, we use the same form of sigmoid activation throughout the network. The training process was divided in two parts. In the first part, the weights were initialized to small random values with zero mean and the network was then trained with pattern vectors from the training set. The output nodes were monitored during training. For any training pattern from class ω_j, the output unit corresponding to that class had to be high ($\geq$0.95) while, simultaneously, the output of all other nodes had to be low ($\leq$0.05). If more than one output is labeled high, or if none of the outputs is so labeled, the choice is one of declaring a misclassification or simply assigning the pattern to the class of the output node with the highest numerical value.

EXPERIMENTS AND ANALYSIS

Experimental setup: The aspect graph representation identifies regions of the viewing sphere where "equivalent views" and neighborhood relations on the viewing sphere generate a graphical structure of views. The basic idea of aspect graph can be used to represent the object with various poses. We can extract multiple 2D views of a 3D object in different view regions and use these view images to simulate the photos of the object with various poses or changes in 3D viewpoint. Figure 6 shows the generation process of the project images.

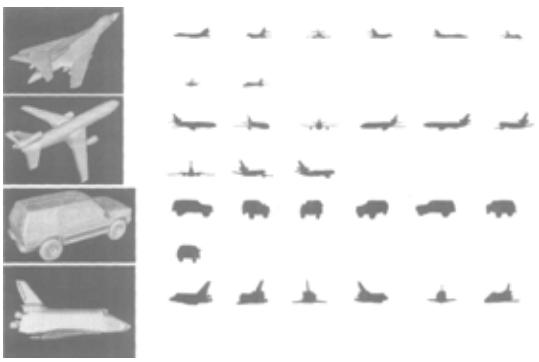

Fig. 6: 2D views to present the objects

Fig. 7: The models used in the experiment. They are F16, F22, F117 and M1237 in order

The Princeton Shape Benchmark (PSB) (Shilane *et al.*, 2004) is always used to investigate the performance of the methods for 3D object representation. We evaluate the performance of the recognition system on 4 object classes from PSB, namely F16, F22, F117 and M1237. Project images are sampled every 5º from 90 to 60th northern latitude and every 2 degree around each latitude, as can be seen from Fig. 6. In the 90th northern latitude, θ the view angle along vertical direction, is assigned to 0º. The angle θ in the 85th northern latitude is assigned to 5º and the rest can be deduced accordingly. So, for each latitude, there are 360/2 = 180 images, we select randomly 120 images from them as the training set, Since there are four classes and seven latitude each class, the size of the whole training set and the whole test set are: 120×7×4 = 3360, (180-120)×7×4 = 1680, respectively. Figure 7 shows four view images of the four models with θ = 0º.

We adopt the probability of misclassification as the evaluation criteria. The number of misclassified patterns divided by the total number of patterns tested gives the probability of misclassification, which is a measure commonly used to establish an object recognition system based on neural network performance.

Comparison with baseline methods: As baselines for comparison, we implemented three additional recognition systems. These systems simply use signature based distance versus angle, moment invariant and Fourier descriptor as feature vectors. Since using the same learning algorithm as our system and differ only in the representation, they provide a good basis for judging the importance of representation in learning.

In this part, the project images are sampled every 2 degree where θ = 0º (the viewpoint is just above the model) and we select randomly 100 images from 360/2 = 180 images as the training set. Since there are four classes, the size of the whole training set and the whole test set are: 120×4 = 480, (180-120)×7×4 = 1680, respectively.

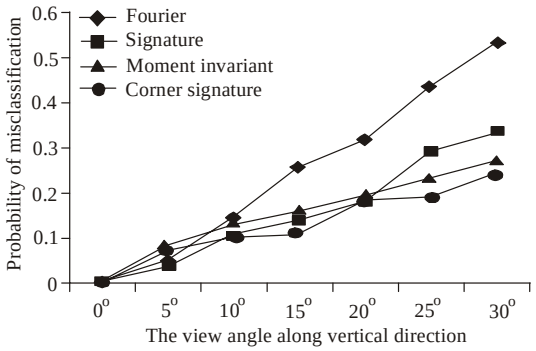

Fig. 8: Performance of the methods with changes in 3D viewpoint

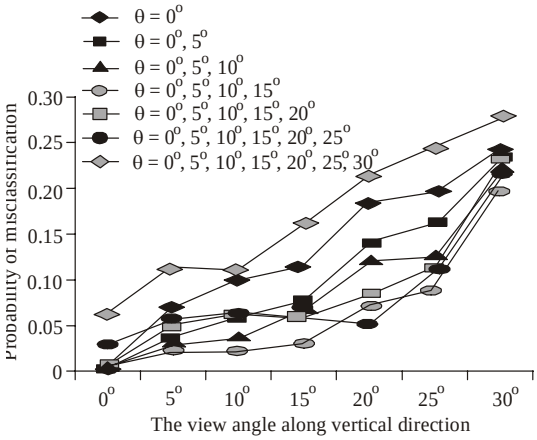

Fig. 9: Improvement in performance by using additive training

The results (using ours and the other three methods) are shown in Fig. 8. The performance of the baseline features and ours in this experiment indicates that the corner-based feature is robust to the changes in 3D viewpoint, i.e., the algorithm we proposed performs comparatively well when recognizing the objects with various poses.

Improved training for recognition: Next, starting with the weight vectors learned by using the data generated with $\theta = 0^o$, the system was trained with another data set generated with $\theta = 5^o$. The recognition performance is then established by running the test samples through the system again with the new weight vectors. Note the significant improvement in performance. Figure 5 shows the results obtained by continuing this retraining and retesting procedure for $\theta = 10^o$, 15^o, 20^o, 25^o and 30^o.

As expected if the system is learning properly, the probability of misclassifying patterns from the test set decreased as the increased because the system is being trained using the data with larger variations in shape. But the classification power only increased when the system was trained within 15 degrees of the closest training view. Otherwise, the network was not able to adapt itself sufficiently to capture non-planar changes and occlusion effects for 3D rigid targets. The Fig. 9 shows the improvement in performance by using additive training.

CONCLUSION

In this study, we want to seek for a segmentation and recognition method of rigid image targets which is automatically and with as little human intervention as possible. Target segmentation by watersheds offers a framework that can make effective use of knowledge is a significant advantage of this method. The markers obtained from the codebook bring a priori knowledge to bear on the segmentation problem. And the codebook is constructed base on the visual words, which are chosen from the suitable local features by clustering algorithms.

Then, we have presented an approach for recognizing rigid targets in images using corner signature. In our approach, the feature space is optimized and the redundant information is reduced. According to comparison with three classical features, the corner signature is efficient for building rigid target class representation. And our method performs relative well when the viewpoint changes in a large range.

Much future study remains with invariant and distinctive image features. An attractive aspect of the invariant local feature approach to recognition is that there is no need to select just one feature type and the best results are likely to be obtained by incorporating many different features, all of which can contribute useful matches and improve overall robustness.

ACKNOWLEDGMENT

This study was supported in part by the Research Foundation for Youth Scholars of Beijing Technology and Business University (QNJJ2011-38) and Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality (PHR201008239).

REFERENCES

Belongie, S., J. Malik and J. Puzicha, 2000. Shape context: A new descriptor for shape matching and object recognition. In NIPS, pp: 831-837.

Berg, C. and J. Malik, 2001. Geometric blur for template matching. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp: 607-614.

Cao, J., K. Li, C.X. Gao and Q.X. Liu, 2009. efficient object recognition using corner features. International Conference on Computational Intelligence and Security, pp: 344-348.

Cao, J., H.Q. Chen, H.J. Ma and Y. Wang, 2010. Optimization Algorithms for Corner Features. International Symposium on Knowledge Acquisition and Modeling, pp: 437-440.

Cao, J., H.Q. Chen, H.J. Ma and Y. Wang, 2011a. Object classification with local features. Int. Conf. Comput. Network Techn., 7: 553-556.

Cao, J., H.Q. Chen and M.Y. Mao, 2011b. Optimization algorithms for local features. Int. Conf. Autom. Commun. Architect. Mater., 225-226: 921-924.

Gonzalez, R.C. and R.E. Woods, 2007. Digital Image Processing. 2nd Edn., Publishing House of Electronics Industry, Beijing.

Ke, Y. and R. Sukthankar, 2004. PCA-SIFT: A more distinctive representation for local image descriptors. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp: 511-517.

Leibe, B., A. Leonardis and B. Schiele, 2008. Robust object detection with interleaved categorization and segmentation. Int. J. Comput. Vis., 77:(1-3): 259-289.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis., 60(2): 91-110.

Mikolajczyk, K. and C. Schmid, 2005. A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell., 27(10): 1615-1630.

Smith, S.M. and J.M. Brady, 1997. SUSAN-A new approach to low level image processing. Int. J. Comput. Vis., 23(1): 45-78.

Shilane, P., P. Min, M. Kazhdan and T. Funkhouser, 2004. Princeton shape benchmark. Retrieved from: http://shape.cs.princeton.edu/benchmark/

Yu, W., Z. Hou and J. Song, 2011. Color image segmentation based on marked-watershed and region-merger. Acta Electr. Sinica, 39(5): 1007-1012.