2010 Fourth Pacific-Rim Symposium on Image and Video Technology

Shape Classification Using Local and Global Features

Kart-Leong Lim[1], Hamed Kiani Galoogahi[2]
[1]Dept. of Electrical and Computer Engineering, [2]School of Computing
National University of Singapore
Singapore, Singapore
{g0800493, g0900434}@nus.edu.sg

***Abstract*—In this paper, we address the shape classification problem by proposing a new integrating approach for shape classification that gains both local and global image representation using Histogram of Oriented Gradient (HOG). In both local and global feature extraction steps, we use PCA to make this method invariant to shapes rotation. Moreover, by using a learning algorithm based on Adaboost we improve the global feature extraction by selecting a small number of more discriminative visual features through a large raw visual features set to increase the classification accuracy. Our local method is adopted from the popular bag of keypoints approach for shape classification. To integrate the classification results generated based on both local and global features, we use a combining classifier to perform the final classification for a new unknown image query. The experiment results show that this new method achieves the state-of-art accuracy for shape classification on the animal dataset in [8].**

Keywords-Shape Classification; HOG; SIFT; Bag of Keypoints, Adaboost Feature Selection.

I. INTRODUCTION

Shape classification plays an important role in many image processing and computer vision applications to extract, recognize and understand physical structures and objects. Even though many major researches have been done on this field, shape classification is considered as an open problem in image processing and machine vision issues, especially when the shape classes have large variations due to pose, deformation and occlusion or suffer from cluttered and complex background. On the other hand, different shapes that belong to a same class may share similar parts on couture or texture which can be used as a general representation for the class to discriminate from another one. Generally, to represent a class based on its common partial shapes, two sets of concepts have been proposed in shape classification, representing by local and global features.

Local features based methods perform shape classification using a set of simple and frequent features to simulate the shape knowledge of a class gaining finite samples of training data. However, the local features are not discriminative enough to distinguish two relatively similar classes. The aim of using global feature is to utilize more specific and less frequent features to represent more discriminative knowledge of a class domain.

The histogram of oriented gradient (HOG) is one of the most efficient descriptor which is commonly used for both local and global feature description [1-3]. While using as a descriptor, the basic idea in HOG is that an object shape can often be characterized by the distribution of intensity gradient orientations. All methods that use HOG as local or global feature follow a relatively similar scenario: first, the algorithm divides the input image window into small spatial regions which called *"cells"*, then for each cell accumulating a local 1-D histogram of gradient orientations. The combined 1-D histograms are used as a descriptor for each shape domain. Different methods might use different but simple approaches on 1-D gradient orientation histogram to make it invariance to illumination, shading, etc. by combining neighborhood or overlapped cells to construct larger blocks. While, others use HOG descriptor as a basic concept to generate more precise and invariant local and global features and eliminate the less discriminative extracted feature.

In this paper, we use histogram of gradient orientation feature for both local and global representation to perform shape classification. During training phase for both local and global features, we choose a finite set of random rectangular regions in the normalized and aligned object window followed by computing orientation histogram descriptors for each of them effectively. As a result, for each class, the features are constructed by a batch of the generated orientation histograms. Because the global features must be enough discriminative for each class, we then apply Adaboost [4] to select most discriminative histogram features to learn each class classifiers. For local representation we use a different scenario. For local feature extraction we use small random region on edge map of images to extract local histogram orientations just using edge pixels. Extracting local feature using edge map is inspired by the fact that the orientation histogram on non edge region is not informative enough for shape representing. Because the local histogram features are less discriminative and more common through different classes; we use a quantized set of the extracted local features to train the classifiers for each class. Given an unknown image query, using the local and global features, each local and global classifier generates a classification score. Eventually, this method uses a final classifier to integrate the two generated local and global based scores to perform the final classification. The reminder of this paper is organized as follow. In section 2 we review some recent related approaches for shape

978-0-7695-4285-0/10 $26.00 © 2010 IEEE
DOI 10.1109/PSIVT.2010.26

IEEE computer society

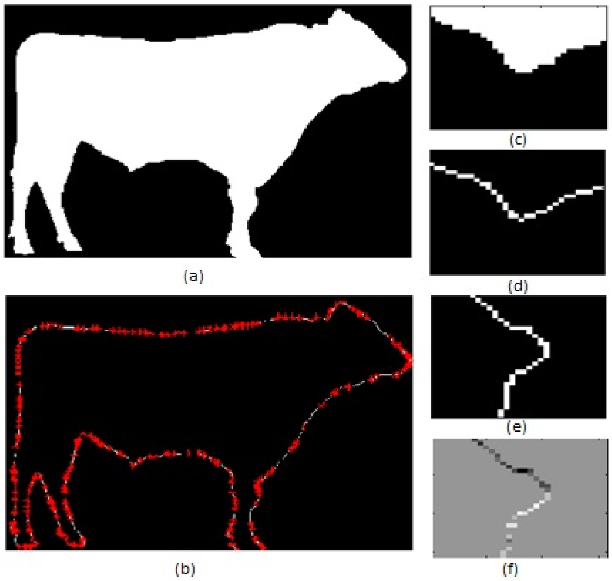

Figure 1. (a) original image, (b) the random generated points, red points, (c) selected sub region around a trial random point, (d) edge map of c, (e) sub region edge after aligning with peak local orientation, (f) gradient orientation map of e.

classification and object recognition. The detailed proposed method is described in section 3. In section 4 we present the result of experiments and analyze the performance of the proposed method in compare to some existing approaches. Some final comments and conclusions are made in section 5.

II. Related Works

The major point in shape classification problem is how to find an appropriate descriptor to represent the discriminative features of each class. As an efficient image descriptor, the histogram of gradient orientation is used efficiently by *SIFT* [1] for object classification [5] and human detection [2].

As a baseline image matching approach, *Lowe* [1] proposed an efficient local descriptor by assembling a high dimensional vector representing the image gradient orientation within a local region in the image which is called *SIFT* descriptor. *Lowe* also discussed about an approach using *SIFT* features for object matching that shown that it is invariant to image scaling, rotating, changing in 3D viewpoint and illumination [1].

Inspired by *SIFT*, *Csurka* and *et al.* [5] proposed the bag of keypoints method for generic visual categorization. To choose the most efficient local feature, using a vector quantization algorithm, this method assigns extracted *SIFT* descriptors based on HOG to a set of clusters. Then, it constructs a bag of keypoints according to the number of patches assigned to each cluster, cluster and corresponding local feature with low patches number will be discarded. The bag of keypoints used as the feature vector for shape classification. According to the idea behind the bag of keypoints method, we can answer the question why global feature based on HOG is not used to construct the bag of keypoints. In spite of the fact that global features are more distinctive than local features, there are less frequent patterns for global features. In the other word, for constructing a bag of keypoints, we need batches of features to be common for all not for a specific class.

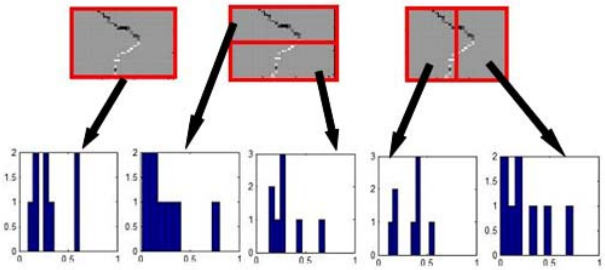

Figure 2. Three different masks for calculating orientation histogram and five extracted histograms for the global region.

On the contrary, global feature can be used when shape classification needs more distinctive and less frequent features. Local feature is not discriminative enough to distinct two different classes, especially when these two class are relatively similar, e.g. “cow” and “deer”. *Viola* and *Jones* [6] proposed a fast method for object detection based on simple rectangular feature called “*Haar-like Features*”. In fact, in this method, object detection procedure classifies image based on the value of simple two-rectangular, three-rectangular and four-rectangular masks that results in a huge set of extracted features. Due to the large size of extracted features, this method uses a learning algorithm based on Adaboost which selects a small number of critical visual features from the large primary features set.

Laptev [7] proposed a novel idea for object detection based on boosted histograms. In this method, *Laptev* uses HOG as global feature to describe each class shape. Given each input image, this method generates a set of random sub image and then calculates 1-D orientation histogram for each extracted sub image. In fact, each class is represented by a set of HOG. Because the extracted features are huge, this method use Adaboost classifier to choose the more discriminative features.

As a different approach, *Bai and et al* [8] use an integration of local and global features extracted from couture and skeleton, respectively. This idea is based on the fact that contour and skeleton provide complementary information for shape understanding. Generally, using contour based approaches are useful to capture local shape information and represent shape information, but they are sensitive to articulation, permanent and non-rigid shape deformation. In addition, extracting shape couture from a cluttered and complex background is another challenging problem that must be addressed in this kind of approaches. On the other hand, methods that use skeleton approaches to define global feature of shapes can cope with non-rigid deformations, but are efficient for rough structural. Moreover, skeleton approaches are vulnerable to self-

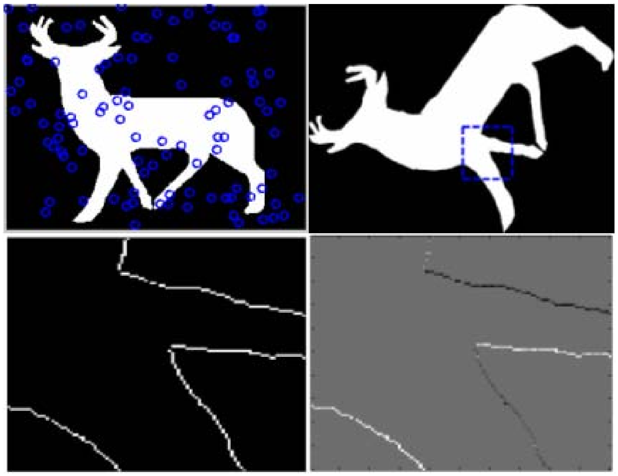

Figure 3. **Top left:** original image with random generated points. **Top right:** the image after aligning with its peak global orientation. The box is a region around a random point. **Bottom left:** the gradient magnitude image of the selected sub region **Bottom right:** the gradient orientation image of the selected sub region.

occlusion and huge shape deformation.

III. Proposed Algorithm

A. Local and Global Features

Image intensity is not as robust to illumination change as gradient of intensity. This is investigated in [1,6]. Our image feature extraction borrowed concepts from HOG [6], SIFT [1] and Ivan's version of HOG [7]. Our feature is a window of a random sub-region in an image that captures the gradient orientation, which we would compute the orientation histogram. For edge detection, we use Sobel mask with one pixel width thinning. Likewise in SIFT [1]; the author makes the SIFT feature rotation invariant by aligning the window to its peak orientation.

We find the peak orientation of the window by using Principal Component Analysis (PCA) on the edge pixel coordinates *(Y_i, X_i)* to get the axis along the largest eigenvalue [9]. The direction of the eigenvalue relative to the mean of the object contour allows us to compute the peak angle of the window. We then apply a matrix rotation to the original texture image about the center of the window. Then, we recomputed Sobel edge detection on the rotated image texture before calculating the feature descriptor, which is a 1x40 vector from a concatenation of five 8-bins gradient orientation histogram (Fig. 2).

Our terms local and global feature refer to features compute on small and large window sizes typically of 31x31 and 100x100 respectively. For local feature we only select random keypoints along edge points, while global feature are computed anywhere within the image. Fig. 1 shows how this method extracts local features. Given an input image as Fig. 1(a), Local feature extraction starts by computing the image edge map followed by generating random pixels on edge boundary, Fig. 1(b). Then, by choosing a 31x31 small cell around each candidate point, Fig. 1(e). Finally, using the gradient orientation of aligned sub region, a 1x8 orientation histogram is calculated which is used for local description. As for global feature, the feature extraction process is similar to local feature except that due to CPU cost, we perform peak orientation once on the entire image by finding the average over the orientation angles of 100 random points. This is illustrated in Fig. 3.

By this approach, we are capturing simple curves and lines for local feature and complex curves for global features. Another meaning to this is that features computed from small window sizes tend to be more commonly found between different classes while the reverse is that larger windows which are less commonly found in other class and will tend to capture more within-class image signatures. Thus, local feature is an ideal candidate for bags of keypoints model [5] which uses a visual codebook to contain all the common features used to describe an object class, while global feature can be selected by Adaboost learning to pick the more distinctive features for class separation.

B. Local Feature Classifier

For each given class, the local features are defined based on the concept of histogram of gradient orientation at 8 different orientations. Raw local feature extraction starts by selecting a set of small cells on each image edge map that belong to the class. Selecting small NxN cells on image map is performed by generating *T* random (x,y) edge pixels coordination as cell centers. By choosing random small cells on image edge map, we guarantee to select appropriate sub region with nonzero histogram to describe the local region more efficiently. In this case, assuming that for each image we generate *l* small cells, each image can be represented by 1x8 dimension feature matrix. Generally, local feature classification contains four main steps for classification in the algorithm:

1. Local feature extraction from each image
2. Apply clustering on the entire feature database to obtain a set of clusters
3. Represent each image by the count of features occurring in each cluster i.e. a histogram, where each bin is now a cluster
4. Construct a classifier based on the histogram of each image for each class.

After extracting a set of local orientation histogram features from each image in the data set, all the features are clustered into *K* clusters. The reason why we cluster the extracted local features is that for each image this algorithm will generate around 100 to 500 random cells to extract local features. Assuming there are 50 images per class, then for 20 given classes, there are about say worst case 50x20x500 = 10^3x500 histogram features for training is not optimal and might result in overtraining. Therefore, some form of quantization is required. The clustering algorithm used here is the simple K-Mean algorithm. After extracting a set of local features from each image in the data set, all the features are clustered into *K* clusters. The K-Mean clustering performs a quantization such that the entire feature space is only represented by a specific set of clusters and the center of each cluster is used as the representative for the cluster.

After generating the clusters for the features of a database, histogram features from each image is tossed back on the same representative set of clusters by nearest neighbor such that it produces a histogram where the bins are the clusters and the count of each bin refers to how many times features are matched to each bins. Thus, each image is now represented by a histogram which can be used as input to train a classifier.

C. Global Feature Classifier

In [6], the concept of feature selection by Adaboost [4] learning is successfully applied to face detection as well as in object detection [7]. The main idea in Adaboost feature selection is to find a set of features that can draw the best separation between positive and negative training images.

Since we do not have any domain knowledge about what kind of mask is best suited for our problem, we simply treat every random window as a possible feature. We then apply Adaboost to select a set random window that best represent the class features. The flowchart for our Adaboost feature selection is as follows:

1. Find *F*, a set of raw features found from all random windows in the same class images.
2. Compute matrix *WF*, where row is the total number of training images (positive and negative) and column is *F*. The value of each element in WF_{ij} is a set of random windows that have similar Cosine distance value to F_j that occurred in image W_i.
3. We filter F_j that are biased or having too few samples for both the positive and negative training samples from W_i.
4. Perform Adaboost for $t=1:T$ iterations, where T is the desired number of features:
   I. Sample draw with replacement a random set of images W_t from training set W, according to D_t, the distribution draw of W_t. The first iteration uses uniform distribution for D_1.

   For $j=1:F$ iterations

   II. Compute with a weak learner, e.g. a Mean Square Error classifier, the error from the misclassified samples of W_t.

   III. Choose the feature F_j with the lowest error and apply the standard Adaboost equations for updating the weight D_{t+1} in [6].
5. After the T^{th} iteration, a set of F_j selected features is computed.

The column vectors of the selected F_j in WF are the training samples of F_j for classification later.

After the Adaboost feature selection, we use a cascaded one-versus-all KNN learner for classification. For an unknown image within X pre-defined classes, a 300 random windows of 1x40 global features is first extracted and then put into each KNN classifier for computation. Each of the 1x40 will generate a label positive or negative for every KNN classifier. The majority vote of positive labels will decide the class label given the label of the KNN classifier.

D. Final Classifier

The reason for ensemble system is that no single base level classifier performs better than the other for different scenarios. Especially in our problem where some class result is poorer than the other classifiers. Some of the common tactics for combining class label from different classifiers include majority voting, weighted averaging etc. However, they need an odd numbers of classifiers. In our case we have 2 classifiers. In order to solve this problem, we compute a vector score from each classifier from -1 to +1 for each class. It means that for each image, output from each base level classifier a confidence score against every class. For each image the output for each base-level classifier is a 1xL vector for L classes. The 2 vectors from both classifier is combined using $F=W_1*F_1+W_2*F_2$. This is input into the meta-level classifier. Also, given the ground-truth, we can therefore construct a meta-level classifier to learn the decision for classifying the query at the combined level. Our choice of the combining classifier is a linear SVM classifier.

IV. Experiments

We use the same animal dataset[1] as tested by [8] for our experiment. There are 20 classes of binary images containing contours of animal shapes. The reason why we have used this dataset is because it is a clean and well segmented dataset so we do not require any additional preprocessing task before feature extraction. Moreover, the dataset is relatively challenging as some classes exhibit very similar contour to other classes such as deer and cow and some classes like monkey and rat have very unpredictable shape for classification. We split our dataset randomly into half for testing and half for training. As for the images used in the training process, we use 50 of the training images for positive class and 3 random images each from the other remaining 19 negative class, giving 107 images for training in both the SVM classifier and Adaboost feature setup. We also resize all images to within 640x480.

For local feature extraction, we first extract 300 random keypoints on the edge contour. For each local keypoint, we take a window size of 31x31 to compute both the PCA peak orientation and as well as to rotate an extended cropped region of the 31x31 window. This is to avoid having artificial edge created by the window when cropping some region along borderline. The original curve is preserved by cropping the original size of the 31x31 window after texture rotation to the Y-axis direction. After that the gradient orientation can be extracted by edge detection in the window. We tried 100 and 300 random windows per image and found the latter to give about 2% overall improvement. We also tried using 1x8 and 1x40 for local feature descriptor and found the improvement to be about 4% improvement. As for K number of clustering in K-means, we tried K=50, 100, 150, 200, 300, 500 and found the best K to be at 100. We only used linear SVM for our local feature classifier. We construct one-versus-all SVM classifiers for each class. The input size to each SVM is a 107x100 which refers to 107

[1] http://sites.google.com/site/xiangbai/animaldataset

training samples (50 positive, 57 negative) and each sample is a 1x100 vector. The 1x100 refers to the occurrence of each 1x40 feature to the nearest K=100 cluster bins (each bin is 1x40) using cosine similar distance, for each 300 windows per image. Thus, for local classifier, the parameters affecting the result is i) the number of windows per image, ii) the size of the window, iii) the size of the descriptor, iv) the K size for clustering and the sample size of training data.

For global feature extraction, we first generate 100 random windows and for each, we compute PCA for its peak orientation. Then we average the 100 angles and do a one-time image rotation of the entire original binary image (texture). As this is a lossy operation in both local and global case, i.e. we get holes in the rotated texture; we apply a flood-fill operation to repair the texture image. The reason for this is because the texture rotation operation is very slow for big crop region so we need to use an approximation. This is quite different from using the entire image for PCA peak orientation as this is a more robust way to deal with non-smooth noisy edges. As described earlier, for each class, we run Adaboost algorithm using the training images to extract feature samples which is Mx40, where M refers to the total number of samples from all 107 training images that supports this feature. M is different for each class. The classification is straight forward by using a cascade of 20 classes KNN for 20 animal classes. Likewise for global classifier, the parameters to tune are as mentioned earlier but more importantly the Adaboost settings such as the filtering step of Fj in 3) of the *global feature classifier* section and the cosine similar measure threshold (which in our case is 0.9), the number of T iterations (T is 200 in our case), the choice of the weak learner and the classifier (which in our case is KNN).

Our experimental result for test data on the animal dataset for our local, global and final classifier is shown in Table 1. In comparison with [8], where their contour shape and skeleton shape methods refer to local and global shapes, likewise we have a local and global feature extraction and classifier. We do not have any class result that underperformed below 50% as seen in “cat” and “monkey” classes in [8]. The lowest we achieved is 52% for “monkey” using our local method while the CS method in [8] underperformed at 21%. This suggests that the bag of feature approach combined with our local feature extraction is more robust to complex shape for classification. There is a similar trend in our result when compared to [8] in terms of the difficulty of the class for most cases. However, our method could not achieve the top results for some classes in [8] e.g. classes “spider” and “tortoise”. Our global result is 6.5% poorer than our local method. This is probably due to the fact that there are fewer training samples within class than between class. Another observation seen is that our result in global generally exhibit poorer results for some four-legged animal shapes e.g. “cat”, “cow”, and “leopard”.

In addition, as can be seen in Table 2, the proposed method achieved much better result in both areas, especially in the local method. We attribute this unique result to our efficient feature extraction and learning methods.

V. Conclusion

In this paper, we integrate the use of the Histogram of Oriented Gradient [2,7] in the Bag of Keypoints [5] and the Adaboost feature selection [6] methods for shape classification that leads to outperforming the experimental results of using the Contour and Skeleton shape methods in [8] for the animal dataset. We further improved our result on shape classification by combining both results from the two methods using an ensemble classifier. Furthermore, contrary to the limitation of using contours and skeleton in binary images, our method can also be applied to grayscale images as demonstrated earlier by [5] and [7] in both object categorization and face detection problems.

TABLE I. THE OVERALL ACCURACY OF NEW METHOD VS.[8]

Our Method	**Local**	**Global**	**Combine**
	74.90%	68.38%	80.37%
[8]	**CS**	**SP**	**CS & SP**
	71.70%	67.90%	78.40%

References

[1] D. Lowe, “Object Recognition from Local Scale-Invariant Features”, In Proc. Seventh International Conference on Computer Vision, Greece, 1999.

[2] N. Dalal, B. Triggs, “Histograms of Oriented Gradients for Human Detection”, In Proc. Computer Vision and Pattern Recognition, 2005.

[3] K. Mikolajczyk, C. Schmid, “A Performance Evaluation of Local Descriptors”, In Proc. Computer Vision and Pattern Recognition, 2003.

[4] Y. Freund, R. E. Schapire, “A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting”, Journal of Computer and System Sciences, Vol. 55, ed. 1, pp. 119-139, 1997.

[5] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray, “Visual Categorization with Bags of Keypoints”, In Workshop on Statistical Learning in Computer Vision, ECCV, 2004.

[6] P. Viola and M. Jones, “Rapid Object Detection using a boosted cascade of Simple Features”, CVPR, 2001.

[7] I. Laptev,”Improving Object Detection with Boosted Histograms”, Journal of Image and Vision Computing, vol. 27, ed. 5, pp. 535-544, 2009.

[8] X. Bai, W. Liu, and Z. Tu, “Integrating Contour and Skeleton for Shape Classification”, Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, ICCV 2009.

[9] W. Yi, S. Marshall, “Principal Component Analysis in Application to Object Orientation”, Journal of Geo-Spatial Information Science, vol. 3, ed. 3, pp. 76-78, 2008.

TABLE II. THE RESULT OF THE NEW METHOD IN COMPARISON OF [8].

		Bird	Butterfly	Cat	Cow	Crocodile	Deer	Dog	Dolphin	Duck	Elephant
Our Method	**Local**	80%	86%	72%	64%	72%	96%	84%	70%	70%	92%
	Global	70.5%	70.5%	57.4%	56.5%	56.5%	64.1%	62.5%	84.6%	54.8%	90.2%
	Combine	83.3%	88.4%	72.9%	66.5%	72.9%	97.2%	85.3%	84.7%	70.3%	95%
[8]	**CS**	76%	89%	39%	70%	54%	69%	69%	87%	83%	95%
	SP	55%	89%	37%	80%	60%	65%	62%	64%	79%	90%
	CS&SP	76%	93%	48%	80%	66%	79%	75%	89%	89%	97%
		Fish	**Flyingbird**	**Chicken**	**Horse**	**Leopard**	**Monkey**	**Rabbit**	**Mouse**	**Spider**	**Tortoise**
Our Method	**Local**	60%	66%	84%	94%	64%	52%	74%	56%	98%	64%
	Global	73.2%	59.8%	78.8%	78.2%	54.2%	63.8%	73.5%	68.6%	82.5%	67.5%
	Combine	76.9%	70.3%	90.8%	96.3%	66.6%	64%	82%	71.4%	99.7%	72.9%
[8]	**CS**	70%	57%	89%	96%	56%	21%	81%	52%	98%	83%
	SP	51%	35%	86%	77%	64%	33%	72%	82%	94%	81%
	CS&SP	74%	65%	94%	97%	65%	33%	87%	84%	100%	90%