Object Category Classification Using Occluding Contours

Jin Sun[1], Christopher Thorpe[2], Nianhua Xie[1], Jingyi Yu[2], and Haibin Ling[1]

[1]Computer and Information Science Department, Temple University, Philadelphia, PA, USA
[2]Department of Computer and Information Science, University of Delaware, Newark, DE, USA

Abstract. *Occluding contour* (OC) plays important roles in many computer vision tasks. The study of using OC for visual inference tasks is however limited, partially due to the lack of robust OC acquisition technologies. In this work, benefit from a novel OC computation system, we propose applying OC information to category classification tasks. Specifically, given an image and its estimated occluding contours, we first compute a distance map with regard to the OCs. This map is then used to filter out distracting information in the image. The results are combined with standard recognition methods, bag-of-visual-words in our experiments, for category classification. In addition to the approach, we also present two OC datasets, which to the best of our knowledge are the first publicly available ones. The proposed method is evaluated on both datasets for category classification tasks. In all experiments, the proposed method significantly improves classification performances by about 10 percent.

1 Introduction

Occluding contour (OC) is well known to play important roles in many vision tasks [1, 2]. Unlike regular photograph, an occluding contour image removes the effects of illumination, texture, and appearance while maintaining important edge and silhouette information. In computer vision, researchers have been seeking to develop new contour-based visual inference algorithms for many years [4]. In many visual inference tasks, a big challenge is to locate foreground object boundaries from the sea of all kinds of edge contours. Despite the known importance of OC, acquiring high quality OC in complex environment has been a long-standing challenging task [2].

In this paper we study the method and efficacy of using OC information for visual category classification, which is among the most important vision tasks [17]. We first use a novel multi-flash based OC acquisition device to get the initial OC estimation. This step provides us occluding contours that are more accurate than those from other existing methods. Once the OC data is ready, they can be used to improve visual inference tasks such as category classification.

The basic idea is to use occluding contours for feature filtering. Similar strategy appeared in [3]. Regions that are close to an OC are more likely to contain valuable shape related information and less pruning to distracting texture noises. Therefore, OC can be used to trim local visual features and then prepare a "purified" shape-related feature set for high level vision tasks such as visual recognition, detection, tracking, etc. Specifically, for an image and its estimated OC image, a distance map is generated

from the occluding contours. The distance map is used to filter out distracting local features in the image. The improved feature set is then combined with standard bag-of-visual-words model for visual category classification.

Another contribution of this paper is the benchmark datasets, which are the first such datasets to the best of our knowledge. We designed two datasets with both color images and OC images. The first dataset simulates the ideal occluding contours by manually picking OCs from normal edges maps (Canny edges [5]). In contrast, the occluding contours in the second dataset are automatically computed from the OC-Cam introduced in Section 2. We conducted category classification experiments on both datasets. In all experiments, the OCs information in the proposed method could significantly help improve classification performances.

The rest of the paper is organized as follows. Section 1.1 summarizes related work. Then, we introduce the OC acquisition device in Section 2. After that, the proposed OC-based category classification method is described in Section 3. Section 4 presents the experiments. Finally, Section 5 concludes the paper.

1.1 Related Work

As mentioned above, OC image could eliminate many distracting effects while maintaining important edge and silhouette information. To acquiring OCs, traditional passive image processing methods are often not robust enough to classify scene edges (e.g., occluding contours vs. material or texture edges) [8]. In particular, when foreground objects are surrounded by complicated background, it would be highly difficult to identify the occlusion boundaries.

Recent advances in computational photography have suggested that active illumination techniques can utilize shadows to effectively extract occlusion boundaries. For example, aerial imagery techniques can first detect shadows in a single intensity image and then infer building heights by assuming the ground geometry and surface reflectance models [11–13]. It is also possible to strategically cast shadows onto scene objects to recover their geometry [19]. In our work, shadows of objects are produced by multi-flash camera, which we refer readers to [18] for a complete review of this device.

In computer vision, researchers have been seeking to develop contour-based visual inference algorithms for many years. For example, contour information has been widely used in object recognition and localization tasks [4, 20, 7, 16, 15]. Most previous studies either assume that shapes of target objects are known, or work directly on the contours obtained from low- or middle-level edge extraction processes. Our work is different in that we explicitly use occluding contours achieved through the hardware directly.

Among many visual inference tasks, we choose visual category classification to demonstrate the effectiveness of using OCs. Category classification is an important research topic and has been attracting a large amount of research attention recently [17]. Our method is closely related to the bag-of-visual-words model [21, 10], which have been demonstrated excellent performance on several benchmark datasets [23, 22]. The proposed method can be viewed as an extension of these methods in the aspect of feature selection.

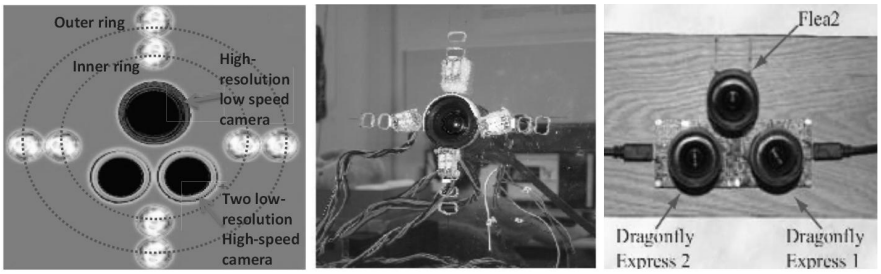

Fig. 1. The Occluding Contour Camera (OC-Cam) system design. (a) An OC-Cam uses one high resolution infrared camera and a pair of high speed visible light cameras. They are surrounded by multiple rings of controllable infrared LED lights. (b) The infrared camera and the LEDs form a multiple-flash camera. (c) The infrared camera and the visible camera pairs form a hybrid speed camera.

Fig. 2. Examples of occluding contours achieved by the OC-Cam system. Left: original normal images. Middle: corresponding OC images. Right: result of Canny edges [5] for comparison.

2 Extracting Occluding Contours

In this section we briefly introduce the occluding contour acquisition devices used in this study and the postprocessing steps. It is worth noting that both acquisition and postprocessing are fully automatic.

2.1 Occluding Contour Camera

Our solution for acquiring the OC data is to construct a novel *Occluding Contour Camera* (OC-Cam). The OC-Cam extends the previously proposed multi-flash camera that composes of a single image sensor with four flashes evenly distributed about the camera's center of projection, as shown in Figure 1. To acquire the contour data, the multi-flash camera takes successive photos of a scene, each with a different flash turned on. The location of the shadows generally abuts depth discontinuities and changes along with the flash position. All the depth edge pixels hence can be detected by analyzing shadow variations. For example, turning on the left flash will result in the shadows to the right of the depth edge. We can then traverse the image horizontally and identify the pixels that transition from the non-shadow region to the shadow region.

The major limitation of the multi-flash camera is that it is difficult to determine the proper camera-flash baseline, i.e., the distance the flash lies from the center of projection

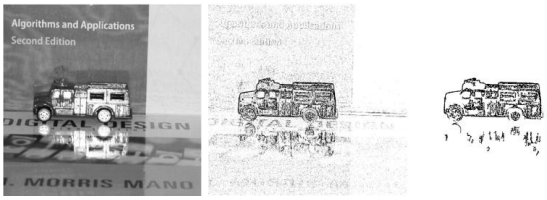

Fig. 3. Left: original normal images. Middle: original occluding contours. Right: results after postprocessing.

of the camera. For example, the shadows may appear detached from the boundary when the baseline is too large or may disappear when the baseline is too small. Since our goal is to acquire the occluding contours for various types of objects, it is important that we dynamically adjust the flash baseline, e.g., for acquiring both the internal and the external occluding contours of the object.

To achieve this goal, our OC-Cam mounts multiple rings of flashes around the central camera to support dynamic flash-camera baselines. In our implementation, we synchronized the LED flashes and the central viewing camera using the APIs provided by the PointGrey Research. The APIs allows the programmer to configure the camera, trigger image capture and also gives the programmer access to general purpose registers on the camera that can act as input/output ports depending on the configuration. Specifically, we use these registers as a four bit output port to send signals to the flash hardware. During the capture process each shot will take one picture at the given frame rate while at that time one of the flashes is illuminating the scene. For moderate frame rates such as twenty frames per second, synchronizing the flash sequence with the shutter is extremely important.

In order to control multiple rings of flashes with the same four bit port architecture, we further modify the control hardware: instead of using each bit to directly control a given flash, all flashes will be triggered sequentially from one bit. To do this, the pulse from one bit on the port will increment a counter. The output of the counter is fed into a decoder that indexes each flash. Therefore, the binary output from the counter can select an output line on the decoder which triggers the appropriate flash. We then use another bit to control which ring of flashes is used. Several images from the system are shown in Figure 2.

2.2 Postprocessing

The original OC image contains noises and irrelevant broken lines, hence a postprocessing is needed for further usage. First, an image filter with certain threshold (100/255 in gray level in the experiments setting) is convolved with the original image to produce binary image where black pixels indicate edge and white pixels indicate irrelevant background. Then the morphologic operators, closing and opening, are conducted on the image successively. Intuitively, closing joins the broken lines into connected line components (2x2 neighborhood pixels in the experiment setting) while opening removes

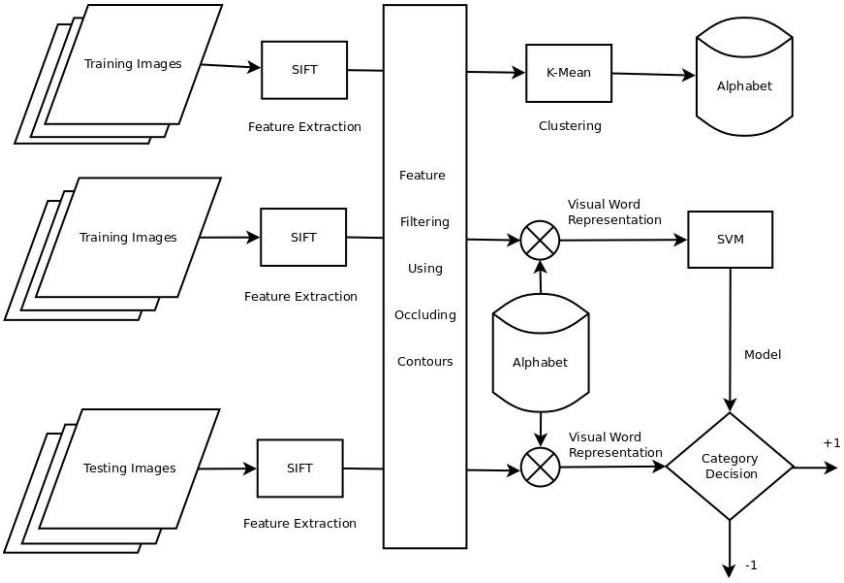

Fig. 4. Flow chart of OC guided category classification.

those connected components less than certain amount of pixels in area (100 pixels in the experiments setting). Figure 3 shows some results of the postprocessing.

3 Category Classification Using Occluding Contours

In this section we explore how to bring the rich shape information carried in OCs into category classification tasks. The overview of the process is shown in Figure 4.

3.1 Feature Filtering Using Occluding Contours

We propose to advance the state-of-the-art visual recognition algorithms by exploring the role of OCs as a *feature filter*. Specifically, OCs can help high-level vision tasks to get "purified" shape related features. This "purified" feature set in turn leads to improved object representation.

Let an input image be $I : \Lambda \rightarrow [0, 1]$, where $\Lambda \subset \mathbb{R}^2$ is the grid I defined on. The feature extraction of I is represented by a process $\mathcal{F}(I)$, which results in a set of local features. Without loss of generality, we denote the feature set as

$$\mathcal{F}(I) = \{(\mathbf{x}_i, \mathbf{f}_i)\}, \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^2$ indicates the position of the i^{th} feature and $\mathbf{f}_i \in \mathbb{R}^{n_f}$ indicates the n_f-dimensional feature descriptor. Specifically, in our experiment SIFT [14] is used, such that n_f=128.

(a) (b) (c) (d) (e)

Fig. 5. Use occluding contours for feature filtering. (a) The original image. (b) The original image with local features. (c) The OC image. (d) The distance map. (e) Original image with filtered local features.

The OC image of I is then denoted as $I_{oc} : \Lambda \rightarrow [0, 1]$. Our task is to use I_{oc} to trim $\mathcal{F}(I)$. A natural strategy is to use I_{oc} directly by eliminating any feature $(\mathbf{x}_i, \mathbf{f}_i)$ such that $\mathbf{x}_i$ is within a distance of an OC pixel. Precisely, the new feature set $\mathcal{G}$ is defined by:

$$\mathcal{G}(\mathcal{F}(I), I_{mask}) = \{(\mathbf{x}_i, \mathbf{f}_i) \in \mathcal{F}(I) : I_{mask}(\mathbf{x}_i) < \tau\}, \tag{2}$$

where $I_{mask}(\mathbf{x})$ is the distance transform map of $I_{oc}(\mathbf{x})$ and τ is the distance threshold. In particular our experiment uses Euclidean distance. In the new feature set $\mathcal{G}$, feature descriptors will stay close to occluding contour therefore provide better description of the target objects. An example of the filtering process is shown in Figure 5.

3.2 Category Classification Using Bag-of-Visual-Word Model

We follow the idea of Bag-of-Visual-Word approach [21] to represent the images as histogram of visual words. The independent features are generated by SIFT 128-dimensional feature descriptor. After that the alphabet of visual words, i.e. codewords dictionary, is formed by k-means clustering. The new image thus could be represented by histogram of visual words in the alphabet. The main difference here is that we apply the OC information to filter out irrelevant features whenever possible, as shown in Figure 4.

The discriminative method Support Vector Machine (SVM) is used in our approach as classifier. For implementation, we choose the LibSVM package [6] and a Gaussian kernel defined by

$$K(\mathbf{s}_i, \mathbf{x}) = \exp\left(-\frac{||\mathbf{s}_i - \mathbf{x}||^2}{2\sigma^2}\right), \tag{3}$$

where $\mathbf{s}_i$ denotes support vectors, $\mathbf{x}$ represents the feature representation (histogram of visual words) of the input image and σ is the covariance parameter for the Gaussian kernel.

4 Experiments

To evaluate the proposed method, two datasets are created containing both color images and corresponding OC images. The proposed method is conducted on both datasets in comparison with the original bag-of-visual-word method. In the following, we use BOW as abbreviation for the bag-of-visual-word and BOW+OC for the proposed method.

Fig. 6. Example images misclassified by the standard BOW but correctly classified using our proposed BOW+OC method.

Table 1. Category classification experiments on the Category-16 dataset.

Method	BOW	BOW+OC
Classification rate (%)	43.57±1.97	49.82±2.27

4.1 Synthetic Occluding Contours

We first create a dataset containing 16 categories selected from the Caltech 256 dataset [9]. Each of the 16 categories contains 20 images. For each image in the dataset, we generate a "simulated" occluding contour image by first using Canny edge detector and then manually removing non-occluding contours. In other words, for each image I in the dataset, there is a corresponding OC mask I_{oc}. In the rest of the paper we call this dataset *Category-16*.

To demonstrate how OCs can help with category classification tasks, both BOW and BOW+OC frameworks are conducted on the Category-16 dataset. For the experiment, we randomly divide the dataset into training and testing sets, with 10 of total 20 images per category for training and the rest for testing. The experimental result is summarized over 5 random splits. The average classification rate is listed in Table 1. It shows that OC, even when used in a very simple way, can substantially improve recognition rate. Fig. 6 shows several examples that are misclassified by BOW but correctly classified by BOW+OC.

4.2 Real Dataset with Complicated Background

Another dataset we build contains five categories: car, cow, cup, dog and horse. Each of the five categories contains 24 images (accompanied with OC images) taken from six different objects. For every object, images are shot from four poses: 0°, 90°, 180° and 270° horizontal rotating from the default pose. For each image in the dataset, we generate the occluding contour image from the device introduced in Section 2. Therefore every image I in the dataset has a corresponding gray level OC image I_{oc}. In the rest of the paper we call this dataset *Category-5*. Figure 7 shows example images of this dataset. We will make this dataset publicly available after publishing this paper.

Similar to the experiment on the synthetic dataset, both BOW and BOW+OC frameworks are conducted on the Category-5 dataset to demonstrate how OCs can help with the category classification tasks. To make experiment result consistent, we still randomly divide the dataset into training and testing sets, with 12 of total 24 images per category for training and the rest for testing. The experimental result is summarized over 100 random splits. The average classification rate is listed in Table 2. The result has shown that OC method has significant improvement in recognition rate: around

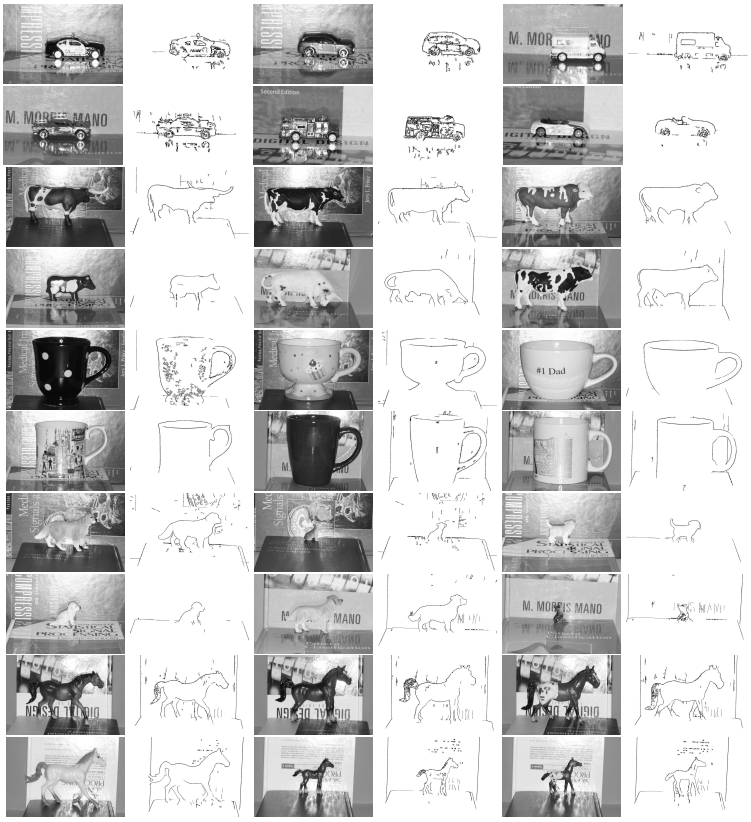

Fig. 7. Example images of the Category-5 OC dataset, six objects per class and one image pair (color image and OC image) per object.

Fig. 8. Example images misclassified by the standard BOW but correctly classified using our proposed BOW+OC method.

10%. Examples in Figure 8 show some examples that are misclassified by BOW but successfully classified by BOW+OC.

Table 2. Category classification experiments on the Category-5 dataset.

Method	BOW	BOW+OC
Classification rate (%)	56.17±7.78	66.50±8.22

Fig. 9. Example images misclassified by the BOW+OC but correctly classified using standard BOW method.

The above two experiments show clearly that OC information can be used to improve the performance of visual classification. It is also worth studying when the proposed method "hurts" the performance. In Figure 9 we show some examples which are misclassified by BOW+OC but successfully classified by BOW. One possible reason of misclassification by BOW+OC could be that sometimes it over-eliminates feature descriptors from the image. The setting of τ, i.e. the distance threshold, currently is determined empirically. This problem becomes noticeable when the input images are of certain types: background is solo-colored, background is uniformly textured, etc. These types share one property in common: feature descriptors are aggregated close to the object and are little scattered around the background in the OC image. Receiving those images as input, the BOW+OC method might eliminate some valuable feature descriptors around the object hence reduce the recognition rate while on the other hand standard BOW method will benefit from keeping all features. Though the limitation it appears in these scenarios, BOW+OC actually satisfied our expectation: when the standard BOW method can handle object classification in uniform background but suffer from complicated background, BOW+OC performs much better recognition rate according to the experimental results above.

5 Conclusions and Future Work

This paper investigates using the shape information from Occluding Contour (OC) to improve visual inference tasks, with focus on category classification. To this end, a new method is proposed that uses occluding contours as a feature filter to improve the image representation used in category classification. The improved representation is then combined with the bag-of-visual-words model for classification tasks. The proposed method clearly improves the performance on two datasets.

The applications of occluding contours are by no means limited to category classification. In fact, we expect the study in this paper to motivate rich future work toward different fields in computer vision, such as object localization. The datasets presented in this paper can therefore serve as benchmarks for future study as well. Aside from application of OC information in visual inference, we are also interested in improving the process of OC acquisition.

Acknowledgment Ling is supported by NSF under Grant IIS-1049032. Thorpe and Yu are supported under NSF Grant IIS-CAREER-0845268 and an Air Force Young Investigator Award.

References

1. M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. *CVPR*, 2008.
2. D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2004.
3. Yong Jae Lee and Kristen Grauman. Shape discovery from unlabeled image collections. *CVPR*, 2009.
4. I. Biederman, G. amd Ju. Surface vs. edge-based determinants of visual recognition. *Cognitive Psychology*, 20:38–64, 1988.
5. J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:679–714, 1986.
6. C. Chang and C. Lin. "LIBSVM: a library for support vector machines", *www.csie.ntu.edu.tw/˜cjlin/libsvm*, 2001.
7. V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):36–51, 2008.
8. D. Forsyth. Computer Vision - A Modern Approach. Prentice Hall, 2002.
9. G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
10. L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *CVPR*, 2005.
11. A. Huertas and R. Nevatia. Detecting buildings in aerial images. *Comput. Vision Graph. Image Process.*, 41(2):131–152, 1988.
12. R. IRVIN and D. MCKEOWN. Methods for exploiting the relationship between buildings and their shadows in aerial imagery. volume 19, pages 1564–1575, 1989.
13. C. Lin and R. Nevatia. Building detection and description from a single intensity image. *Comput. Vis. Image Underst.*, 72(2):101–121, 1998.
14. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
15. C. Lu, L. J. Latecki, N. Adluru, X. Yang, and H. Ling. Shape guided contour grouping with particle filters. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV)*, 2009.
16. S. Maji and J. Malik. Object detection using a max-margin hough transform. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
17. Toward Category-Level Object Recognition. Ponce, J.; Hebert, M.; Schmid, C.; Zisserman, A. (Eds.) *Lecture Notes in Computer Science*, Vol. 4170, Springer, 2006.
18. R. Raskar, K. han Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Transactions on Graphics*, 23:679–688, 2004.
19. S. Savarese, H. Rushmeier, F. Bernardini, and P. Perona. Shadow carving. *Computer Vision, IEEE International Conference on*, 1:190, 2001.
20. J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, pages 503–510, 2005.
21. J. Willamowski, D. Arregui, G. Csurka, C. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors.
22. N. Xie, H. Ling, W. Hu, and X. Zhang. Use Bin-Ratio Information for Category and Scene Classification. *CVPR*, 2010.
23. J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.