Active View Selection for Object and Pose Recognition

Zhaoyin Jia
Electrical and Computer Engineering Department
Carnegie Mellon University

Abstract

In this paper the author proposed an algorithm of actively selecting the views in the multi-view object and pose recognition problem. The proposed method tries to exploit the relation between the different view point of an object, model the object by multiple HoG (histogram of gradient) features and uses the Adaboost algorithm to learn a better view selection scheme as well as linking the different view points. Experiment shows the proposed algorithm can achieve better recognition accurancy comparing to randomly selecting the view point.

1. Introduction

Object recognition is one key step to achieve the ultimate goal of artificial intelligence, and thus becomes one of the areas that have been extensively investigated. With the development of the machine learning algorithm and the appearance of many fancy vision features, the object detection and recognition have been greatly advanced. [3] [4] [9] [6] However, most of the work have been done in detecting the object in a single image. Proceeding to the multiple images, there are some works focusing on the relation between the multiviews while trying to recognize the objects [12] and the feature distribution in multiple images [7]. Recently some models for 3D localization and pose esimation across multiview images appeared. [11] In this work, not only the similar features in the multiple images are considered helpful in object recognition, but also the 3D relationship between the multi-view are explored. The multi-view images are linked by a homography and then a better performance is achieved in estimating the category as well as the pose.

However there are more relations to make use of. Imagine a robot that is roaming in a room, trying to recognize some specific objects by continuously taking the images, which is a typical problem of mutli-view object recognition. The information that we have access to could be those multiple images, and furthurmore, the action/movement of the robot should be also available to us. Such extra information could certainly help us in pose estimation, for we have the pose of the robot to the first image. Once we gain a high believe in one image of category and pose recognition, the rest undetermined images could be solved by using the pose of the robot. For instance when looking at the back view of a toaster, it is even hard for human to identify it, but after moving 180^o around the object and seeing the front view, both the object and pose could be easily identified. Fig.1

(a) initial view (b) turning 90^o (c) turning 180^o

Figure 1. When facing the back view of a toaster, it is difficult to achieve the object recognition (a). However after turning around object and find a distinct view, (b) (c), all the poses could be identified

Besides that, the human beings are somehow doing the active search. When looking at a particular object from one view, we are not supposed to sequentially take the images with a constant degree interval such as $45^o, 90^o, 135^o \ldots$. More likely we choose the next action based on the previous recognition results, and by combining all the accessed views in the past we can make a better classification in both category and pose. We are inspired by this action and this paper is aiming to simulate such human behavior.

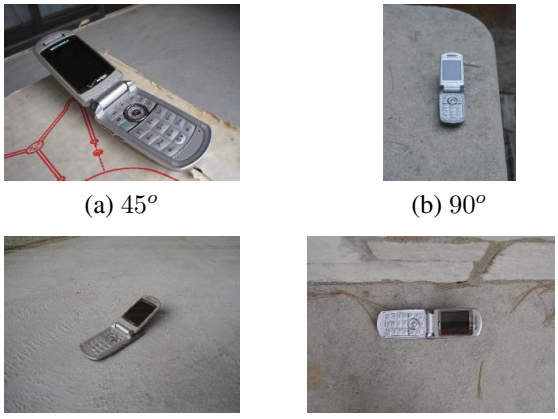
Researchers have done some similar works in this area. One application of such work is the Curious George [8], a robot exploring the the environment actively while identifying the objects. In this work although the 3D environment is attentivly searched, there is no feedback when doing the object recognition and the images are taken with a fixed degree intervals. In [1] the authors measure the similarity between multi-views and choosing the most different view in this similarity space. With the advanced machine learning algorithms, [10] uses the reinforcement Q learning technique to train the action.

In this paper we introduce an active search framework for multiview object and pose reocgnition. The problem

could be divided into two phases. One is object/pose recognition in a single image, which is a typical problem in computer vision. Another is training the actions based on the previous views. We use Histogram of Gradient and Support Vector Machine to model this single image recognition problem [3], and Adaboost to learn the actions. The paper will introduce these two steps sequentially. Because HoG feature only serves as a basis for multi-view recognition, the later part in the paper is paid more attention.

2. Database

When doing the active search, an ideal platform could be a controlled robot that could localize itself and equipped with a camera taking the images of the unknown environment, but in this situation the controlling ,localization as well as the mapping could become obstacles before solving the vision problem. As the proposed algorithm is focused on vision only, we use the UIUC Dataset of 3D object categories database [11] to simulate the behavior of the robot. In this database not only the multi-view of one object could be accessed, but also the view angles are available.2 In the database for one object there are 8 pictures of one object with 45 degree intervals, and 2 to 3 variations in height and scale. Therefore the action of 'rotating the object for 45^o' could be done by accessing the next image in one object image sequence, and then getting rid of all the troubles in dealing with the robot. However the algorithm in this paper could be implemented on real robot or a camera array, when the relatively view angle is accessible to the user.



(a) 45^o (b) 90^o

(c) 135^o (d) 180^o

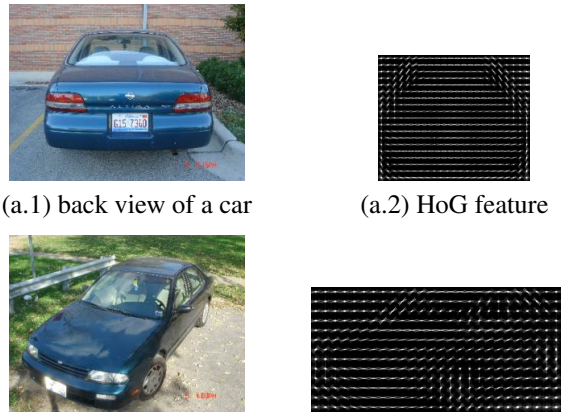
Figure 2. The dataset contains multiple views of one object and the view angles

3. Recognition in single image

We use HoG and SVM to do a single image classification. HoG is a dense representation of a certain object category. We follow the construction in [12] as the basis feature for classification. The first step is dividing the image into 8 by 8 non-overlapping pixel regions, known as cells, and for each of the cells a 1D histogram of gradient orientation is accumulated over pixels. HoG feature is useful in object recognition for such histograms largely maintain the shape of the objects, and also because it stors the histogram of the gradient orientation, such feature can handle some deformation of the object in the image.

We use nine orientation bins to discretize the gradient of each pixel. Each pixel could vote for the orientation, and the strength of the vote is dependent on the gradient magnitude. Then one 8×8 cell is tranformed into a histogram of gradient of nine channels. Then these nine channels are normalized with respect to the gradient energy in the neighborhood. We use 2×2 blocks that containing the cell adn normalize the given cell with the total energy in each blocks. Such operation captures the information of the neighbor cells, and leads to a 9×4 feature vector.

However, despite the robustness of the HoG feature, in [12] object from different views should still be modeled as different categories. For example, car from back views and from side views make a great deal of difference so that if modeling them together, the performance of the classifier should drop drastically. Fig.3.



(a.1) back view of a car (a.2) HoG feature

(b.1) side view of a car (b.2) HoG feature

Figure 3. Although HoG can handle some deformation, but still some degree deviation of the object results in quite different HoG features.

Therefore we model HoG of the object from different views as one independent training and testing category. Although the training and testing complexity have increased linearly to the number of the views, the recognition performance will be enhanced. Furthermore, modeling different veiws of the object will enable us to identify the pose of the object in the given image. So after modeling the object of different views, we train the one vs. all Support Vector Machine [2] for classification, e.g. for each object of one specific view, we use the samples under this constrains as

the true positives, and the samples of different object, or the same object but different viewing angles as the true negatives, and then do a binary classification.

4. Training the action

Suppose we have m different category of objects, and n different view of each objects, with the previous modeling we should have mn different SVM models after all, so we should have mn different hypothesis. But given one testing image, only one hypothesis is correct. We can control the robot to rotate for a certain degree, and get new images to gain more information to verify the proposed hypothesis. Therefore the problem switchs to whether we can determine a set of efficient actions, so that the true hypothesis would gain more beliefs, and the other $mn - 1$ false hypothesis could be wiped out.

The idea of active search is trying to learn the optimal sequence given the previous images. And if we consider the SVM result on each view as one classifier, we can combine the information from multi-views by combining the SVM result of those different views. It is identical to combine the different weak classifiers to form a strong classifier, which is the basic idea of boosting algorithm in machine learning. As a result we use the Adaboost algorithm to achieve this combination as well as the optimal sequence.

Adaboost [5] is a machine learning algorithm that minimize the exponential loss error. Similar to the other boosting algorithm, it exhaustly searches the optimal weak classifier for classification and weight different classifiers to combine a strong one. The trick in the algorithm is that it focuses on the 'hard instances' that could not be easily classified, and gradually picks the weak classifier to increase belief margin of these 'hard instances', while sacrificing the belief margin of the 'easy instances', and overall the exponential classification error is decreased.

4.1. training instance

The action taken next step is based on the previous actions and images acquired. In other word, the next wanted image should be the image from a view angle that could most disambiguate the category/pose confusion. To learn that, we should build a set of training instance. And after we run the boosting algorithm on this set, a set of actions could be given to achieve the goal of active search.

To simplify the problem, we assume that there is only one image given: the initial image. And each HoG/SVM is evaluate on this image, and then mn hypothesis are obtained but only one hypothesis k is true. The next step has $n - 1$ choice. To form the true positive instance, we put the following $n - 1$ in the right order relative to the true hypothesis k. And the true negative instances are the initial image is not true for hypothesis k, either the initial pose is wrong, or the category is not correct. However, the following sequence should be in right order, so that 'moving 45^o' is truly rotating the robot in the correct way.

For example, we are going the learn the action taken when the given image is the back view of the toaster, note as $t(0^o)$. The true positive instance would be $t(0^o), t(45^o), t(90^o), t(135^o), \ldots$. And the true negative instance would be of the same category, but wrongly initial step, such as $t(45^o), t(90^o), t(135^o), t(180^o), \ldots$, or the wrong category such as car: $c(0^o), c(45^o), c(90^o), c(135^o), \ldots$. However the image sequence should be still in the right order, because the action taken by the robot is exactly known. We use all the true positives in the database and randomly choose true negatives two times in number comparing to the true positives.

4.2. training phase

4.2.1 weak classifier

The boosting algorithm uses a combination of weak classifiers and in this paper we use the positive or negative decision stump of the SVM response on each image as the weak classifier. In detail, we can have all the true positives and true negatives instances from the previous section piled in big image matrix. The row represents images of 'moving 45^o' actions, and the column is different instances. Remembering each instance includes a set of images, one image could be considered as a feature in identifying the initial hypothesis. We have mn hypothesis, and therefore mn SVM responses, which is the belief margin of each hypothesis on this image. We make a positive or negative decision stump t on these SVM responses. Supposing we are trying to evaluate one hypothesis k, and for positive decision stump t, if the SVM response of hypothesis k on this image is greater than t, then we classify the instance as true for hypothesis k, otherwise it is false.

4.2.2 Adaboost Training

For each training instance i we have a weight w_i, and the weight is initialized identical between instances. Once the decision stump on one image is made on the j step, we can calculate the weighted training error ε_j:

$$\varepsilon_j = \sum_i w_i[y_i \neq h_j(x_i)] \tag{1}$$

where y_i is the true prediction of the current hypothesis k, $h_j(x_i)$ is the prediction of the current classifier on the instance x_i, e.g. the output of the current decision stump. The weighted error is evaluated on all the possible action and threshold t, and we choose the action with the minimum weighted error ε_{min} as the current weak classifier.

(a) (b) (c)

Figure 4. (a) Form the image matrix with the row as the possible action and column as the different instances. Each instance is equally weighted. (b) For one decision stump on SVM response, the weighted for each instance is changed accordingly. (c) Choose the optimal action that minimize the weighted error and then an action is learned.

The weight w_i is a key step to shift the focus of the combined classifiers of the boosting algorithms to those 'hard instance' in the training examples. w_i is initialized as identical for all the training instance, such as $w_i = 1$. Then first we can compute the weight α_i for current decision stump h_i:

$$\alpha_j = 0.5 \ln \frac{1 - \varepsilon_{min}}{\varepsilon_{min}} \tag{2}$$

And the rule for updating the weight of each instance for the $j + 1$ step is

$$w_i = w_i e^{-y_i \alpha_i h_j(x_i)} \tag{3}$$

And we again select the weak classifier that minimize the weighted error. After doing it iteratively the Adaboost algorithm can give a sequence of action that could optimally verify the hypothesis k, and in this way the action for verifying the hypothesis k with initial image is then learned. Such process is shown in Fig.4.

4.3. Testing phase

After training we obtain a set of weak classifiers $\{h_j\}$ and their weights $\{\alpha_j\}$, and each h_j represents the actions that should be taken for one hypothesis given the initial image. During the testing we follow the action from h_j, which tells us where the next image should be obtained, and the combined classifier is:

$$h(x_i) = \sum_j \alpha_j h_j(x_i) \tag{4}$$

However remembering that the weak classifier is a simple decision stump of SVM response on mn hypothesis, for one image that is evaluated it is possible to have multiple hypothesis become true. So here we must apply the contrain that only one hypothesis could be true for one testing image. If such case happens, we enforce that constrain by returning to examine the SVM response of those hypothesis that predicted true, and pick-up the maximum SVM response as the predicted hypothesis, if it is hypothesis k that is being verified, then it is true, otherwise hypothesis k is false for this testing image.

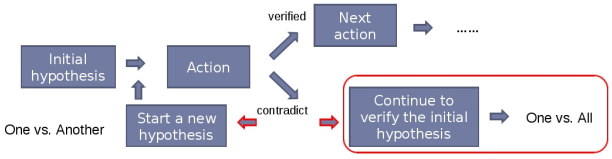


Figure 5. There are two strategies during the active search when the current predicted hypothesis contradicts with the initial hypothesis. Currently we continue to verify the initial one and the final output is a one vs. all classifier.

Also there are two strategies during the process of the active search when the current prediction of the initial hypothesis is false, e.g. there is a contradiction between the current predicted hypothesis and the initial hypothesis. Fig.5. In this situation, one can start a new hypothesis that has the maximum belief on the images that already obtained, then the final output of the classifier would be a one vs. another classifier that could give the exact prediction of the given image sequence. Another strategy is to continue verifying the initial hypothesis, and the output of the overall classifier is whether the initial hypothesis is true or not. Then it forms a one vs. all binary classifier. In this paper we use the second strategy.

5. Experiment

We use the 9 categories (car, toaster, cell phone, bicycle, iron, stapler, shoe, head) in the UIUC 3D object dataset, and for one object it has 8 different views. We use roughly 2000 images to model the SVM and HoG model for different categories and views, 2000 images as the true positives to train the actions, and 3000 images as the true positives

for testing. During the testing phase we randomly choose two times in number of true negatives with wrongly category or initial image. We try to identify the pose as well as the category of the testing instance.

The baseline method we use is what currently used in the robot application [8]. When obtain a new image, we choose the maximum SVM response as the predicted hypothesis. If the hypothesis of the new image contradicts with the previous one, we compare the maximum SVM response across the images and still pick up the maximum one. The baseline uses random image sequence.

Fig.6 shows the category and pose recognition accuracy for the baseline and proposed algorithm. The performance of the two methods start the same at the initial step, but the proposed algorithm have a significant boost when acquiring more images, and finally results in a better performance. One contribution in this better result of the proposed method is it learns the optimal action taken based on the previous results, so the recognition accuracy is increased more rapidly than randomly chosen the actions. Another reason is that during the boosting process, more information of the previous image is passed through the combination process, and thus gains a better performance at last.

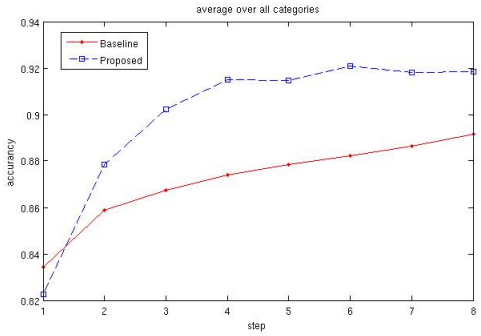


Figure 6. Blue line: the performance in recognition accuracy of the proposed algorithm. Red line: the baseline method

We also take a close look on how the proposed method perform on different categories. In Fig.7 it shows the performance of two categories 'iron' and 'stapler'. The proposed algorithm all have a better performance after several steps. We also examine the adaboost classification margin of the hard instance, e.g. those instance that are wrongly classified at first. If they are wrongly classified, we put their margin to negative by multiply by -1 for true positive and 1 for true negative. If they are correctly classified we put their margin to positive in the same way. Fig.7 shows that when obtaining the new image using the proposed algorithm, there is a great spur in belief margin, and therefore the classification performance becomes better.

We also did an experiment evaluating how fast the active search algorithm can identify the category and pose of the object. We calculate the margin of the classification at each step, and stop acquiring the new image if the belief margin is over 0.5. In Fig.8 the result shows that the proposed algorithm uses fewer steps to achieve the same belief margin, and therefore it is a faster active search method comparing to the baseline.

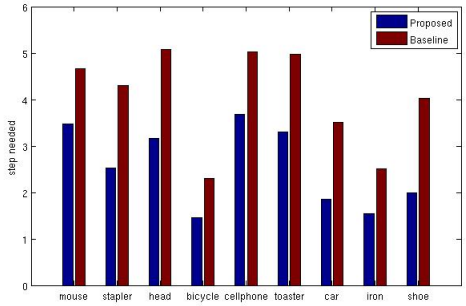


Figure 8. Blue bar: the average step needed to get 0.5 belief margin for proposed method. Red bar: the average step to achieve the same margin for baseline method

Another thing might be interesting is investigate what actions the algorithm are picking up given the initial image. Fig 9 shows the histogram of the angle rotated at the second step after given the initial image. It shows that the most frequent angle is around 90^o at both side rather than turning 180^o. That is because for many object, such as car, cellphone, toaster etc, turning 180^o of the object results in a very similar image, such as the back view and front view of a car, therefore it is not a very discriminative view to select. However for those categories turning 90^o the object becomes quite different and then extra information is gained in recognition.

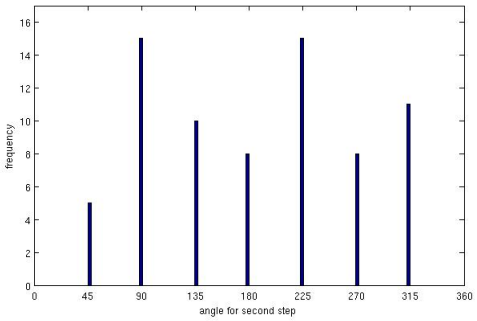


Figure 9. Histogram of angle rotated at the second step.

6. Conclusion

In this paper we introduce an algorithm to achieve the active search in multi-view object and pose recognition problem. We use HoG and SVM to build the basic classifier, and then implement the active search algorithm by making

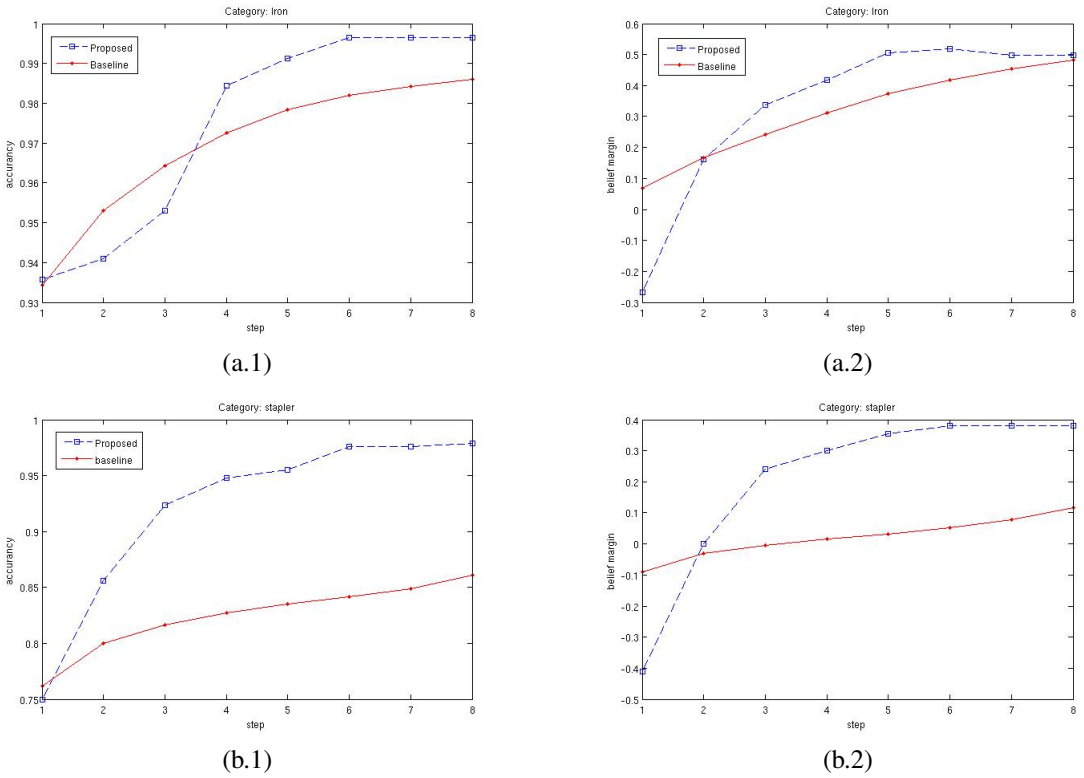


Figure 7. (a.1) Testing accuracy on 'iron' category. (a.2) The margin on hard instance of 'iron' category. (b.1) Testing accuracy on 'stapler' category. (b.2) The margin on hard instance of 'stapler' category. Bluelines: proposed algorithms. Redlines: baseline algorithms.

use of the adaboost on the training samples to learn the optimal action for recognition. Result shows that our method has a better performance than randomly sequence in recognition accuracy. Also the proposed method can obtain a fast recognition, for it uses fewer images to achieve the similar performance.

One future work of the active search would be boosting in features. Currently the extra information gained across the multiple image is only limited to the response of HoG and SVM models. However it is possible to build a part model of the object, and then it will enable us to pick up the most distinct part across all the multi-views, and the recognition performance could be enhanced by acquiring new parts. Another extension would be the testing on real applications. When implemented on real robot or camera array, the active search space could be extended into scale, height besides the view angle, and the space is continuous other than strict 45^o intervals. Also some tracking techniques could be applied when doing the recognition.

References

[1] S. Abbasi and F. Mokhtarian. Automatic view selection in multi-view object recognition. In *ICPR*, pages Vol I: 13–16, 2000. 1

[2] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. Svm and kernel methods matlab toolbox. Perception Systmes et Information, INSA de Rouen, Rouen, France, 2005. 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages I: 886–893, 2005. 1, 2

[4] P. F. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8, 2008. 1

[5] Freund and Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *JCSS: Journal of Computer and System Sciences*, 55, 1997. 3

[6] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999. 1

[7] D. G. Lowe. Local feature view clustering for 3D object recognition. In *CVPR*, pages 682–688. IEEE Computer Society, 2001. 1

[8] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. A. Baumann, J. J. Little, and D. G. Lowe. Curious george: An attentive semantic robot. *Robotics and Autonomous Systems*, 56(6):503–511, 2008. 1, 5

[9] K. P. Murphy, A. Torralba, D. Eaton, and W. T. Freeman. Object detection and localization using local and global features. In *Toward Category-Level Object Recognition*, pages 382–400, 2006. 1

[10] L. Paletta and A. Pinz. Active object recognition by view integration and reinforcement learning. *Robotics and Autonomous Systems*, 31(1-2):71–86, 2000. 1

[11] S. Savarese and F. F. Li. 3D generic object categorization, localization and pose estimation. In *ICCV*, pages 1–8, 2007. 1, 2

[12] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. J. V. Gool. Towards multi-view object class detection. In *CVPR*, pages II: 1589–1596, 2006. 1, 2