Automatic Database Creation and Object's Model Learning

Nguyen Dang Binh and Thuy Thi Nguyen

Institute for Computer Graphics and Vision
Graz University of Technology, Austria
{binh,thuy}@icg.tugraz.at

Abstract. This paper proposes a new framework to automatically generate visual object database meanwhile efficiently learn the object's model. The system is of important need for the problems of object detection and recognition. Our main idea is to acquire the huge amount of video data actively, and seeks out opportunities to autonomously exploit information from object samples. We employ autonomous learning approach based on online boosting technique, which allows to combine an object detector trained on a single initialized input image with tracking to extract object samples for learning. The autonomous learning process with interactive learning strategy allows to adaptively improve the learning object model while generating informative samples. Our method allows to generate thousands of object samples within hours from large video databases or from live camera, thus saving time and labor's efforts. We will show that the proposed method can extracts well-localized, diverse appearances of object examples from video sequence through only one initialized input sample, and builds robust object model. In addition to requiring very little human intervention, a significant benefit of this method is that it does not require pretraining. In the experiments, the approach is evaluated in detail for creating data sets and learning for the problems of human hand gesture recognition and face detection. In addition, to show the generality, results for different objects are also presented.

Keywords: Database creation, object's model learning, online boosting, autonomous learning, object detection, pattern recognition.

1 Introduction

In the last decade, computer vision has been being a fast growing research field. A lots of researches have been focused on acquiring knowledge about visual object of interest by training of detectors but only a little attention has been paid to efficiently labeling and acquiring suitable training data. Training a reliable object model requires large dataset, where positive and negative samples are usually obtained by hand labeling from large number of images. This costs a lots of time and labor efforts. In addition, the construction of appearance-based object detection systems is challenging because a large number of training examples must be collected and manually labeled in order to capture variations in object appearances. The number of variables that may

D. Richards and B.-H. Kang (Eds.): PKAW 2008, LNAI 5465, pp. 27–39, 2009.

be relevant in the database distribution is immense such as viewing angle, scale variation, lighting condition, background clutter, etc. With so many variables, there is no assurance that the training database has taken all the relevant variables into account or that their distribution will be the same as will be found in the application context.

The popular method for training a visual object classifier is to use a supervised learning algorithm (e.g. Adaboost [3, 4], neural network [5], or support vector machine [6]) with large hand-labeled set of object and non-object images. Most of these approaches have some drawbacks: First, at the preparation phase, manual labeling of training data is mandatory. For reliably training an object detector a large training data, in the order of several thousand image patches [7], is required. Labeling such a large training set by hand is costly. For the scope of many projects, creating a training set of over a thousand images is unrealistic. It is important to keep the human supervisory effort to a practical level. Second, for offline learning strategy, learning process is performed offline before the classifier is used for detection/recognition task. So, after the training the classifier remains fixed, and any further training is not possible. However incremental, online learning is desirable because then the classifier needs to know only what is actually necessary for the specific task. The classifier is time-adaptive and online learning can continue as long as the task is performed.

Recent years, online boosting learning [1, 17] has become widely used in computer vision community. Although online boosting allows efficiently training and improving the detector with a small training data set, learning an object detector is performed by an interactive process of just clicking to select positive and negative object samples on the current image. The evaluation the current classifier is done at a time. Thus the classifier can not update temporal information that represent contiguous variances of object samples, i.e. an object moving under camera or video sequences (e. g hand gesture or body gestures).

Obtaining a set of rich and informative training samples is challenging problem, especially for positive examples. To reduce the labeling effort, there have been a number of approaches which used two phases in a co-training fashion [7, 8]. Usually, at the first phase, a classifier is trained on small set of training data. Then at the second phase, the classifier is employed to detect object patches which will be used as positive samples. [8] even completely avoid hand labeling by using motion detection to obtain the initial training set. New examples are acquired by applying a detector obtained by online learning. Negative examples (i.e. examples of images not containing the object) are usually obtaining by a bootstrap approach [9]. A drawback of these approaches is one has to train an initial classifier. Moreover, the classifier tends to be biased by the examples used to train the initial model. Thus, many potential new examples, which represent different views of the same object, may not fit to the learned model.

By combining a tracking process with learning in a framework, the variation of object appearances can be tracked through image sequences, which provide samples for learning. Using a classifier and a tracker together, we take the advantage of the temporal continuity of video sequences to validate both tracking and classification, one for the other, while generating additional training examples. Our approach does not rely on pre-trained classifiers to bootstrap the learning process. We start with only one initial input image sample.

The main contribution of this paper is an autonomous learning of object's model that overcomes the above limitations. The basic idea is to design an automatic labeler, which can be seen as an oracle, to generate databases, i.e. positive and negative training samples, meanwhile incrementally learn the object's model. We develop an autonomous online learning algorithm based on online boosting [1], which allows to update the existing classifier based on one positive sample delivered by the oracle at each iteration to learn the object model and generate the object's database.

2 Related Work

There are numerous methods for moving object detection use motion as their primary source of information. Levin et al. [7] applied motion information to reduce the hand-labeling effort. They initially trained two classifiers, one on background difference images and the other one on intensity images with a small number of hand-labeled examples. Then the two classifiers use unlabeled data to iteratively improve each other. Our approach differs from [7] in that we use an autonomous classifier as tracker instead of the second classifier to identify informative training examples. Likewise, Javed et al. [17] used co-training to improve the performance of an initial classifier by selecting new training examples using PCA. Both systems needed a non negligible amount of supervision for labeling during initial training. Sivic et al. [13] applied tracking to obtain training samples. They used a face detector [14] that is trained by boosting orientation-based features. A conservative detection threshold is used to obtain low false positive rate. The consequence is many faces are not detected and the false negative rate is increased. [12] initialized an affine covariant region tracker to compute face representation from the tracked patches. First, they searched to localize facial features such as eyes, tip of the nose, and the center on the mouth. Then the object representation is built from five overlapping SIFT descriptor at the detected features. The drawback is to learn the model for the feature position and appearance, thousands hand-labeled face images is needed. The strength of the learned classifier-based detection approach is that it selects the object model using a learning algorithm, based explicitly on the model's ability to discriminate between object and non-object training examples. Hewitt and Belingie [15] proposed a method to learn a face representation, where a tracker serves for verification. The tracker locates the face correctly whereas the initial classifier may fail. Wu et al. presented an approach to online (re)-training of a detector based on the output of a coarse detector using boosting. As boosting focuses on difficult examples during training, it may be unstable if some examples are wrongly labeled. The method [6] and [7] also need supervision for initial stages, and it can only learn objects having the appearance similar to the samples used in initial training. Most of these mentioned approaches have been applied in one context only (e.g. pedestrian or car detection).

Several limitations should be addressed are: First, a pre-trained classifier is needed to initialize the learning process. Besides, a simple tracker may make some errors and select wrong samples, which must be verified manually before feeding to the learning process. Hence, non negligible human supervision effort is necessary. Second, tracker provides the labels during tracking, which would allow online learning,

still the models are trained off-line. In addition, there is no verification process on samples learned so far. Finally, there is no attempt to collect generated samples as the automatic acquisition of training data to build the objects databases.

In this paper we introduce an autonomous learning framework that based on online boosting learning. It requires no initial or pre-training. The tracker is initialized once, e. g with a good positive sample on the first single input image. Afterward, no user interaction is needed. In particular we employ online boosting for both learning and tracking, which allows to learn online an object detector and generate object samples. The idea is to use tracking information for selecting the most valuable positive and negative samples. An existing classifier is directly updated and evaluated on current image. The thus obtained detection results are the true positives and false positives as negatives. These samples are used to update the classifier and stored to the object's database. This proposes a simplification of the sample's generation process, in which a computer can train itself to detect and distinguish individual objects. It is a mean for reducing human effort needed to prepare the training set by training the object model. The process can perform real-time for processing images from a live camera or video sequence.

The paper is organized as follows. Section 1 is given to introduction. Section 2 presents the related work. Section 3 describes our framework for learning object model and generation of training data. Experiments and results are shown in section 4. Section 5 is for conclusion and future work.

3 Learning Framework

3.1. On-Line Learning

We employ the on-line boosting learning for feature selection as proposed in [1]. In the following, we briefly summarize the method. The main idea of boosting learning for feature selection is that each feature f_j corresponds to a single weak classifier h_j and that boosting selects an informative subset of *N* features, where a weak classifier has to perform only slightly better than random guessing (i.e., the error rate of a classifier for a binary decision task must be less than 50%). In fact, various different feature types may be applied but similar to the seminal work of Viola and Jones [3] in this work we use Haar-like features, which can be calculated efficiently using integral data-structures.

In the off-line case boosting for feature selection can be summarized as follows: given a training set of positive and negative samples $\chi = \{\langle x_1, y_1\rangle,...,\langle x_L, y_L\rangle | x_i \in R^m, y_i \in \{-1,+1\}\}$ where $x_i \in R^m$ is a sample and $y_i \in \{-1,+1\}$ is the corresponding label, a set of possible features $F = \{f_1,...,f_M\}$, a learning algorithm $\Im$, and a weight distribution *D*, that is initialized uniformly by $D(i) = \frac{1}{L}$. In each iteration *n, n = 1,..., N*, all features $f_j, j = 1,...,M$ are evaluated on all samples $(x_i, y_i), i = 1,...,L$ and hypotheses are generated by applying the learning algorithm $\Im$ with respect to the weight

distribution D over the training samples. The best hypothesis is selected and forms the weak classifier h_n and the weight distribution D is updated according to the error of the selected weak classifier. The process is repeated until *N* features are selected (i.e., *N* weak classifiers are trained). Finally, a strong classifier H is computed as a weighted linear combination of all weak classifiers h_n.

Contrary, during on-line learning each training sample is provided only once to the learner. Thus, all steps described above have to be on-line and the weak classifiers have to be updated whenever a new training sample is available. On-line updating the weak classifiers is not a problem since various on-line learning methods exist, that may be used for generating hypotheses. The same applies for the voting weights α_n that can easily be computed if the errors of the weak classifiers are known. The crucial step is the computation of the weight distribution since the difficulty of a sample is not known a priori. Thus, the basic idea is to estimate the importance λ of a sample by propagating it through the set of weak classifiers [18]. In fact, λ is increased proportional to the error e of the weak classifier if the sample is misclassified and decreased otherwise.

Thus, the work-flow for on-line boosting for feature selections selection can be described as follows: a fixed number of *N* selectors $s_1,...,s_N$ are initialized with random features. A selector s_n can be considered a set of M weak classifiers $\{h_1,...,h_M\}$, that are related to a subset of features $F_n = \{f_1,...,f_k\} \in F$, where F is the full feature pool. The selectors are updated whenever a new training sample $\langle x, y \rangle$ is available and the selector $s_n(x)$ selects the best weak hypothesis according to the estimated training error from the importance weights of the correctly and incorrectly classified samples seen so far. Finally, the weight α_n of the n-th selector s_n is updated, the importance λ_n is passed to the next selector s_{n+1}, and a strong classifier is computed by a linear combination of N selectors:

$$H_{on}(x) = sign\left(\sum_{n=1}^{N} \alpha_n \cdot s_n(x)\right) \tag{1}$$

Thus, contrary to the off-line version, an on-line classifier is available at any time of the training process.

3.2 Autonomous Online Learning

We will present our autonomous online learning algorithm to learn incrementally an object's model and efficiently generate training data. The learning process begins with only one initialized example using online boosting that has been discussed in detail in Section 3.1.

First, to initialize the classifier, a selected image region is assumed to be a positive sample. We have one-click to select target object as positive sample $\langle x,+1 \rangle_{t=0}$ and

$Pos_{t=0} = Pos_{t=0} \cup \{\langle x,+1\rangle\}$. The current target region is used for a positive update of the classifier $C_{t=0}$. Given this positive sample, an initial classifier $C_{t=0}$ is trained. The classifier is evaluated, and once the target object has been detected (the best of the detection) at time t, it is considered to be a positive image sample $\langle x,+1\rangle_{t=1}$ for updating of the classifier. At the same time, false positives are determined and used as negative samples $\{\langle x_1,-1\rangle,...,\langle x_n,-1\rangle\}_{t=1}$ for update. These negative samples are obtained by taking regions of the same size as target window from the false positives in the surrounding background: $Neg_t = Neg_{t-1} \cup \{\langle x,-1\rangle\}$. Using these samples to update, several iterations of the online boosting algorithm are carried out. Thus the classifier adapts to the specific target object and at the same time it is discriminative against its surrounding background. At time t, the classifier C_{t-1} is applied on the current image I_t. Thus the obtained detection result is verified by the tracking result T_t that robustly represents the object-of-interest. Based on this verification, the valuable samples (see Figure 1), i.e., the reported false positives (blue bounding boxes), are identified. In addition, such selected samples are labeled. These samples are fed back into the discriminative classifier as positive and negative examples, respectively, and we get a better classifier C_t. Obviously, the number of negatives is theoretically infinite if a non-integer positive grid is used. The current C_t classifier is evaluated at the surrounding region of interest and so obtains for each sub-patch a confidence value which implies how well the underlying image patch fits the current model. Afterwards we choose the best of the detection as maximum obtained confidence and shift the target window to the new maxima location, and $Pos_t = Pos_{t-1} \cup \{\langle x,+1\rangle\}$. Next, the classifier has to be updated in order to adjust to possible changes in appearance of the target object and to become discriminative to a different background. The current target region is used for a positive update of the classifier while surrounding false positive regions are taken as negative samples and $Neg_t = Neg_{t-1} \cup \{\langle x,-1\rangle\}$. To cover as many negative as possible we maintain the same set of positives but bootstrap a new set of negatives that pass all previous strong classifier (i.e. false positive). This update policy has proved to allow stable learning and tracking in natural scenes. As new frames arrive, the whole procedure is repeated and the classifier is therefore able to adapt to possible appearance changes and in addition becomes robust against background clutter.

The idea is to employ online boosting technique to adaptively learn an object representation/discriminative classifier from only one initialized example. To actually learn the object representation we develop autonomous online learning algorithm based on online boosting learning algorithm [1] but any other online learning method my be applied.

Algorithm 1. Online Autonomous Learning and Data Generation

Input: - An empty discriminative classifier $C_{t=0}$;

- Video sequence or image set

Output: Classifier C_t; Positive set Pos_t and Negative set Neg_t and Ground truth;

1: Initialize parameters for the classifier $C_{t=0}$ and train with 1-click on initial object sample;

2: Initialize positive and negative sets: $Pos_{t=0} = \{\}$ and $Neg_{t=0} = \{\}$

3: **while** Non-Stop-Criteria **do**

4: Evaluate C_{t-1} on current image frame I_t obtain J detection x_j and display results;

5: Predict and determine true positive: T_{t-1};

6: **For** j=1,…, J **do**

If $T_{t-1} \approx x_j$ then

begin

//Use true positives samples to update the classifier;

•Update(C_{t-1}, x_j,+1) follows algorithm 2;

// Automatic true positive labeling: adding true positive to Pos_t set ;

• $Pos_t = Pos_{t-1} \cup \{\langle x,+1\rangle\}$;

end

Else $T_{t-1} \neq x_j$ then //Determine false positives on current image I_t;

begin

//Use false positives as negative samples to update the classifier;

•Update(C_{t-1}, x_j,-1) follows algorithm 2;

// Automatic negative labeling: adding negative samples to Neg set ;

• $Neg_t = Neg_{t-1} \cup \{\langle x,-1\rangle\}$;

end

7: **End for**

8. **End while**

3.3 Image Representation and Features

In our work, we use efficient integral image representation for fast calculation of objects features. The features include Haar wavelet [3], local orientation histogram [19] and a simplified version of local binary patterns [20] as a representation, which can be fast computed on integral images. The computation of these feature types can be done very efficiently. For online learning a weak classifier h_j for feature j we

Algorithm 2. Online Learning for Feature Selection

Input: - Training example $\langle x, y \rangle$, $y \in \{-1,+1\}$;

- Strong classifier C_{t-1};

- Initialized weight $\lambda_{n,m}^{corr} = 1$; $\lambda_{n,m}^{wrong} = 1$;

Output: Strong classifier C_t

1: Initialized the importance weight $\lambda = 1$

2: **For** n=1,…,N **do** // for all selectors

for m=1,…., M **do** //update the selector s_n^{sel}

- $h_{n,m}^{weak} = update(h_{n,m}^{weak}, \langle x, y \rangle, \lambda)$; // update each weak classifier
- if $h_{n,m}^{weak}(x) = y$ then $\lambda_{n,m}^{corr} = \lambda_{n,m}^{corr} + \lambda$;

  else $\lambda_{n,m}^{wrong} = \lambda_{n,m}^{wrong} + \lambda$;

  end if
- $e_{n,m} = \frac{\lambda_{n,m}^{wrong}}{\lambda_{n,m}^{corr} + \lambda_{n,m}^{wrong}}$;

end for

$m^+ = \arg\min_m (e_{n,m})$; //choose weak classifier with the lowest error

$e_n = e_{n,m^+}$; $s_n^{sel} = h_{n,m^+}^{weak}$;

if $e_n = 0$ or $e_n > \frac{1}{2}$ then exit

end if

$\alpha_n = \frac{1}{2}.\ln\left(\frac{1-e_n}{e_n}\right)$; //calculate voting weight

// update importance weight

if $s_n^{sel} = y$ **then** $\lambda = \lambda.\frac{1}{2.(1-e_n)}$;

else $\lambda = \lambda.\frac{1}{2.e_n}$;

end if

$m^- = \arg\max_m (e_{n,m})$; $\lambda_{n,m^-}^{corr} = 1$; $\lambda_{n,m^-}^{wrong} = 1$;

Get new h_{n,m^-}^{weak};

3. **End for**

first build a model by estimating the probability $P\left(1 \middle| f_j(x)\right) \sim N(\mu^+, \sigma^+)$ for positive labelled samples and $P\left(-1 \middle| f_j(x)\right) \sim N(\mu^-, \sigma^-)$ for negative labelled samples, where $f_j(x)$ evaluates this feature on the image x. The mean and variance are

incrementally estimated by applying a Kalman filtering technique. Next, to estimate the hypothesis for Haar-Warelets, we use either simple threshold $h_j(x) = p_j.sign\left(f_j(x) - \theta_j\right)$ where $\theta_j = \left|\mu^+ + \mu^-\right|/2$, $p_j = sign\left(\mu^+ - \mu^-\right)$ or a Bayesian decision criterion:

$$h_j(x) = sign\left(P(1|f_j(x)) - P(-1|f_j(x))\right) \approx sign\left(g(f_j(x|\mu^+, \sigma^+) - g(f_j(x|\mu^-, \sigma^-)\right)$$

where $g(x|\mu, \sigma)$ is a Gaussian probability density function. For histogram features (orientation histograms and LBPs), we use nearest neighbour learning D (e.g. Euclidean):

$$h_j(x) = sign\left(D\left(f_j(x), p_j\right) - D\left(f_j(x), n_j\right)\right)$$

The cluster centers for positive p_j and negative n_j samples are learned by estimated the mean and the variance for each bin separately. All modules are based on the same type of classifier that is trained using the same features (For more details see [1]).

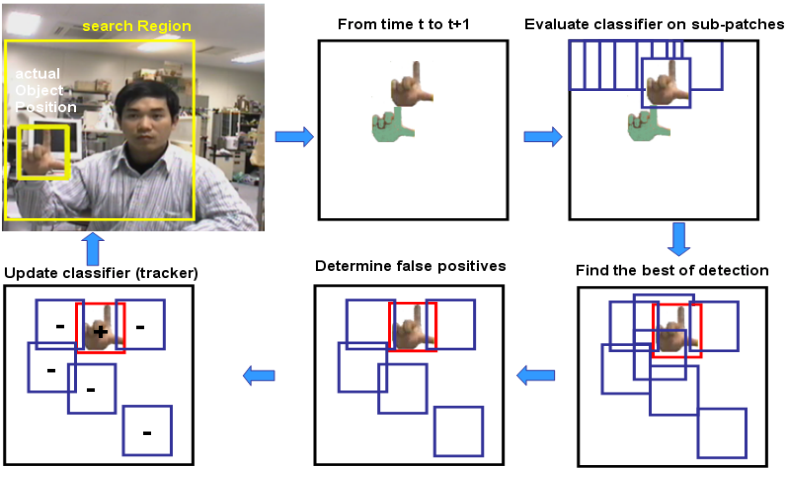

Fig. 1. Autonomous online learning – The main steps of the object's model learning are illustrated. Suitable updates for learning an object detector: Positive (red bounding box) and negatives (blue bounding boxes) as false positives are selected to update at time t.

3.4 Refinement of Generated Data

As we employ autonomous learning to train an object model, the classifier starts from only one initialized hand labelled training sample and performs update autonomously. At the first iteration, the classifier is updated once and then performs evaluation on current image to classify object and non-object classes. Since it is just updated only once, at this step, it has little knowledge about appearance of the object. So, it is rather "weak" in discriminating object class. Therefore, it may produce some wrongly

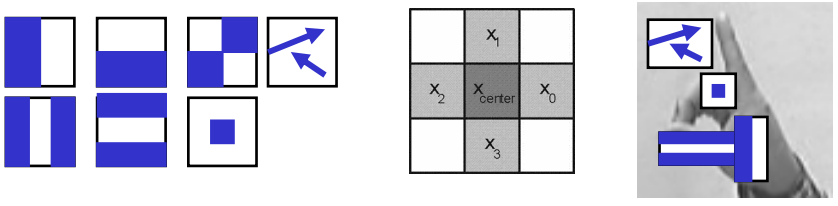

Fig. 2. Basic image features used. Haar-like features from Viola and Jones [3] and in addition orientation histograms (with 16 bins) from [19] and local binary patterns (LBP) [20] as features.

classified object sample, which is a "bad" sample to the database. The same situation may occur even after few iterations, i.e. the classifier has been updated just on few samples and does not cover various changing appearances of the object. Fortunately, because of our intelligent updating strategy, after sufficient number of iterations, the classifier becomes strong, which means the discriminative power has been increased, it can generate reliable samples to update itself and contribute to the database. What we are considering is the "bad" samples that have been added to the database at early stage of learning. The solution would be: to use the strong classifier, after training, to refine the generated training samples in the database. This can be easily done by applying the trained classifier on collected samples in the database. Bad samples, i.e. samples detected with low confident, will be removed. Thus, this results in a more "clean" data.

4 Experiments

In this section, we will demonstrate the capability of our proposed framework for the problem of data acquisition and training an object model from raw data. We conducted experiments on various objects with different complexities. In the following, we will present experiments and results mainly for the problem of human hand and face object learning for detection. But generic object types can also be applied. All experiments are set up as follows: each object model is represented by a classifier, which contains 150 weak classifiers, 50 selectors. First, we randomly initialize parameters for each classifier. Then a specific classifier is train for each object type with one click to initialize the object sample. Resulted video sequences are available on request.

4.1 Data sets

We have performed intensive experiments on several data sets with different complexities. The data sets are typically specific for different applications in computer vision. They include public available data and our recorded video. The goal of this section is to illustrate the effectiveness and robustness of our framework. First, we performed experiments on public available data sets to show our approach ability over very recent proposed approaches for tracking [21, 22]. The data sets include

two typical challenging video sequences which we have selected from [22][1]. Second, to show the robustness of our framework, we perform experiments on several data sets to learn complex objects, include: deformed, articulated object such as hand movement recorded outdoor by a moving camera[2]; object in a context of very similar appearance, such as textured object on the same textured background. Third, further more, we show that our system is able to learn hand gestures model with hands of different persons. Data set for the third experiments includes video sequences which we recorded by a low resolution webcam camera. In the experiments we will show that our framework is able to: autonomous online learning for object's model from simple to complex objects quite well even in difficult circumstance, especially during the object's model learning the system also generates positive and negative samples for building data set.

* *Hand data* – video sequences: The first sequence shows a hand moving from the dark towards bright scene with rapid movements and postures changes, camera motion, changing of lighting conditions and arbitrary backgrounds. Other sequences contain hand movements with different gestures/postures in complex backgrounds.

* *Face data*: The sequence shows a person moving from dark to bright area with pose changes, illumination changes and clutter background.

* *Cars data*: The sequence shows cars in real scenario of out door environment [22].

4.2 Experiment Results

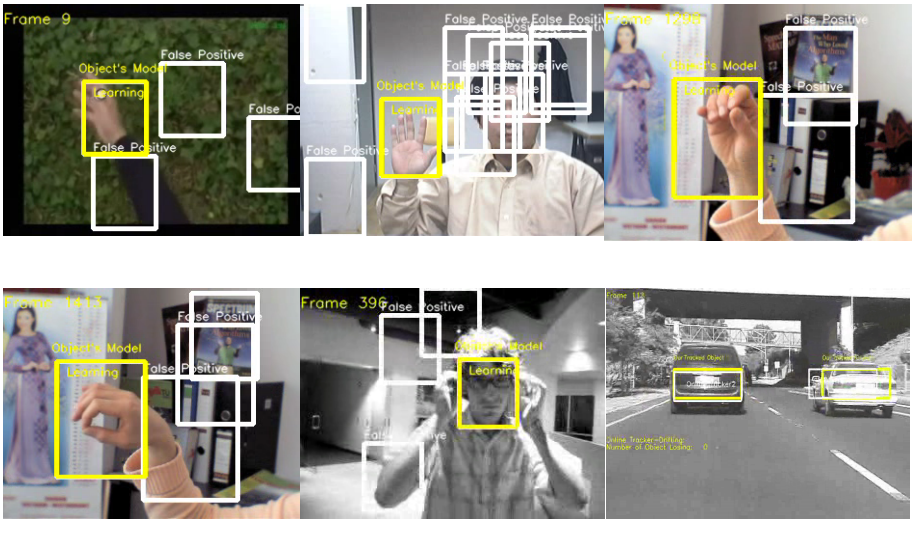

Fig. 3. Tracking results of autonomous object's model learning: true positive sample (yellow box), and false positive samples as negative (white box)

[1] http://www.cs.toronto.edu/~dross/ivt/
[2] http://www.movesinstitute.org/~kolsch/HandVu/VisionBasedHandTracking.wmv

Table 1. The result of object's model learning, number of labeled training samples have been collected (number of positives, negatives) and the detection rates of the test video sequences.

Data sets	Number of image frames have been annotated	Number of training samples	Number of training samples have been generated			Detection rate
			Number of Positive samples	Number of false Positive samples	Number of Negative samples extracted randomly from background	
Hand set 1	659	95	659	67	1343	100%
Hand set 2	600	67	600	190	1328	100%
Hand set 3	1977	129	1977	20	552	99.6%
Hand set 4	2130	135	2130	26	576	99.2%
Face set	462	26	462	126	1498	100%
Car set	659	164	659	183	1473	100%

5 Conclusion

In this paper, we have presented a new framework for generating visual object database meanwhile efficiently learning the object's model. Our system takes the advantages of the online boosting learning approach to build an autonomous learning object's model with reduction to minimal data acquisition effort. The system is composed of two main modules which are co-operated. The learning module, which is a discriminative classifier, learns efficient object representation online. The availability of the online classifier with intelligent update strategy in combining with tracking information allows to collect data samples during learning and to build a visual object database naturally. We have applied our framework for the problems of learning a detector for several object of interest, include: human hand, face and car models. The model learning module outperformed state-of-the-art online boosting learning approach in term of accuracy and stability. Database of each object have been generated efficiently, which contain informative, rich representation of considering object. Experiments have shown various capable applications of our proposed framework for visual data acquisition and object's model learning.

For future work, we plan to study more about the generalization ability of the autonomous learning algorithm. We also plan to use our result for further study of transfer learning. Moreover, multi tasks learning with automatic knowledge acquisition is a topic of interest.

References

1. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proc. CVPR, IEEE, vol. 1, pp. 260–267 (2006)
2. Hertz, T., Bar-Hilled, A., Weinshall, D.: Learning distance functions for image retrieval. In: Proc. CVPR, IEEE, vol. 2, pp. 570–577 (2004)
3. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. CVPR, IEEE, vol. I, pp. 511–518 (2001)

4. Tieu, K., Viola, P.: Boosting image retrieval. In: Proc. CVPR, IEEE, pp. 228–235 (2000)
5. Rowley, H., Baluja, S., Kanade, T.: Neural Network-based Face Detection. IEEE Trans. On PAMI 20(1), 23–38 (1998)
6. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods - Support Vector Learning (1998)
7. Levin, A., Viola, P., Freund, Y.: Unsupervised improvement of visual detectors using co-training. In: Proc. IEEE CVPR, vol. I, pp. 626–633 (2003)
8. Nair, V., Clark, J.: An unsupervised, online learning framework for moving object detection. In: Proc. IEEE CVPR, vol. II, pp. 317–324 (2004)
9. Sung, K., Poggio, T.: Example-based learning for view-based face detection. IEEE Trans. on PAMI 20(1), 39–51 (1998)
10. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and Practice of Background Subtraction. In: Proc. of ICCV, pp. 255–261 (1999)
11. Elgamal, A., Harwood, D., Davis, L.: Non-parametric Model for Background Substraction. In: Proc. of ECCV (2000)
12. Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. In: Proc. ECCV, vol. I, pp. 85–98 (2004)
13. Sivic, J., Everingham, M., Zisserman, A.: Person spotting: Video shot retrieval for face sets. In: Leow, W.-K., Lew, M., Chua, T.-S., Ma, W.-Y., Chaisorn, L., Bakker, E.M. (eds.) CIVR 2005. LNCS, vol. 3568, pp. 226–236. Springer, Heidelberg (2005)
14. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust detectors. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 69–82. Springer, Heidelberg (2004)
15. Hewitt, R., Belongie, S.: Active learning in face recognition: Using tracking to build a face model. In: Proc. IEEE Workshop on Vision for Human-Computer Interaction (2006)
16. Wu, B., Nevatia, R.: Improving part based object detection by unsupervised, online boosting. In: Proc. IEEE Computer vision and Pattern Recognition (2007)
17. Javed, O., Ali, S., Shah, M.: Online detection and classification of moving objects using progressively improving detectors. In: Proc. IEEE CVPR (2005)
18. Oza, N.C., Russell, S.: Experimental comparisons of online and batch versions of bagging and boosting. In: Proc. ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining (2001)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings, CVPR, San Diego, CA, USA, vol. 1, pp. 886–893 (2005)
20. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)
21. Kolsch, M., Turk, M.: Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop, pp. 158–166 (2004)
22. Ross, D., Lim, J., Lin, R., Yang, M.H.: Incremental Learning for Robust Visual Tracking, the International Journal of Computer Vision. Special Issue: Learning for Vision (2007)