ROBUST OBJECT TRACKING WITH RADIAL BASIS FUNCTION NETWORKS

R. Venkatesh Babu

Department of Electrical Engineering,
Indian Institute of Science,
Bangalore, India

S. Suresh *Anamitra Makur*

School of Electrical and Electronics Engineering,
Nanyang Technological University,
Singapore

ABSTRACT

Visual tracking has been a challenging problem in computer vision over the decades. The applications of visual tracking are far-reaching, ranging from surveillance and monitoring to smart rooms. In this paper we present a novel object tracker based on fast learning Radial Basis Function (RBF) networks. Here, the object and background pixel-based color features are used to develop object/non-object RBF classifiers. The posterior probability information of these classifiers are used for developing an efficient object model for tracking in the subsequent frames. The performance of the proposed tracker is tested with many video sequences of real-life complexity and compared against the color-based mean-shift tracker. The proposed tracker is illustrated to be suitable for real-time robust object tracking due to its low computational complexity.

Index Terms— Visual Tracking, Neural Networks, RBF-Neural Networks, Object Tracking

1. INTRODUCTION

The objective of object tracking is to faithfully locate the targets in successive video frames. The major challenges encountered in visual tracking are cluttered background, noise, change in illumination, occlusion and scale/appearance change of the objects. Considerable work has already been done in visual tracking to address the aforementioned challenges [1].

In the last few decades, neural networks have been successfully used in a number of applications such as pattern recognition, remote sensing, dynamic modeling and control and medicine [2, 3, 4]. The increasing popularity of neural networks in many fields is mainly due to their ability to learn the complex nonlinear mapping between the input-output data and generalize them. Also, neural networks make no prior assumption about the statistics of input data and can construct complex decision boundaries [5]. These properties makes neural network an attractive mathematical tool to many practical problems. As one of the most popular neural network models, radial basis function network attracts lots of attentions on the improvement of its approximation ability as well as the construction of its architecture.

In neural network learning algorithms [6, 7], gradient search methodology has been widely used for network parameters update. Hence, the learning process is computationally intensive and may require several hours to train the network. Also, one has to select proper learning parameters (learning rate and epoch) to avoid local minima problems. Higher training time and issues in learning parameter selections leads to development of an alternative algorithm which can be implemented real time. Recently in [8], fast learning neural algorithm called 'Extreme Learning Machine' (ELM) is presented for a single hidden layer neural network (SLFN).

In fast learning algorithm [9], it is shown that for SLFN with radial basis function, the random selection of centers and width's of hidden neurons and analytically calculated output weight can approximate any continuous function to desirable accuracy. This algorithm overcomes many issues in traditional gradient algorithms such as stopping criterion, learning rate, number of epochs and local minima. Since the algorithm is computationally less intensive and has good generalization ability, it is suitable for real time applications. In fact, the performance of fast learning algorithm on many real world problems have been compared with the other neural network approaches in [8] and its performance has been found to be better.

Learning-based tracking algorithms were rarely used for general purpose object tracking [10]. This is due to the difficulty in adapting the neural networks for tracking purpose. Adapting a tracking problem into a classification problem gives a wider scope for modeling the objects using neural networks. Hence in this paper, we propose a novel robust object tracking algorithm using fast learning RBF networks. The object model is developed using two RBF classifiers namely object and non-object. The features used for building RBF classifiers are simple color features. The posterior probability estimated by the neural network is used for target model development [11]. The target localization is achieved by recursively maximizing the posterior probability estimated by the RBF networks. The performance of the tracker is tested with challenging video sequences and compared against the well known mean-shift tracker.

The paper is organized as follows: Section 2 describes the overview of the proposed tracker. Section 3 presents the details of main system modules such as object background separation, feature extraction, objet modeling. The basics of RBF network for modeling the object is presented in section 4. Experimental results and discussions are presented in Section 5. Finally, Section 6 concludes the paper.

2. SYSTEM OVERVIEW

In this paper we present a novel object tracker using RBF networks. The RBF network is trained for classifying object and non-object (background) pixels. The color and location information of every object and surrounding background pixels are used as features for training the network. The training process uses fast learning algorithm [8] and it incurs very low computational effort to develop the neural classifiers. The posterior probabilities of the classifiers are used to develop an efficient object model. Fig. 1(a) illustrates the development of object model using RBF networks. Initially the object of interest is localized by the user by drawing a rectangle around it. The object-background separation module separates the object from the surrounding background pixels by estimating the likelihood map. The feature extraction module, extracts the basic color information

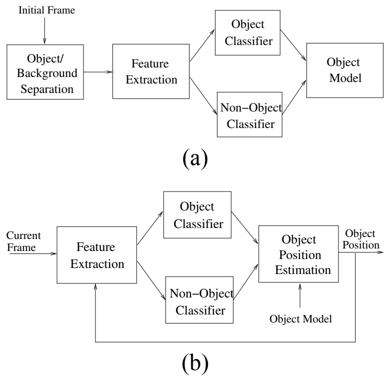


Fig. 1. (a) Object Model development : Training phase (b) Object tracking : Testing Phase.

of the labeled object and background pixels. The object and non-object classifier were tuned to maximize classification accuracy for object and background pixels correspondingly. Only the probability information of reliable object pixels, which are classified as object in both classifiers, are used for building target object model. The tracking phase is illustrated in Fig. 1(b). Object localization starts at the center of the object window in the frame where it was previously tracked. In order to find the object pixels, the extracted features from this location are tested with both object and non-object classifiers. The displacement of the object is given by the shift in centroid of the object pixels. The object location is iteratively shifted and tested until the convergence. The cumulative displacement indicates the shift in object location for the current frame.

3. RBF NETWORKS-BASED OBJECT TRACKER

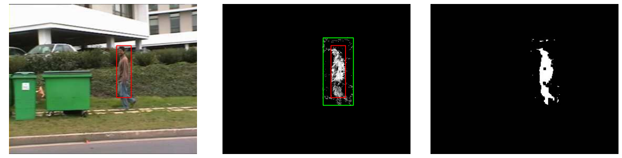
The following subsections explain the main modules of the proposed RBF Networks-based tracking system. In the following subsection, we detail the procedure for separating the foreground and background pixels.

3.1. Object-background separation

Efficient object tracking heavily depends on how well the object is modeled free from background pixels. In order to obtain a reliable object model, we separate the object region from the surrounding background in the first frame of the video sequence. The object-background separation is used for labeling the object and background pixels. The R-G-B based joint probability density function (*pdf*) of the object region and that of a neighborhood surrounding the object is obtained. This process is illustrated in Fig. 2. The region within the red rectangle is used to obtain the object pdf and the region between the green and red rectangles is used for obtaining the background *pdf* [12]. The resulting log-likelihood ratio of foreground/background region is used to determine object pixels. The log-likelihood of a pixel considered within the outer bounding rectangle is (green rectangle in Fig. 2) obtained as

$$L_i = \log \frac{\max\{h_o(i), \epsilon\}}{\max\{h_b(i), \epsilon\}} \tag{1}$$

where $h_o(i)$ and $h_b(i)$ are the probabilities of ith pixel belonging to the object and background respectively; and ϵ is a small non-zero value to avoid numerical instability. The non-linear log-likelihood



(a) (b) (c)

Fig. 2. (a) Initial frame with object boundary (b) likelihood map L (c) Mask obtained after morphological operations (T).

maps the multimodal object/background distribution as positive values for colors associated with foreground and negative values for background. The binary weighting factor M for ith pixel is

$$M_i = \begin{cases} 1 & \text{if} \quad L_i > \tau_o \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where, τ_o is the threshold to decide on the most reliable object pixels. Once the object is localized, by user interaction or detection in the first frame, the likelihood map of the object/background is obtained using (2). In our experiments, τ_o is set at 0.8.

3.2. Feature Extraction

Feature extraction is one of the computationally intensive modules in most of the classification problems. In order to perform object tracking in realtime we need to extract features that can distinguish classes effectively and incurs minimal computational load. In the proposed tracker the feature used for modeling the object are simple pixel color based features. The pixel-based color features such as R-G-B and Y-Cb-Cr are extracted from the given video frame.

4. RADIAL BASIS FUNCTION BASED OBJECT MODEL

In this paper, the object model development is converted to a classification of object pixels from the non-object (background) pixels. The object model is developed using two radial basis function (RBF) classifiers, namely 'Object Classifier' and 'Non-object Classifier'. Here, the object RBF classifier is developed such that it maximizes the classification of number of object pixels in the region of interest. Similarly, the non-object RBF classifier maximizes the classification of number of non-object pixels.

In general, a two-class problem can be stated in the following manner. Suppose, we have the N observation samples $\{U_i, T_i\}_{i=1}^{N}$, where $U_i = [u_{i1}, \cdots, u_{id}] \in \Re^d$ is a d-dimensional feature of ith sample and $T_i \in \{-1, +1\}$ is its coded class label. If sample U_i is assigned to object class then T_i is one otherwise (background class) it is -1. The objective of the classification problem is to estimate the functional relationship between the random samples and its class labels from the known set of data. Here, we use radial basis function classifiers developed using fast learning algorithm [9] to approximate the functional relationship.

It has been proved in literature that the neural network based classifier model developed using mean square error loss function can approximate the posterior probability very well [11]. Since, the fast learning algorithm also uses least square estimate to minimize the error, the output of RBF network approximates the posterior probability. The posterior probability of pixel U_i obtained using the object classifier is

$$\hat{p}_o\left(U_i|c\right) \quad \approx \quad \frac{\max(\min(\hat{Y}, 1), -1) + 1}{2} \tag{3}$$

Similarly, the posterior probability of object pixel U_i obtained using the non-object classifier is

$$\hat{p}_b\left(U_i|c\right) \approx \left(\frac{1-\max(\min(\hat{Y},1),-1)+1}{2}\right) \quad (4)$$

Here, two classifiers are used to estimate the posterior probability of the object pixels reliably. Since, 'object classifier' maximizes the number of object pixels, it might include some of the background pixels as object. Similarly, the 'non-object classifier' might include some object pixels as background. The objective here is to identify the object pixels with high confidence. Hence, we neglect the pixels which are assigned different class labels by object and non-object classifiers. The posterior probability of these two classifiers are used to obtain the object model $\hat{p}_t$ as given below.

$$\hat{p}_t\left(U_i|c\right) = \min\left[\hat{p}_o, \hat{p}_b\right] \quad (5)$$

The posterior probability of the target model is used for localizing object in the subsequent frames.

4.1. Fast Learning Radial Basis Function Classifier

In this section, we present a brief overview of the fast learning algorithm for radial basis function network [9]. Radial basis function network is three layered feed-forward network. The first layer is linear and only distributes the input signal, while the next layer is nonlinear and uses Gaussian functions. The third layer linearly combines the Gaussian outputs.

Using universal approximation property, one can say that the single hidden layer feed-forward network with sufficient number of hidden neurons can approximate any function to any arbitrary level of accuracy. Let $\mu_i \in \Re^d$ and $\sigma_i \in \Re^+$ be the center and width of ith Gaussian hidden neuron, α be $1 \times K$ the output weights and N be the number of pixels. The output $(\hat{Y})$ of RBF network with K neurons has the following form:

$$\hat{Y} = \sum_{j=1}^{K} \alpha_j \exp\left(\frac{-\|U-\mu_j\|}{\sigma_j}\right), \quad (6)$$

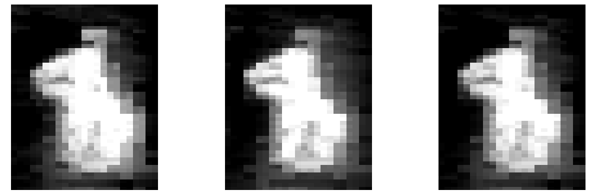
Equation (6) can be written in matrix form as

$$\hat{Y} = \alpha\, Y_H \quad (7)$$

Here, Y_H (is of dimension $K \times N$) is called the hidden layer output matrix of the neural network; the ith row of Y_H is the ith hidden neuron output with respect to inputs $U_1, U_2, \cdots, U_N$. For most of the practical problems, it is assumed that the number of hidden neurons are always less than of number of training samples. In fast learning algorithm, for a given number of hidden neurons (K), it is assumed that the center μ and width σ of hidden neurons are selected randomly. The output weights are estimated analytically as

$$\alpha = TY_H^\dagger \quad (8)$$

where $Y_H^\dagger$ is the Moore-Penrose generalized pseudo-inverse of Y_H. The network output $\hat{Y}$ is used for estimating the posterior probability $(\hat{p}^k)$ using equations (3)-(5).



(a) (b) (c)

Fig. 3. Posterior probability of pixels for a given object window. (a) Object Classifier (b) Non-object Classifier (c) Object Model.

4.2. Object Localization

In this section we explain the development of object model from RBF classifiers and object localization based on the estimated posterior probability. The posterior probability of the current object window estimated by the object and non-object classifiers for the object window (starting frame) of PETS video sequence are shown in Fig. 3(a)-(b) respectively. The corresponding classification matrix are given below.

$$C_o = \begin{bmatrix} 730 & 47 \\ 38 & 358 \end{bmatrix} \qquad C_b = \begin{bmatrix} 747 & 69 \\ 21 & 336 \end{bmatrix} \quad (9)$$

From (9), we can observe that, object classifier maximizes the classification accuracy for object class $(C_o(2,2) = 358 > C_b(2,2) = 336)$. Similarly non-object classifier maximizes the classification accuracy for the background class $(C_b(1,1) = 747 > C_o(1,1) = 730)$. The object model $(\hat{p}_t)$ is developed using (5) and is shown in Fig. 3(c). This target model uses only reliable object pixels which are classified as object in both classifiers.

Let X_c^k be the object center, X_i^k be the candidate pixel locations (centered around X_c^k) and $\hat{p}_c^k(i)$ be the corresponding posterior probability of ith candidate pixel at kth iteration. The posterior probability at kth iteration $(\hat{p}_c^k)$ is obtained by testing the features obtained from locations X_i^k. Now the new location of the object center is estimated as the centroid of posterior probability $(\hat{p}_c^k)$ weighted by target model $(\hat{p}_t)$.

$$X_c^{k+1} = \frac{\sum_i X_i^k \hat{p}_c^k(i)\hat{p}_t(i)}{\sum_i \hat{p}_c^k(i)\hat{p}_t(i)} \quad (10)$$

The iteration is terminated when the change in centroid location for any two consecutive iteration falls below a threshold t_s. Typical value of t_s used in our experiments is in the range of 0-2.

4.3. RBF-Networks Based Tracker Algorithm

1. Select the object to be tracked in the initial frame by selecting a rectangle window around it.
2. Separate the object from background based on the object likelihood map.
3. Extract the object and background features from the labeled object and neighboring background pixels.
4. Obtain the object model using the RBF- network.
5. Test for object in the next frame starting from the previously obtained object location.
6. Recursively obtain the object location for the current frame using (10).
7. Go to step 5.

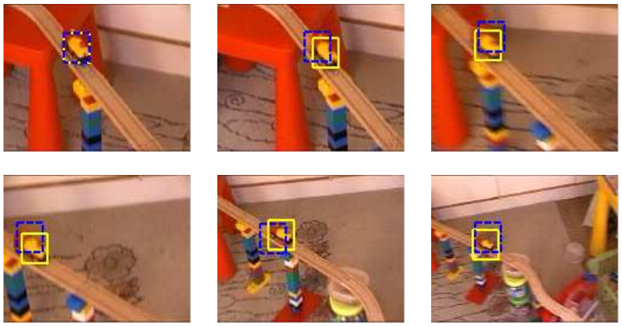


Fig. 4. Tracking result of proposed system (solid yellow) against mean-shift (dashed blue) tracker for ‘train’ sequence. Frames shown 1, 10, 30, 60, 100 and 150.

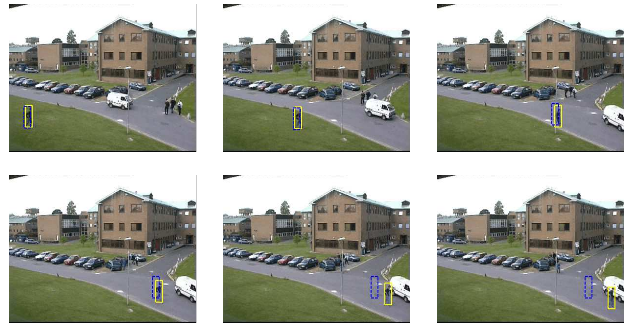


Fig. 5. Tracking result of proposed system (solid yellow) against mean-shift (dashed blue) tracker for ‘pets’ sequence.

5. EXPERIMENTS AND DISCUSSIONS

The proposed algorithm has been tested on several complex video sequences. The ’train’ and ’walk’ videos used in our experiments are shot by a handheld camcorder, and hence include a wide variety of camera operations. For example the train sequence contains lots of jerky motion with camera pan, tilt and zoom operations. We have compared the performance of the proposed tracker against the mean-shift (MS) tracker [13], which is known for robust object tracking in cluttered environment. Fig. 4 shows the tracking result for the proposed and MS tracker for a toy train moving in a cluttered background. Though both the proposed and the mean-shift tracker tracks the object, the accuracy of the MS tracker degrades very much during large displacement of the object due to camera panning operation. The last 2 frames of Fig. 4 shows how the accuracy of MS tracker reduces with camera zoom-out operation. The proposed algorithm tracks the object with a better precision during the aforementioned camera operations. Fig. 5 shows the tracking results for a PETS sequence. The proposed algorithm tracks the object through out the sequence with better accuracy compared to the MS algorithm. The MS tracker fails at the end part of the sequence. The proposed tracker has been tested for various speeds by temporally downsampling the video. For example the result shown in Fig. 5 is for downsampling factor 4. These results clearly show that the fast learning neural networks are well suited for tracking objects in real-world scenario. Further, the RBF tracker uses, on an average, 1-2 iteration to localize the object and the computational effort required to calculate the posterior probabilities in the testing phase is negligible. Hence the proposed tracker is suitable for real-time applications.

6. CONCLUSIONS

In this paper we have proposed an object tracker using RBF-networks. The fast learning algorithm is used to develop the object model at the initial frame using two RBF classifiers. The object localization is achieved using the target model and the posterior probability of the current object window. The performance of the proposed tracker is compared with the well-known MS tracker for various complex video sequences. The proposed tracker provides better tracking accuracy compared to MS tracker. Since the testing phase of RBF network incurs very low computational burden, the proposed tracker is suitable for real-time applications.

7. REFERENCES

[1] A. Blake, “Visual tracking: a very short research roadmap,” *Electronics Letters*, vol. 42, no. 5, pp. 254–256, 2006.

[2] Yi Lu Murphey and Yun Luo, “Feature Extraction for a Multiple Pattern Classification Neural Network System,” *Pattern Recognition Proceedings*, vol. 2, pp. 220–223, 2002.

[3] M. Voultsidou, S. Dodel, and J. M. Herrmann, “Neural Networks Approach to Clustering of Activity in fMRI Data,” *IEEE Trans. on Medical Imaging*, vol. 24, no. 8, pp. 987–996, Aug. 2005.

[4] E. Trentin and M. Gori, “Robust combination of neural networks and hidden markov models for speech recognition,” *IEEE Trans. on Neural Networks*, vol. 14, no. 6, pp. 1519–1531, Nov. 2003.

[5] C. Lee and D. A. Landgrebe, “Decision boundary feature extraction for neural networks,” *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 75–83, 1997.

[6] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris, and D. M. Hummels, “On training of radial basis function classifiers,” *Neural Networks*, vol. 5, pp. 595–603, july-august 1992.

[7] J. Moody and C. J. Darken, “Fast learning in network of locally-tuned processing units,” *Neural Computation*, vol. 1, pp. 281–294, 1989.

[8] G.-B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, 2006.

[9] G.-B. Huang and C.-K. Siew, “Extreme learning machine with randomly assigned rbf kernels,” *International Journal of Information Technology*, vol. 11, no. 1, pp. 16–24, 2005.

[10] S. Avidan, “Support vector tracking,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 184–191.

[11] R. Rojas, “A short proof of the posterior probability property of classifier neural networks,” *Neural Computation*, vol. 8, no. 2, pp. 41–43, 1996.

[12] R. T. Collins, “Mean-shift blob tracking through scale space,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, 2003, pp. 234–241.

[13] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 5, pp. 564–577, 2003.