Interaction Between Object Detection and Multi-Target Tracking

Wang Zhiming, Bao Hong
School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China
wangzhiming@ies.ustb.edu.cn

***Abstract*—Object detection and target tracking are two basic tasks in video analysis and understanding. Though both of them were studied widely and deeply, interaction between them worth more efforts. We give an interaction framework between PNN based object detection and meanshift target tracking. Detection results were used to accelerate tracking speed by decrease search steps, and tracking results were used to guide the updating of background model for motion detection. Performances of both multi-object tracking and motion detection were improved.**

Keywords-Object detection, Target tracking, Background model, Meanshift, Neural network

I. INTRODUCTION

Intelligent video processing has been widely used in various environments such as office building, airport, subway, etc. One of the most important tasks of video processing system is to detect moving objects from video. It is the base for succeeding object tracking, activity recognition, and behavior understanding. Various background models were proposed for background subtraction, including single Gaussian model, mixture of Gaussian (MoG), non-parametric model based on Bayesian classification [1], neural network [2, 3], etc. But motion detection always suffers from complex background with light changing, background disturbance, and object shadows.

Multi-target tracking has been proved another tough task in intelligent video processing. Many multi-target tracking researches focus on improvement of tracking algorithm, or combining different tracking techniques such as mean shift and particle filter. For example, Khan [4] combines particle filter and anisotropic mean shift, and objects were partitioned into non-overlapping sub-regions to enhance the tracking robustness to partial occlusions.

Perera [5] used nearest-neighbor data association strategy to initialize targets and using the Hungarian algorithm to solve the one-to-one correspondence assignment problem for multi-object tracking. As Hungarian algorithm needs high computation complexity, Reilly [6] divided the scene into grid cells and the Hungarian algorithm is then used to estimate the association of detections in every cell, which reduced computation dramatically. Prokaj [7] proposed a tracklet inferring algorithm for multi-object matching based on Basyesian network and MAP estimation.

But none of them take full advantage of interaction between detection and tracking. For example, tracking results were not feedback to motion detection model.

In this paper we proposed an interaction algorithm between motion detection and object tracking, try to improve performance of multi-object tracking as well as motion detection results.

II. FRAMEWORK OF PROPOSED ALGORITHM

In our interaction framework, motion detection is achieved by PNN (Probability Neural Network), and Object tracking is implemented by meanshift tracking algorithm. As shown in Fig. 1.

In the first frame, objects were initialized by motion detection and object segmentation. In succeeding frames, every detected region was matched with exist object, and meanshift tracking was accelerated by match results. Motion detection background model was also updated by tracking result, which improve the precision of next frame's motion detection result.

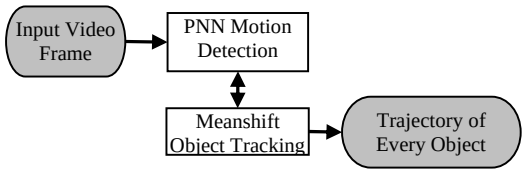


Figure 1. Framework of interaction between detection and tracking

III. PNN BASED MOTION DETECTION

In PNN based motion detection [8,9], every pixel is classified into foreground or background by a hybrid PNN and WTA based neural network. Fig. 2 gives the structure of PNN based motion detection neural network for every pixel.

The whole network contains four layers. The first layer is the input layer, which just accepts the pixel value (HSV).

The second layer, called feature layer, transforms HSV value to some feature data more suitable for classification.

$$(x_V, x_S, x_H) \Rightarrow (x_V x_S \cos(x_H), x_V x_S \sin(x_H), x_V) \quad (1)$$

The third layer is pattern layer, which is a Parzen probability estimator. Every pattern neuron represents a pixel pattern, and it was used as an independent estimator. It gives the conditional probability of current pixel (with given features) belongs to this pattern. The probability is computed by multiply the output of the pattern neural with weight adaptively learned during online processing.

Published by Atlantis Press, Paris, France.

Prior conditional probability of one pattern node belong to background p(B|$\mathbf{b}_i$) was stored in connect weights from pattern neuron to classification neuron. Conditional probability of current pixel belongs to a pattern node ($\mathbf{b}_i|\mathbf{x}$) was estimated by:

$$p(\mathbf{b}_i \mid \mathbf{x}) = \exp(-d^2(\mathbf{x},\mathbf{b}_i)/2\sigma^2) \quad (2)$$

where, $\mathbf{x} = \{x_H, x_S, x_V\}^T, \mathbf{b}_i = \{u_{iH}, u_{iS}, u_{iV}\}^T$ give pixel value and model value of the *i*th pattern neuron. d($\mathbf{x}$,$\mathbf{b}_i$) is the distance between $\mathbf{x}$ and $\mathbf{b}$, defined by:

$$d(\mathbf{x},\mathbf{y}) = \| (x_V x_S \cos(x_H), x_V x_S \sin(x_H), x_V) - (x_V x_S \cos(x_H), x_V x_S \sin(x_H), x_V) \|_2 \quad (3)$$

σ is a smoothing parameter. *N* is number of pattern neurons.

The fourth layer, called output layer, includes two neurons with different functions. One is classification neuron, which is a WTA (winner take all) neuron. It selects the maximum value from all of its inputs, and gives the result by comparing this maximum value with a given threshold.

Another neuron in output layer is an activation neuron, which also works in a WTA manner, but only gives the index of pattern neuron with maximum probability (output). All weights from pattern neuron to the activation neuron is 1.

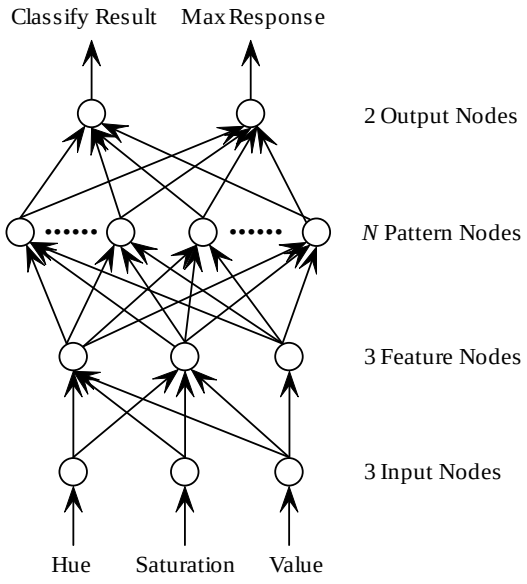


Figure 2. Neural network for motion detection

Response of classification neuron is defined by:

$$O_1 = \begin{cases} 1 & \max\{\ p_i(B \mid \mathbf{x})\} \geq \theta_1 \\ 0 & otherwise \end{cases} \quad (4)$$

$$p_i(B \mid \mathbf{x}) = p(\mathbf{b}_i \mid \mathbf{x}) \cdot p(B \mid \mathbf{b}_i) = p_i w_i \quad (5)$$

p_i (i=1,2,…,N) is output of pattern layer neurons, and w_i (i=1,2,…,N) is corresponding connecting weight to classification neuron. If one of its input (pattern node output multiply its weight) greater than threshold θ_1, current pixel is classified to background (output '1'). Otherwise, it is classified into foreground (output '0').

Response of activation neuron is defined by:

$$O_2 = \begin{cases} \arg\max\{p_i\}, & \max\{p_i\} \geq \theta_2 \\ 0 & otherwise \end{cases} \quad (6)$$

If max(p_i) (i=1,2,…,N) grater than a predefined threshold θ_2, it output the index of the pattern neuron with maximum output. Otherwise, '0' is give for none of the pattern neuron is activated. The output of activation neuron is used to guide the thereafter weight updating process.

After pixel classification, model parameters were updated for every pixel. Weights between pattern neuron and classification neurons were updated according to following rule:

$$\begin{cases} w_i^{t+1} = \min(1, w_i^t + \beta^t), & i = i_{\max} \\ w_i^{t+1} = (1 - \frac{\beta^t}{N}) \cdot w_i^t, & otherwise \end{cases} \quad (9)$$

w_i^t is weight of *i*th pattern neuron in time *t*, β^t is learning rate in time *t*, $i_{\max}$ is the maximum index of pattern neuron outputted by activation neuron. If none of the patterns is activated, all of the weights were reduced.

Learning rate is a very important for model update. Small value makes network adapt to scene change slowly, but large value often makes slowly moving object being misclassified to background.

In [9], learning rate was computed by the ratio of motion different and total pixel number:

$$\beta^t = \min(\beta_{\max}, \beta_{\min} + \Delta n^t / n) \quad (8)$$

$\beta_{\max}, \beta_{\min}$ are upper and lower boundary of learning rate, and satisfies $0 < \beta_{\min} < \beta_{\max} < 1$. n gives the overall pixel number respectively, and Δn_t is the absolute different pixel number between foreground pixels in current frame and last frame. (8) means if the total motion pixel changes dramatically, learning rate should be large.

IV. Object Tracking by Meanshift

Mean shift tracking [10] performs mean shift algorithm on probability distributions. Color object was represented as probability distribution by color histogram. Color histogram in HSV space was calculated and binned into 1D histogram. As image sequences change over time, mean shift dynamically search the position and size with highest probability.

Mean shift tracking maximize following Bhattacharyya coefficient between two histograms:

$$\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}),\hat{\mathbf{q}}]=\sum\sqrt{\hat{p}_u(\hat{\mathbf{y}})\hat{q}_u}$$

Here $\hat{\mathbf{q}}=\{\hat{q}_u\}_{u=1,\ldots,m}, \sum_{u=1}^{m}\hat{q}_u=1$ is estimated *m*-bin histogram of the target model, **y** is the candidate location, and $\hat{\mathbf{p}}(\mathbf{y})=\{\hat{p}_u(\mathbf{y})\}_{u=1,\ldots,m}, \sum_{u=1}^{m}\hat{p}_u=1$ is estimated at a given location **y** from the *m*-bin histogram of target candidate. Maximization of $\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}),\hat{\mathbf{q}}]$ results following tracking algorithm.

Mean shift tracking algorithm:

1. Compute the m-bin histogram of the target model:

$$\hat{\mathbf{q}}=\{\hat{q}_u\}_{u=1,\ldots,m}, \sum_{u=1}^{m}\hat{q}_u=1$$

2. For every search size:

2.1 Compute the m-bin histogram of the estimated target at location $\mathbf{y_0}$:

$$\hat{\mathbf{p}}(\mathbf{y}_0)=\{\hat{p}_u(\mathbf{y}_0)\}_{u=1,\ldots,m}, \sum_{u=1}^{m}\hat{p}_u=1$$

2.2 Compute the weight by

$$w_i=\sum_{u=1}^{m}\delta[b(\mathbf{x}_i)-u]\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}}$$

2.3 Derive the new location of target by mean-shift:

$$\hat{\mathbf{y}}_1=\frac{\sum_{i=1}^{n_h}\mathbf{x}_i w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0-\mathbf{x}_i}{h}\right\|^2\right)}$$

2.4 If $\|\hat{\mathbf{y}}_1-\hat{\mathbf{y}}_0\|<\varepsilon$ stop;

Otherwise $\hat{\mathbf{y}}_0=\hat{\mathbf{y}}_1$, go to step 2.1.

3. Find the best size with maximum Bhattacharyya coefficient:

$$\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_1),\hat{\mathbf{q}}]=\sum\sqrt{\hat{p}_u(\hat{\mathbf{y}}_1)\hat{q}_u}$$

g(.) is a normalized Gaussian kernel for spacial distance weight, and *h* is the sigma parameter, n_h is the pixel number in a given search size.

V. INTERACTION BETWEEN DETECTION ADN TRACKING

The detail flow chart of interaction between detection and tracking is show in Fig. 3, which include following steps:

1. Detect all foreground pixels by PNN and label every motion region by binary image segmentation.

2. Match every region to current tracking regions by spacial distance, object size and Bhattacharyya coefficient between color histogram of two regions.

3. Track every non-matched region with mean shift tracking algorithm.

4. Update background PNN model based on both detection result and tracking result.

If a tracking region in previous frame is matched with a motion region in current frame, it needn't track at all, which will save a great deal of computation. On the other hand, if a region is missed by motion detection (for example, a person stand still for a long time), it can be find easily by mean shift tracking.

In the original PNN background model, background update after every frame. If a moving object stays for a long time, it will gradually blend into background. Finally, it will lose in foreground detection. But in our model, as long as it has been tracked successfully, background model for these pixels won't be changed, and it could be detected as well.

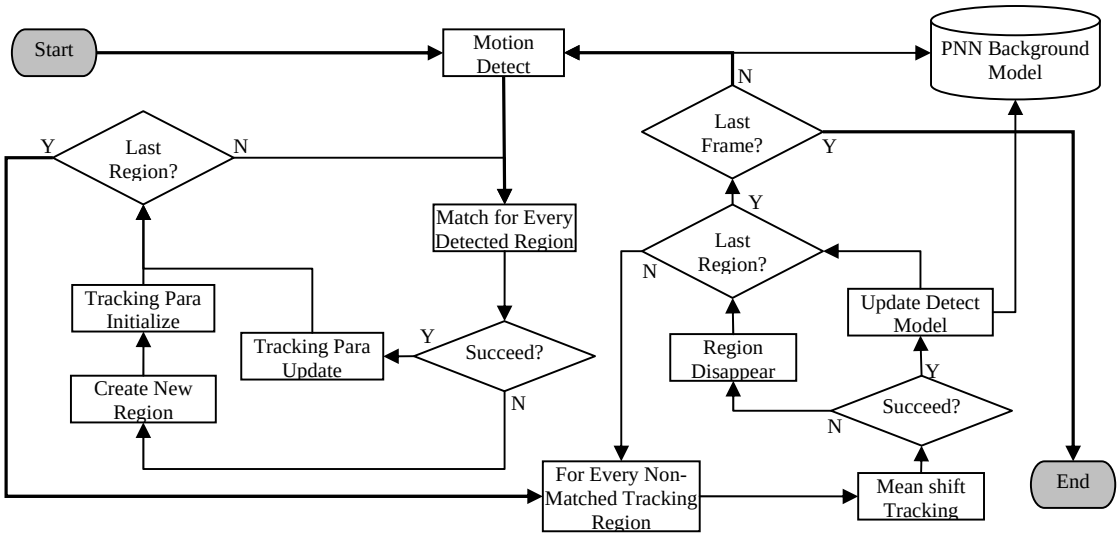


Figure 3. Framework of interaction between detection and tracking

I. EXPERIMENTAL RESULTS

Two experiments were carried out to validate efficiency of the proposed algorithm. The first is to compare the mean search steps for object tracking with and without proposed interaction strategy. The second is to compare the motion detection accuracy with and without interaction.

The first experiment was taken on three image sequences. One is an indoor image sequences provided by National Research Council of Naples from Italy (MSA)[3], two others are bi-channel image sequences provided by Ohio State University [11], which include 6 couples of visible color image sequence and thermal gray image sequence from two scenarios, and we used two of them from different scenarios (OTCBVS1 and OTCBVS4).

Detect and track examples are show in figure 4. Frame number for every image sequence, mean target number per frame, and mean search steps per frame or per target are listed in table 1. Due to some error report, mean target number is greater than true target number. But the relative search steps comparison make some sense. It can be found in table 1 that with our interaction strategy mean track steps decreased dramatically, reduced to about one fifth to one tenth.

The second experiment was taken on MSA[3]. It includes 528 frames of visible color image sequence show a man walked across and shows some actions, and put left a black bag in the scenery. Without interaction, the bag will gradually blend into background after it was left there still. But with detection and tracking interaction processing, it will not update the background as long as the bag was successfully tracked. Figure 5 show the motion detection results between with and without interaction. Obviously, motion detection results were improved with interaction.

Figure 4. Test image sequences and detect and track examples for three test image sequences

TABLE 1. MEAN SEARCH STEPS IN TRACKING COMPARISON BETWEEN WITH AND WITHOUT INTERACTION

Methods	Video	Frame Number	Mean Target Number Per Frame	Mean Search Steps Per Frame	Mean Search Steps Per Target
Without Interaction	MSA	528	1.697	77.186	45.484
	OTCBVS1	1054	5.119	58.807	11.489
	OTCBVS4	1506	0.572	6.711	11.739
With Interaction	MSA	528	0.945	8.788	9.299
	OTCBVS1	1054	5.289	12.557	2.374
	OTCBVS4	1506	0.572	0.819	1.433

II. CONCLUSIONS

An interaction algorithm between motion detection and multi target tracking was given in this paper. Motion regions were detected by PNN based neural network background model, and targets were tracked by meanshift algorithm. In the process of tracking, every target is matched to all detected object first by distance, size and texture. It greatly reduced search steps for every target. In the process of motion detection, target tracking results were used to guide the updating of the background model. Experimental results on three image sequences from different scenarios validated the efficiency of proposed algorithm. Mean search steps for every target were reduced to about one fifth or one tenth of original tracking algorithm, and motion detection results were improved evidently when there is a stand still object.

Further research works including more intelligent background update strategy and target separation when there is a heavy occlusion or overlap between targets.

ACKNOWLEDGMENT

The research is financially supported by National Natural Science Foundation of China under grant No. 61040038.

REFERENCES

[1] L. Li, W. Huang, I. Gu, *et al*, "Statistical modeling of complex backgrounds for foreground object detection," IEEE Transaction on Image Processing, 2004, 13(11): 1459~1472.

[2] D. Culibrk, O. Marques, D. Socek, *et al*. "Neural Network Approach to Background Modeling for Video Object Segmentation", IEEE Transactions on Neural Networks, 2007, 18(6): 1614~1627.

[3] L. Maddalena, A. Petrosino, "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications", IEEE Transactions on Image Processing, 2008, 17(7): 1168~1177.

[4] Z. H. Khan, I. Y. H. Gu, A. G. Backhouse, Robust Visual Object Tracking Using Multi-Mode Anisotropic Mean Shift and Particle Filters, IEEE Transactions on Circuits and Systems for Video Technology, 2011, 21(1):74~87.

[5] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 666–673, 2006

[6] V. Reilly, H. Idrees, and M. Shah. Detection and tracking of large number of targets in wide area surveillance. Proceedings of the European Conference on Computer Vision (ECCV), vol. 6313, pp. 186~199, 2010.

[7] J. Prokaj, M. Duchaineau, and G. Medioni. Inferring Tracklets for Multi-Object Tracking. 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2011.

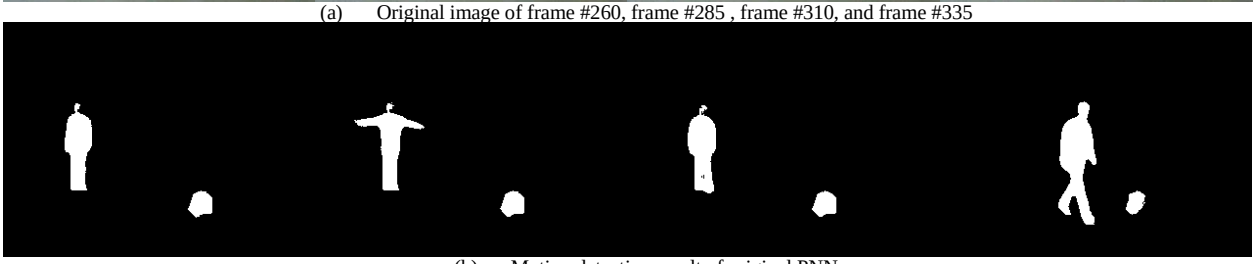
[8] WANG Zhiming, ZHANG Li, BAO Hong, PNN Based Motion Detection with Adaptive Learning Rate, 2009 International Conference on Computational Intelligence and Security, Beijing, China, Dec. 11~14, 2009.

[9] WANG Zhiming, BAO Hong, ZHANG Li, Adaptive Background Model Based on Hybrid Structure Neural Network, Acta Electronica Sinica, 2011, 39(5): 1053~1058.

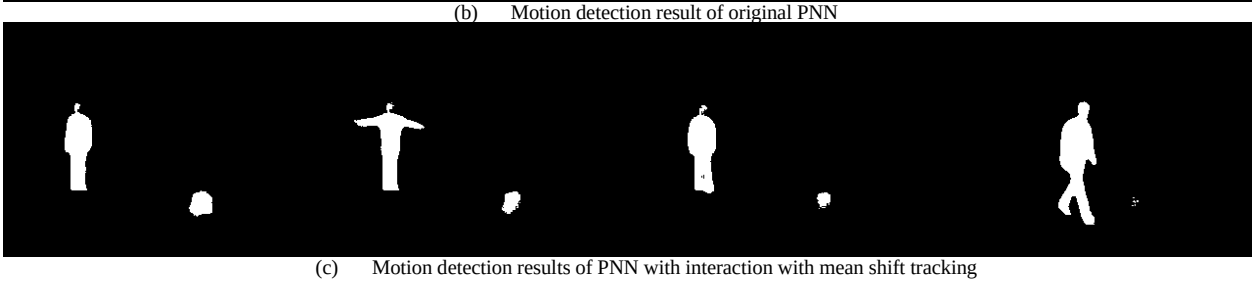
[10] D. Comaniciu, V. Ramesh, Mean shift and optimal prediction for efficient object tracking, Proceedings of International Conference on Image Processing, 2000, Vol. 3, pp.70~73.

[11] J. Davis, V. Sharma, “Background-Subtraction using Contour-based Fusion of Thermal and Visible Imagery”, Computer Vision and Image Understanding, 106(2007): 162~182.

(a) Original image of frame #260, frame #285 , frame #310, and frame #335



(b) Motion detection result of original PNN



(c) Motion detection results of PNN with interaction with mean shift tracking

Figure 5. Motion detection results comparison