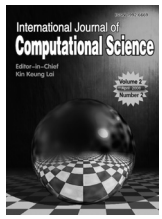
International Journal of Computational Science
1992-6669 (Print) 1992-6677 (Online) www.gip.hk/ijcs

2008, Vol. 2, No. 2, 274-285.



Object Location and Track in Image Sequences by Means of Neural Networks †

Zhenghao Shi[1, 2, 5]*, Yuyan Chao[4], Lifeng He[3], Kenji Suzuki[1], Tsuyoshi Nakamura[2], Hidenori Itoh[2]

[1] Department of Radiology, The University of Chicago, 5841 South Maryland Avenue, MC 2026,Chicago, IL 60637, USA
zhshi@ieee.org, suzuki@uchicago.edu
[2] School of Computer Science and Engineering, Nagoya Institute of Technology, 464-8555, Japan
{tnaka, itoh}@juno.ics.nitech.ac.jp
[3] Graduate School of Information Science and Technology, Aichi Prefectural University, Nagakute, Aichi, 480-1198, Japan
he_4005@yahoo.co.jp
[4] Faculty of Environment and Information Management, Nagoya Sangyo University, Owariasahi-Shi, Aichi 488-8711, Japan
chao@nagoya-su.ac.jp
[5] School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, 10048, China

Abstract. Object location and track in image sequences is an important task in computer vision, which has many applications. Major challenges of object track have been, and continued to be, improvement of its accuracy and real-time performance. In this paper, a novel BP-neural-network-based object location approach is proposed, in which a threshold for the object-matching quality is used for determining whether the object is present in a given frame. To simplify the network structure, a directional wavelet transform (DWT) is used for extracting image features, which can reduce the size of the input patterns. In order to further improve the computation speed of the method, the information on the position of the target object in the previous frames is used for predicting the position of the target object in the current

† This work was supported by Hori Information Science Promotion Foundation.
* Corresponding Author. Email: zhshi@ieee.org.

frame. Experiments indicate that the proposed method is more accurate in target detection and more computationally efficient than conventional methods.

Keywords: object location, object track, neural networks, image sequences .

1 Introduction

Interests in object location and track in image sequences have increased significantly over the past few years, and many methods have been proposed. The two mainstream methods are Template matching methods and feature invariant approaches [1]-[6]. Template matching methods store several patterns of objects and describe each pattern by image pixels. The correlation among an input image and the stored patterns is computed for detecting objects. The principles are simple. However, the sum of the image pixels is great and the intensity value of each pixel is sensitive to changes in absolute intensity, contrast and illumination. Hence, template matching techniques usually need expensive computation and give low accuracy results. Feature invariant approaches, on the other hand, use symbolic features derived from intensity images. Such features include points defined by local intensity, extremes, edges, corners, and regions. Because feature-based techniques allow simple comparisons between the attributes of features, they are generally faster and accurate than template matching methods. However, since they rely on single scale segmentation to extract features (e.g., edges, corners, regions), difficulty of finding feature correspondences across images is increased by segmentation errors [5] [6]. Thus when local image feature information is insufficient, the method will fail to work.

In recent years, ANN (Artificial Neural Network) has attracted considerable attention for object location and track in image sequences because of its capability of high-speed information processing and uncertainty information processing [7]-[11]. Nasrabadi and W. Li [8]-[9] and Shi et.al.[10] used a two dimensional Hopfield network to perform a sub-image isomorphism to obtain the optimal compatible matches between the two images with application in object recognition. N.SANG [11] used a relaxation labeling method to perform invariant matching between patterns.

Differing from above works, this paper focuses on the application of BP (Back Propagation, BP) neural network [12] for object location in image sequences. A BP neural network has the advantages that it is easy to understand and can be efficiently implemented in real-time hardware. Compared with the neural network based methods mentioned above, the proposed algorithm in this paper has a very good advantage. It uses the location predicted from previous image sequences as the candidate position for detection and location the target object and uses a threshold for the object-matching quality for determining whether the object is present in a given frame, which is very beneficial to reduce computational cost and to improve robustness to object lost. By employing the capabilities of the BP network in functional approximation and generalization to learn the non-structured knowledge required, higher computing speed and higher accuracy in real time

object location are both achieved. Experimental results (performed under different conditions) indicate that the proposed method is very promising.

The remainder of this paper is organized as follows: Section 2 describes the proposed method in detail. Section 3 shows the experimental results conclusions of this paper with remarks and suggestions for future work are shown in Section 4.

2 The Proposed Method

2.1 Design of Neural Networks

Design of the best neural network for a considered application should be constrained by the trade off among the training time, the required memory, the computational complexity and the computational time, other than the probability of success.

The objective of our research is to location and track an object in the frames grabbed from a movie clip playing at the speed of 25 frames per second. In order to guarantee the real time behavior of the systems, the location should be performed as fast as possible.

Base on what mentioned above, a three layers BP (Back Propagation, BP) neural network (namely an input layer, a hidden layer, and an output layer) is constructed to perform the operation in this paper. It has been noted that a back propagation neural network (BPNN) with one (or more) sigmoid-type hidden layer(s) and a linear output layer can approximate any arbitrary function [12].

In principle, a BP neural network may be trained to locate an object in images directly. However, for even a moderate image size, the network can be quite complex. For example, if the images were 128 x 128, and if all the pixels are directly put into the neural network, the number of inputs of the network would be 16384, it will be very difficult to process the images in real-time with a standard PC. Therefore, a preprocessing stage must be incorporated to reduce the size of the input pattern.

For the purpose of reducing the size of an input pattern, there are many methods can be employed to perform this task, such as principal component analysis, factor analysis and DWT (Directional Wavelet Transform, DWT) [13]-[16]. Usually, the edges and textures of images sometimes have strong directionality, which is usually very useful for image analysis, especially in the problem of real time object tracking, since the real time images are usually small, directionality of the image features are more obvious and can be used for image matching. However, the directionality of image edges and textures usually appears in the local change of an image or entirety along some directions. This kind of properties can not be incarnated by the image features extracted with a conventional method, such as principal component analysis, factor analysis. However the direction wavelet transform is well able to describe this kind of image properties, which has very strong capability in resisting grayness reversal and noise [16]. Due to this reason, in this paper, DWT is selected to perform this task. The DWT can be depicted as following [16]:

$$DWf(s,\gamma,\theta)=\int_R\int_R f(x,y)\psi_s(x\cos\theta+y\sin\theta-\gamma)dxdy \tag{1}$$

Its corresponding discrete formation is written as

$$\{(s,\gamma_i(s,\theta),\theta)\}_{s\in S,\theta\in\Omega,1<i<L} \tag{2}$$

Where $\psi_s(x)=(\frac{1}{s})\psi(\frac{x}{s})$ denotes a conventional wavelet function. $f(x,y)$ is an arbitrary function of a linear space $B=\{f(x,y)\in L^2(R^2)\}$, $x\cos\theta+y\sin\theta-\gamma=0$ denotes a line in R^2, which represents a direction of the directional wavelet transform. γ and θ reflect the mean value of local variable rates of $f(x,y)$ in the direction V= ($\sin\theta$, $\cos\theta$) along the line $x\cos\theta+y\sin\theta-\gamma=0$. s denotes the directional wavelet transform scale. S and Ω denote the selected scale and direction angle set. For each $s\in S,\theta\in\Omega$, a value set corresponding to the L extremes of $\{DWf(s,\gamma,\theta)\}_{,\gamma\in Z}$ is extracted. All extracted value sets consist of the image feature set, its feature number is determined by the following expression:

$$Num_{feature}=S\times\Omega\times L \tag{3}$$

Since $DWf(s,\gamma,\theta)$ is only dependent upon the local varieties γ, θ, and integrals of an image in a certain direction V=($\sin\theta$, $\cos\theta$) along the line $x\cos\theta+y\sin\theta-\gamma=0$, the DWT not only holds the local analysis ability of conventional wavelet transform in spatial and frequency domain, but also holds directional analysis ability, these ensure that even if there exist grayness reversal and noise in an image, the DWT corresponding to each image feature point is still stable. This indicates that the DWT of an image is a robust image feature.

In this paper, experimental parameters are chosen as: S={1, 2}, Ω={0, 0.3925, 0.785, 1.175, 1.57, 1.9625, 2.355, 2.7475}, L=3, so for each image block, the feature number is 48. Therefore the number of input neurons in the proposed network is 48.

The output layer of the network is designed according to the need of the application output. Since the output of the neural network is expected to detect and location a special object in image sequences, so the number of output neuron is designed as 2, which is expected to produce the row and column coordinates of the target.

Hidden layer automatically extracts the features of the input pattern, and reduces its dimensionality further. There is no definite formula to determine the number of hidden neurons. In this research, the following trial-and-error process was used to identify the number of neurons in the single hidden layer:

First, the initial neurons number in hidden layer is given by an empirical equation:

$$h=\sqrt{I+O}+\alpha \tag{4}$$

Where h is the number of hidden layer, I and O are the numbers of the input layer and output layer respectively, α is a constant and $\alpha\in[1,10]$. Then the number is updated based on experiment results.

By use of the results of the experimental analysis described above, the number of hidden units was determined to be 10 units.

Thus, the numbers of units in the input, hidden, and output layers were 48,10 and 2, respectively.

2.2 Object Location and Track by BP Neural Network

To locate and track an object in image sequences based on the neural network designed above include two stages (As shown in figure 1):

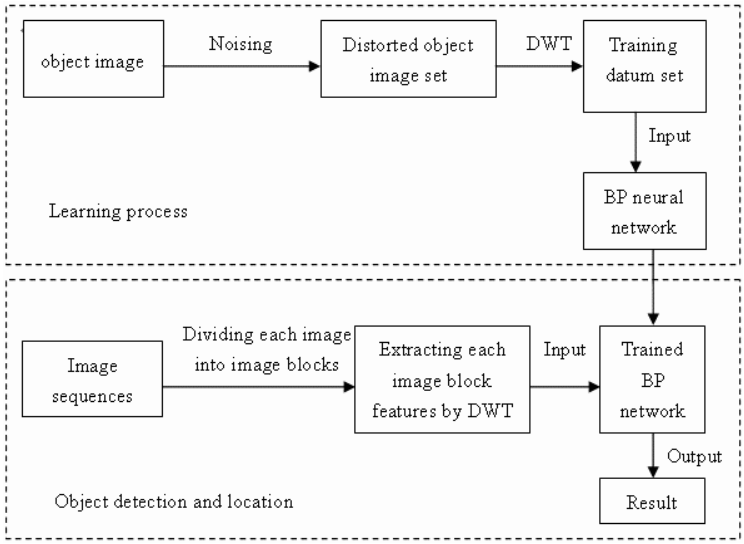


Fig. 1. Location and track an object in image sequences by BP networks

(1) BP Neural Network Learning

In the learning process, we first produce distorted images of an object image to form a training image set. Then DWT is used to extract these images features, and form training data. Subsequently, the training data is processed through input layer, hidden layer and output layer (called *forward propagation*). The output of hidden layer and that of output layer are:

$$b_j = f(\sum_{i=1}^{48} W_{ij} \bullet a_i - \theta) \qquad (j = 1,2,...,10) \tag{5}$$

$$c = f(\sum_{j=1}^{10} V_j \bullet b_j - \gamma) \tag{6}$$

Where b_j denotes the j^{th} output of hidden layer, W_{ij} represents the connection weight from node i of input layer to node j of hidden layer, a_i denotes the i^{th} input of input layer, θ denotes the

threshold of hidden layer, c represents the output of output layer, V_j represents the connection weight from node j of hidden layer to hidden layer, γ denotes the threshold of output layer.

The status of neurons in every layer affects status of neurons in the next layer only. If there is an error between the desired output and the actual output, i.e., the network does not produce the desired output, then the backward propagation begins, which tends to feedback the error and adjust the weight values for each layer. Errors between the desired output and the actual output are calculated by

$$E = \frac{1}{2}\sum (y_k - c_k)^2 \tag{7}$$

Where y_k denotes the desired output of the k^{th} sample, and c_k denotes the actual output of the k^{th} sample. The BP algorithm adjusts the weight values for each layer in the steepest descent direction, which can be shown as followings:

$$\Delta W_{ij}(n+1) = \eta \delta_i a_i + \alpha \Delta W_{ij}(n) \tag{8}$$

Where η is the learning rate, α is a flat factor and belongs to (0, 1), δ_i denotes the correction error of each node, a_i is the i^{th} input value.

The whole process is repeated for each of the sample cases, then back to the first case again, and so on. The cycle is repeated until the overall error value drops below some pre-determined threshold.

(2) Location and Track an Object in Image Sequences

Once the BP network is trained, it can be used for positioning an object in image sequences directly. This stage is carried out by comparing template features with features extracted from sub-images of each frame image according to the following procedures: Each frame of image sequences is divided into small image blocks according to the reference object image size firstly, where the block is shifted pixel by pixel inside the scan area. Then image features of these image blocks is extracted by DWT and fed to the trained BP network, then forward propagation is done, the process looks for the correspondences that match each region to the most similar one, a threshold on the match quality is used to determine whether the object is present in a given frame.

Since the successive image sequences do not differ much due to the high temporal sampling rate, poison of target between adjacent images do not differ significantly. In order to improve computational speed and to reduce computational cost, the position information of the target object in previous images is used to predict the position of the target object in the current image in our research. We first predict the current position of the target object based on a trajectory computed from 3 previous image sequences, and construct a plausible bounding box centered at this position. Next, we find all threshold network responses from the trained BP networks. If there is a strongest response from the network in some a spatial location, we choose that location as the candidate position for detection and location the target object. Otherwise, we use the location

predicted from previous image sequences as the candidate position for detection and location the target object. Especially for the later case, if the object has not be found in several successive frames, then the object is lost, and object detection can be given up. The number of images that the network waits before giving up detecting the object depends on how long the object has successfully been followed. Thus the network can detect and locate an object that is hidden from view for a substantial time period.

3 Experiments

3.1 Materials

For the BP neural network is trained in a supervised manner, so the desired output for every training pattern must be included in dataset. In our research, 12 traffic image sequences are adopted as dataset, which are downloaded from the website of Group Prof. Dr. H.-H. Nagel[17]. These image sequences contain the variations in the background and target illumination and in the scene content and included various types of traffic object of various appearance and sizes.

All experiments were done on a standard 1.8GHz PC with 256 MB RAM and MATLAB 6.5.

3.2 Training

For selecting the training object images for the proposed neural network, we classified objects images into several groups based on the visual appearance of patterns in terms of appearance, size, contrast, and background. We selected one images from each group, and obtained ten typical object images. We trained the neural network with the ten object images as teaching images and with eight out of twelve image sequences as training cases. The sizes of the train region in the teaching image were 16 x 16 pixels. The slope of the linear function of the output units of the neural networks, and the learning rate for training the neural networks were 0.01 and 0.001, respectively. With the parameters above, the training of the neural networks used in this paper required a CPU time of 16.5 hours on the workstation mentioned above. After training, the trained neural network was applied to the entire database to obtain scores for all images. The time for applying the trained neural network to object location and tracking was negligibly small (as shown in table 1).

3.3 Evaluation

This section consists of three parts. First, we perform experiments testing the effectiveness of the proposed method, then we present experiments to test the generality of the proposed method, and

lastly, we present results comparison to other approaches to evaluate the computational performance of the proposed method.

Figure 2 shows the track process of a gray car in a cluttered scene, which has been used as a training case in our research. The detected region is shown surrounded by rectangular bounding boxes. From the figure, it can be seen that the proposed neural networks works very well, even in the case that the car significantly overlapped with other ones (e.g. figure 2(b)).

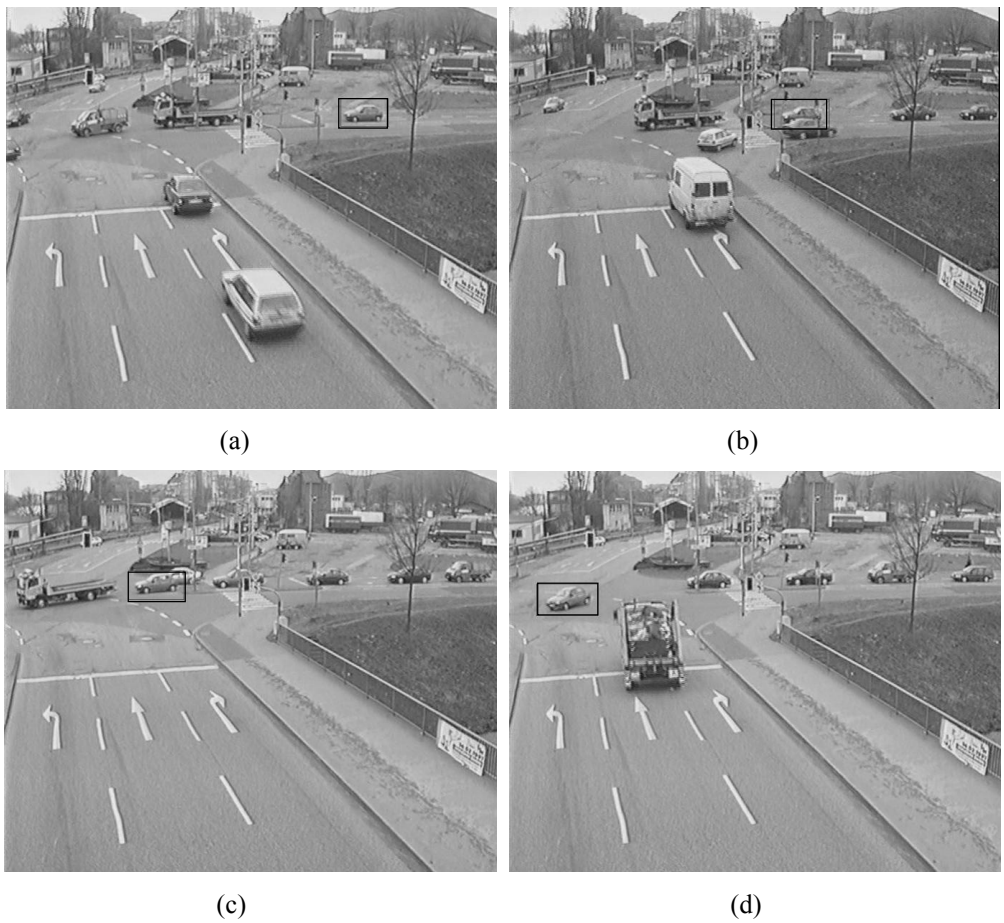


Fig. 2. The track process of a gray car in a cluttered scene

To investigate the generalization ability (performance for non training cases) of the proposed neural networks method, we evaluated its performance with non training cases alone. Figure 3 illustrates the track process of a white microbus in a cluttered scene, which is obtained by a still camera. In this example, the object turns over the road and its shape and size changes gradually through the sequence as it moves further away from a viewer view point. As can be seen, Even though the microbus changes in pose occur, it is robustly detected and located.

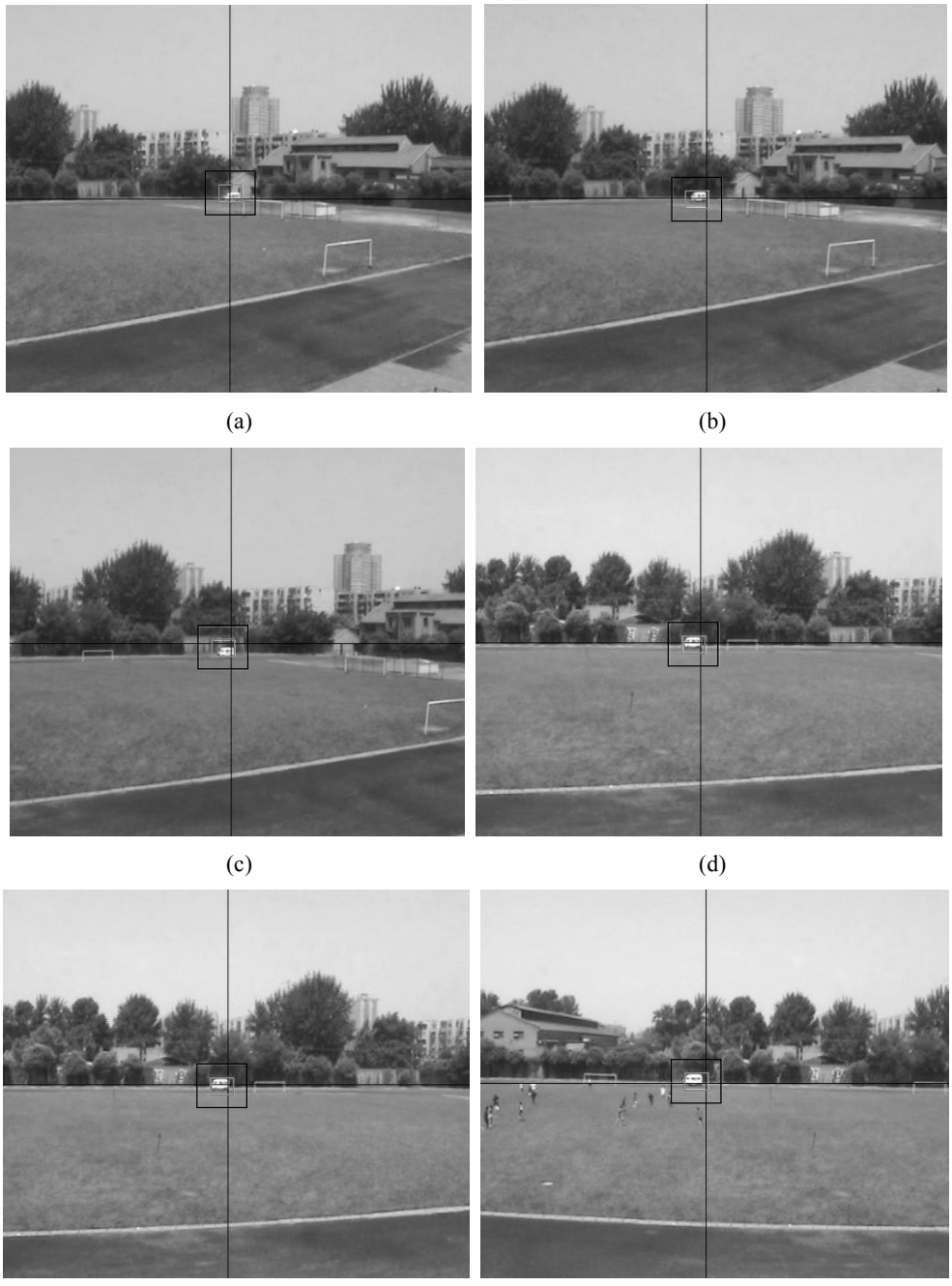


Fig. 3. Detection and location a microbus in a nontraining image sequences while it turns over and changes its size and shape

We also compared the computational performance of the proposed method with the classical template matching method and Hopfield neural networks based method [15]. Figure 4 shows the tracking trace using template matching method and our proposed method respectively, which is correspondence to the video shown in Figure 3. From figure 4, it can be seen that the value of coordinate y of the tracking trace obtained by the proposed method is closer to the true trace than that of the classical template matching method.

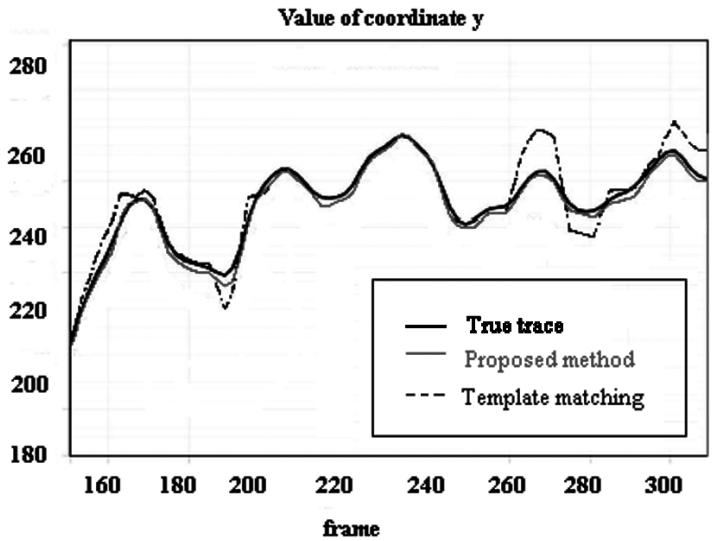


Fig. 4. The tracking trace using different methods

Table 1. Comparison of computational performance

Method	Average time to process a frame	Location ration
Template matching	0.0291 sec.	85%
Hopfield Neural networks based method	0.0056 sec.	95%
Proposed method	0.0044 sec.	96%

Table 1 shows the average location ration for all frames of the image sequences used in figure 3 and the average per-frame processing time for a frame size of (160×120) using different methods. From Table 1 we can see that the neural networks (e.g. Hopfield neural network and neural network used in our study) based methods had almost the same high location ration, which was 95% and 96%, respectively, while the location ration of template matching based method was very lower compared with that of them, only 85%. The average time of template matching based object location method and Hopfield neural network based method was 0.0291 seconds and 0.0056 respectively, and that of the proposed method in this paper is only 0.0044 seconds. The results

clearly indicate that the performance of object location method proposed in this paper is superior to other ones.

4 Conclusions

In this paper, we described an efficient and novel approach for detecting and track an object in image sequences based on BP neural network, where a threshold on the match quality is used to determine whether or not the object is present in a given frame. In order to simply the network structure, DWT is used for extracting image features. By DWT, the size and dimensionality of the input pattern can be reduced. At the same time, in order to improve the computation speed of the method, position information of the target in previous frames is used to predict the position of the target in the current frame. Experimental results indicate the efficiency and the effectiveness of the proposed method.

Unlike many methods using the low-level features of the video frames, the proposed method is not sensitive to the small change in luminance. Moreover, it has high precision as shown in our experiments.

But it should be noticed that training of the proposed neural network took a long time, i.e., about 16.5 hours for each pattern. There are many methods [18]-[19] for accelerating the convergence speed of the BP algorithm. These methods can be applied to our modified BP algorithm, and the time for training can be shortened. It should also be noticed that training with our proposed neural networks method may be trapped at local minima, because our modified neural networks was based on the standard BP algorithm. Therefore some effective amelioration for it must be done. There are many methods [20]-[21] for avoiding local minima for the BP algorithm. By use of these methods, the performance of the proposed neural networks might be improved by avoiding possible local minima and this will be discussed in our other papers in future.

Acknowledgements

The authors are grateful to the anonymous referees for their constructive and helpful comments. This work was supported by Hori Information Science Promotion Foundation.

References

1. M.-H. Yang, J. Kriegman, and N. Ahuja: Detecting Faces in Images: a Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(1) (2002) 34-58.
2. Benjamin Coifman, David Beymer and Jitendra Mailik: A Real-time Computer Vision System for Vehicle Tracking and Traffic Surveillance. *Trasportation Research: Part C* 6(4) (1998) 271-288.

3. F. Bartolini, V. Cappellini, C. Giani: Motion Estimation and Tracking for Urban Traffic Monitoring. *Proceedings 3rd IEEE International Conference on Image Processing ICIP'96*, 787-790, Lausanne, Switzerland. (1996)
4. Ming-Hsuan Yang, Narendra Ahuja, Mark Tabb: Extraction of 2D Motion Trajectories and its Application to Hand Gesture Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(8) (2002) 1061-1074.
5. J.K. Aggarwal and N. Nandhakumar: On the Computation of Motion from Sequences of Images - a Review. *Proceedings IEEE* 76 (1988) 917-935.
6. A. Mitchie and P. Bouthemy: Computation and Analysis of Image Motion: A Synopsis of Current Problems and Methods. *International Journal of Computer Vision* 19(1) (1996) 29-55.
7. Zhenghao Shi, Shitan Huang, Yaning Feng: Artificial Neural Network Image Matching. *Microelectronics and Compute* 20 (2003) 18-21.
8. N.M. Nasrabadi, W. Li: Object Recognition by a Hopfield Neural Network. *IEEE Trans. Systems, Man, and, Cybernetics* 21(6) (1991)1523-1535.
9. Wenjing Li, Tong Lee: Hopfield Neural Networks for Affine Invariant Matching. *IEEE Transactions on Neural Networks* 12(6) (2001) 1400-1410.
10. Zhenghao Shi, Yaning Feng, Linhua Zhang, and Shitan Huang: Hopfield Neural Network Image Matching Based on Hausdorff Distance and Chaos Optimizing. In: *Proc of International Symposium on Neural Networks*, 848-853, Chongqing, China. (2005)
11. N. SANG etc: Relaxation Matching by the Hopfield Neural Network. *Proc. of SPIE* 2664 (1996) 182-190.
12. Rumelhart, D., Hinton, G., Williams, R.: Learning Representations by Back Propagation Errors. *Nature* 323 (1986) 533-536.
13. K. Levi and Y. Weiss: Learning Object Detection from a Small Number of Examples: The Importance of Good Features. In *Proc. CVPR* (2004) 53-60.
14. B. Han and L. Davis: On-line Density-based Appearance Modeling for Object Tracking. In *Proc. ICCV* 2 (2005) 1492-1499.
15. JunghUA Wang, ChihpING Hslao: On Disparity Matching in Stereo Vision via a Neural Network Framework. *Proc. Natl. Sci. Counc. ROC (A)* 23(5) (1999) 665-678.
16. Jie Zhou, Jiaxiong Pang, and Mingyew Ding: Image Matching Method Based on Wavelet Feature. *Pattern Recognition and Artificial Intelligence* 9(6) (1996)125-129.
17. http://i21www.ira.uka.de/image_sequences
18. R.A.Jacobs: Increased Rates of Convergence through Learning Rate Adaptation. *Neural Network* 1 (1988) 295-307.
19. R. Battiti: Accelerated Back-propagation Learning: Two Optimization Methods. *Complex Syst.* 3 (1989) 331-342.
20. W.Finnoff: Diffusion Approximations for the Constant Learning Rate Backpropagation Algorithm and Resistance to Local Minima. *Neural Computing.* 6(2) (1994) 285-295.
21. Y.Shang and B.W.Wah: Global Optimization for Neural Network Training. *Computer* 29(3) (1996) 45-54.