Growing Competitive Network for Tracking Objects in Video Sequences

J.M. Ortiz-de-Lazcano-Lobato[1], R. M. Luque[1],
D. López-Rodríguez[2], and E.J. Palomo[1]

[1] Department of Computer Science, University of Málaga, Málaga, Spain
{jmortiz,rmluque,ejpalomo}@lcc.uma.es
[2] Department of Applied Mathematics, University of Málaga, Málaga, Spain
dlopez@ctima.uma.es

Abstract. The aim of this paper is to present the use of Growing Competitive Neural Networks as a precise method to track moving objects for video-surveillance. The number of neurons in this neural model can be automatically increased or decreased in order to get a one-to-one association between objects currently in the scene and neurons. This association is kept in each frame, what constitutes the foundations of this tracking system. Experiments show that our method is capable to accurately track objects in real-world video sequences.

1 Introduction

In video surveillance systems, accurate and real-time multiple objects tracking will greatly improve the performance of objects recognition, activity analysis and high level event understanding [1,2,3,4].

Segmentation and tracking of multiple objects is important not only for visual surveillance, but also for other video analysis applications such as video compression, indexing, video archival and retrieval systems [5,6], as well as in robotics [7], human-machine interfaces [8], ambient intelligent systems [9] and augmented reality applications [10].

Most of the work on tracking for visual surveillance is based on change detection [5,11,12,13] or frame differencing [14] if the camera is stationary.

The most popular approach for visual tracking is the adaptive tracking of coloured regions, with techniques such as the particle filtering of coloured regions [10,15] and the Kalman/mean-shift [16], which uses the well known mean-shift algorithm [17] to determine the search region, and the Kalman filter to predict the position of the target object in the next frame.

In this paper, the use of growing competitive neural networks (GCNNs) [18] to perform object tracking is proposed. These networks are derived from the usual competitive neural networks (CNNs) [19]. Their main particularity consists in that this kind of network is able to generate new process units (neurons) when needed, in order to get a better representation of the input space.

In general, CNNs are suitable for data clustering, since each neuron in a CNN is specifically designed to represent a single cluster. In the field of object tracking

M. Kolehmainen et al. (Eds.): ICANNGA 2009, LNCS 5495, pp. 109–118, 2009.

in video sequences, such clusters correspond to moving objects. Thus, it seems reasonable to use CNNs as trackers.

However, due to the dynamic nature of a video sequence, objects are constantly appearing and disappearing from the scene, and the method used to track objects should take care of this situation. Consequently, since classical CNNs are not able to manage a variable number of clusters (that is, objects in the scene), the use of GCNNs become a good approach for tracking.

Another advantage of using a neural model to tackle this problem is that neural networks are intrinsically parallel, which allows the processing of all objects in the scene at the same time.

The rest of this paper is structured as follows: in section 2 standard competitive neural networks are briefly described. Section 3 is devoted to the segmentation algorithm, while section 4 explains the tracking system. Finally some experimental results and conclusions are presented in sections 5 and 6 respectively.

2 Standard Competitive Neural Network

The standard competitive neural network [19] forms the kernel of the presented segmentation and tracking modules. Some modifications, which are explained in next sections, have been added to the network in order to obtain a good performance when it is used for these two processes.

The standard competitive neural network has a unique layer of N process units or neurons. This number of neurons is fixed a priori by the user. In each time instant t, an input pattern $\boldsymbol{x}(t)$ is presented to the network and a competition process among the neurons starts. The neuron whose weight vector $\boldsymbol{w}_j$ is closest to the input pattern in the input space is declared the winner. Therefore, the winner neuron is the one which best represents that input pattern:

$$c(t) = \arg \min_{1 \leq j \leq N} \{\|\boldsymbol{x}(t) - \boldsymbol{w}_j(t)\|^2\} \tag{1}$$

Once the winner neuron has been determined, the weight vector $\boldsymbol{w}_{c(t)}$ must be updated in order to incorporate some knowledge from the pattern to the network. Only the winner neuron has gained the right to learn something from the pattern and, thus, it is the only neuron which will be updated in the instant time t. Thus, the standard competitive updated rule is

$$\boldsymbol{w}_i(t) = \begin{cases} \boldsymbol{w}_i(t-1) + \alpha\ (\boldsymbol{x}(t) - \boldsymbol{w}_j(t-1)) & \text{if } i = c(t) \\ \boldsymbol{w}_i(t-1) & \text{otherwise} \end{cases} \tag{2}$$

where $\alpha \in [0, 1]$ is named the *learning rate* and determines how important is the information extracted from the current input sample with respect to the information already known from previous training steps.

3 Object Segmentation

Almost every visual surveillance system starts with motion detection. Motion detection aims at segmenting regions corresponding to moving objects from the

rest of an image. Subsequent processes such as tracking and behaviour recognition are greatly dependent on it. It usually involves environment modelling, segmentation motion, shadow detection and object classification.

Many works based on motion detection and, more concretely, on background subtraction using fixed cameras as the CCTV cameras installed in public transport can be found in the literature [4,2,5]. All of these methods try to effectively estimate the background model from the temporal sequence of the frames and, subsequently capture objects in motion of the scene.

In this step a video object segmentation algorithm is used [20], which aims at classifying the pixels in a video sequence as belonging to foreground or background. This algorithm uses a neural network model to learn pixel colour statistics from the frames observed in the video sequence. Concretely, the recent history of the RGB colour space of each pixel is modelled by an unsupervised competitive neural network.

The learning rule used by each network to model the input space is the standard competitive learning rule, which updates the synaptic weight $\boldsymbol{w}_c$ of the winning neuron according to Eq. (2).

This neural network is formed by at least 2 neurons (foreground and background), although more neurons can be added for multimodal backgrounds. Note that one neuron represents the foreground and the rest of neurons represents the background. The B most activated neurons are used to model the background, whereas the rest of neurons correspond to foreground objects. This value B is computed as the amount of neurons whose number of activations $n_{a_1}, \ldots, n_{a_B}$ verify $\frac{n_{a_1}+\ldots+n_{a_B}}{N} > T$ for a prefixed threshold T, where N is the total number of activations of all neurons, as proposed in [4]. In this work, we have used $T = 0.7$.

When the segmentation results have been obtained, additional techniques as shadow detection [4] and morphological operations are necessary to obtain clear foreground regions.

4 The Tracking Module

The tracking module is based on a growing competitive neural network, which follows an online training process. The number of neurons of the network is not fixed, and is changed depending on the amount of objects which must be tracked by the system in each time instant.

Every object appearing in the video frame is assigned to a neuron. This neuron becomes the responsible for identifying and representing the object, as well as predicting its location in future instants if needed. New neurons are created when not previously detected objects appear in the image, whereas some neurons are killed if they are useless, that is, when the objects associated to the neurons leave the scene.

On the other hand, it is desirable that most of the problems related to this tracking phase can be tackled and solved by the network. Some of these problems are those produced by object occlusion, and those due to some deficiencies in the

segmentation phase, such as the appearance of spurious objects in the middle of the video frame and the sudden disappearance of objects which should remain in the scene several frames later.

In algorithm 1 the main steps of the algorithm are shown. Each step is described more detailedly in the following sections.

4.1 Adapting the Network to the Tracking Application

Object tracking is a task which must be solved in real time. The video may be considered as a data stream, where only information related to the current frame is known. Data from previous video frames are forgotten unless the network uses some memory mechanism to store that previous data, or at least a part of it. Therefore, the tracking system must be able to perform in an online way. In a time instant t the system will be provided M training patterns (or input patterns) $\boldsymbol{x}_i(t)$, $i \in \{1 \dots M\}$. These patterns correspond to M objects which were detected by the segmentation module in the video frame sampled in time instant t.

The proposed solution considers that each neuron represents an object in the frame and tracks this object through the input space frame by frame. Hence, the cluster defined by each neuron does not only enclose different object patterns in the frame, but possible variations of the object pattern. Each cluster can be seen as the expected value for the object pattern between consecutive frames. This way, the learning rate should be fixed to a large value, for example 0.9. Otherwise, the network cannot adequately detect changes in the object, and thus, the object may not be identified.

On the other hand, objects in the middle of the scene cannot vanish into thin air. They leave the scene after reaching and trespassing one of the four frame borders. Occlusions may seem an exception to that fact. But, even when an object is partially or totally covered by another object, the former one continues in the scene. This knowledge can be added to the tracking module and, consequently, to the network, producing several changes in the classical competitive model.

First, the *memory capacity* of the neurons has been *augmented*. Each neuron, j, stores a log, H_j, which contains the known information about the object assigned to the neuron during the last K video frames. Hence, if the object disappears for a short period of time then some object properties such as its location can be predicted by means of the previous information that is known about it and is kept in the log.

Second, *every neuron* is allowed to *modify its weight vector in each training step*. If the neuron wins the competition then it uses the input pattern that made the competition arise. On the contrary, each neuron which does not win the competition *predicts the object features* based on the knowledge stored in its log, and this estimated pattern is used to modify its weight vector. Thus, the update rule is defined as:

$$\boldsymbol{w}_j(t) = \begin{cases} \boldsymbol{w}_j(t-1) + \alpha\ (\boldsymbol{x}(t) - \boldsymbol{w}_j(t-1)) & \text{if } j = c(t) \\ \boldsymbol{w}_j(t-1) + \alpha\ (\hat{\boldsymbol{x}}_j(t) - \boldsymbol{w}_j(t-1)) & \text{otherwise} \end{cases} \tag{3}$$

where $\hat{\boldsymbol{x}_j}(t)$ is the pattern predicted by neuron j.

The estimated pattern $\hat{\boldsymbol{x}}_j(t)$ is obtained by summing the current object pattern, stored in the weight vector of the neuron, and the averaged change observed in that pattern and computed for the K previous frames.

$$\hat{\boldsymbol{x}}_j(t) = \boldsymbol{w}_j(t-1) + \frac{1}{K-1}\sum_{i=1}^{K-1}\left(H_j(i+1) - H_j(i)\right) \tag{4}$$

with $H_j(i)$ the entry which was written down, in the log of the j-th neuron, $t-K-1-i$ frames ago.

Finally, the competition rule has also been slightly modified. This time a mask vector $\boldsymbol{m} \in \{0,1\}^D$, with D dimension of the input space, has been added in order to let the user choose which object components should be considered when calculating the quantisation error during the competition process. Then,

$$c(t) = \arg\min_{1\le j\le N}\{\|\boldsymbol{m}\cdot(\boldsymbol{x}(t) - \boldsymbol{w}_j(t))\|^2\} \tag{5}$$

where $\cdot$ means the componentwise product.

4.2 Neurons Birth and Death

The size of the neural network layer should not be fixed a priori because the number of objects which are present in the scene varies from one frame to another. Hence, the proposed network is formed by $n(t)$ neurons in a time instant t, and a mechanism to add new neurons to the network and to eliminate the useless neurons is needed.

When an unknown object appears in the scene, none of the existing neurons is able to represent it accurately and the quantisation error is expected to reach a high value, compared with the error obtained for correctly identified objects. Thus, a new neuron should be created in order to track that new object. A user-defined parameter $\delta \in [0,1]$ has been utilised. It is related to the maximum relative error permitted in the quantisation process. The parameter δ manages the neurons birth by means of the check

$$\forall j \in \{1\ldots n(t)\} \quad \frac{\|\boldsymbol{x}(t) - \boldsymbol{w}_j(t)\|}{\|\boldsymbol{x}(t)\|} > \delta \tag{6}$$

Notice that if δ is assigned a low value, then (6) ensures that the neurons are updated only if the input pattern $\boldsymbol{x}(t)$ is very close to $\boldsymbol{w}_j(t)$ in the input space. That is, when $\boldsymbol{x}(t)$ and $\boldsymbol{w}_j(t)$ are patterns representing the same object in different frames.

Once the neuron is created, its memory structures are initialised. The input pattern responsible for the birth of the neuron is assigned to the weight vector of the neuron and to the first entry in the neuron log.

$$\boldsymbol{w}_j(t) = \boldsymbol{x}(t)\ ;\ H_j(1) = \boldsymbol{x}(t) \tag{7}$$

On the other hand, if an object leaves the scene then the neuron which represents it should be destroyed. For this purpose, each neuron has a counter C_{die} which

Algorithm 1. Main steps of the tracking algorithm

Input: Time instant t and the features of the segmented objects $x_i(t)$
Output: Labelling of the segmented objects
foreach *Segmented object* $x_i(t)$ **do**
  Compute winner neuron by means of Eq. (5);
  if *Eq. (6) is satisfied* **then**
    Create a new neuron. Initialize it;
  else
    Update the network using equation Eq. (3);
  end
end
Refresh the counter values belonging to the neurons which win a competition;
Decrement all neurons counter values by one;
Check out neuron counters and destroy neurons whose counter value is zero;

means the *lifetime* of the neuron, measured in number of training steps, i.e., frames. Each training step, the counter value is decreased by one and, if the value reaches zero then the corresponding neuron is removed. Every time a neuron wins a competition its counter value is changed to the initial value. Therefore, only neurons associated to objects which are not longer in the scene are destroyed, since it is very unlikely for these neurons to win a competition.

5 Results

In this section the results of our tracking approach to detect and track rigid objects are showed. Some traffic sequences are used to prove the effectiveness of our method. In these sequences some common problems appear, such as occlusions, stopped car in the scene or errors happened in the segmentation phase, which must be satisfactorily solved by a robust tracking algorithm. They are provided by a video surveillance online repository [21] and the Federal Highway Administration (FHWA) under the Next Generation Simulation (NGSIM) program.[1]

Each object pattern has nine components: the x-coordinate and y-coordinate of the object centroid, the 2-d coordinates of the upper left corner of the box which bounds the object, the length and width of that bounding box; and the RGB color components. In our experiments the mask vector is set to hide all object components except for the centroid.

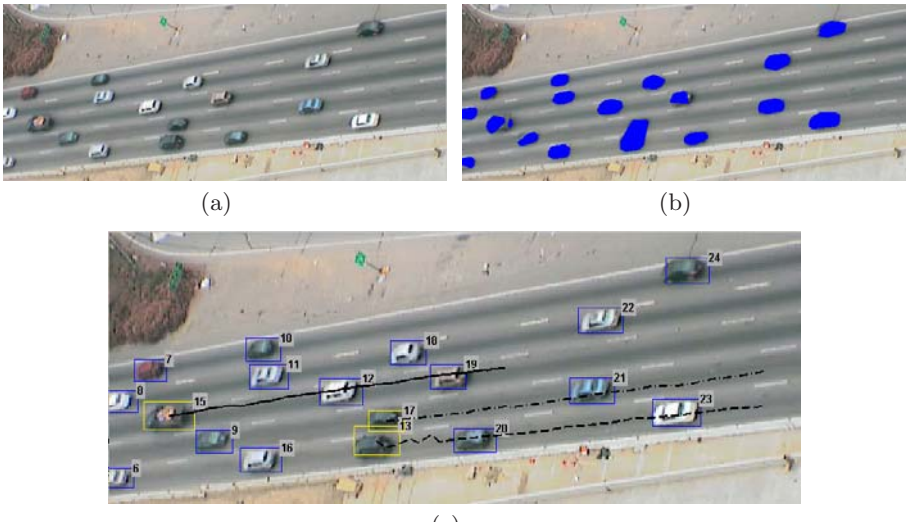
In figure 1, the objects of the sequence are detected and tracked along several frames. In figure 2(b) errors in segmentation phase are observed. Two objects are overlapped at the bottom of the image and one object is divided in two blobs at the left. These problems are solved in fig. 2(c) as it can be observed in the objects identified by the numbers 15, 17 and 13.

Other two different sequences are showed in fig. 3 and 4. Figure 3 contains occlusions of the objects in motion caused by background objects, such as trees.

[1] Datasets of NGSIM are available at http://ngsim.fhwa.dot.gov/

(a) (b)

Fig. 1. Two frames (209, 218) of a traffic sequence are viewed in which the objects are identified and tracked



(a) (b)

(c)

Fig. 2. Problems in the segmentation phase, which are observed in 2(b), are solved in the tracking phase. The trajectory of some objects are plotted in 2(c).

Objects identified in 3(a) by 142 and 154, are robustly obtained in 3(d) despite the segmentation results showed in 3(c).

6 Conclusions

In this work we have presented a new algorithm for moving object detection and tracking in video sequences. This is an important part of video surveillance systems, since these systems need a good starting point to analyse object behaviour. With a reliable tracking algorithm, objects can be easily identified in the video sequence and, using other analysis tools, the behaviour of these objects can be studied, and the system can determine whether there are suspicious/dangerous objects or not.

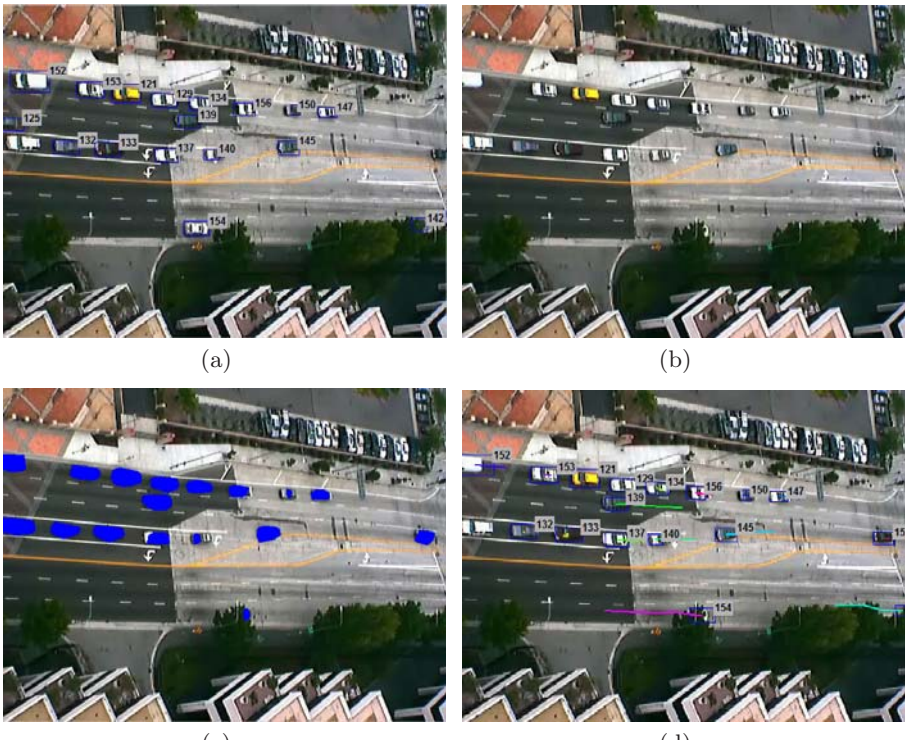


Fig. 3. Stopped cars and car occlusions caused by some trees are observed in this scene. 3(a) shows the frame 740 of the sequence and its identified objects. In 3(b) (frame 747) an occlusion of the car 154 is observed. 3(c) shows the results of segmentation step. In 3(d) the object 154 is correctly tracked.

Fig. 4. Another sequence obtained in [21]. Results of our tracking approach can be observed in 4(a) and 4(b) (frames 245 and 255).

The algorithm proposed in this paper is based on the use of a subtype of the well-known competitive neural networks: the growing competitive neural network (GCNN), which allows the creation and removing of neurons, which are identified to the objects in the scene. Since the number of objects in a video sequence can change from frame to frame, it seems reasonable to permit a change in the number of process units of the network. Thus, a better representation of the foreground objects is obtained.

This new neural model is able to predict the features of each object (location...), by using a log which stores all information known for every object in the last few frames. This allows to deal with several problems produced at the segmentation phase, such as object occlusion or fusion.

Experimental results show that our approach is a reliable and accurate method to detect objects in video sequences publicly available in Internet. In addition, segmentation derived problems can be robustly tackled by this method.

Our future work covers aspects of behavioural analysis, as it is the next logical step in a surveillance system.

Acknowledgements

This work is partially supported by Junta de Andalucía (Spain) under contract TIC-01615, project name Intelligent Remote Sensing Systems.

References

1. Amer, A., Dubois, E., Mitiche, A.: Real-time system for high-level video representation: Application to video surveillance. In: Proceedings of the SPIE International Symposium on Electronic Imaging, pp. 530–541 (2003)
2. Haritaoglu, I., Harwood, D., Davis, L.: w^4: Real-time surveillance of people and their activities. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 809–830 (2000)
3. Lv, F., Kang, J., Nevatia, R., Cohen, I., Medioni, G.: Automatic tracking and labeling of human activities in a video sequence. In: Proceedings of the 6th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (2004)
4. Stauffer, C., Grimson, W.: Learning patterns of activity using real time tracking. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 747–767 (2000)
5. Elgammal, A., Davis, L.: Probabilistic framework for segmenting people under occlusion. In: Proceedings of the 8th IEEE International Conference on Computer Vision, pp. 145–152. IEEE Computer Society, Los Alamitos (2001)
6. Menser, B., Brunig, M.: Face detection and tracking for video coding applications. In: Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, pp. 49–53 (2000)
7. Böhme, H., Wilhelm, T., Key, J., Schauer, C., Schröter, C., Gross, H., Hempel, T.: An approach to multi-modal human-machine interaction for intelligent service robots. Robot. Auton. Syst. 44, 83–96 (2003)
8. Darrell, T., Gordon, G., Harville, M., Woodfill, J.: Integrated person tracking using stereo, color, and pattern detection. Internat. J. Comput. Vision 37, 175–185 (2000)

9. Hayashi, K., Hashimoto, M., Sumi, K., Sasakawa, K.: Multiple-person tracker with a fixed slanting stereo camera. In: 6th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 681–686 (2004)
10. Grest, D., Koch, R.: Realtime multi-camera person tracking for immersive environments. In: IEEE 6th Workshop on Multimedia Signal Processing, pp. 387–390 (2004)
11. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Trans. Pattern Analysis and Machine Intelligence 19(7) (1997)
12. Krahnstover, N., Yeasin, M., Sharma, R.: Towards a unified framework for tracking and analysis of human motion. In: Proc. IEEE Workshop Detection and Recognition of Events in Video (2001)
13. Siebel, N., Maybank, S.: Fusion of multiple tracking algorithm for robust people tracking. In: Proc. European Conf. Computer Vision, pp. 373–387 (2001)
14. Lipton, A., Fujiyoshi, H., Patil, R.: Moving target classification and tracking from real-time video. In: Proc. DARPAIU Workshop, pp. 129–136 (1998)
15. Nummiaro, K., Koller-Meier, E., Van Gool, L.: An adaptive color-based particle filter. Image Vision Comput. 21, 99–110 (2003)
16. Comaniciu, D., Ramesh, V.: Mean shift and optimal prediction for efficient object tracking. In: IEEE Int. Conf. Image Processing (ICIP 2000), pp. 70–73 (2000)
17. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 142–149 (2000)
18. Alahakoon, D., Halgamuge, S.K., Srinivasan, B.: Dynamic self-organizing maps with controlled growth for knowledge discovery. IEEE Trans. Neural Networks 11(3), 601–614 (2000)
19. Ahalt, S., Krishnamurthy, A., Chen, P., Melton, D.E.: Competitive learning algorithms for vector quantization. Neural Networks 3, 277–290 (1990)
20. Luque, R., Domínguez, E., Palomo, E., Muñoz, J.: A neural network approach for video object segmentation in traffic surveillance. In: Campilho, A., Kamel, M.S. (eds.) ICIAR 2008. LNCS, vol. 5112, pp. 151–158. Springer, Heidelberg (2008)
21. Vezzani, R., Cucchiara, R.: Visor: Video surveillance online repository. In: BMVA symposium on Security and surveillance: performance evaluation (2007)