Bull. Nov. Comp. Center, Comp. Science, 42 (2018), 67–75

Tracking objects by the Bayesian network

M.S. Tarkov, M.I. Osipov

Abstract. An algorithm for tracking objects in a videostream based on the use of a hierarchical Bayesian network is proposed. A specific feature of the algorithm proposed is the use of multi-dimensional scaling, which made possible to significantly reduce the network training time. The algorithm is resistant to temporary monitored object disappearances. It is able to monitor several objects against a complex background and well parallelized.

1. Introduction

The video stream is a frames sequence on which an object is imprinted, also, given a set of the object images in different angles. This set plays the training data role. It is required, using the training data, to solve the object tracking task, that is, to determine the object location relative to a current frame, or, if the object is not present in the frame, to make a conclusion about its absence.

When considering the objects tracking task in a video stream, one has to face the problem of the qualitative analysis object images. These images can drastically differ from each other and represent a complex combination of smaller and simpler objects, which makes the analysis, as a whole, extremely difficult. To successfully solve the tracking task, the tracking system is based on a priori data. In this case, a simple comparison of the object in the analyzed image with the standards is extremely inefficient due to a huge number of versions of the object visual representation at various angles.

To solve the above problems, it is possible to use the results obtained in the biologists' studies of the human brain recognition and memorization mechanism. Studies have shown that for these processes the neocortex is responsible. It is the brain area that makes up the bulk of its cortex. It has turned out [1] that a multilayer neurons network is used for memorization, each layer being responsible for the objects perception of a certain complexity. This means that, when working with visual data, a hierarchical principle is used. Thus, as biologists believe, the human brain solves the problem of storing a large number of images.

In [2, 3], the hierarchical approach is realized by the Bayesian probability network, which helps in establishing cause-effect relationships, and this feature is used to establish the semantic connection between the image and its fragments. The hierarchy levels number, the objects complexity into which the image is divided, and many other parameters determine the network

efficiency. The objective of this paper is to find the optimal versions of the networks structure for the object tracking in the videostream.

2. The Bayesian networks

The Bayesian networks (BN) are a powerful data mining tool. The BN is a mathematical model, essentially representing a directed acyclic graph, whose vertices are some statements and arcs are the cause-effect relations between them. The BN operation occurs in the following assumptions framework [4]:

1. Each vertex is an event described by a random variable that can have several states.
2. The connections weights between the vertices are determined by a conditional probabilities table or a conditional probabilities function.
3. The vertices states probabilities that do not have incoming arcs are unconditional.

In other words, vertices are random variables in the BN and arcs are probabilistic dependencies defined through the conditional probability tables.

When using the BN, one of the basic concepts is the random variables distribution density partitioning into a simpler densities product (a smaller random variables number). The basic cases have two and three nodes with different cause-effect relations [4]:

1. The independent variables x and y: $p(x, y) = p(x)\, p(y)$.
2. The dependence of y on x: $p(x, y) = p(x)\, p(y \mid x)$.
3. The dependence of y and z on x: $p(x, y, z) = p(x)\, p(y \mid x)\, p(z \mid x)$.
4. The dependence of z on x and y: $p(x, y, z) = p(x)\, p(y)\, p(z \mid x, y)$.

If the BN nodes are conventionally merged into layers of nodes corresponding to the same type statements, and the statements types in the layers form a certain hierarchy, we obtain a hierarchical BN [3]. Such a network is capable of implementing the logical inference. Based on the listed types of relationships, one can build a BN with an arbitrary number of layers.

The BN classes can be associated with the image classes. The main thing is that the links between them have a causal nature. In the images case, such a relationship can be a nesting relation, that is, the parent vertices correspond to the images fragments of the class to which the child vertex corresponds. We are interested in the probability of an image belonging to a class corresponding to a child vertex, provided that some fragments (parent vertices) are found in this image, and some are not. Having determined such connections between the BN layers, we can use this model for the qualitative image analysis.

If a representative from the image class corresponds to the BN top and shows the same type of specific objects, for example, cars, houses or trees, then this node probability value corresponds to finding an object in the input image from the class onto which this vertex is mapped. Thus, the BN not only answers the question of which image class is in front of us, but it also justifies its answer by the fact that according to the nodes values, there are certain objects types in the image.

To determine the links in the BN, a statistical sampling of the images, onto which the network nodes are mapped, and the information on the imposition of classes in each other is necessary. The hierarchical BN used to solve this problem consists of the three layers (Figure 1):

- At the first layer, there are nodes $s_1, \ldots, s_K$ corresponding to elementary fragments with the sizes from 10×10 to 30×30 pixels of the next layer images.
- The second layer consists of the nodes $obj_1, \ldots, obj_M$ corresponding to simple objects images, for example, the human body parts.
- At the third layer, there are nodes corresponding to the types of objects being monitored. Values of these nodes are the probabilities with which the network maps the input image onto the corresponding type.

The third layer can contain one or more nodes. The tasks that solve more than one node at the third node are multitracking tasks [3]. In this publication, the monitored object is single and, accordingly, the third layer contains a single node.

The learning process includes the training data analysis, which determines the number of the second and first layers nodes, and the relationships between layers. In addition, as a training result, the conditional probabilities values are determined, which, in fact, are the connections weights.

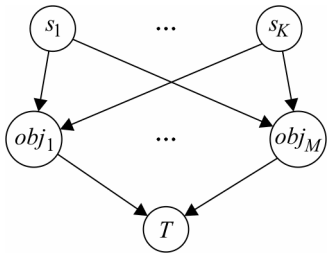

Figure 1. The Bayesian network structure

3. An algorithm for solving the object tracking problem using a three-layer Bayesian network

The three-layer network topology is determined in the learning process. At the initial stage, the following is known about it (see Figure 1):

- Nodes of the first layer are marginal, that is, we consider their probabilities as unconditional ones.

- The number of the third layer nodes is equal to that of the monitored objects number, one node in this case.
- Each node is the parent of all the next layer nodes (if any).

In the object recognition mode, the network functioning begins from the first layer. When training the network, the weight coefficients are calculated in the direction from the third layer to the first one (in Figure 1 from bottom to top).

There is a set of training images T of the size K. At the first stage, it is necessary to extract individual objects from them. The extraction can be done automatically by clustering. However, this stage is not completely automated.

After the objects are extracted, the training data are represented by a set of pairs $s_i = (id_i, t)$, where id_i is the object identifier, which can represent both the number and the meaningful name of an object (hand, head, etc.), and t is an integer specifying the object type (in our case $t = 1$). Assigning an ID to an extracted object is not always possible automatically, sometimes it is necessary to manually specify object identifiers. This can be a time-consuming process, but the consequence is the classification error reduction.

After the set $T = \{s_i = (id_i, 1),\ i = 1, \ldots, K\}$ is defined, the sets obj_i, $i = 1, \ldots, M$, where M is the number of the extracted objects identifiers being formed. These sets are defined as follows:

$$obj_i = \{s :\ id = id_i\}, \quad i = 1, \ldots, M.$$

In other words, the same identifier objects groups are formed. Thus, in the second layer, each node represents a set of obj_i objects. Therefore, the number of the second layer nodes is equal to that of the extracted objects identifiers. The connections weights are given by the formula

$$p(obj_i \mid t) = \frac{|obj_i|}{|T|}, \quad i = 1, \ldots, M.$$

They are equal to the probabilities of finding the corresponding objects, provided that we have an image of a monitored object. It is also necessary to calculate unconditional probabilities for each object class. Their values are obtained using the expression

$$p(obj_i \mid t) = \frac{|obj_i|}{\sum_{j=1}^{M} |obj_j|}, \quad i = 1, \ldots, M.$$

Further, from the objects images, 10×10 to 30×30 pixels fragments are extracted. Images that are the centers of object clusters are broken up into fragments, after which all the fragments are also clustered, for example, by

the k-means method. We denote the metric chosen for clustering $d(x, y)$. The centers of fragments clusters form the dictionary CB. The number of the first layer nodes is equal to the dictionary L power. After the dictionary formation, we represent the available objects classes in the form of sets

$$obj_k = \{\tilde{c}_i = (c_i, m_i),\ i = 1, \ldots, L_k\}, \quad k = 1, \ldots, M,$$

where c_i is the center of the cluster with an extracted fragment, $m_i = 1, \ldots, M$ is the cluster number. After forming the sets obj_i, we define the group of sets C_i as follows:

$$C_i = \{\tilde{c} :\ c = c_i\}, \quad i = 1, \ldots, L.$$

The sets C_i are the fragments clusters. The first layer nodes correspond to these clusters. The weights of the connections between the first and the second layers are calculated by the formula

$$p(C_i, obj_j) = \frac{|\{\tilde{c} \in C_i :\ m = j\}|}{|obj_j|}, \quad i = 1, \ldots, L, \quad j = 1, \ldots, M.$$

These weights are equal to the probability $p(C_i, obj_j)$ of finding the ith fragment, provided that the jth class of objects is found in the image. The video stream frame processing algorithm has the following form. The next frame of the video stream is given. The image must be divided into elementary fragments, whose size coincides with the fragments represented by the constructed BN first layer nodes. Next, the image is traversed by a window whose size is close to the object size.

Fragments, that fall into the window, form one of the test sets. After the tour, we obtain a pool of the sets $C^i_{\text{test}} = \{c^{i1}_{\text{test}}, \ldots, c^{iN_i}_{\text{test}}\}$, $i = 1, \ldots, n$, that are matched with the dictionary. Each element of the sets C^i_{test} is replaced by the element of the dictionary closest to the metric d. Elements that are too far away from all the dictionary elements are removed from the set in question. As a result, we obtain the sets $\widetilde{C}^i_{\text{test}} \subseteq CB$, $i = 1, \ldots, n$, each of them corresponding to a specific area in the frame. To estimate the probability of finding an object in the considered area, it is necessary to apply the corresponding set to the input of the BN.

Consider the network functioning on a certain set $\widetilde{C}^I_{\text{test}}$ of input data. The probability values at the first layer nodes are given by the expression

$$p(c_i) = \begin{cases} 1, & c_i \in \widetilde{C}^I_{\text{test}}, \\ 0, & \text{otherwise}, \end{cases} \qquad i = 1, \ldots, L.$$

The second layer probabilities are given by the formula

$$p(obj_i \mid c_1, \ldots, c_L) = \frac{p(obj_i, c_1, \ldots, c_L)}{p(c_1, \ldots, c_L)} = \frac{B_i \prod_{j=1}^{L} p(c_j \mid obj_i)}{\prod_{j=1}^{L} p(c_j)},$$

where $B_i = p(obj_i)$ is determined in the course of training. The joint probability for the third layer node and the second layer nodes associated with it has the form

$$p(T, obj_1, \ldots, obj_M) = p(T) \prod_{j=1}^{M} p(obj_j \mid T) = \prod_{j=1}^{M} p(obj_j \mid T).$$

Here $p(T)$ is the unconditional probability of finding the corresponding object type, and in our case it is equal to one. At the third layer node, the probability of detecting the monitored object in the area under consideration

$$p(T \mid obj_1, \ldots, obj_M) = \frac{p(T, obj_1, \ldots, obj_M)}{p(obj_1, \ldots, obj_M)} = \frac{\prod_{j=1}^{M} p(obj_j \mid T)}{\prod_{j=1}^{M} p(obj_j \mid c_1, \ldots, c_L)}$$

is calculated.

When all input sets are processed by the network, one should select a set with the maximum probability (MAX block in Figure 2 below). Next, one needs to compare the maximum value with a predetermined threshold. If it is above the specified threshold, then the area, to which the set with the maximum probability corresponds is the area where the monitored object is located in the frame, otherwise the algorithm makes a conclusion about the absence of the monitored object in the considered frame. The above threshold is selected empirically or based on the considerations related to a specific nature of the problem being solved.

4. Solving the task of tracking an object

A video of a person's movement in the office room is given (Figure 2). The person's images being tracked are also given for training the system. It is necessary to determine the person's location in the current video frame. If a monitored object is not detected in the frame, one should report about it. To reduce volume of calculations, all manipulations are carried out with monochrome images. The size of the frame is 352×228 pixels.

The problem is solved by a three-layer BN (see Figure 1). To reduce the computation time, the data dimension is reduced by the multi-dimensional scaling method [5, 6], which maps the points of objects under study onto a new space of a lower dimension. In the case under consideration, the points are mapped onto a plane. In this case, the object points are arranged so that the pairwise distances between them in the new space differ as little as possible from the measured pairwise proximity measures of the analyzed

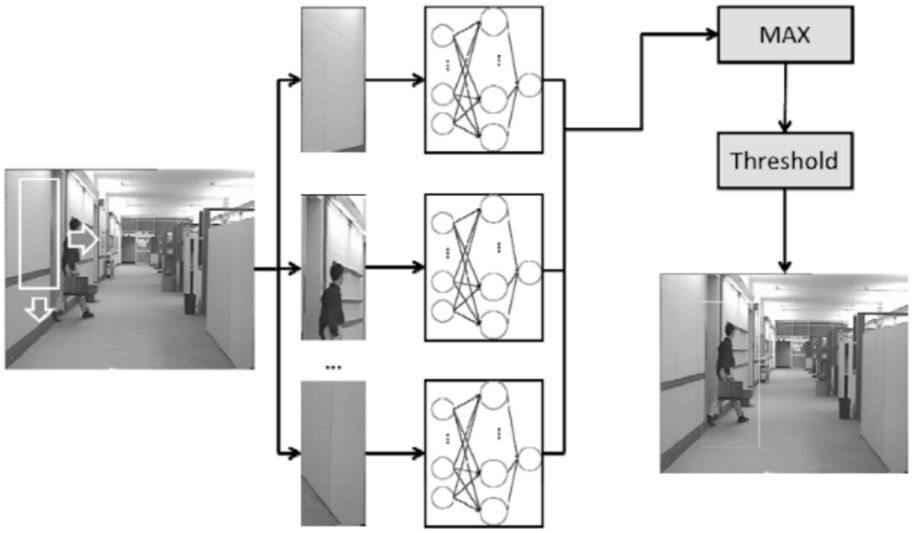

Figure 2. The scheme of the frame processing algorithm

objects in the original space. As a measure of the closeness of images, the Euclidean distance is used. This approach allows one to process frames independent of each other. To evaluate the quality of the algorithm, we used a comparison between the coordinates of the object and the coordinates calculated by the algorithm of the tracking algorithm from the OpenCV library [7].

The table shows the times of training on a processor with a clock speed of 2.1 GHz using multi-dimensional scaling and without it for different volumes of the training sample. Using the data dimension reduction in learning by the multi-dimensional scaling has given the acceleration in learning by two orders of magnitude. This is a key point, since the tracking systems are to be able to periodically retrain the improvement of efficiency. The processing time of the frame of the trained BS is 0.2 seconds.

Using the multi-dimensional scaling leads to a slight ($2 \div 3\,\%$) increase in the average deviation from the monitored object center. When adjusting the frame rate to save computational resources (taking every fifth frame), the average deviation increases by less than $3\,\%$. Thus, it is possible to significantly reduce the amount of computation with a loss of accuracy that is acceptable in solving a particular problem.

The BN training time in seconds

Training data amount (in frames)	5	7	12
Training time without scaling	335	714	2657
Training time with scaling	4.47	7.69	22.8

The processing frames independence from each other gives the algorithm stability to disappearance of a temporary monitored object from the frame. This situation can occur, for example, when the object one is searching for is temporarily closed by another object. The proposed algorithm has a good parallelizability and after some transformations it can be used to track several objects.

5. Conclusion

An algorithm for tracking an object in a videostream based on the use of a hierarchical Bayesian network (BN) is proposed and investigated. A characteristic feature of the algorithm proposed is using the multi-dimensional scaling which made possible to significantly reduce the network training time. The algorithm has the following advantages:

1. Ability for tracking several objects (multi-tracking) against a complex background.
2. When examining the next frame, no information is needed about the previous frame.
3. The algorithm is resistant to a temporary disappearance of the monitored object from the frame.
4. Good parallelism.

The price of the algorithm advantages is its complexity and the demands for learning the BN. However, the BN ability to perform the intelligent data analysis makes the BN indispensable in solving a wide range of problems.

References

[1] Hawkins J., Blakeslee S. On Intelligence: How a New Understanding of the Brain Will Lead to the Creation of Truly Intelligent Machines. — New York: Owl Books, 2005.

[2] Nieto M., Unzueta L., Barandiaran J., et al. Vehicle tracking and classification in challenging scenarios via slice sampling // EURASIP J. Adv. Sig. Proc. — 2011. — Vol. 95. — P. 1–17.

[3] Nillius P., Sullivan J., Carlsson S. Multi-target tracking-linking identities using Bayesian network inference // IEEE Computer Society Conference on Computer Vision and Pattern Recognition. — 2006. — Vol. 2. — P. 2187–2194.

[4] Koski T., Noble J.M. Bayesian Networks. An Introduction. — John Wiley & Sons, 2009.

[5] Borg I., Groenen P.J.F. Modern Multidimensional Scaling. Theory and Applications. — Springer Science + Business Media, Inc., 2005.

[6] Tarkov M.S., Chiglintsev E.A. Reducing the dimensionality of the data in the problem of diagnosing thyroid disease // Optical Memory and Neural Networks (Information Optics). — 2012. — Vol. 21, No. 2. — P. 119–125.

[7] Bradski G., Kaehler A. Learning OpenCV. — USA: O'Reilly Media, Inc., 2008.