The Sixteenth International Symposium on Artificial Life and Robotics 2011 (AROB 16th '11),
B-Con Plaza, Beppu,Oita, Japan, January 27-29, 2011

Video Object Segmentation Using Color-Component-Selectable Learning for Self-Organizing Maps

Shin-ya Umata, Naotake Kamiura, Ayumu Saitoh, Teijiro Isokawa and Nobuyuki Matsui
Division of Computer Engineering, Graduate School of Engineering,
University of Hyogo, 2167 Shosha, Himeji, 671-2280, Japan

Abstract: In this paper, self-organizing-map-based video object segmentation is proposed, assuming that either Y-quantification or HSV-quantification can be systematically selected. Given a video sequence, the value of probability density function is calculated for each component value according to kernel estimation at the first fame. Some areas randomly chosen from the background are then examined, using each component value, whether it is misjudged that they include the target object. The quantification is determined so that occurrence frequency of the above false extraction can be reduced. The data presented to maps are generated, based on the selected quantification. Experimental results show that the proposed method well recognizes the target object.

Keywords: self-organizing maps, block-matching-based learning, video object segmentation

I. INTRODUCTION

A number of video object segmentation algorithms [1]-[4] have been developed, based on soft computing. In [3] and [4], fast block-matching-based self-organizing maps (SOM's) referred to as T-BMSOM's are employed. In them, a rectangular area covering a target object in a frame of a given video sequence is spilt into units with some pixels. The area is referred to as a window. A unit has a vector with element values associated with Y color component. The color attribute of the window is therefore quantified by the above vectors of all units in it. The map for segmentation is constructed by using such vectors as training data. The vectors corresponding to units in the subsequent frame under segmentation are also presented to the map. The map then judges whether the unit corresponding to the presented vector belongs to the target object. The adequacy in quantifying color attributes is thus strongly related to segmentation capability of the map. It is clear that the above Y-quantification does not always fit to arbitrary video sequences.

This paper proposes map-based video object segmentation, assuming that the quantification is selectable. The HSV-quantification is prepared in addition to Y-quantification. The quantification selection is made with the first frame in a given video sequence. The value of probability density function is calculated for each component value according to kernel estimation[5], using the area with the target object. The function value associated with some component value determines the label to be assigned to that component value. The label specifies whether the component value with it is relevant to the target object. Some rectangles are next randomly clipped from the background. They are examined, using labeled component values, whether units in them are accurately classified as part of the background. The quantification selection depends on the number of units accurately classified in such manner. The training data are generated according to the selected quantification, and a map for segmentation is constructed by T-BMSOM learning using them. Experimental results reveal that the quantification selection works well for the segmentation without excessive overextraction.

II. PRELIMINARIES

In this paper, T-BMSOM learning[3] is adopted. A block is defined as shown in Fig. 1. The average of neuron reference vectors in it is given as its reference vector. The Euclidean distance between a block reference vector and the presented data is calculated to find a winner. For the $N\times N$-sized map, the maximum (or minimum) block size is $(N-1)\times(N-1)$ (or 2×2).

Straightforwardly adopting the concept of blocks brings about increase in computational time complexity for learning. In [3], the decision-tree-like search of winner and batch learning process are employed to overcome this issue. The former chooses $(N-2)$ candidates for winner, each time a member of the training data set is presented to the $N\times N$-sized map. The first candidate is one of the four $(N-1)\times(N-1)$-sized blocks. The $(N+1-s)$-th candidate is chosen out of the four $(s-1)\times(s-1)$-sized blocks included in the same $s\times s$-sized block, which has been most recently determined as the $(N-s)$-th candidate. The candidate

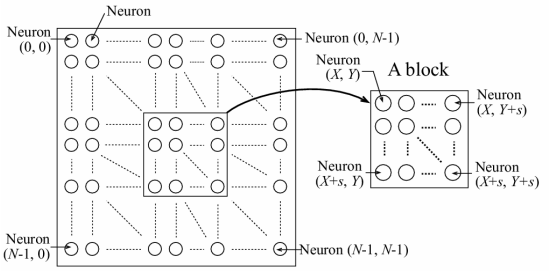

Fig. 1. An $N \times N$-sized map and a block it

with the shortest distance to the presented data is formally determined as winner block for it. The vectors to be updated are those of neurons in the winner.

The batch learning process is summarized as follows. The values associated with modifications on each neuron reference vector are stored for every member in the training data set. Once an epoch is complete, all of the neuron reference vectors are simultaneously updated. The learning termination condition is specified by the number of epochs. Detailed steps of T-BMSOM learning are described in [3].

III. MAP-BASED VIDEO OBJECT SEGMENTATION

The proposed extraction quantifies the first frame in a given video sequence by vectors employed as members of a training data set. T-BMSOM learning is then applied. In the resultant map, the following two clusters are formed: the cluster with neurons mainly firing for the data generated from the moving object and that mainly firing for the data generated from the background. The moving object in each subsequent frame is extracted using the constructed map, and an extraction result appears on the computer screen in the form of a window-like area including the target object.

A. Generation of training data

In the first frame, a rectangular window including a target object is defined as shown in Fig. 2. T-BMSOM learning uses only the data generated from this window for map training. The first window (i.e., the window in the first frame) is formed as follows. In the following, an $n \times n$-sized set of pixels is considered to be a unit. The background difference is applied to the first frame, and units belonging to the target object are systematically extracted. The minimum rectangle with which all units of the extracted target object can be covered is then specified, provided that its center equals the center of gravity of the extracted object. Let $leng_B$

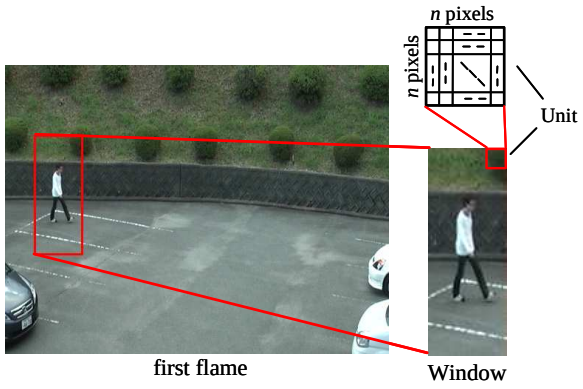

Fig. 2. Window with a target object

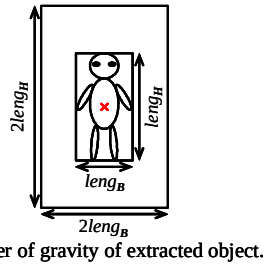

Fig. 3. Window with center of gravity of extracted object as its center

(or $leng_H$) denote the base (or height) length associated with the minimum rectangle. Fig. 3 depicts a schematic representation of the window. The window has the center of gravity of the extracted object as its center.

In [3], the unit has a vector with two element values. Y values of YUV color components of all n^2 pixels in the unit are averaged, and the resultant value is given as one element value. The other equals the standard deviation for the Y component associated with pixels in the unit. The training data associated with the above unit corresponds to the above vector with average and standard deviation. Y-quantification hereinafter denotes the above approach quantifying color attributes based on Y component colors.

If some quantification scheme is unsuitable for a given video sequence, it is highly likely that the target object is lost at some point in time when object segmentation is in progress. To overcome this issue, data are generated on condition that HSV-quantification is also available. When HSV-quantification is adopted, the average and the standard deviation are calculated for each component as well as when Y-quantification is adopted. The number of elements is six for each of the data generated from the units.

A method for determining the quantification scheme is described as follows. It is solely applied to the first frame. The value of probability density function is

calculated for every component value, according to the kernel estimation[5]. Let the *l*-th window denote the window with the target object in the *l*-th frame. The calculation is executed by using units belonging to the target object in the first window. Let t_u^q denote the average of *q*-component values in the *s*-th unit belonging to the target object, where $q \in \{Y, H, S, V\}$, $1 \le u \le W$ and *W* is the number of units in the target object. The density function is given as follows:

$$\hat{p}^q(t) = \left(1/Wh^q\right)\sum_{s=1}^{W} K\left(\left(t - t_u^q\right)/h^q\right), \quad (1)$$

where $K(t)$ and h^q are kernel function[5] and bandwidth, respectively. They are as follows:

$$K(t) = e^{\left(-t^2/2\right)}/\sqrt{2\pi}, \quad (2)$$

$$h^q = 0.9\sigma^q / W^{1/5}, \quad (3)$$

where $\sigma = min(SD, IQR/1.349)$, *SD* is the sample standard deviation and *IQR* is the inter-quartile range. A label is next assigned to each *q*-component value. The label implies whether the *q*-component value is relevant to the target object. Let LB_t^q denote the label assigned to some *q*-component value, t^q. It is determined according to the following equation:

$$LB_t^q = \begin{cases} TO & if \quad \hat{p}\left(t^q\right) > th^q, \\ BG & if \quad \hat{p}\left(t^q\right) \le th^q, \end{cases} \quad (4)$$

where th^q equals the average of values of $\hat{p}^q(t)$ calculated for the first window. If $LB_t^q = TO$ (or BG), it is considered that the value t^q strongly characterizes the color attribute of the target object (or the background).

Next, *WN* rectangles are randomly clipped from the background area excluding the first window, provided that the size of each rectangle is equal to the size of the first window. The average of the *q*-component values associated with pixels is calculated for any unit in each chosen rectangle, and the label that Eq. (4) assigned to the value equal to the average is checked. If the label is *TO* (or *BG*), the unit with this average has high probability of being incorrectly (or correctly) judged as the target-object part (or the background part). This scheme is similarly applied to any unit in the clipped *WN* rectangles. Let N_{BG}^q denote the number units with high probability of being correctly judged as the background part, where $q \in \{Y, H, S, V\}$. For N_{BG}^H, N_{BG}^S and N_{BG}^V, if at least two values are larger than N_{BG}^Y, the data based on HSV-quantification are employed for training a map; otherwise, the data based on Y-quantification are employed. For example, HSV-quantification is chosen if $N_{BG}^Y < N_{BG}^H$ and $N_{BG}^Y < N_{BG}^S$.

B. Construction of maps and object extraction processing

Once T-BMSOM learning is complete, neurons are labeled as follows. All of the members in the training data set are presented again to the just trained map. Each of the members has either 'target object (*TO*)' or 'background (*BG*)' as its label. Let NL_{TO}^i (or NL_{BG}^i) denote the firing frequency for the *i*-th neuron while all of the members with label *TO*'s (or *BG*'s) are presented. Let BN^i denote the label to be assigned to the *i*-th neuron. It is given as follows:

$$LBN^i = \begin{cases} TO & if \quad NL_{TO}^i > NL_{BG}^i, \\ BG & if \quad NL_{TO}^i \le NL_{BG}^i, \end{cases} \quad (5)$$

The map is unavailable to extract the target object until all of the neurons are labeled in the above way.

Let us briefly explain object extraction processing. Once the segmentation is complete for the (*l*–1)-th frame, the data to be presented is generated from each of the units in the *l*-th window, on condition that coordinates of four vertexes of the *l*-th window are equal to those of the (*l*–1)-th window. This *l*-th window is said to be nonconclusive. The data is generated according to the quantification determined in the first frame. If a winner for the data associated with some unit is a neuron labeled *TO* (or *BG*), it is judged that the corresponding unit belongs to the target object (or background). After arbitrary units are similarly examined, the nonconclusive *l*-th window area is updated using the extracted object as shown in Fig. 3. The updated area is fixed as the conclusive *l*-th window. Extracting the target object is similarly conducted for each of the subsequent frames.

IV. EXPERIMENTAL RESULTS

T-BMSOM learning produces well-trained maps [3], [4] on condition that training data sets are updated during learning, compared to conventional SOM learning. The method based on T-BMSOM learning is therefore applicable as follows.

1) The *l*-th map is constructed, using the training data set generated form the *l*-th window.
2) The *l*-th map classifies units in the (*l*+1)-th window.
3) The training data set is updated according to the classification result associated with the (*l*+1)-th window. The (*l*+1)-th map is constructed, using the updated training data set.

If the map constructed with the training data set generated from some window has enough capability of appropriately extracting the units in several of its

subsequent windows, incremental learning conducted in the above manner can be skipped for such windows. This capability is clearly useful in reducing the loads imposed on the computer. The proposed segmentation is therefore evaluated on condition that only the training data set generated from the first window is available. Experimental conditions are as follows: the frame with 640×480 pixels, the unit with 8×8 pixels, 5×5-sized map and learning termination condition is 20 epochs. These are similarly employed in [4]. *WN* is set to 100. The proposed segmentation is compared with the segmentation in [4]. They are implemented on computer (CPU: Athlon 2.0GHz, Memory: 1.0GB).

The proposed segmentation and the segmentation in [4] are applied to sequences referred to as Video 1, Video 2 and Video 3. Figure 4 depicts some frame in each sequence and extraction results. Recall that the method in [4] is solely based on Y-quantification, whereas the proposed scheme uses HSV-quantification in addition to Y-quantification. The proposed method generates data based on Y-quantification for Video 1, and data based on HSV-quantification for Video 2 and Video 3. Note that the extraction results obtained by the method in [4] correspond to the cases of data generation solely based on Y-quantification.

In Figs.4 (a)-(c), units, each of which is judged as part of the target object (pedestrian), are marked by dots. For the results achieved by the proposed method, the pedestrian appropriately appears in the rectangular window. On the other hand, the method in [4] judges a number of units to be actually included in the background as those of the pedestrian, as shown in Figs.4 (b) and (c). It is thus revealed that the quantification selection works well for generating data presented to maps.

V. CONCLUSIONS

This paper proposed video object segmentation, using a map constructed on condition that training data were generated in accordance either with Y-quantification or with HSV-quantification. The quantification selection is made with the first frame in a given video segmentation. Each component value is labeled either as *TO* or as *BG*, and N_{BG}^{q}'s are calculated, using labeled component values, as measures correlating with probability that units in the background are rightly classified. The quantification selection depends on N_{BG}^{q}'s. Experimental results established that the selection is useful in generating data tolerant to the overextraction that occurs in the background.

In future, the proposed method will be modified so that extraction accuracy can be improved.

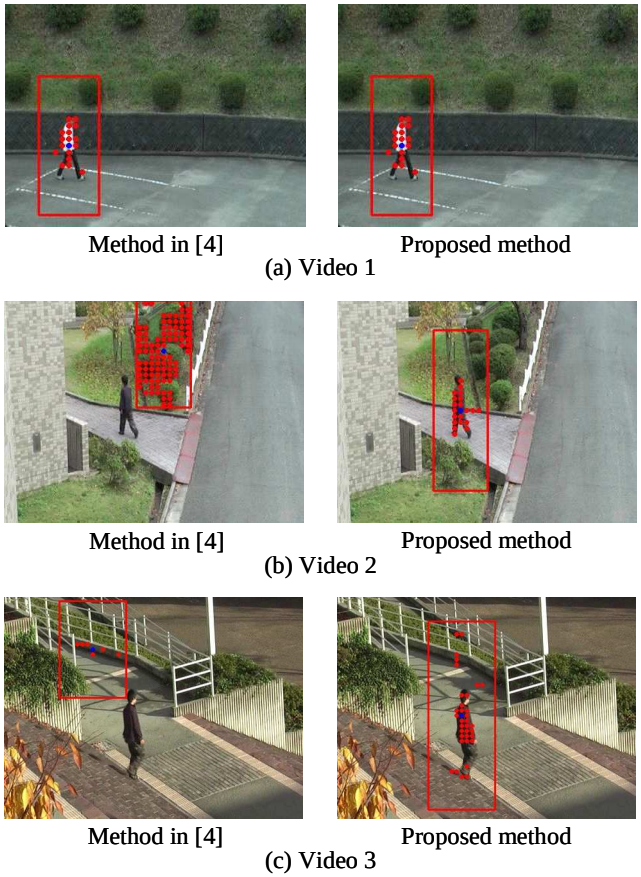

Fig. 4. Object segmentation results for each quantification

REFERENCES

[1] H.Mochamad, H.C.Loy and T.Aoki, "Semi-Automatic Video object Segmentation Using LVQ with Color and Spatial Features," IEICE Trans. INF. & SYST., vol. E88-D, no. 7, pp. 1553-1560, 2005.
[2] H.Grabner and H.Bischof, "On-line Boosting and Vision," In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 260-267, 2006.
[3] T.Isokawa, K.Iwatani and A.Ohtsuka et al., "On Self-Organizing Maps Learning with High Adaptability under Non-Stationary Environments," Proceeding of SICE-ICASE International Joint Conference 2006, pp.4575-4580, 2006.
[4] N.Kamiura, Y.Ohki and A.Saitoh et al., "On Video Object Segmentation Using Fast Block-Matching-Based Self-Organizing Maps," Proc. Of IEEE TENCON 2008, Hyderabad (CDROM), 2008.
[5] B.W.Silverman, "Density Estimation for Statistics and Data Analysis," London: Chapman and Hall, 1986.