A Region-Based Object Tracking Scheme Using Adaboost-based Feature Selection

Fan-Tung Wei, Sheng-Ting Chou
Department of Computer Science & Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan
{wft94,cts95m}@cs.ccu.edu.tw

Chia-Wen Lin
Department of Electric Engineering
National Tsing Hua University
Hsinchu 30013, Taiwan
cwlin@ee.nthu.edu.tw

***Abstract*—This paper presents an object extraction system for video surveillance applications that require pixel-wise extraction accuracy. The proposed mechanism is composed of two trackers. The first tracker extract video objects by using Adaboost on pixel-based global seed features. it can provide more detailed segmentation of target. The second tracker applies bidirectional labeling on regions as well as uses Adaboost on region-based local seed features to refine the object masks obtained from the first tracker. The system is featured with an interactive tool which allows users to deal with serious object occlusion situations. A confidence measure is proposed to minimize the effort of human interactions.**

I. INTRODUCTION

In many object tracking applications, objects are represented in some primitive geometric shapes, such like ellipses [1-3] and/or rectangles [4-7]. This kind of representations are simple to gather feature values from targets, and easy to be modified by translation, affine, or projective transformation. However, they are not well suited for characterizing non-rigid targets. On the other hand, there are some more accurate representations of targets, such as template [3,7], skeleton, and more complicated models [8]. Those representations usually have much detailed information of a target, but suffer from high complexity. Furthermore, since they are specific on particular targets, it is hard to extend them for general purposes.

Once the representation scheme is chosen, we tend to find the target in incoming frames to achieve tracking. An intuitive thought is to find the most similar one in the incoming frames. One can define some novel probabilistic likelihoods as kernel functions to characterize an incoming frame as a distribution. After that, we can find the optimal match using manifold mechanisms. For example, Lanz proposed a hybrid joint-separable model to cooperate with particle filtering [8].

On the other hand, we can use non-target regions to help find out the target [1,4,5]. The main difference from above is that we explicitly use samples as opposites but not just as conceptual dissimilarity. In such scenario, we treat tracking problems as classification problems. After defining target and non-target samples, we collect them in incoming frames and tending to separate them from each other, by classification schemes or more complicated mechanisms. This kind of tracking strategies usually work well due to richer features are available to be extracted, especially for cases with supervised training. In [3], Avidan used an optic-flow based support vector machine to train and classify vehicles. However, for non-rigid target applications, it is difficult to define such opposite samples; also there are lacks of time and samples for training when the applications require real-time and on-line processing.

To provide more detailed segmentation for further analysis, many tracking methods require a background model [2,3]. These methods gain favors from useful background information but are restricted on innate needs of backgrounds. Specifically, if the background is not still and/or is time-varying so that it cannot be built correctly, those trackers will turn out to fail. The seed features mechanism [4], on the other hand, provides an opportunity to segment an object well without the need of establishing any background model.

This work is aimed at offering a tool to extract video objects with pixel-wise accuracy, rather than characterizing objects with simple geometric shapes, so that the tool can be used in video editing and video surveillance applications that require accurate segmentations. Based on the Adaboost feature selection approach proposed in [4], we propose a novel dual-tracker algorithm to achieve pixel-wise object extraction.

The rest of this paper is organized as follows. Section 2 elaborates on the proposed object extraction algorithm. The experimental results are shown in Section 3. Finally, conclusion is drawn in Section 4.

II. PROPOSED METHOD

Fig. 1 shows the framework of the proposed algorithm. Initially, to reduce the computational complexity, the target object is manually located with a simple bounding rectangle as ROI (region of interest) at the first frame. The incoming frame along with the ROI information is fed into the dual-tracker module which consists of the pixel-wise tracker and region-wise tracker. These two trackers are both built on top of the Adaboost feature selection algorithm [4], ina which one tracker uses pixel-wise features and the other uses region-wise features. The region-wise tracker adopts a *K*-means clustering scheme to compensate for shortcoming of the pixel-wise tracker. The two object masks obtained from both trackers are post-processed using a morphological filter and are then combined to obtain the final object mask.

Since it is very difficult, if not impossible, to deal with any object occlusion situation while tracking/extracting an object using existing automatic tracking/extraction methods, our approach provides manual refinement tools to handle failure situations due to object occlusions. We propose a method of automatically identifying unreliable object extractions to trigger human interactions so as to minimize the effort of human interaction.

978-1-4244-1684-4/08/$25.00 ©2008 IEEE

A. *Pixel-wise Tracker*

This tracker is a modified version of the method proposed in [4]. Through the seed features selection and discriminability evaluation mechanism, the fittest features for test sequences can therefore be obtained.

We define our seed features as:

$$F = \left\{ w_1 R + w_2 G + w_3 B \,\middle|\, w_i \in [-2,-1,0,1,2], i = 1,2,3 \right\} \quad (1)$$

where each corresponding weight w_i is chosen as an integer ranging from +2 to -2 for the three color components: R (red), G (green), and B (blue), respectively.

In order to separate foreground pixels from background ones, we also use the tuned feature defined in (2) as an integrated expression and define thresholds to separate them from each other.

$$L(i) = \log \frac{\max\{p(i), \delta\}}{\max\{q(i), \delta\}} \quad (2)$$

where $p(i)$ represents the distribution of foreground and $q(i)$ is the distribution of background, i denotes the bin index, and δ is a small value to prevent dividing by zero or taking the log of zero.

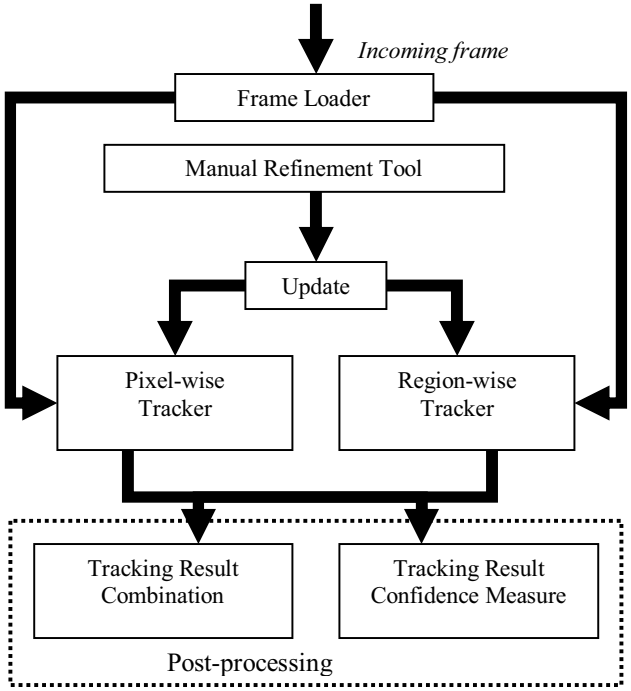


Fig. 1. Framework of the proposed method.

Every seed feature generates a corresponding tuned feature, which is then used for classifying foreground and background pixels. The hypothesis for pixel classification is defined as:

$$H(i) = \begin{cases} foreground & L(i) > \theta_{\text{obj}} \\ background & \text{otherwise} \end{cases} \quad (3)$$

where θ_{obj} denotes the threshold to separate foreground pixels from background. Our method uses three different thresholds to expend the solution space as follows:

$$\theta_{\text{obj}} = \left\{ \max\{p(i)\}/2, \min\{-q(i)/2\}, 0 \right\} \quad (4)$$

The value zero is a trivial threshold where the feature values of foreground pixels equal those of background pixels. The other two thresholds represent different ratios of predominance. Through this multi-threshold mechanism, we have a total of 147 features to construct the solution space.

Fig. 2 illustrates an example of thresholding the tune features. $\Theta_{\text{obj}} = \max\{p(i)/2\}$, zero, and $\min\{-q(i)/2\}$ are represented in green, blue and yellow, respectively and the tuned feature is shown in dark-red line. The background and foreground value curves are represented in orange and in cyan, respectively.

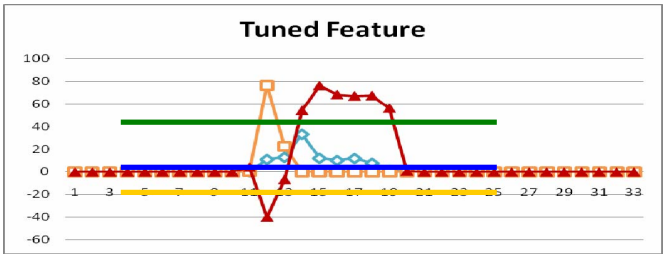


Fig. 2. Three thresholds on the tuned features.

frame #0 frame #132 frame #168 frame #186

Fig. 3. An example of object classification using seed features with Adaboosting.

We use the Adaboost algorithm to combine the tuned features, attempting to find more accurate segmentation result from them. One major advantage of this tracker is that seed features are capable of surviving during occlusion. Different from typical tracking methods that represent targets in simple geometric shapes, seed features have natural color thresholds instead of these artificial bounds, thereby having a potential to accurately extract targets from the frame, even if there exist serious occlusions as shown in Fig. 3. Because there usually does not exist a perfect threshold, we choose a threshold at 10%, meaning that if the confidence of a color/ region is not high enough, the region is classified as background.

B. *Region-wise Tracker*

The proposed pixel-wise tracker, however, may still result in false-positive and false-negative classifications. We propose a region-wise tracker, which consists of two K-means clustering operations followed by a post-processing procedure as elaborated below:

1) Regionalization: the tracker first performs K-means clustering to achieve regionalization. For each frame, every pixel $\mathbf{p}$ is represented in a five-dimensional feature vector,

$$\mathbf{p} \equiv [x \; y \; R \; G \; B] \quad (5)$$

where x and y represent the pixel's horizontal and vertical location coordinate, respectively. R, G, and B represent its three color components. A pixel in the frame is classified into mean $\mathbf{r}_i$ if $\mathbf{r}_i$ is the closest color. The distance between the mean and the pixel is defined as follows:

$$D(\mathbf{p}, \mathbf{r}_i) = (1-\alpha) D_{\text{location}}(\mathbf{p}, \mathbf{r}_i) + \alpha D_{\text{color}}(\mathbf{p}, \mathbf{r}_i), \quad 0 \le \alpha \le 1 \quad (6)$$

where D_{location} and D_{color} denote the spatial-location distance and the color distance, respectively, which are measured in the Euclidean distance as follows:

$$D_{\text{location}}(\mathbf{p}_1,\mathbf{p}_2)=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2} \quad (7)$$

$$D_{\text{color}}(\mathbf{p}_1,\mathbf{p}_2)=\sqrt{(R_1-R_2)^2+(G_1-G_2)^2+(B_1-B_2)^2} \quad (8)$$

To reduce classification complexity and regionalize pixels, we define the candidate region set for each pixel as

$$\Im_i \equiv \{\mathbf{r}_i \,|\, D_{\text{location}}(\mathbf{p}_i,\mathbf{r}_i)<d,\ \forall j\} \quad (9)$$

In (6), α represents the weight to balance these two different kinds of distances. It plays an important role in regionalization. A large α implies color consistency is emphasized while localization constrain is relaxed. We define a measure of color variety as

$$\text{var}(\Im)\triangleq\frac{1}{3n_i}\sum_{\mathbf{r}_j\in\Im_i}\left[\left(R_j-\overline{R}\right)^2+\left(G_j-\overline{G}\right)^2+\left(B_j-\overline{B}\right)^2\right] \quad (10)$$

where n_i represents the number of pixels in $\Im$. If var($\Im$) is large, we increase α to stress on color consistency.

2) Region tracking by bi-directional labeling: this step tracks each region by finding the correspondance of the region between the current and previous frames using bi-directional labeling. The tracking can be divided into the following three cases:

Case 1: region $\mathbf{r}_j^t$ in the current frame corresponds to one or more regions $\{\mathbf{r}_i^{t-1}\}$ in the previous frame after performing the forward labeling in (11), and all these corresponding regions have the same label. If so, we conider $\mathbf{r}_j^t$ has the same label as that of $\{\mathbf{r}_i^{t-1}\}$.

$$label(\mathbf{r}_j^{t-1})=\left\{label(\mathbf{r}_i^t \,|\, D(\mathbf{r}_j^{t-1},\mathbf{r}_i^t)=\min_k D(\mathbf{r}_j^t,\mathbf{r}_k^{t-1}))\right\} \quad (11)$$

Case 2: for $\mathbf{r}_j^t$, if more than one forward corresponding region $\{\mathbf{r}_i^{t-1}\}$ is found using forward labeling, but the labels of $\{\mathbf{r}_i^{t-1}\}$ are not consistent, we apply the backward labeling in (12) to find the backward correponding region and label $\mathbf{r}_j^t$ accordingly.

$$label(\mathbf{r}_j^t)=\left\{label(\mathbf{r}_i^{t-1} \,|\, D(\mathbf{r}_j^t,\mathbf{r}_i^{t-1})=\min_k D(\mathbf{r}_j^t,\mathbf{r}_k^{t-1}))\right\} \quad (12)$$

Case 3: if no corespondiong region of region $\mathbf{r}_j^t$ can be found in the previous frame by forward labeling, $\mathbf{r}_j^t$ is labeled by (12).

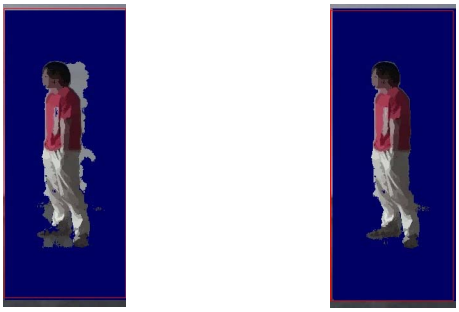
The kernel distance function in this bi-directional lableing is similar to one in regionalization step:

$$D(\mathbf{r}_j^t,\mathbf{r}_i^{t-1})=(1-\beta)D_{\text{location}}(\mathbf{r}_j^t,\mathbf{r}_i^{t-1})+\beta D_{\text{color}}(\mathbf{r}_j^t,\mathbf{r}_i^{t-1}) \quad (13)$$

3) Region-based Adaboosting: After labeling each region, we extract the mean color of each region, and use these color information to generate seed features. Again, Adaboosting is adopted in on-line learning of the region-based features.

C. Post-Procesing and Manual-Refinement Tools

After obtaining the two labeled segmentation masks ϕ_{local} and ϕ_{global} using the pixel-wise and region-wise trackers, respectively, we perform morphological filtering to remove isolated label points. After the post-processing, the two filtered segmentation masks are combined to obtain the final results. The proposed algorithm can do a good job in extracting video objects most of the time. It, however, may lead to some incorrect segmented regions when a serious object occlusion occurs which is in general very difficult to resolve using existing automatic segmentation techniques without any human interaction.



(a) (b)

Fig. 4. (a) Original segmentation mask and (b) the result after manual refinement.

Therefore we provide two brush-like refinement tools to quickly refine the segmentation results. Users can simply drag this tool on the current frame to change several regions at the same time, and regions which are partially or fully passed through are adjusted accordingly. Fig. 4 illustrates an example of an object mask before and after using the manual-refinement tool. Note that, the result after performing manual refinement is very close to "ground truth" and thus can be fed back to update seed features coefficients in Adaboost.

It is not practical to perform manual refinement on each frame. To minimize the effort of manual interaction, we propose a confidence measure to automatically identify which frames are not segmented well and thus require manual refinement. We use the following XOR operation to calculate the mismatch between the pixel-wise and region-wise segmentation masks, which is considered the index of uncertainty.

$$u=\text{XOR}(\phi_{\text{local}},\phi_{\text{global}})=\sum_i \text{XOR}(\mathbf{p}_{\text{local}}^i,\mathbf{p}_{\text{global}}^i) \quad (14)$$

where XOR(·) represent the binary exclusive OR function. A frame with a large u indicates that the final result is unreliable, thus manual refinement should be applied.

III. Experimental Results

In our experiments, the video format of test sequences is ITU-R 601 (720×480) with interlacing. Figs. 5 and 6 show two of our test videos with and without camera motions, respectively. The first experiment is to track the right-hand-side person in Fig. 5. We can observe that the object extraction results (indicated in white color) are pretty good before object occlusion. During the occlusion period (see frame #150, #159), the mismatch value calculated by (14) becomes significantly large. Therefore, the manual-refinement tool is employed to confirm the real target for these frames, and the refined mask is fed into the Adaboost mechanism so as to update both trackers. The refined result has great improvement of confidence. Even with some serious occlusions, the proposed method may still be able to be tracked properly.

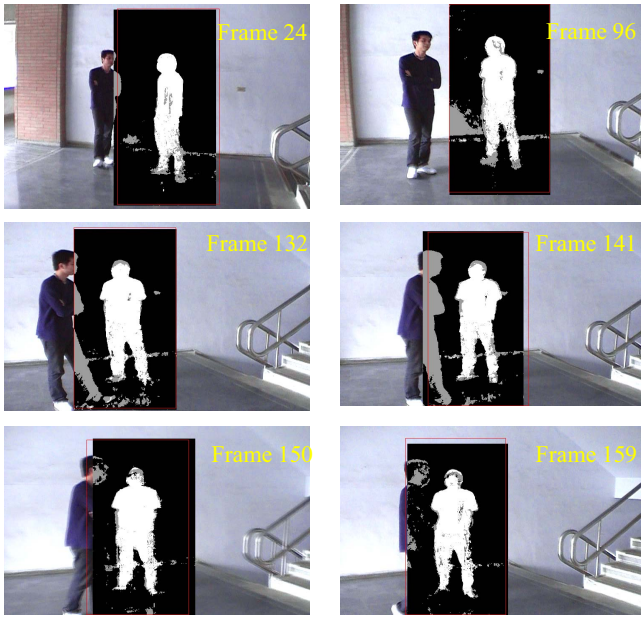


Fig. 5. Object extraction result of right-hand-side person in the *Two-Person* sequence. The white area indicates foreground pixels classified by both trackers; the gray area is the uncertain region; the black area is background.

Fig. 6. Object extraction result of right-hand-side person in the *Hall* sequence using the proposed method.

The test sequence shown in Fig. 6 has no camera motion (i.e., the background is still), and the two people walk around each other which leads to significant changes in shapes and sizes of objects. We only track the right-hand-side person because his shape changes a lot during the interaction while the left-hand-side one has relatively stable feature values. In case of some occlusion regions that cannot be successfully segmented, the manual-refinement tool can be used to fix the results.

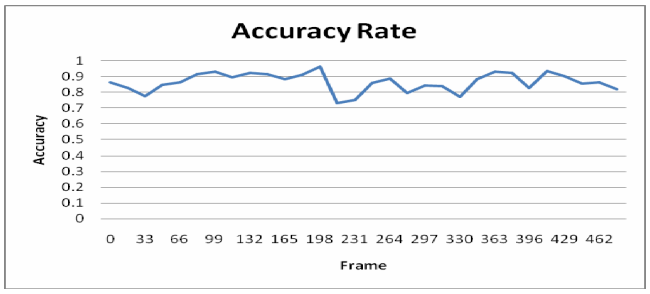


Fig. 7. Accuracy ratio with the proposed method for the *Hall* sequence.

IV. Conclusion

We proposed an object extraction scheme with pixel-wise accuracy. The proposed scheme mainly consists of two trackers. The pixel-wise tracker extracts an object using Adaboost-based global color feature selection. The region-wise tracker first performs *K*-means clustering to regionalize each frame, and then achieves region tracking by using a bidirectional labeling scheme. Subsequently, the region-wise tracker uses Adaboost with regional seed features to obtain region-wise segmentation result. Finally, the two object extraction results are fused to achieve accurate segmentation. Manual refinement tools are also provided to deal with serious occlusion situations. Experimental results show that the proposed algorithm achieves robust object extraction even with some object occlusions.

Acknowledgement

This work was supported in part by the Ministry of Economic Affairs (MOEA), Taiwan, R.O.C., under grant 96-EC-17-A-02-S1-032 and by the National Science Council, Taiwan, R.O.C., under grant 96-2422-H-007-003.

References

[1] C. Hua, H. Wu, Q. Chen, and T. Wada, "A Pixel-wise object tracking algorithm with target and background sample," in *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 739-742, Sept. 2006.

[2] T. Zhao and R. Nevatia, "Tracking multiple humans in complex stuations," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 26, no. 9, pp. 1208-1221, Sept. 2004.

[3] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, vol. 2, pp. 406-413, July 2004.

[4] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tacking features," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631-1643, October 2005.

[5] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064-1072, August 2004.

[6] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261-271, February 2007.

[7] Y. Wu, T. Yu, and G. Hua, "Tracking appearances with occlusions," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, vol. 1, pp. 789-795, 2003.

[8] O. Lanz, "Approximate Bayesian multibody tracking," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 28, no. 9, pp. 1436-1449, Sept. 2006.