Consecutive Tracking and Segmentation Using Adaptive Mean-shift and Graph Cut

Junqiu WANG and Yasushi YAGI
The institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 560-0047
Email: jerywang@public3.bta.net.cn, yagi@am.sanken.osaka-u.ac.jp

***Abstract*— We present an effective tracking and segmentation algorithm in which tracking and segmentation are carried out consecutively. Object tracking in video sequences is difficult since the appearance of an object tends to change. An adaptive tracker that employs color and shape features is adopted to conquer this problem. The target is modeled based on discriminative features selected using foreground/background contrast analysis. Tracking provides overall motion of the target for the segmentation module. Based on the overall motion, we segment object out using the effective graph cut algorithm. Markov Random Fields, which are the foundation of the graph cut algorithm, provide poor prior for specific shape. It is necessary to embed shape priors into the graph cut algorithm to achieve reasonable segmentation results. The object shape obtained by segmentation is used as shape priors to improve segmentation in next frame. We have verified the proposed approach and got positive results on challenging video sequences.**

I. Introduction and Related Work

Object tracking and segmentation in video sequences are important for a number of tasks, such as human-computer interface, video surveillance, and intelligent robotics. There are three common steps in these tasks: detecting moving objects, tracking and segmenting objects from frame to frame, and behavior analysis based on segmented objects. The goal of our work is to track and segment objects in video sequences.

Tracking and segmentation can be challenging even in a stationary camera setting. Moving pixels can be extracted by change detection techniques such as adaptive background subtraction [14], [15] or frame differencing [13] when image sequences are captured by a stationary camera. The detected moving pixels are grouped into blobs according to their connectivity. Such blobs do not have object shape constraints. The extracted blobs can be merged with backgrounds or similar objects. Moreover, the appearance of an object in a video sequence tends to change due to different viewpoints and illumination changes. The variations bring trouble to these change detection techniques.

Tracking and segmentation are more difficult when video sequences are captured by a moving camera, in which the background is dynamic and hard to be modeled. Tracking and segmentation in video sequences with dynamic backgrounds have many important applications such as wearable surveillance, mobile robots and intelligent vehicles. In this work, we do not use the stationary background assumption. Therefore our approach is applicable in sequences with dynamic backgrounds.

In this work, tracking and segmentation are carried out consecutively. The tracking module computes bounding boxes of an object and foreground likelihood ratio images. Based on the likelihood ratio images and shape priors, the segmentation module computes the foreground segmentation results in the current frame.

Mean-shift algorithm is adopted in this work to do blob tracking. To conquer the problem of variations of object appearance and illuminations, we update the target models adaptively, instead of using one object model throughout video sequences. The widely used color features for the mean-shift algorithm and its variations are not always discriminative enough for target localization because illumination and viewpoint tend to change [16]. In addition, the background may have color similar to the target. We present an adaptive tracking algorithm that integrates color and shape-texture features. Good features are selected and applied to represent the target according to their descriptive ability.

Since segmentation is extremely difficult especially in dynamic and cluttered backgrounds, Boykov and Jolly [2] proposed an effective graph based segmentation algorithm for interactive segmentation. This algorithm and its variations have achieved excellent results in interactive segmentation [2][12] and 3D reconstruction [10]. The graph cut algorithm segments images by energy minimization. After an initialization that labels a few pixels as object and background, an energy function based on both boundary and region information is defined and minimized. The energy function consists of two terms: data term and smoothness term. The data term indicates individual labeling preference of pixels based on observed intensities and pre-specified likelihood function. Smoothness term encourage spatial coherence by penalizing discontinuities between neighboring pixels.

The traditional graph cut algorithm segments images using pixel color information. However, automatic segmentation based on color distributions alone is extremely challenging. Markov Random Fields, which are the foundation of the graph cut algorithm, provide poor prior for specific shape. Graph cut-based segmentation can give wrong shape due to ambiguous edges. It is necessary to embed shape priors into the graph cut algorithm to achieve reasonable segmentation results. Based on tracking results, shape priors are learned

online and incorporated into our algorithm to alleviate the problem brought by diffuse edges or similar objects in close proximity to one another.

The remainder of the paper is organized as follows. Section II gives an overview of the proposed approach. Section III describes the adaptive mean-shift tracking module. Section IV presents the graph cut segmentation algorithm in which shape priors are incorporated. We evaluate the performance of the proposed method in Section V. Section VI concludes the paper.

II. Overview

We address object tracking and segmentation in an integrated framework in which tracking and segmentation are conducted consecutively.

The input to our algorithm is a sequence of images $i : \mathcal{P} \times \mathcal{T} \rightarrow \Re$, for each pixel $p \in \mathcal{P}$ and each time $t \in \mathcal{T} = \{0, 1, 2, \ldots, T-1\}$. $i_P \in \Re^T$ denotes the temporal profile of the pixel p. The first frame of the sequence is initialized by labeling the target. The initialization can be done using the interactive segmentation algorithm [12]. The labeled object provides shape prior $\mathcal{S}_0$ for the next frame.

The tracking module computes the bounding boxes in consecutive frames using the adaptive mean-shift algorithm. The bounding boxes represent the global motion of the object.

The segmentation module computes foreground/background segmentation by minimizing the energy function considering data term, smoothness term, and shape priors term. We represent the output as a labeling $f : \mathcal{P} \rightarrow \mathcal{L}$. The shape priors are learned based on the segmentation results. Our algorithm aims at finding a labeling that matches pixels of similar temporal profile while minimizing discontinuities.

III. Adaptive Multi-cue Tracking

Our tracking formulation is based on the basic mean-shift algorithm [6], with a feature selection from color and shape-texture cues. The target model is updated based on feature selection results. The adaptive tracking algorithm effectively conquers the problem brought by the target appearance variations.

A. The Basic Mean-shift algorithm

The mean shift algorithm is a robust non-parametric probability density estimation method for climbing density gradients to find the mode of the probability distributions of samples. It can estimate the density function directly from data without any assumptions about underlying distribution. This virtue avoids choosing a model and estimating its distribution parameters. The mean-shift algorithm has achieved considerable success in object tracking [6].

The traditional mean-shift tracker has two main drawbacks, the first of which is the appearance constancy assumption that is often violated in practical applications due to variations of viewpoints or illumination changes. We develop an adaptive mean-shift tracking algorithm in which discriminative features are selected from shape-texture and color cues. A color cue is described by color histograms and shape-texture cue is represented by gradient orientation histograms. The use of orientation histograms has been found effective in gesture recognition. It has achieved great success since the invention of Scale Invariant Feature Transformation approach [11].

Basically mean-shift algorithm is a blob tracker in which only object translation in image sequences is estimated. This is the second drawback of the basic mean-shift algorithm. It has been extended to multi-dimensional tracking algorithms by simultaneously estimating position, rotation [17], and scale [4]. However, few algorithms can deal with exact object segmentation. In this work, the tracking module provides global motion of the target. The exact labeling of the target is computed by the segmentation module.

B. Log-Likelihood Ratio Images

We compute color and gradient orientation histograms for target representation. The likelihood ratio images are computed based on the weighted histograms. The likelihood ratio produces a function that maps feature values associated with the target to positive values and those associated with the background to negative values. The frequency of the pixels that appear in a histogram bin ($p^{(b_{in})}$) is calculated as $\zeta_f^{(b_{in})} = p_f^{(b_{in})}/n_f$ and $\zeta_b^{(b_{in})} = p_b^{(b_{in})}/n_b$, where n_f is the pixel number of the target region and n_b the pixel number of the background.

The log-likelihood ratio of a feature value is given by

$$L^{(b_{in})} = \max(-1, \min(1, \log \frac{\max(\zeta_f^{(b_{in})}, \delta_L)}{\max(\zeta_b^{(b_{in})}, \delta_L)})), \qquad (1)$$

where δ_L is a very small number (δ_L is set to 0.001 in this work). The likelihood image for each feature is created by back-projecting the ratio into each pixel in the image.

C. Feature Selection

We use color and texture cues for the modeling of a target. In the candidate feature set, the color cue consists of 7 different features: the color histograms of R, G, B, H, S, r, and g, while the shape-texture cue consists of a gradient orientation histogram. Given m_d features for tracking, the purpose of the feature selection module is to find the best subset feature of size m_m, and $m_m < m_d$. Feature selection can help minimize the tracking error and maximize the descriptive ability of the feature set.

We find the features with the largest corresponding variances. Based on the equality [5] $\text{var}(x) = E[x^2] - (E[x])^2$, the variance of Equation(1) is computed as

$$\text{var}(L; p) = E[(L^{b_{in}})^2] - (E[L^{b_{in}}])^2.$$

The variance ratio of the likelihood function is defined as [5]:

$$\text{VR} = \frac{\text{var}(B \cup F)}{\text{var}(F) + \text{var}(B)} = \frac{\text{var}(L; (p_f + p_b)/2)}{\text{var}(L; p_f) + \text{var}(L; p_b)}. \qquad (2)$$

We evaluate the discriminative ability of each feature by calculating the variance ratio. These features are ranked according

to the discriminative ability by comparing the variance ratio. The feature with the maximum variance ratio is taken as the most discriminative feature.

D. *Target Localization in Adaptive Tracking*

The proposed tracking algorithm combines the top two features through back-projection [3] of the joint histogram, which implicitly contains certain spatial information that is important for the target representation. We calculate the joint histogram of the target with the top two features,

$$p_f^{(b_{in}^{(1)},b_{in}^{(2)})} = C \sum_{\mathbf{x}_i \in R_f} k(\|\mathbf{x}_i\|)\delta[h(\mathbf{x}_i) - b_{in}^{(1)}]\delta[h(\mathbf{x}_i) - b_{in}^{(2)}], \tag{3}$$

and a joint histogram of the searching region

$$p_b^{(b_{in}^{(1)},b_{in}^{(2)})} = C \sum_{\mathbf{x}_i \in R_b} k(\|\mathbf{x}_i\|)\delta[h(\mathbf{x}_i) - b_{in}^{(1)}]\delta[h(\mathbf{x}_i) - b_{in}^{(2)}]. \tag{4}$$

where h is histogram computed on the target and $k()$ is an Epanechnikov kernel function [6] that evaluates pixel contributions to the mode seeking.

We get a division histogram by dividing the joint histogram of the target by the joint histogram of the background,

$$p_d^{(b_{in}^{(1)},b_{in}^{(2)})} = \frac{p_f^{(b_{in}^{(1)},b_{in}^{(2)})}}{p_b^{(b_{in}^{(1)},b_{in}^{(2)})}}. \tag{5}$$

The division histogram is normalized for the histogram back-projection. The pixel values in the image are associated with the value of the corresponding histogram bin by histogram back-projection. The back-projection of the target histogram with any consecutive frame generates a probability image $p = \{p^i\}_{i=1\ldots n_h}$ where the value of each pixel characterizes the probability that the input pixel belongs to the histograms.

Since we are using an Epanechnikov profile the derivative of the profile, $g(x)$, is constant. The target's shift vector in the current frame is computed as

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i p^i}{\sum_{i=1}^{n_h} p^i}. \tag{6}$$

The tracker assigns a new position to the target by using

$$\hat{\mathbf{y}}_1 = \frac{1}{2}(\hat{\mathbf{y}}_0 + \hat{\mathbf{y}}_1). \tag{7}$$

If $\|\hat{\mathbf{y}}_0 - \hat{\mathbf{y}}_1\| < \varepsilon$, this position is assigned to the target. Otherwise, let $\hat{\mathbf{y}}_0 = \hat{\mathbf{y}}_1$ and compute the Equation(6) again. In our algorithm, the number of the computation is set to less than 15. In most cases, the algorithm converges in 3 to 6 loops.

IV. Segmentation Using Shape Priors

Boykov and Jolly [2] proposed an effective graph cut method (min-cut/max-flow) for interactive segmentation based on the powerful combinatorial optimization method [9]. Segmentation is formulated in terms of energy minimization in the graph cut algorithm. The cost function is obtained in a context of MAP-MRF estimation [8]. The purpose of the min-cut/max-flow is to seek the labeling of image pixels ($\mathcal{P}$) by minimizing energy:

$$E(L) = E_{smooth}(L) + E_{data}(L),$$

where $L = (L_1, \ldots, L_{|P|})$ is a binary vector whose components specify label assignment; E_{smooth} measures the smoothness of neighboring pixels; and E_{data} measures the disagreement between labeling and the observed data. E_{smooth} and E_{data} are formulated respectively as

$$E_{smooth}(L) = \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(L_p, L_q),$$

and

$$E_{data}(L) = \sum_{p \in P} D_p(L_p),$$

where $\mathcal{N}$ contains all unordered pairs of neighboring pixels; $V_{p,q}$ measures the smoothness of interacting pairs of pixels; D_p is determined by the fitness of p given the observed data.

A. *Data Term*

The form for the data term requires knowledge of some information about the object and the background. In particular, suppose that we know the probability distribution over intensity for both the object and the background, the data term is computed using

$$D_p(L = 0) = -\log p(I|H_b),$$

and

$$D_p(L = 1) = -\log p(I|H_o).$$

These distributions can be learned based on the segmentation results of the previous frame. We built two histogram of the color information of the target region and the background region. The data term is computed based on these distributions.

B. *Smoothness Term*

The smoothness term can be written as

$$V = \gamma \sum_{(p,q) \in \mathcal{P}} [L_p \neq L_q] \frac{e^{-(I(p)-I(q))^2/2\sigma^2}}{\|p - q\|}$$

where $[\varphi]$ denotes the indicator function taking values 0, 1 for a predicate φ, $(p, q) \in \mathcal{P}$ is a set of pairs of neighboring pixels, and $\|p-q\|$ denotes the Euclidean distance of neighboring pixels, γ is a coefficient for weighting of the smoothness term. This energy encourage coherence in regions of similar grey-level.

Fig. 1. Generation of shape priors for the segmentation. (a) The silhouette provided by the gait recognition module; (b) the Edge detection result; and (c) the Euclidean distance transform result.

C. *Min-cut/max-flow Segmentation With Shape Priors*

A shape prior term is added into the energy function to embed shape priors into the min-cut/max-flow algorithm. The energy function now is defined as

$$E(L) = E_{smooth}(L) + E_{data}(L) + E_{shape}(L). \qquad (8)$$

The shape priors is obtained based on segmentation results of the previous frame. The min-cut now includes the shape fitness as well as smoothness and data initial labeling. The energy function E_{shape} is penalized if the segmented contour deviates from the boundary of the silhouette.

The shape prior can be represented by a distance transform result[7]. In Fig. 1, edges are detected in the silhouette image using Canny edge detector. Then the Euclidean distance transformation [1] of the edge image is computed. The cost function of shape priors is well described in the transformed image where costs depend on the distance from the edges. The shape prior energy is written as

$$E_{shape} = \sum_{(p,q)\in\mathcal{N}} [L_p \neq L_q] \frac{\psi(p) + \psi(q)}{2},$$

where ψ is a value on the transformed image.

Segmentation is achieved by minimizing energy function described in Eq. 8.

V. Experimental Results

The proposed object tracking and segmentation algorithm is applied to a variety of sequences to verify the performance. The proposed approach is compared with segmentation by thresholding likelihood ratio images. Given a threshold, likelihood ratio images are transformed into binary images where target and its background are labeled. This is an efficient method that has been used in tracking for target modeling.

We have applied our approach on many image sequences. Two of them are showed in Fig. 2 and Fig. 3 respectively.

The car sequence is captured by a moving camera. Hence there are consistent motions for both foreground and background, which may cause trouble for both pixel-wise background subtraction [14] and layer-based motion segmentation approaches [15]. Our algorithm is not affected by such motions. The segmented region using our algorithm provides coherent region and boundaries of the car. In contrast, the results from the thresholding likelihood ratio method are not satisfying because there are many foreground pixels are labeled as background. The comparison demonstrates that our approach outperforms the thresholding likelihood ratio method.

The person talking sequence is captured in an indoor environment in which a person is talking while moving his body freely. The segmentation results can be useful in human computer interface and teleconference. In Fig. 3, the proposed approach provides better segmentation results. However, there are some pixels are mislabeled by our approach (especially in $T80$ and $T98$). The reason is that the proposed approach suffers from "pollution" of shape priors. When the shape priors computed by the segmentation module deviate from the true shape, it can accumulate and misguide the segmentation. One possible solution to this problem is to learn accurate object shapes offline and apply the learned priors to specific frames.

Timing The proposed algorithm is efficient. When it is applied to the car sequence (images of size 360×240, the tracking module spends 0.03 seconds for each frame and the segmentation module uses 0.09 seconds. The segmentation in the man sequence is a little more expensive than that in the car sequence since the target is much larger. It takes 0.19 seconds to segment the man. These experiments demonstrate that our algorithm is efficient and effective. All the tests are conducted on an Intel Centrino 1.6GHz laptop with 1GB RAM.

VI. Conclusion and Future Work

We have devised an object tracking and segmentation algorithm that compute the global movement and local refinement. The global movement of an object is given by the tracking module and the graph cut segmentation module computes the detail object segmentation. Shape priors are learned based on the tracking and segmentation results. They are incorporated into the graph cut algorithm, which improves the segmentation performance.

To improve the performance of our approach, we will learn accurate shape priors offline. We believe that the application of these priors can bring better tracking and segmentation results.

References

[1] A. Boregefors. "Distance transformations in digital images", *Computer Vision, Graphics and Image Processing*, Vol. 34(3), pp. 344-371, 1986.

[2] Y. Boykov and M-P. Jolly. "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *Proc. of Int'l Conf. on Computer Vision*, pp. 105-112, 2001.

[3] G.R. Bradski, "Computer Vision Face Tracking as a Component of a Perceptural User Interface," in *Proc. of the IEEE Workshop Applications of Computer Vision*, pp. 214-219, 1998.

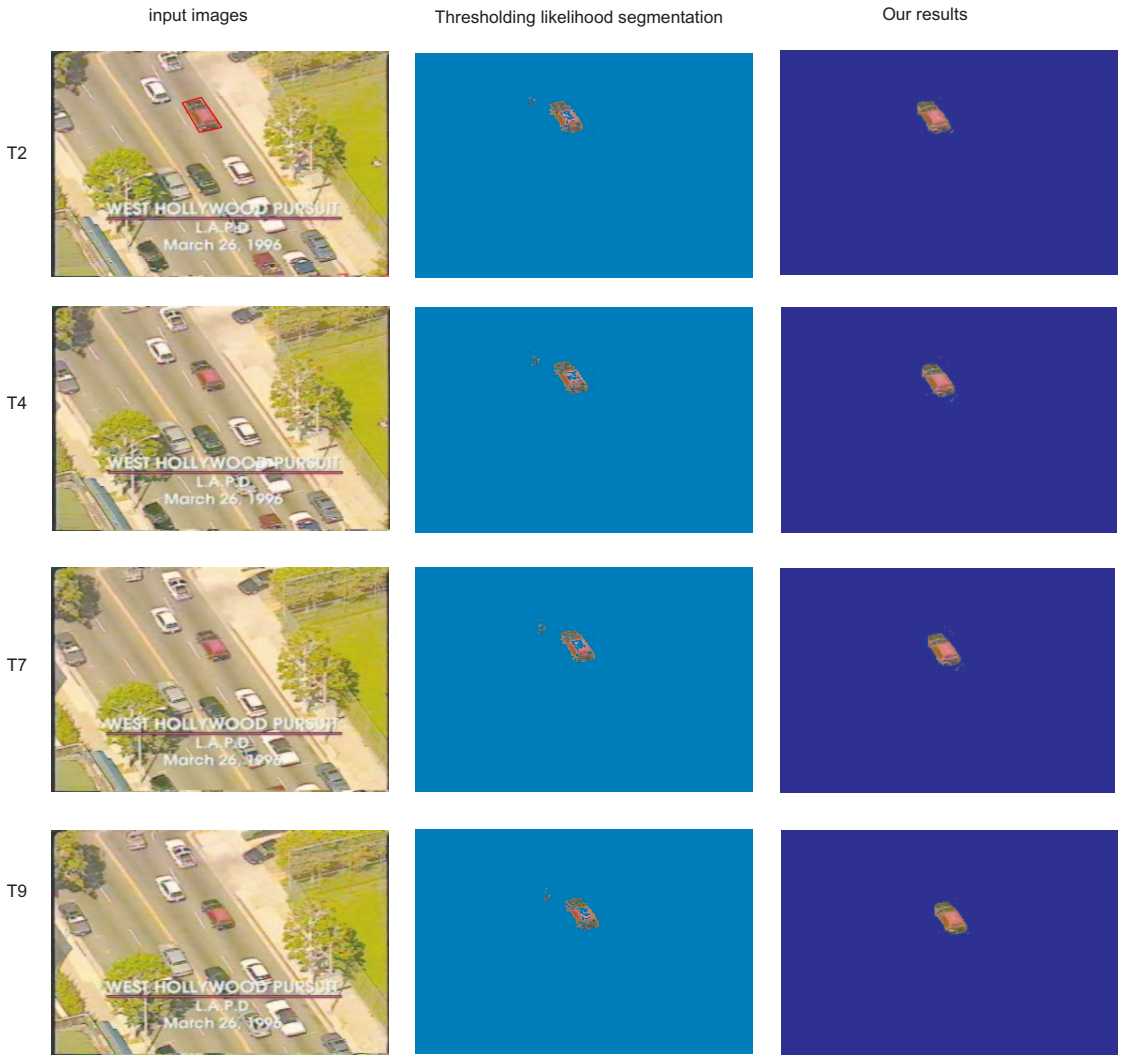

Fig. 2. Comparison of our results on the car sequence with other methods: Left: input images; Middle: Results from segmentation of the manually selected difference image. Right: Results from our algorithm. The camera is not stationary in the sequence.

[4] R. T. Collins, "Mean-shift Blob Tracking through Scale Space," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 234-240, 2003.

[5] R. T. Collins and Y. Liu, "On-line Selection of Discriminative Tracking Features," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, October 2005.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based Object Tracking," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 25, no. 5, pp. 564-577, 2003.

[7] D. Freedman and T. Zhang. "Interactive graph cut based segmentation with shape priors," in *Proc. of IEEE Conf. on Computer Vision and Patten recognition*, pp. 755-762, 2004.

[8] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721C741, 1984.

[9] D. Greig, B. Porteous, and A. Seheult. "Exact maximum a posteriori estimation for binary images". *Journal of the Royal Statistical Society*, Series B, 51(2):271C279, 1989.

[10] V. Kolmogorov and R. Zabih, "Multi-camera Scene Reconstruction via Graph Cuts", in *Proc. the 7th European Conference on Computer Vision (ECCV 02)*, May, 2002.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[12] Carsten Rother, Vladimir Kolmogorov, Andrew Blake, GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts, *ACM Transactions on Graphics (SIGGRAPH)*, August 2004.

[13] A. Rowe, C. Rosenberg, I. Nourbakhsh, "A second generation low cost embedded color vision system," in *Workshop in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPRW'05)*, pp. 136, 2005.

[14] C. Stauffer, E. Grimson, "Learning pattern of activity using real-time tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747-757, 2000.

[15] H. Tao, H.S. Sayhney, and R. Kumar, "Dynamic Layer Representation with Applications to Tracking," in *Proc. of the IEEE Conf. on Computer Vision and Patter Recognition*, vol. 2, pp. 134-141, 2000.

[16] J. Wang and Y. Yagi. "Integrating shape and color features for adaptive real-time object tracking", *2006 IEEE Int'l Conf. on Robotics and Biomimetics*, 2006.

[17] Z. Zivkovic and B.J.A. Krose, "An EM-Like Algorithm for Color-Histogram-Based Object Tracking," in *Proc. of the IEEE Conf. on Computer Vision and Patter Recognition*, pp. 798-803, 2005.

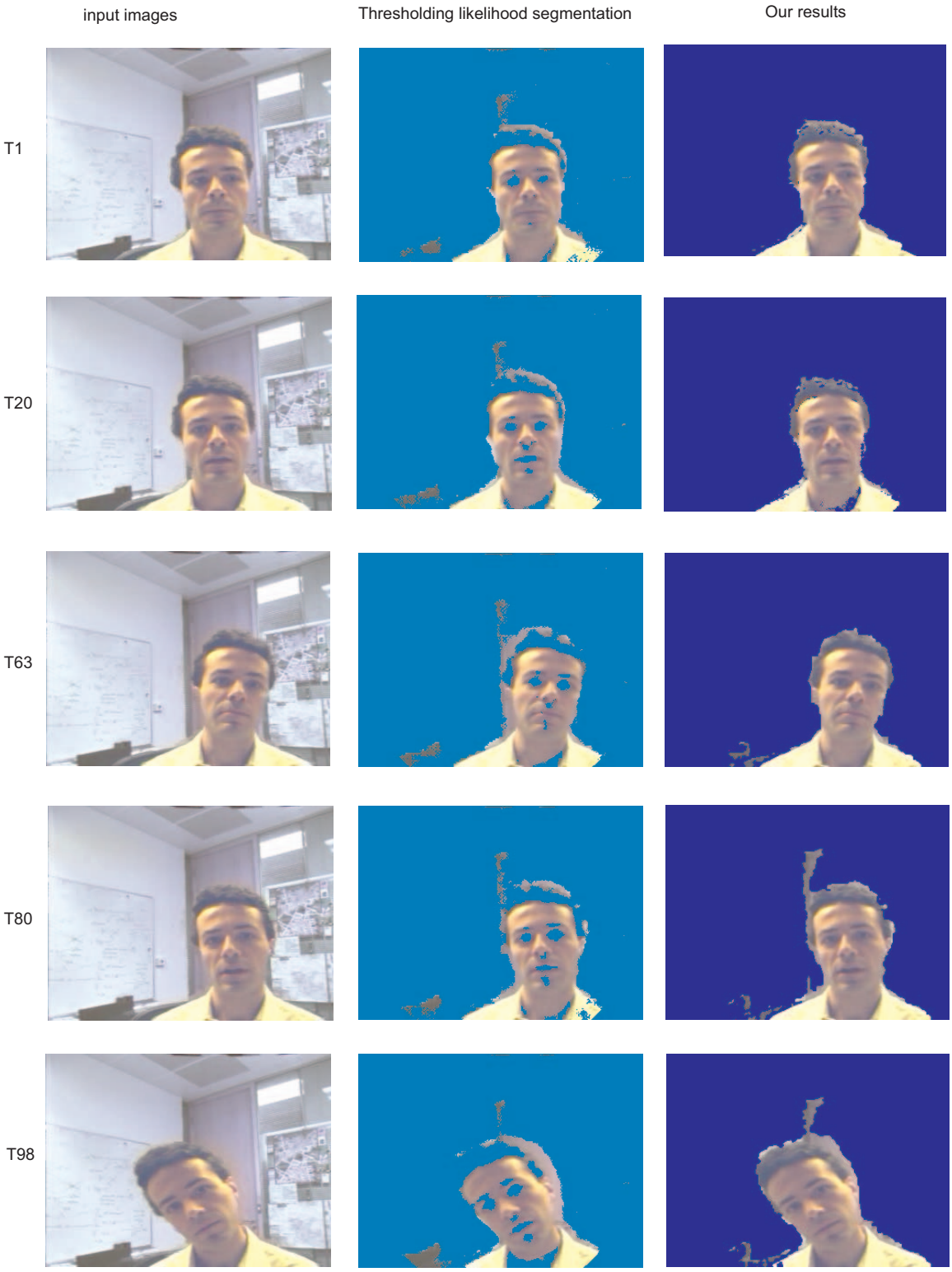

Fig. 3. Comparison of our results on the person talking image sequence with other methods: Left: input images; Middle: Results from segmentation of the manually selected difference image. Right: Results from our algorithm.