

BILAYER VIDEO SEGMENTATION FOR VIDEOCONFERENCING APPLICATIONS

Alessandro Parolin¹, Guilherme P. Fickel², Claudio R. Jung², Tom Malzbender³, Ramin Samadani³

¹ Graduate School of Applied Computing, Universidade do Vale do Rio dos Sinos

² Institute of Informatics, Universidade Federal do Rio Grande do Sul

³ Multimedia Communications and Networking, HP Labs Palo Alto

E-mails: alessandro.paronin@gmail.com, {gpfickel, crjung}@inf.ufrgs.br, {tom.malzbender, ramin.samadani}@hp.com

ABSTRACT

This paper presents a new bilayer video segmentation algorithm focusing on videoconferencing application. A face tracking algorithm is used to guide a generic Ω -shaped template of the head and shoulders. A region of interest (ROI) is created around the generic template, and an energy function based on edge, color and motion cues is used to define the boundary between the person and the background. Our experimental results indicate that the silhouettes can be effectively extracted in common videoconferencing scenarios.

Index Terms— biliayer video segmentation, dynamic programming, videoconferencing systems.

1. INTRODUCTION

Videoconferencing systems allow audiovisual communication from groups of people far from each other. They encounter applications in the corporative domain (offline meetings saving travel time and money) and in the generic purpose market (relatives traveling or living away are allowed to have a more immersive communication when compared to telephone calls).

In particular, natural communication and collaboration is crucial in running any business or enterprise. In the corporate environment, there may be several participants in a virtual meeting, in different rooms. For a more immersive sensation, a uniform background would be more desirable than a mosaic of different background scenarios. In this context, background removal using a single monocular camera (which is still the most common setup nowadays) may support natural, seamless interaction between remote participants.

Although background removal has been extensively studied for several applications, most of existing approaches rely on a training stage where the background is learned (e.g [1,2]). In videoconferencing applications, however, there is no time

for such training, and new techniques are required. This paper presents a new approach for human figure segmentation in video sequences. The proposed approach relies on a face tracking algorithm that guides a head-shoulders generic template. The template defines a region of interest (ROI) that contains the boundary of the participant, and an energy function based on edge, color and motion cues is used to extract the silhouette of the participant.

The remainder of this paper is organized as follows. Section 2 presents an overview of existing methods for bilayer video segmentation, and the proposed approach is described in Section 3. The experimental results are shown in Section 4, and the conclusions are drawn in the last Section.

2. RELATED WORK

There are several applications that require the identification of background and foreground objects, such as tracking, surveillance, background substitution. Most existing approaches employ a set of training frames, where all background pixels should be visible at most times(e.g [2]). In our application, the system usually starts with the person in front of the camera, so that traditional background subtraction techniques, such as those based on Gaussian mixture models [1] can not be applied. There are some bilayer video segmentation approaches that do not require such training period, and some of them are briefly described next.

In [3], a bilayer segmentation algorithm using stereo cameras was proposed. The goal of this paper was to combine the usual characteristics of monocular segmentation systems such as color, contrast and temporal derivatives with the stereo matching information. This method proved to be better than algorithms based in stereo or color/contrast alone, while its computational cost remains relatively low.

The method proposed in [4] uses the spatial, color and temporal information to achieve bilayer video segmentation. The prior on segmentation is represented by a second order, temporal, Hidden Markov Model (HMM), together with a spatial Conditional Random Field (CRF), favouring coherence except where contrast is high. Layer segmentation and

This work was partially funded by Hewlett-Packard Brasil Ltda., with resources from the Brazilian Informatics Law (Law no. 8.248, 1991). Author Cláudio Jung would like to acknowledge Brazilian agency CNPq for supporting his work.

explicit occlusion detection are achieved by binary graph cut, segmenting foreground and background. Their approach does not present shape constraints, and the obtained results are visually good. However, there are some drawbacks: it requires a manual initialization of the foreground and, in order to achieve better results, some weights of the CRF must be tuned with a training period, in which ground-truth data must be available.

The approaches described in [5, 6] proposed a new motion representation referred as “motons” that combines both motion and spatial information. Those motons are used to estimate a segmentation likelihood, which is learned by random forests. In the end of the algorithm, a CRF fuses the motion, color and contrast, and a final segmentation is achieved with min-cut. This method achieves good results, even in the presence of background motion. However, similarly to [4], it requires a set of labeled data for training, and the dependence on the training data and the video sequence to be processed should still be better evaluated.

Lee and collaborators [7] proposed a bilayer video segmentation algorithm by adaptive propagation of global shape and local appearance. Similarly to [4], their approach is based on minimizing an energy map of a Markov Random Field (MRF), performed through branch-and-mincut. The shape priors proposed by the authors increase the robustness of the segmentation, but it requires the insertion of manually segmented keyframes. Furthermore, the computational cost is not small (according to then authors, it takes 1 ~ 10 seconds to process each frame).

Despite the existence of some approaches for on-the-fly bilayer video segmentation, their computational cost is still relatively high. Furthermore, they involve some kind of training data (as in [4–6]) or manual intervention (as in [7]). The method described in this work is fully automatic, and it is described next.

3. THE PROPOSED APPROACH

The proposed approach consists of finding the head-shoulder silhouette contour within a ROI, which is defined based on anthropometrical measurements and a face tracker. A graph is built within the ROI, and its edges relate to an energy function devised to be large along the silhouette and small otherwise. The contour is then computed as the maximum cost path in a graph.

3.1. Face Tracking and Template Fitting

In videoconferencing systems, the participants are expected to face the camera most of the time, so that face tracking algorithms can be used to provide an estimate location of the participants. In this work, the face tracker proposed in [8] is used, since it is robust w.r.t. head tilts/turns, and it also tracks the size of the faces.

Given the position and size of the face at a given frame, a generic Ω -shaped template representing the head-shoulders region is re-scaled and superimposed to the frame, defining the ROI. Figure 1(a) shows the result of the face tracker, and Figure 1(b) illustrates the initial Ω -shaped ROI.

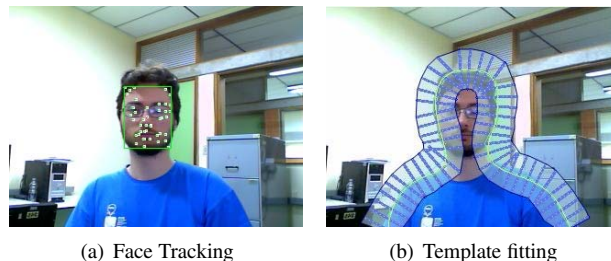


Fig. 1. Example of face tracking and placement of the Ω -shaped ROI.

3.2. Obtaining the silhouette

Given the template-based ROI, the silhouette of the participant is a curve inside the ROI that maximizes a certain energy function, designed to be large on the boundary of the participant, and small otherwise. In this work, the problem of finding the silhouette is formulated as a maximum cost path in a graph.

A set of N line segments are placed orthogonally to the template (dark blue segments in Figure 1(b)), and their length is based on the radius of the tracked face (more precisely, the length is half the radius of the face). Along each line segment i , a set of M equidistant nodes $n_j(i)$ are placed. Each node $n_j(i)$ is connected to three other nodes $n_{j-1}(i+1)$, $n_j(i+1)$ and $n_{j+1}(i+1)$ in the adjacent line segment $i+1$. Figure 2 shows an example using $M = 4$ nodes per segment, and all possible edges (green arrows) connecting nodes (blue dots) in adjacent line segments.

The definitions above describe a directed graph $G = (V, E)$, where V is the set of vertices (nodes), and E the set of edges. For each edge e connecting nodes n_1 and n_2 , a weight $w(e)$ is assigned by computing the average of an energy function E_f considering a raster pixel scan in the line segment between n_1 and n_2 . The silhouette of the participant is obtained as the path $\langle n_{j_1}(1), n_{j_2}(2), \dots, n_{j_N}(N) \rangle$ that presents the maximal weight, where j_k is the selected node for the k^{th} line segment. Such path can be easily obtained using Dijkstra’s algorithm, which is a dynamic programming technique used to solve the shortest path problem in an oriented graph [9].

3.3. Energy Functions

We use three different cues to compute the energy that guides the shortest path: edge, color and motion. The individual energy functions are described next.

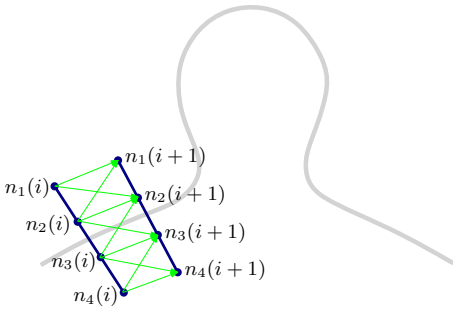


Fig. 2. Illustration of the graph built based on the Ω -shaped template.

3.3.1. Edge Information

Image edges play an important role when identifying different objects in a scene, since they tend to present strong responses along the boundaries of the objects. However, texture and image artifacts (e.g. noise) may produce strong intensity edges, which may lead to erroneous boundaries.

In this work, the luminance component of each frame (normalized to the range $[0, 1]$) is initially blurred with an isotropic Gaussian filter, to attenuate noise and textural information (an 11×11 filter with standard deviation $\sigma = 2.5$ pixels was used, considering 320×240 images). Then, the Sobel operator is applied to the smoothed image, generating an edge gradient image ∇I .

In general, the magnitude of the gradient $|\nabla I|$ is expected to be large along the silhouette of the participant. However, other objects in the background (particularly those with high contrast edges and close to the person) also produce large values for $|\nabla I|$, and they may be included in the optimal path if inside the ROI.

In this work, instead of considering just the magnitude of the gradient, we also take into account the orientation of an edge. In fact, the orientation of ∇I should be orthogonal to local orientation of the contour. In our problem, the silhouette is represented by a path, and the local orientation of the silhouette is given by edges that connect adjacent nodes in the path. For a pixel \mathbf{x} belonging to a candidate edge e in the graph with normal unit vector $\nu(e)$, the influence of the image edge map E_e is given by

$$E_e(\mathbf{x}) = |\nu(e) \cdot \nabla I(\mathbf{x})| = |\nabla I(\mathbf{x})| |\cos \alpha|, \quad (1)$$

where α is the angle between $\nu(e)$ and $\nabla I(\mathbf{x})$. Hence, edges with higher contrast (i.e. large $|\nabla I(\mathbf{x})|$) and coherent with the local contour (i.e. smaller angle α) tend to be prioritized when computing the optimal path.

It is important to mention that edge detectors designed specifically for color images could be used. However, we performed some experiments with the DiZenko operator [10], and results were very similar to the grayscale edge map (but at a higher computational cost).

3.3.2. Motion Information

Although edge information is useful to identify object boundaries in a scene, there may be many strong edge responses in cluttered backgrounds. Assuming that the background is stationary, motion cues can be very useful to determine background/foreground or foreground/background transitions.

In this work, we explore the pixel-wise difference between temporally adjacent frames. If $I(\mathbf{x}, t)$ and $I(\mathbf{x}, t + 1)$ denote the luminance component of pixel \mathbf{x} at frames t and $t + 1$, respectively, the motion energy function $E_m(\mathbf{x})$ is given by

$$E_m(\mathbf{x}) = \begin{cases} 1, & \text{if } |I(\mathbf{x}, t) - I(\mathbf{x}, t + 1)| > T, \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where T_m is a threshold used to discard temporal differences related to noise. In this work, T_m was calibrated experimentally for our camera, and we defined $T_m = 10/255$ ¹.

It is important to observe that a binary image was used to embed motion information. In fact, the temporal differencing presents larger values when objects with high relative contrast move. In our case, we just want to detect pixels where motion occurred, and not prioritize motion relative to higher contrast objects.

Despite its simplicity, the temporal differencing is extremely fast and produce very good results when the participant of the videoconference moves (which happens quite often in common situations). In fact, we also explore the motion map to update the template that defines the ROI. When motion is detected in a percentage of pixels within the ROI (set experimentally to 2% in this work), we assume that the resulting maximum cost path provides a reliable approximation of the silhouette, and the template is updated. With the update procedure, the initial generic Ω -shaped template is adjusted to the shape and pose of the participant as he/she moves.

3.3.3. Color Information

Another important cue do discriminate foreground and background is color. Although histograms are widely used to code color information, they do not carry information on the spatial distribution of pixels that contribute to each histogram bin. An alternative that also embeds spatial information is the spatiogram [11], which stores for each bin the number of votes (i.e. the number of pixels with the corresponding color), and also the mean and the covariance matrix computed with the spatial coordinates of all pixels used to build the bin. Offering improvements over a simple histogram, the mean and covariance used in the spatiogram provide a good explanation of the spatial pixel occurrence for Gaussian-like distributions, but present limitations to explain multimodal distributions.

In this work, we propose a better codification of the spatial distribution using Kernel Density Estimation (KDE) [12], that

¹For a normalized luminance image I .

extends the concept of spatiograms. In our approach, the RGB color space is initially divided into N_c^3 uniform bins, and the image is also spatially divided into $N_s = N_x \times N_y$ rectangular regions R_i , for $i = 1, \dots, N_s$. For each region R_i , we build a 3D structure $h_i(b) = \langle n_i^b, \mu_i^b \rangle$ containing the number of pixels n_i^b in R_i that contribute to the b^{th} bin, and also the mean μ_i^b of the coordinates of such pixels.

In KDE, a kernel centered at each observation is used to obtain a continuous PDF of the data. For a given pixel \mathbf{x} with quantized color b , the KDE-smoothed distribution is given by

$$p(\mathbf{x}, b) = \frac{1}{N_p} \sum_{i=1}^{N_s} g(\mathbf{x} - \mu_i^b) n_i^b, \quad (3)$$

where N_p is the number of pixels in the image, and $g(\mathbf{z})$ is the Gaussian kernel. More precisely, $g(\mathbf{z})$ is given by

$$g(\mathbf{z}) = g(z_x, z_y) = \exp \left\{ -\frac{z_x^2}{2\sigma_x^2} - \frac{z_y^2}{2\sigma_y^2} \right\}, \quad (4)$$

where σ_x and σ_y control the spread of the kernel. In this work, we used $\sigma_x = S_x/3.5$ and $\sigma_y = S_y/3.5$, where S_x and S_y are the dimensions of the rectangular spatial bins.

The posterior probability that a given pixel \mathbf{x} with quantized color b belongs to the foreground is:

$$p(\text{fg}|\mathbf{x}) = \frac{p_{\text{fg}}(\mathbf{x}, b)}{p_{\text{fg}}(\mathbf{x}, b) + p_{\text{bg}}(\mathbf{x}, b)}, \quad (5)$$

where $p_{\text{fg}}(\mathbf{x}, b)$ and $p_{\text{bg}}(\mathbf{x}, b)$ are the KDE-expanded spatiograms of the foreground and background obtained in the previous frame, respectively.

The color-based energy E_c is obtained by computing the edge map of $p(\text{fg}|\mathbf{x})$ using the Sobel operator, which tends to produce lower values in both foreground and background pixels, and larger values along the boundary.

3.3.4. The Final Energy Map

Given the three energy maps described above, the final energy is computed through

$$E_f = w_e E_e + w_m E_m + w_c E_c, \quad (6)$$

where $w_e = 0.1$, $w_m = 0.7$ and $w_c = 0.2$ were defined experimentally (the motion cue was assigned the largest weight).

Figures 3(a)-(c) illustrates the edge, motion and color based energy functions related to the frame shown in Figure 1. As it can be observed, the edge map produces several strong responses due to background objects. The motion energy captures the horizontal transitions between the person and the foreground (the person was moving sideways at this frame), but there are also some responses in the interior of the face and the shirt. The color energy based on KDE-extended spatiograms captures well the boundary of the person, despite

the relatively small color difference between the shirt (blue) and the door behind the person (bluish green). The combined energy map adds up the individual energies when they overlap, as shown in Figure 3(d).

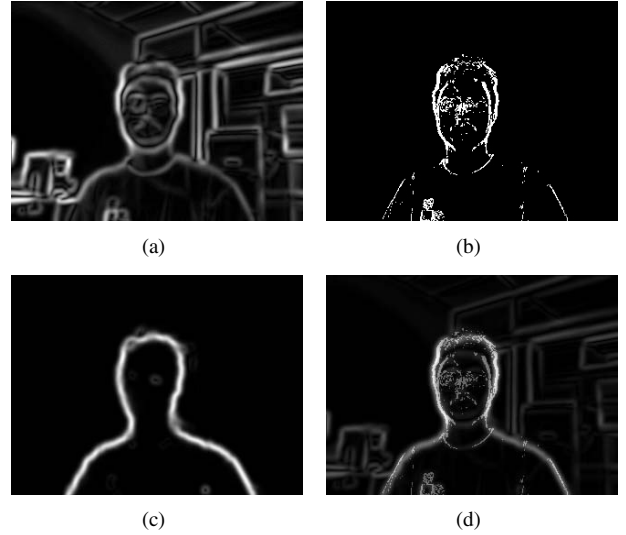


Fig. 3. Energy maps related to (a) edge, (b) motion and (c) color cues. (d) Combined energy map.

Figure 4(a) shows the results of the maximum cost path algorithm applied to the final energy map of Figure 3(d). An example of the extracted foreground pasted into a synthetic background is shown in Figure 4(b).

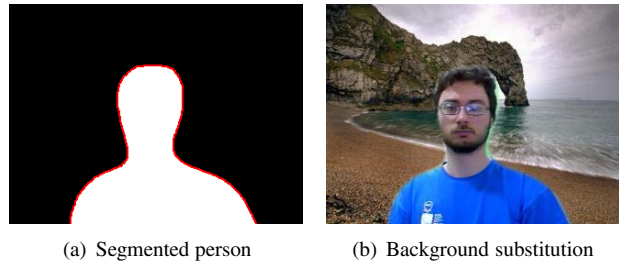


Fig. 4. Example of foreground extraction and background substitution.

4. EXPERIMENTAL RESULTS

The proposed method was tested with different subjects and different background distributions, using $N = 45$ line segments to compute the maximum cost path, each one with $M = 10$ nodes. The KDE spatiogram was built using $N_c = 8$ color bins and $N_s = 5 \times 5 = 25$ spatial bins. We also compared our results with the approach of Criminisi and collab-



Fig. 5. Top: Some frames of the “Chia” sequence. Middle and bottom rows: extracted silhouettes pasted into a plain black background, using [4] and the proposed approach, respectively.

orators [4]², using the same training set³ for all experiments (training specifically for each video sequence would improve the result, at the cost of requiring manual segmentation).

Figure 5 illustrates the result of the proposed approach for some frames of the “Chia” video sequence⁴. The first row shows the original frames, while the second and third rows illustrate the segmentation results using [4] and our approach, respectively. As it can be observed, our technique segments the person and the chair (that presents motion, and attracts the curve in the maximum weight path algorithm). On the other hand, significant portions on the face of the woman are mistakenly segmented as background when using [4].

Figure 6 presents analogous results for the “Fickel” video sequence. Again, the proposed approach correctly segmented the upper body of the participant in most frames, while the application of [4] produced holes on the face. However, it is important to mention that Ω -shaped template used in this work is suited for the segmentation of the head-shoulders region only, and it fails when the arms are present (as in the last column of Figure 6). In such cases, the arms are not segmented, and an “envelope” is obtained after the segmentation procedure. The approach described in [4] does not have this theoretical limitation, but its result was not impressive for our video sequence (see last column, second row of Figure 6).

As shown in Figure 4(b), the background may be replaced when the silhouette is extracted. Since the segmentation procedure is not perfect, alpha matting approaches could be used to obtain the transparency (alpha) values close to the object boundaries. We computed the alpha values within a dilated

version of the contour using [13], and Figure 7 shows some examples of background substitution using alpha matting. As it can be observed, results are visually very good.

The solution is currently implemented in C++ using the OpenCV library. For 320×240 video sequences, the system runs at approximately 23.5 FPS (without the alpha matting procedure) in a notebook powered by a dual core Intel Core2Duo 2.20GHz processor, with 4GB RAM, and Windows Seven OS. It should be noted that several portions of the code can be optimized (including GPU programming, in progress), which could reduce execution time.

5. CONCLUSIONS

This paper proposed a new video bilayer segmentation algorithm for videoconferencing applications based on edge, motion and color cues. Our experimental results indicate that the proposed approach produces smoother contours when compared to competitive approaches (e.g. [4]), and geometrically coherent with a head-shoulders figure. On the other hand, it focused on the segmentation of the upper body only, which is acceptable for most personal videoconferencing situations.

As future work, we intend to explore GPU programming to reduce execution time, and to include a progressive background learning process to reduce the influence of the energy map related to background-related pixels. We also intend to explore fast recent alpha matting algorithms [14] for seamless background replacement that may allow the execution of the whole pipeline in real-time.

6. REFERENCES

- [1] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Pro-*

²Obtained at <http://vision.caltech.edu/projects/yiw/FgBgSegmentation/>

³Available at http://research.microsoft.com/en-us/um/people/antcrim/data_i2i/i2idatabase.zip

⁴Available at <http://vision.ucsd.edu/~leekc/HondaUCSDVideoDatabase/HondaUCSD.html>

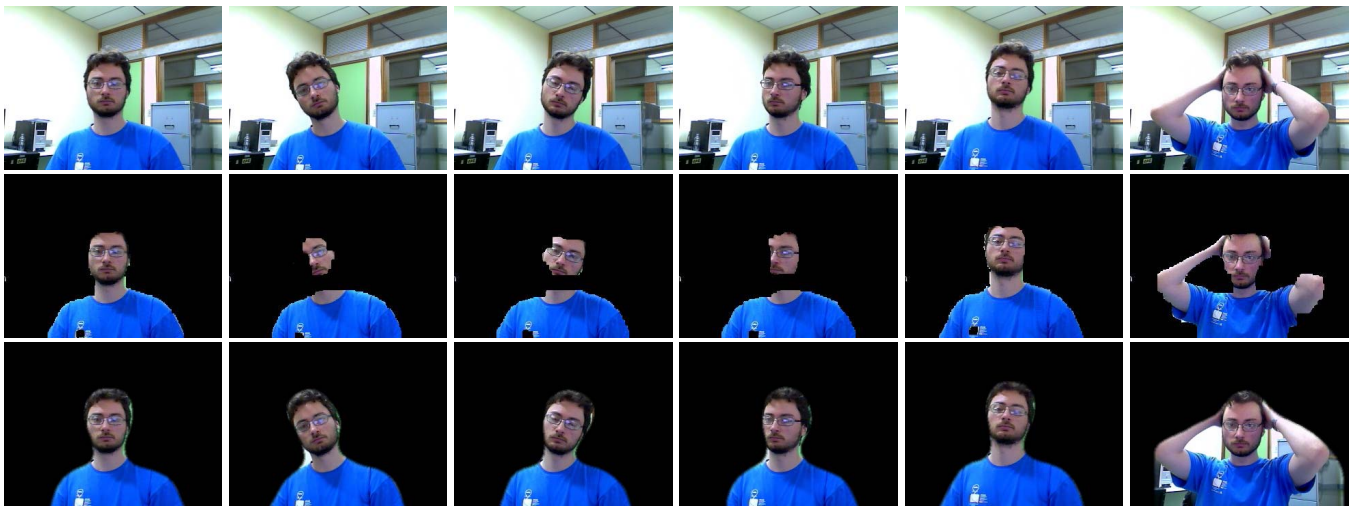


Fig. 6. Top: Some frames of the “Fickel” sequence. Middle and bottom rows: extracted silhouettes pasted into a plain black background, using [4] the proposed approach, respectively.



Fig. 7. Examples of background substitution using alpha matting.

ceedings of IEEE CVPR, vol. 2, 1999, pp. 246–252.

- [2] S. Zhang, H. Yao, and S. Liu, “Spatial-temporal non-parametric background subtraction in dynamic scenes,” in *Proceedings of ICME*, 2009, pp. 518–521.
- [3] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, “Bi-layer segmentation of binocular stereo video,” in *Proceedings of IEEE CVPR*, vol. 2, jun. 2005, p. 1186 vol. 2.
- [4] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, “Bilayer segmentation of live video,” in *Proceedings of IEEE CVPR*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 53–60.
- [5] P. Yin, A. Criminisi, J. Winn, and I. Essa, “Tree-based classifiers for bilayer video segmentation,” in *Proceedings of IEEE CVPR*, 2007.
- [6] —, “Bilayer segmentation of webcam videos using tree-based classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. PrePrint, pp. 1–1, 2010.
- [7] S. Lee, I. D. Yun, and S. U. Lee, “Robust bilayer video segmentation by adaptive propagation of global shape and local appearance,” *Journal of Visual Communication and Image Representation*, vol. 21, pp. 665–676, October 2010.
- [8] J. Bins, C. R. Jung, L. L. Dihl, and A. Said, “Feature-based face tracking for videoconferencing applications,” in *Proceedings of IEEE ISM*, 2009, pp. 227–234.
- [9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, 2nd ed. Cambridge, London: McGraw-Hill Book Company, 2001.
- [10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (2nd Edition)*. Prentice Hall, January 2002.
- [11] S. T. Birchfield and S. Rangarajan, “Spatiograms versus histograms for region-based tracking,” in *Proceedings of IEEE CVPR*, vol. 2, 2005, pp. 1158–1163.
- [12] J. Hwang, S. Lay, and A. Lippman, “Nonparametric multivariate density estimation: a comparative study,” in *IEEE Transactions on Signal Processing*, vol. 42, no. 10, October 1994, pp. 2795–2810.
- [13] A. Levin, D. Lischinski, and Y. Weiss, “A closed form solution to natural image matting,” in *Proceedings of IEEE CVPR*, vol. 1, 2006, pp. 61–68.
- [14] E. S. L. Gastal and M. M. Oliveira, “Shared sampling for real-time alpha matting,” *Computer Graphics Forum*, vol. 29, no. 2, pp. 575–584, May 2010, proceedings of Eurographics.