AN INTEGRATED GRAPH-BASED FACE SEGMENTATION APPROACH FROM KINECT VIDEOS

Jixia Zhang[1]*, Haibo Wang*[2]*, Shaoguo Liu*[1]*, Jiangyong Duan*[1]*, Ying Wang*[1]*, Chunhong Pan*[1]

National Laboratory of Pattern Recognition, CASIA[1]
The Robotics Research Center, Shandong University[2]
jixiazhang@gmail.com

ABSTRACT

In this paper, we present an Integrated Semi-Supervised Graph (IntSSG) approach to automatically segment face from color-depth video. In the first step, IntSSG performs skin color detection and online SIFT matching to initialize some face and non-face pixels. Then, the labels of these pixels are refined by conducting adaptive depth thresholding. Finally, based on a semi-supervised graph framework, IntSSG segments face by propagating the refined labels to other pixels. Experimental results show that IntSSG is able to accurately segment faces in difficult situations such as large pose changes and illumination variations.

Index Terms— Skin detection, face segmentation, depth

1. INTRODUCTION

Face segmentation from video has wide applications in many fields, such as human computer interaction, video editing [1], virtual reality [2]. However, this remains a challenging problem. On the one hand, efficient face segmentation requires a full face output without any holes and noises. On the other hand, face segmentation must overcome the problems of large head pose variation, occlusion and illumination changes that often exist throughout a video sequence.

Face segmentation was usually conducted through the stepwise use of face color detection and region segmentation [3, 4, 5, 6]. Chai et al. [7] detected skin color pixel with a generic skin model, and then performed region-based regularization to link separated regions and find the most likely facial region. By contrast, in [5], following the classification of each pixels, a binary partition tree is utilized to merge detected face pixels or regions. In [6], color Gaussian Mixture Models are adopted to decide the possibilities of each pixel belonging to face or background, which is then followed by a graph-cut based segmentation to segment face. Although skin detection is insensitive to head pose variations, it tends to fail when the illumination condition is different to the one when skin detector is trained.

Some approaches employ a top-down scheme to conduct face segmentation. These approaches adopt a global face model to localize face first, and then segment face boundary. For example, Luo et. al [8] utilized a blob model to locate head-and-shoulder and then segment face with a shape model. However, the specified blob and shape models limit its practical applications. In [9], face is localized with a learned frontal face detector. As reported in [9], this approach may fail when face is not near-frontal.

In this paper, we present IntSSG, an effective approach to automatically segment faces from color-depth videos. By treating face segmentation as a semi-supervised graph problem, IntSSG successfully integrates face pixel detection and segmentation in a single computation framework. In its first step, IntSSG jointly employs skin color detection and online SIFT matching to detect possible face and non-face pixels. Second, IntSSG takes advantage of depth cue to remove the potential wrong labeled pixels in them, by which only most reliable pixels are preserved as prior labels. Finally, with the preserved labeled pixels as prior, IntSSG relies on graph propagation to find all face pixels [10]. The main novelties of IntSSG lies in (1) face pixel classification and region segmentation are unified in a single computational framework, without any pre- or post-processing steps, and (2) skin color, temporal coherence, and depth cues are employed together to guarantee that prior face pixels are correctly detected before propagating to other pixels.

The remainder of this paper is organized as follows. The proposed segmentation method is presented in Section 2. In Section 3, experimental results and analysis are reported. Finally, conclusions are drawn in Section 4.

2. AN INTEGRATED SEMI-SUPERVISED GRAPH-BASED FACE SEGMENTATION APPROACH

This section presents details of the Integrated Semi-Supervised Graph-based (IntSSG) face segmentation approach. With color and depth images from Kinect, IntSSG integrates skin detection, SIFT matching, depth reasoning and smoothness propagation in a unified semi-supervised graph framework.

2.1. Main Framework

Let $\mathbf{I}^t$ be the color image frame obtained at time t, and $\mathbf{D}^t$ be the corresponding depth image. Denote the color value of one pixel by $\mathbf{x} \in \mathbb{R}^3$, and its class label by $y \in \{1, -1\}$ (1 for face class and -1 for non-face class). The goal of IntSSG face segmentation is to classify each pixel as either face or non-face class, i.e., estimating the labels $\mathbf{Y}^t = \{y_1^t, y_2^t, \ldots, y_n^t\}$ for all the n pixels.

IntSSG follows a semi-supervised graph-based scheme [10] to obtain $\mathbf{Y}^t$: first annotate a small portion of pixels as prior labels, then propagate the prior labels to all pixels. More precisely, all the labels are estimated by minimizing the following energy function:

$$E_s + \lambda E_f, \tag{1}$$

This work was supported in part by the National Natural Science Foundation of China under Grants 61175025, 61272049, 61005013, and 61203279, and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA06030300).

where E_s is the propagation energy or smoothness measure, and E_f is the fitting energy of prior labels. E_s plays the role of transferring prior labels to other pixels. Usually E_s is represented by a Laplacian regularization function in the label space. E_f ensures that the estimated labels of annotated pixels do not differ from their given prior labels. The weighting constant λ balances two energies.

IntSSG works as follows. Skin color detection and SIFT keypoint matching are combined to first annotate partial pixels. In case that wrong labeled values are included, possible outliers are then rejected by investigating the depth cues. The labels of the remaining pixels are treated as prior labels. After that, the smoothness measure is retrieved from the color image. At last, the semi-supervised segmentation is conducted incorporating the two energies in Eqn. 1. The flowchart is illustrated in Fig. 1.

The uniqueness of IntSSG lies in two aspects. First, unlike [7], IntSSG integrates skin detection, temporal SIFT matching, and depth reasoning in a unified graph based framework. Second, although IntSSG is based on semi-supervised graph, the labeling process is fully automatic. Interactive sketching in [10] is avoided.

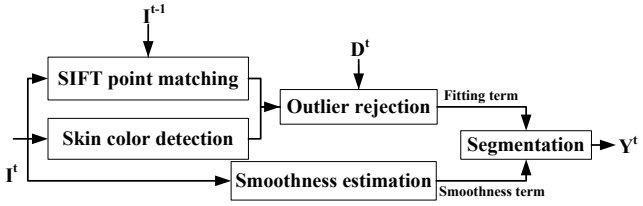

Fig. 1. The flowchart of the method. $\mathbf{I}^t$ and $\mathbf{D}^t$ represent the color and depth image at frame t respectively, with $\mathbf{I}^{t-1}$ the color images for frame t-1. Skin color detection and SIFT point matching between two frames provide labels for partial pixels. The possible outliers are rejected with the aid of depth image. A smoothness term is estimated from the color image. With the labeled pixels and smoothness constraint, the labels for all pixels $\mathbf{Y}^t$ are calculated.

2.2. Smoothness Term

Smoothness term is to ensure that neighboring pixels have similar labels. Among the various definitions [10, 11], we utilize the one in [10] based on two specific reasons. First, it defines a discriminative smoothness term, which is particularly suitable for face segmentation. Second, the nonlinear essence between pixel feature and labels are captured in the local spline model of [10]. The smoothness term is usually defined as the Laplacian kernel of unlabeled data [11]

$$E_s = \mathbf{y}^T \mathbf{M} \mathbf{y}. \tag{2}$$

To construct the Laplacian matrix $\mathbf{M}$, we follow [10] that local pixel colors and labels are correlated with a spline regression function. Unlike linear regression [11], spline regression handles the exception that spatially continuous pixels have very different labels. Correspondingly, the segmented image is more accurate with [10] especially at the boundaries.

2.3. Generating Prior Labels

Prior labels are automatically generated by two complementary techniques: skin color detection and SIFT keypoint matching. Skin color is a relatively unique and robust feature to localize face. Even under severe pose and emotion variances, face can be still identified by its color. On the other hand, skin color is easily affected when lighting condition changes. Accordingly, skin color detection alone is inadequate to provide accurate prior labels. Fortunately, some features like gradients does not change heavily when lighting varies among successive frames. Therefore, besides skin detection, we match SIFT keypoints detected at two successive frames such that the labels at previous time can also be used at current time. By combining the two methods, reliable prior labels are generated in facing various challenges such as large pose, lighting and facial emotion variations.

The mathematical summary of labeling terms is as follows

$$E_f = \sum_{i=1}^{n_1} (y_{l_i} - \hat{y}_{l_i}^{tc})^2 + \sum_{j=1}^{n_2} (y_{m_j} - \hat{y}_{m_j}^{ts})^2, \tag{3}$$

where n_1 and n_2 are the numbers of prior labels of skin color detection and SIFT keypoint matching, respectively. $\hat{y}_l^{tc}$ and $\hat{y}_m^{ts}$ are their prior labels, while y_l and y_m are the labels to derive. The implementation details of skin detection and SIFT matching are as follows:

Skin Color Detection. The robust detection of human skin remains a challenge although many efforts have been made. Learning-based skin detection is recently proven efficient [12]. Since we prefer small-scale training, we utilize the most recent linear regression tree based skin detection approach [12]. The tree structure of this approach allows hierarchical discriminations of skin and nonskin pixels and only requires a small number of training data.

The linear regression tree for skin detection is constructed as follows. At each internal node, we linearly regress each color feature $\mathbf{x}$ into its label y, and splits the node into its left or right child node with a threshold. At a leaf node, we define the likelihoods for skin $P(\mathbf{x}|y=1)$ and nonskin $P(\mathbf{x}|y=-1)$ as the frequencies that skin and nonskin pixels arrive this node. The final segmentation decision is discriminatively expressed as

$$\hat{y} = \begin{cases} 1, & \dfrac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=-1)} > \alpha \\ -1, & \dfrac{P(\mathbf{x}|y=-1)}{P(\mathbf{x}|y=1)} > \alpha \\ 0, & otherwise \end{cases}, \tag{4}$$

where $\hat{y} = 1$ is the label for skin, -1 for nonskin, and 0 for no label. α is the threshold for deciding skin or nonskin points. To train the tree, we manually annotate faces on 10 color images which are captured for each scene and randomly sample 3000 skin and 5000 nonskin pixels from each image. The training procedure takes ≈ 2 seconds implemented by MATLAB.

SIFT Keypoint Matching. The purpose of SIFT keypoint matching [13] is to transfer the labels at previous time to the current time. The actual procedures are

1. Detect SIFT keypoints on the current frame and match them with the face SIFT keypoints detected at previous time;
2. Use RANSAC [14] to iteratively remove erroneous matches and estimate the affine transformation between the current and previous frames;
3. Based on the estimated transformation, the labeled pixels at previous frame are mapped to the current frame, which are correspondingly as candidate prior.

In general face appearance changes slightly between successive frames, thus the previous SIFT points can be easily matched with the currently detected ones. Using affine transformation to mimic head pose helps to remove the spatially wrong SIFT matches. The insensitivity of SIFT to illumination guarantees that we still have enough prior labels when skin detection fails due to lighting variations.

2.4. Removing the Outlier Labels

In some regions, the face and background pixels are very similar. Thereby, the labels obtained by skin detection and SIFT matching may contain some errors. With the smoothness term, these errors are further propagated to more pixels, which finally leads to a visibly wrong face segmentation. To solve this problem, it is vital to reject the outlier pixels introduced in skin detection and SIFT matching.

With the availability of depth map, we use depth thresholding to remove potential outliers. Face lies roughly in a plane when user stays in a front of a camera. The depth values of face pixels are coarsely equal to each other. Therefore the pixel found by skin detection or SIFT matching with a distinguishably different depth value can be safely removed as an outlier. Since the depth sensors in Kinect are immune to visible lights, the influences of lighting variation on skin detection and SIFT matching are further reduced.

The details of removing outlier labels are listed as follows:

1. Detect edge on depth map and segment the map into several connected regions by connecting detected edges;
2. Treat the region where skin points or face SIFT points are mostly located as human region;
3. Conduct over-segmentation on the color image of the human region [15, 16];
4. Merge the over-segmented regions, where the number of detected skin points or face SIFT points is larger than a threshold, as the head region.
5. Dilate and Erode the head region to create prior labels.

3. EXPERIMENT

For validation purpose, we captured face videos with the commodity Kinect camera[1]. The main challenges in these videos include different scenes (indoor and outdoor), large head pose motion and drastic illumination changes.

3.1. Experimental Settings

We compare our integrated semi-supervised graph-based face segmentation approach (IntSSG) with three different segmentation approaches: GrabCut [11], KinectSeg [17] and GrabCutD [18]. GrabCut works only on color image while KinectSeg and GrabCutD utilize both color and depth images. In KinectSeg, the graph is constructed by considering both color and depth features. GrabCutD fuses color and depth directly into four dimensional feature and learn GMMs. The three approaches all require initial inputs from user. Since our own approach is automatic, for fair comparison, we choose the bounding rectangle of the segmented face described in Section 2 as inputs to start these approaches. The original RGB color space is utilized for all the methods.

The key parameters of these approaches are configured as follows. In our own approach, γ is experimentally set to 10000 and α to 1.5. The parameter k which balances color and depth cues in KinectSeg [17] is 0.8. The value of ϕ which is the weight of depth cue in GrabCutD [18] is 0.2.

To quantitatively evaluate the segmentation results, we adopt the commonly used F-Score metric:

$$F = 2 * \frac{precision * recall}{precision + recall}, \quad (5)$$

where $precision$ and $recall$ are the precision and recall rate, respectively. The metric ranges from 0 to 1. A larger F-Score value indicates a more accurate segmentation.

3.2. Results and Analysis

Fig. 2. Segmentation results on the outdoor video. From left to right: segmentation window, the results of GrabCut, KinectSeg, GrabCutD and our IntSSG. From top to bottom are frames 80 103 134 136 163.

Table 1. Mean F-score values of the four methods. IntSSG achieves higher values on both the indoor and outdoor videos illustrating its better segmentation accuracy under pose and illumination variations.

Video	GrabCut	KinectSeg	GrabCutD	IntSSG
Indoor	0.8473	0.8737	0.7660	**0.9346**
Outdoor	0.9105	0.9041	0.8986	**0.9826**

Fig. 2 shows segmentation results on an outdoor sequence. IntSSG succeeds in segmenting the faces no matter how head pose varies throughout the sequence. Obviously the segmented face boundaries of IntSSG are most accurate. For example, ears and hairs are correctly segmented. To reach the same qualities, the comparative approaches often need very selective sketch input. Fig. 3 shows the corresponding F-Scores on this sequence. The incontinuity in the curves comes from the segmentation failures at some frame. The F-Scores of IntSSG are stable whereas the others vary

[1] http://www.xbox.com/en-US/kinect

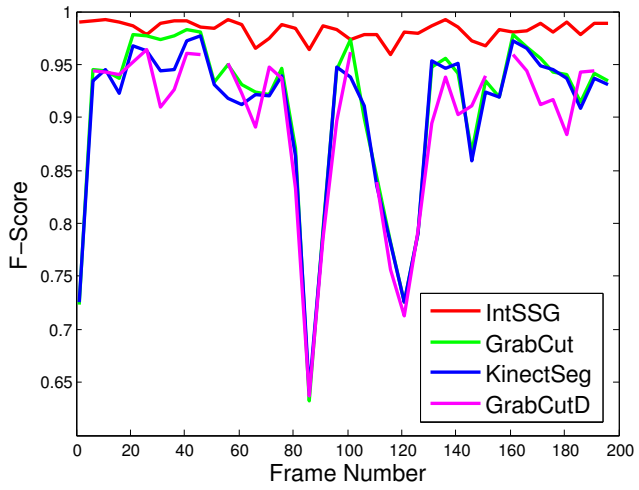

Fig. 3. F-Scores of the four methods on the outdoor video. IntSSG achieves higher and more stable values reflecting that it extracts more accurate faces through the whole sequence.

Fig. 4. Segmentation results on the second video. From left to right: segmentation window, the results of GrabCut, KinectSeg, GrabCutD and IntSSG. From top to bottom are frames 69 71 165 179.

drastically. At frame 85, the F-Scores of GrabCut, GrabCutD and KinectSeg become very small, indicating poor segmentations. This happens mainly because head pose is too large at this moment. However, the segmentation of IntSSG remains good because the skin detection module is insensitive to head pose. Note that we manually annotated faces every 5 frames.

Fig. 4 shows how IntSSG is robust to illuminations. At the first three rows light is strong, but suddenly it becomes weak at the last two rows. Intuitively, the segmented faces of IntSSG are more accurate than those of the other methods. Table. 1 shows the corresponding F-Scores on this sequence. For evaluating these values, we manually labeled faces on every five frames, computed F-Score on each labeled frame, and took the mean of these values as the F-Score value on this sequence. The values in Table. 1 quantitatively supports that IntSSG yields more accurate faces.

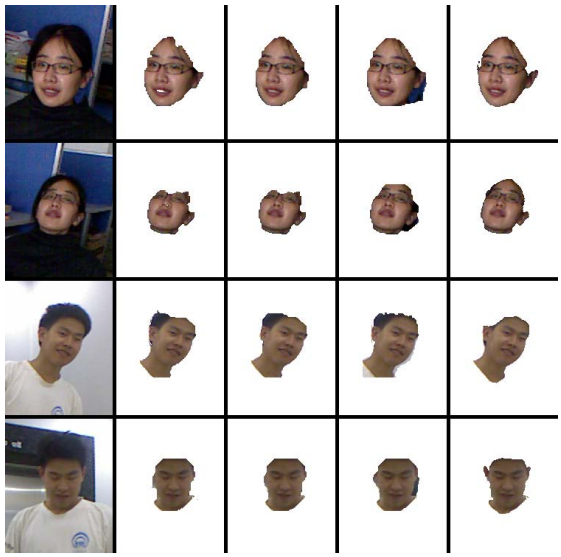

Fig. 5. Snapshots of the results on two other videos by our method. From left to right: segmentation window, the results of GrabCut, KinectSeg, GrabCutD and IntSSG.

More results were obtained on other videos as displayed in Fig. 5. From it, more accurate faces are obtained by our IntSSG.

All the above results show that IntSSG is robust to pose and lighting variations. Comparatively, the three interactive approaches, GrabCut, GrabCutD and KinectSeg, are less accurate. One main reason is that the input to GrabCut, GrabCutD and KinectSeg is a rough face rectangle, which is inadequate to train the GMM color models. Moreover, the depth map of Kinect contains noises and even large black holes. Without a remedy, the data issues are propagated and lead to segmentation inaccuracy in GrabCutD and KinectSeg. Fortunately, our IntSSG can avoid these problems by introducing the outlier removing procedure.

4. CONCLUSION

We have presented an automatic face segmentation method that can be applied in virtual reality or immersive reality. It relied on color and depth information without human intervention. Comparative experiments with several methods demonstrated its efficiency in video segmentation. Accurate face regions were obtained under different poses, scales and illumination changes. These owe to the reliable and robust semantics about face. Future work will address its improvement for solving occlusions especially by hands and for solving multi-faces in the video.

5. REFERENCES

[1] H. Li and K.N. Ngan, "Automatic video segmentation and tracking for content-based multimedia services," *IEEE Commun. Mag.*, vol. 45, pp. 27–33, Jan. 2007.

[2] Jose Maria Buades Rubio, Francisco J. Perales Lpez, and Xavier Varona, "Real Time Segmentation and Tracking of Face and Hands in VR Applications," in *Articulated Motion and Deformable Objects*, 2004, pp. 259–268.

[3] Hayit Greenspan, Jacob Goldberger, and Itay Eshet, "Mixture model for face-color modeling and segmentation," *Pattern Recognition Letters*, vol. 22, pp. 1525–1536, Dec. 2001.

[4] N. Habili, Cheng Chew Lim, and A. Moini, "Segmentation of the face and hands in sign language video sequences using color and motion cues," *IEEE Trans. Circuits Sys. Video Techn.*, vol. 14, pp. 1086 – 1097, Aug. 2004.

[5] Zhi Liu, Jie Yang, and Ning Song Peng, "An efficient face segmentation algorithm based on binary partition tree," *Signal Processing: Image Communication*, vol. 20, no. 4, pp. 295 – 314, 2005.

[6] Kuang chih Lee, D. Anguelov, B. Sumengen, and S.B. Gokturk, "Markov random field models for hair and face segmentation," in *Proc. IEEE Int'l Conf. Automatic Face Gesture Recognition*, Sep. 2008, pp. 1 –6.

[7] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applicaiton," *IEEE Trans. Circuits Sys. Video Techn.*, vol. 9, pp. 551–564, Apr. 1999.

[8] Huitao Luo and A. Eleftheriadis, "Model-based segmentation and tracking of head-and-shoulder video objects for real time multimedia services," *IEEE Trans. Multimedia*, vol. 5, no. 3, pp. 379 – 389, Sep. 2003.

[9] Hongliang Li, King N. Ngan, and Qiang Liu, "Faceseg: automatic face segmentation for real-time video," *IEEE Trans. Multimedia*, vol. 11, pp. 77–88, Jan. 2009.

[10] Shiming Xiang, Feiping Nie, and Changshui Zhang, "Semi-supervised classification via local spline regression," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2039–2053, Nov. 2010.

[11] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut–interactive foreground extraction using iterated graph cuts," in *Proc. ACM SIGGRAPH*, 2004, pp. 309–314.

[12] Jixia Zhang, Haibo Wang, Franck Davoine, and Chunhong Pan, "Skin detection via linear regression tree," in *Proc. IAPR Int'l Conf. Pattern Recognition*, 2012, pp. 1711–1714.

[13] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[14] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, Jun. 1981.

[15] P. Meer D. Comanicu, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, May. 2002.

[16] B. Georgescu P. Meer, "Edge detection with embedded confidence," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1351–1365, Dec. 2001.

[17] Z. Tomori, R. Gargalik, and I. Hrmo, "Active segmentation in 3d using kinect sensor," in *Proc. Int'l Conf. Computer Graphics, Visualization and Computer Vision*, 2012.

[18] Karthikeyan Vaiapury, Anil Aksay, and Ebroul Izquierdo, "Grabcutd: improved grabcut using depth information," in *Proc. ACM workshop on Surreal media and virtual cloning*, New York, NY, USA, 2010, pp. 57–62, ACM.