Automatic Bi-Layer Video Segmentation Based on Sensor Fusion

Qiong Wu, Pierre Boulanger and Walter F. Bischof
Department of Computing Science
University of Alberta
{qiong,pierreb,wfb}@cs.ualberta.ca

Abstract

We propose a new solution to the problem of bi-layer video segmentation in terms of both, hardware design and algorithmic solution. At the data acquisition stage, we combine color video with infrared video, which is robust to illumination changes and provides an automatic initialization of the cue map for foreground-background segmentation. Two algorithms are presented to complete the segmentation, graph cut and contrast-preserving relaxation labeling. Both algorithms use color and edge information. Our experimental results show that the better performance of contrast-preserving relaxation labeling over graph cut, and the parallel characteristics make relaxation labeling superior to graph cut for implementation in GPU hardware.

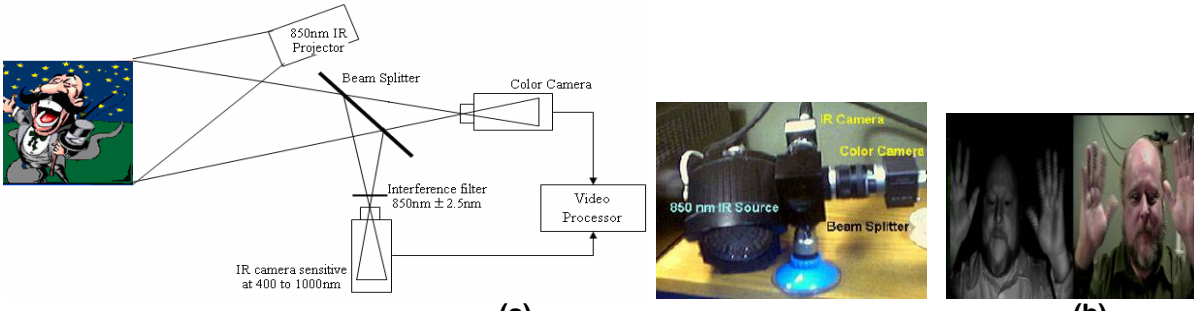
1. Introduction

Many tasks in computer vision involve bi-layer video segmentation. One important application is in teleconferencing, where there is a need to substitute the original background with a new one. Here we present a solution that segments the foreground layer from the natural background automatically, efficiently, accurately and robust to illumination changes.

A large number of papers have been published on bi-layer video segmentation. In order to achieve the goal of automatic image segmentation, more information is needed than what is given in the original image. For example, background subtraction seeks the solution using adaptive thresholding with a background model [1]. Segmentation of stereo video computes depth information to assist in foreground segmentation [2, 3]. These methods all have several limitations: they are not robust to illumination changes, and they have low computational efficiency and segmentation accuracy. Recently, several researchers have used a depth-camera in combination with a regular camera to acquire depth data to assist in foreground segmentation [4, 5]. The way they combine the two cameras, however, involves scaling, resampling and dealing with synchronization problems. There are special video cameras available today that produce both depth and RGB signals, e.g. ZCam [6], but they are very expensive and difficult to integrate with normal video technology.

In a recent paper [7], we proposed to combine infrared (IR) video and color video for foreground segmentation. We combined the IR camera with the color camera in a way that saves us from resampling and synchronization. The IR image is used to initialize the segmentation cue map, a pentamap, and graph cut (GC) is applied to optimize the segmentation result.

In this paper, we use the same data acquisition unit, but we propose to use contrast-preserving relaxation labeling (CPRL) to optimize the segmentation result. We find that, for a number of reasons, CPRL is



(a) (b)

Figure 1: (a) Data acquisition unit (from [10]). (b) An example of a video frame.

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

superior to GC, as will be discussed.

2. Proposed Method

In this section, we describe our data acquisition setup and the segmentation pentamap (see also [7]). We describe CPRL and GC in Section 3 and 4, respectively, and we give the experimental results in Section 5.

2.1. Data Acquisition Setup

Figure 1(a) shows the setup of our data acquisition unit. The foreground is illuminated by an invisible IR source at 850 nm that is captured by an infrared camera tuned to 850 nm, with a narrow-band (850nm ± 25 nm) filter being used to reject all light except the one produced by the IR illuminator. The IR camera and color camera produce a mirrored video pair that is synchronized in time and space, using a genlock mechanism for temporal synchronization and an optical beam splitter for spatial registration. With this system, there is no need to align the images using complex calibration algorithms since they are guaranteed to be coplanar and coaxial. An example of a video frame is shown in Figure 1(b). As one can see, the IR image is a mirror version of the color image. This is due to the beam splitter and can be easily corrected with a simple image transposition.

Our design has many advantages. First, it automatically produces synchronized IR and color video pairs, which saves us from tedious synchronization problems. Second, IR information is independent of illumination changes, hence a cue map can be stably initialized (see below). Third, the IR illuminator adds flexibility to the foreground definition. One can move the IR illuminator around to any object to be segmented. Hence the foreground is defined by the object within certain distance to the IR illuminator rather than to the camera.

2.2. Pentamap Initialization

One advantage of the IR image is that it can be used to predict foreground background areas. The IR image is a grayscale image, in which brighter parts indicate the foreground (illuminated by IR source). Missing foreground parts must be within a certain distance from the illuminated parts. Assuming this distance is τ (in number of pixels), any dark area outside of this distance belongs to the background. The value of τ increases with the distance of the object to the IR source. Suppose the effective distance for the IR source is D_{max}, $\tau=\tau_{max}=f(D_{max})$, and the value of τ does not change during the video capture if the user does not change the camera configuration.

For the foreground/background segmentation, we want to find a binary label vector $f=(f_1,f_2\ldots f_{|P|})$, where $f_n \in \{0,1\}$, with 1 the foreground label and 0 the background label, and $|P|$ the number of pixels. Initial estimates of foreground and background can be obtained from the IR image. After registration, each IR image is an array $I=(I_1,I_2\ldots I_{|P|})$, and the color image is an array $Z=(Z_1,Z_2\ldots Z_{|P|})$, where I_n is a grayscale value, and Z_n is an RGB color vector. An estimate of the foreground area can be found by thresholding the IR image: $MASK=\{p\in P|\ I_p \geq T\}$.

We define a pentamap as follows:

Definition: *A pentamap T partitions the image into five regions, certain foreground (CF), certain background (CB), local foreground (LF), local background (LB) and unknown (U), represented as T: P* $\rightarrow$ $\{T_{CF}, T_{CB}, T_{LF}, T_{LB}, T_U\}$, *with*

$T_{CF}=\{p|\ p\in MASK.erosion(s)\}$

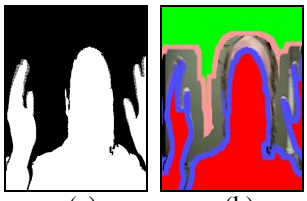
$T_{LF}=\{p|\ p\in MASK- T_{CF}\}$

$T_{CB}=\{p|\ p\in \sim(MASK.dilation(\tau+s))\}$

$T_{LB}=\{p|\ p\in MASK.dilation(\tau+s)- MASK.dilation(\tau)\}$

$T_U=\{p|\ p\in P- T_{CF}-T_{CB}- T_{LF}- T_{LB}\}$.

An example of MASK and the pentamap are shown in Figure 2. Sample foreground/background cues are drawn from T_{LF}/T_{LB}, which is an interior/exterior narrow strip of width s of the foreground. It is reasonable to assume that the color pixels in the unknown area T_U is consistent with the neighborhood regions. The value of s (in the definition of T_{CB} and T_{LB} above) does not change the segmentation result much as long as $s\in[15, 25]$.



(a) (b)

Figure 2: (a) Foreground MASK. (b) Pentamap with red=T_{CF}, green=T_{CB}, blue=T_{LF}, pink=T_{LB}, and the remaining area=T_U.

We use a Gaussian Mixture Model (GMM) to represent the foreground/background color model, which is derived from $\{Z_p|p\in T_{LF}\}/\{Z_p|p\in T_{LB}\}$. Each GMM is represented by a full-covariance Gaussian mixture with M components (M=10). We represent the GMM for the foreground as $K_F=\{K_{F1}, K_{F2}\ldots K_{FM}\}$, and similarly for the background $K_B=\{K_{B1},K_{B2}\ldots K_{BM}\}$.

In our experiments, we used the following parameter values: T=0.004 and τ=55.

3. Contrast Preserving Relaxation Labeling

Relaxation labeling [8] can be used to reduce ambiguities and noise based on the parallel use of local constraints between labels. In image segmentation, each pixel is first assigned a probability vector and a label based on the color information, and the probability vector is updated iteratively based on the local constraints between labels [9]. We apply relaxation labeling to complete our foreground segmentation based on the initial estimates described in Section 2. We propose a new local constraint involving contrast information, which we call contrast-preserving relaxation labeling. As in [9], we proceed in three steps.

Step 1: Initialization. For each pixel, compute a probability vector

$$\Pr^0(p) = [\Pr_1{}^0(p) \quad \Pr_0{}^0(p)]$$

where $\Pr_1{}^0$ is the probability of pixel p belonging to the foreground (f_p=1), and $\Pr_0{}^0$ the probability of belonging to the background (f_p=0). Based on the pentamap, the probability vector for pixels in the unknown area T_U are defined according to the following scheme:

$$\Pr_1{}^0(p) = \begin{cases} 1 & ,\forall p \in T_{CF}, T_{LF} \\ \dfrac{\max\limits_{i=1...M} \Pr(Z_p \mid K_{Fi})}{\max\limits_{i=1...M} \Pr(Z_p \mid K_{Fi}) + \max\limits_{i=1...M} \Pr(Z_p \mid K_{Bi})} & ,\forall p \in T_U \\ 0 & ,\forall p \in T_{CB}, T_{LB} \end{cases} \quad (1)$$

$$\Pr_0{}^0(p) = 1 - \Pr_1{}^0(p) \quad (2)$$

Step 2: Iteration. In the nth iteration, the probability vector $\Pr^n(p)$ for pixel p is updated based on the previous vector $\Pr^{n-1}(p)$ and neighborhood probability vector $\Pr^{n-1}(q)$, $q \in \overline{N}(p)$ where $\overline{N}(p)$ is the 8-connected neighborhood about pixel p.

$$\Pr_i{}^n(p) = \frac{\Pr_i{}^{n-1}(p)(1+Q_i{}^{n-1}(p))}{\sum_{j=0}^{1} \Pr_j{}^{n-1}(p)(1+Q_j{}^{n-1}(p))} \quad (3)$$

where

$$Q_i{}^{n-1}(p) = \frac{1}{card(\overline{N}(p))} \sum_{q \in \overline{N}(p)} C(p,q) * \Pr_i{}^{n-1}(q) \quad (4)$$

with i={0,1}. $C(p,q)$ is the compatibility coefficient, and it uses contrast information as follows:

$$C(p,q) = \begin{cases} 1, & \text{if} \left\| Z_p - Z_q \right\|^2 \le \theta \\ -1, & \text{otherwise} \end{cases} \quad (5)$$

Step 3: Convergence and final labeling. As shown in [8], (3) converges to a consistent labeling as $n \to \infty$. After running for n_f iterations, each pixel is assigned a label with a larger probability component. We found n_f=10 to be sufficient in our experiment.

4. Graph Cut

Many authors have modeled image segmentation as a graph cut problem (e.g., [10]). Here we describe how we build our graph model according to the pentamap defined in Section 2.

A graph G=(V,E) is defined as a set of nodes V with a set of edges E. Besides two terminal nodes OBJ (foreground) and BKG (background), T_{CF} and T_{CB} are represented as two separate nodes, and all other nodes correspond to each pixel in the other area. Edges are defined in Table 1. We build the graph model in this way to prevent an unwanted cut (e.g., across T_{CF}/T_{CB}) and to improve the computational efficiency (more details are given in [7]). In Table 1, K=∞ (a very large value in practice), γ is the relative importance of data term. The data term, which reflects color information, is the same as used in CPRL: $\Pr_0{}^0(p)$ and $\Pr_1{}^0(p)$ are defined in Section 3. The contrast term defined for CPRL is controlled here with the smoothness parameter α.

Edge	Weight	For
{Vi, OBJ}	K	Vi∈ T_{CF},T_{LF}
	0	Vi∈ T_{CB},T_{LB}
	-ln $\Pr_0{}^0(p)$* γ	Vi∈ T_U
{Vi, BKG}	K	Vi∈ T_{CB},T_{LB}
	0	Vi∈ T_{CF},T_{LF}
	-ln $\Pr_1{}^0(p)$* γ	Vi∈ T_U
(Vi,Vj)	$\alpha * \exp(-\left\|Z_i - Z_j\right\|^2 / \beta)$ β is the expectation of $2\left\|Z_i - Z_j\right\|^2$	i∈ $\overline{N}$(j) (8-connectivity), i≠j, V_i∈ $T_{LB}/T_{LF}/T_U$, V_j∈ T_U

Table 1: Edge weight table

5. Experimental Results

We compared the segmentation results produced by CPRL and GC. CPRL has only one parameter, the threshold θ of contrast information in (5). Segmentation results do not change much if θ ∈ [80,400]. GC, on the other hand, has three parameters α, β and γ (see Table 1), and segmentation results are very sensitive to these parameter values, as shown in Figure 3 and Table 2.

CPRL is also superior to GC because CPRL can be computed in a parallel algorithm. CPRL applies the same linear constraint to each pixel, hence all pixels can be processed at the same time, and CPRL can thus be implemented on a GPU. If we have 10 iterations in CPRL, the image needs to be processed in the GPU frame shader ten times. Given that the computation of each iteration can be finished in microseconds, the processing of each image can be done in real-time. We are currently working on a parallel implementation using CUDA.

6. Conclusion

In this paper, we propose to combine IR video and color video for bi-layer segmentation of natural scene. The proposed design can not only produce automatically synchronized video sequences, but can also be used to stably initialize the segmentation map. We presented two algorithms, CPRL and GC, to complete foreground segmentation. Our experiment results show that CPRL produces very good segmentation results and has a better stability with respect to variations of parameter values. Given the parallel nature of CPRL, one can take advantage of fast GPU processing. In our future work, we will implement CPRL in GPU for real-time bi-layer video segmentation.

7. References

[1] N. Friedman, S. Russell, "Image Segmentation in Video Sequences: a Probabilistic Approach", *Proc. 13th Conf. on Uncertainty in Artificial Intelligence*, Aug 1997, pp. 175-181.

[2] C. Eveland, K. Konolige, and R.C. Bolles, "Background modeling for segmentation of video-rate stereo sequences", *Proc. IEEE Computer Vision and Pattern Recognition (CVPR),* Santa Barbara, CA, USA, Jun 1998, pp. 266-271.

[3] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, "Bi-layer Segmentation of Binocular video", *Proc. CVPR,* San Diego, CA, US, 2005, pp. 407 – 414.

[4] N. Santrac, G. Friedland, R. Rojas, "High resolution segmentation with a time-of-flight 3D-camera using the example of a lecture scene", *Technical Report B-06-09, Department of Computer Science, Freie Universität Berlin*, Sep 2006.

[5] O. Wang, J. Finger, Q. Yang, J. Davis, and R. Yang, "Automatic Natural Video Matting with Depth", *Pacific Conference on Computer Graphics and Applications (Pacific Graphics)*, 2007.

[6] G. Iddan and G. Yahav, "3D Imaging in the studio (and elsewhere)", *Proc. SPIE*, 2001, pp. 48-55.

[7] Q. Wu, P. Boulanger and W. F. Bischof, "Robust Real-Time Bi-Layer Video Segmentation Using Infrared Video", *Canadian Conference on Computer and Robot Vision (CRV)*, May 2008.

(a) (b) (c)

Figure 3: Comparison of segmentation results produced by CPRL and GC

CPRL: (a) θ=80 GC: (b) α=2, β=200 and γ=2.2 (c) α=10, β=200 and γ=2.2.

Algorithm	Parameter values	Error rate
GC	α=2, β=200, γ=2.2	0.0268
	α=1, β=200, γ=10	0.0338
	α=10, β=200, γ=2.2	0.0367
CPRL	θ=80	0.0231
	θ=400	0.0241

Table 2: Error rate of GC and CPRL with different parameter values

(a) (b) (c) (d)

Figure 4: Convergence of CPRL.

Pixels in the pink color has the probability $Pr_0^n(p) \in [0.2,0.8]$ at the nth iteration. (a) n=0, (b) n=1, (c) n=2, (d) n=10.

[8] R.A. Hummel and S.W. Zucker, "On the Foundations of Relaxation Labeling Processes", *IEEE Trans. Pattern Analysis and Machines Intelligence,* May 1983, pp. 267-287.

[9] M. W. Hansen and W. E. Higgins, "Relaxation Methods for Supervised Image Segmentation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Sep 1997, pp. 949-962.

[10] Y. Boykov, and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images", *Proc. IEEE Int. Conf. on Computer Vision*, 2001, CD-ROM.