TofCut: Towards Robust Real-time Foreground Extraction Using a Time-of-Flight Camera

Liang Wang[†] Chenxi Zhang[†] Ruigang Yang[†] Cha Zhang[‡]

[†]Center for Visualization and Virtual Environments, University of Kentucky, USA
[‡]Microsoft Research, Redmond, USA

Abstract

Foreground extraction for live video sequence is a challenging task in vision. Traditional methods often make various assumptions to simplify the problem, which make them less robust in real-world applications. Recently, Time-of-Flight (TOF) cameras provide a convenient way to sense the scene depth at video frame-rate. Compared to the appearance or motion cues, depth information is less sensitive to environment changes. Motivated by the fact that TOF cameras have not been widely used in video segmentation application, we in this paper investigate the problem of performing robust, real-time bi-layer segmentation using a TOF camera and propose an effective algorithm named TofCut. TofCut combines color and depth cues in a unified probabilistic fusion framework and a novel adaptive weighting scheme is employed to control the influence of these two cues intelligently. By comparing our segmentation results with ground truth data, we demonstrate the effectiveness of TofCut on an extensive set of experimental results.

1. Introduction

Automatic layer extraction from videos has been one of the most extensively researched topics in computer vision. One of the prime applications is video teleconferencing, where the background content can be replaced by other images or videos. Such a live background substitution module is both fun and aesthetically pleasing, and it can protect privacy information of the users.

Performing bi-layer segmentation on live videos is a challenging task in real-world scenarios. In a typical office environment, illumination may change dramatically due to light switch. The background appearance may vary from time to time, *e.g.*, there may be people walking around behind the user. People may also like to do video chat using laptops, which means casual camera shaking or gradual camera move can happen occasionally.

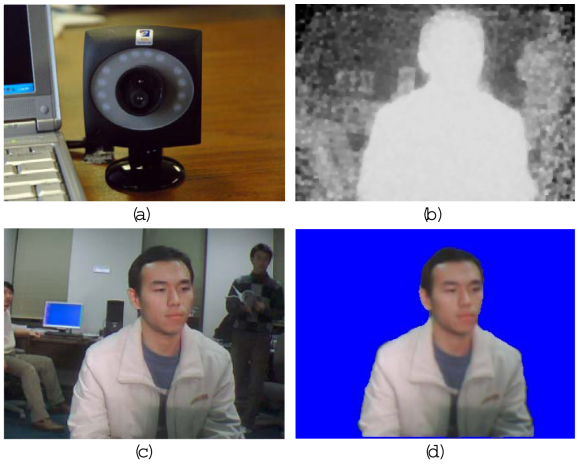

Figure 1. Using TOF cameras for foreground/background segmentation. (a) ZCam from 3DV Systems. (b) The depth image. (c) The color image. (d) Our segmentation result.

This work is related to a sizable body of literature on foreground/background segmentation [4, 8, 12, 13, 15, 14, 18, 19]. People usually make various assumptions to simplify the problem. For instance, in [8, 11, 15, 18], the camera is assumed as stationary and the background appearance is either previously known or near static. These assumptions may be acceptable in personal offices, but can be invalidated in meeting rooms, shared labs, etc. Recently, Zhang *et al*. [20] proposed to use a structure from motion algorithm to estimate depth information, hence they are able to handle dynamic scenes and moving cameras. Nevertheless, their method is an off-line approach, the relative long processing time (6 minutes per frame as reported in the paper) makes it impractical for live video segmentation. In [10, 11, 12], stereo cameras are deployed to compute the scene depth and the fusion of color and depth information lead to improved segmentation results compared to using the color cue along. On the other hand, passive stereo matching remains a difficult vision problem and is prone to errors under low lighting environments or when the scene contains large textureless

regions like the white wall. Unfortunately, it is always the case, especially in man-made scenes like office interiors, that not enough textures exist.

Recently, Time-of-Flight (TOF) cameras start to attract the attention of many vision researchers. TOF cameras are active sensors that determine the per-pixel depth value by measuring the time taken by infrared light to travel to the object and back to the camera. These sensors are currently available from companies such as 3DV Systems [1], Canesta [2] and Mesa Imaging [3] at commodity prices. So far TOF cameras have not been widely used in video segmentation application. Existing TOF camera-based solutions [7, 17] directly take the depth image from the TOF camera and threshold it to compute a foreground mask. These local approaches are simple but are also less robust. Since TOF cameras are characterized by independent pixel depth estimates, the measured depth map can be noisy both spatially and temporally. Such noises are content dependent and hence difficult to remove by typical filtering methods. More critically, when there exist background objects whose depth is close to the foreground layer, thresholding can lead to false segmentation.

In this paper, we investigate the problem of performing robust real-time bi-layer segmentation using a TOF camera. We are motivated by the fact that utilizing TOF cameras for foreground extraction, although seems natural, remains a challenging and unsolved problem in practice. Toward this end, we propose an effective bi-layer segmentation algorithm named *TofCut*. TofCut combines color and depth cues in a unified probabilistic fusion framework. A distinct feature of TofCut is a novel adaptive weighting scheme that is able to adjust the importance of these two cues intelligently over time. The global inference problem can be efficiently solved using the graph cuts method [6]. In summary, our work makes the following main contributions:

- An algorithm designed for TOF camera-based foreground/background segmentation application. The algorithm is tailored for robustness and efficiency and works well under a variety of challenging environments.
- A thorough comparative evaluation using ground truth data is presented to assess the performance of different signals/methods and gauge the progress of TOF camera-based video segmentation.
- A data set that contains 4 video sequences captured with a ZCam from 3DV Systems. Color images, depth maps, and their manually labeled ground truth segmentation results are included. We hope that publicizing this data set can inspire more future work in the domain of vision research with TOF cameras.

The rest of paper is organized as follows. After introducing the problem formulation and notations in section 2, we in section 3 present the TofCut algorithm and the adaptive weighting scheme. In section 4, we provide our experimental comparison of different methods and discuss our results. Finally, we conclude in Section 5 with planned future work.

2. Problem Formulation

Let I^t be the RGB color image at time instance t, and D^t be its corresponding depth map returned by the TOF camera (as shown in Figure 1 (b)(c)). Let Ω be the set of all pixels in I^t. The color and depth values of pixel $r \in \Omega$ are denoted as I_r^t and D_r^t, respectively. For notation clarity we assume I_r^t and D_r^t are color and depth measurements of the same scene point in 3D. In practice for most TOF cameras, the optical center of the color sensor and the depth sensor do not overlap but are very close and the depth map can be warped to align with the color image. In the following, when there is no confusion, we will omit the superscript t for conciseness.

Following the general framework in [5], we formulate the foreground/background segmentation as a binary labeling problem. More specifically, a labeling function f assigns each pixel r a unique binary label $\alpha_r \in \{0(background), 1(foreground)\}$. The optimal labeling can be obtained by minimizing the energy of the form:

$$E(f) = \sum_{r\in\Omega} U(\alpha_r) + \lambda \sum_{(r,s)\in\xi} V(\alpha_r, \alpha_s), \qquad (1)$$

where $\sum U(\cdot)$ is the *data term* that evaluates the likelihood of each pixel belonging to foreground or background. The contrast term $\sum V(\cdot,\cdot)$ encodes the assumption that segmentation boundaries are inclined to align with edges of high image contrast. ξ denotes the set of 8-connected neighboring pixel pairs. λ is a strength parameter that balances the two terms. The contrast term used in our paper is defined as:

$$V(\alpha_r, \alpha_s) = |\alpha_r - \alpha_s| \exp(-\frac{\|I_r - I_s\|^2}{\beta}), \qquad (2)$$

where $\|I_r - I_s\|^2$ is the Euclidean norm of the color difference and β is chosen to be $\beta = 2\langle\|I_r - I_s\|^2\rangle$ ($\langle\cdot\rangle$ indicates expectation) [8, 14].

The color and depth information obtained from the ZCam is combined to form the data term. That is, $\sum U(\cdot)$ consists of two parts:

$$\sum_{r\in\Omega} U(\alpha_r) = \lambda^c \sum_{r\in\Omega} U^c(\alpha_r) + \lambda^d \sum_{r\in\Omega} U^d(\alpha_r), \qquad (3)$$

where $\sum U^c(\cdot)$ is the color term, which models the foreground and background color likelihoods. $\sum U^d(\cdot)$ is the depth term that models depth likelihood of the scene. λ^c and λ^d are two parameters that control the influences of these two terms.

3. TofCut

In this section, we present details of the TofCut algorithm by first introducing how to model the color and depth terms from the TOF camera's measurements. More importantly, by noticing color and depth may have different discrimination capability during different time periods we propose a novel fusion method which is able to adjust the influence of color and depth cues adaptively over time.

3.1. Likelihood for Color

To model the likelihood of each pixel r belonging to foreground or background layer, a foreground color model $p(I_r|\alpha_r = 1)$ and a background color model $p(I_r|\alpha_r = 0)$ are learned from image data. In [4, 12, 14, 15], both likelihoods are modeled with Gaussian Mixture Models (GMMs) and learned using Expectation Maximization (EM). However, a good initialization of the EM algorithm is difficult to obtain, and the iterative learning process of EM will slow the system. Criminisi *et al*. modeled the color likelihoods nonparametrically as color histograms [8]. We notice that the performance of their simplified approach is sensitive to the user specified number of color bins. In this paper we propose to use a hybrid approach. We first construct histograms for the foreground/background pixels respectively, and then build foreground/background GMMs based on the 3D color histograms.

More specifically, two 3D histograms each with H (typically H=8^3) bins in the RGB color space is constructed for the foreground and background separately. Gaussian components for the GMMs are learned using the samples in each bin, denoted as $\{\mu_1^f, \Sigma_1^f, \omega_1^f\}, \ldots, \{\mu_H^f, \Sigma_H^f, \omega_H^f\}$ for the foreground and $\{\mu_1^b, \Sigma_1^b, \omega_1^b\}, \ldots, \{\mu_H^b, \Sigma_H^b, \omega_H^b\}$ for the background, respectively. Here μ is the color mean, Σ is the covariance matrix assumed to be diagonal, and ω_i is the component weight approximated by the corresponding value of the ith color bin. Given a pixel I_r belonging to the bin $\boldsymbol{B}$, the conditional probability $p(I_r|\alpha_r = 1)$ is computed as:

$$p(I_r|\alpha_r = 1) = \frac{\sum_{i\in\aleph} \omega_i^f G(I_r|\mu_i^f, \Sigma_i^f)}{\sum_{i\in\aleph} \omega_i^f}. \quad (4)$$

where $\aleph$ is the index set of $\boldsymbol{B}$'s neighboring bins in 3D. Finally, the color term is defined as:

$$\sum_{r\in\Omega} U^c(\alpha_r) = -\sum_{r\in\Omega} \log p(I_r|\alpha_r). \quad (5)$$

We found the above scheme is quite stable and sufficiently efficient for real-time implementation. Note that both the foreground/background color likelihood models are learned and updated over successive frames, based on the segmentation results of the previous frame. This continuous learning process allows us to estimate the color models more accurately according to the very recent history.

3.2. Likelihood for Depth

Under ideal conditions TOF cameras are capable of relatively accurate measurements. However, in practice the quality of measurements is subject to many factors. The most well known problem is the measured depth suffers from bias as a function of object intensity. That is, dark objects will appear farther in the returned depth map compared to their actual depth w.r.t. the camera. This depth bias will cause dark foreground regions being labeled occasionally as background and result in "flickering" artifacts. In previous literature, researchers usually compensate this bias through a laborious photometric calibration step [9, 21]. In order to alleviate this bias without resorting to the pre-calibration, we take the intensity bias into consideration when building the foreground/background depth models.

The foreground/background depth likelihoods are modeled as Gaussian distributions. Foreground/background pixels from the latest segmented frame are first classified into *dark* and *bright* samples based on an intensity threshold $T = 60$. For each foreground or background model, two Gaussian distributions are learned using the *dark* and *bright* sample sets, respectively. Let $\{\chi^f, \nu^f\}$ and $\{\chi'^f, \nu'^f\}$ represent the two Gaussian components of the foreground depth model, *e.g*., the conditional probability $p(D_r|\alpha_r = 1)$ is:

$$p(D_r|\alpha_r = 1) = \begin{cases} G(D_r|\chi^f, \nu^f) & I_r < T \\ G(D_r|\chi'^f, \nu'^f) & Otherwise. \end{cases} \quad (6)$$

Similarly, $p(D_r|\alpha_r = 0)$ can be defined using the corresponding Gaussian models $\{\chi^b, \nu^b\}$, $\{\chi'^b, \nu'^b\}$ of the background. The depth term can then be written as:

$$\sum_{r\in\Omega} U^d(\alpha_r) = -\sum_{r\in\Omega} \log p(D_r|\alpha_r). \quad (7)$$

3.3. Adaptive Weighting

In previous work such as [12], the color and depth cues are treated equally, *i.e*., λ^c and λ^d are constant over time. However, color and depth may have different discrimination power at different periods. Clearly a robust fusion algorithm should adaptively adjust the importance of different cues over time. For instance, when there are background objects approaching the foreground the depth cue is ambiguous. In that case if the foreground/background colors can be well separated the algorithm should rely more on the color cue. Likewise, if there is a sudden illumination change, the color statistics learned from previous frames is less reliable so depth cue should be favored more. Motivated by this observation, we propose to adaptively adjust the weighting factors to improve the robustness of the segmentation process.

Weighting factors λ^c and λ^d are determined based on the discriminative capabilities of the color and depth models. To measure the reliability of the color term, we compute the Kullback-Leibler (KL) distance between the gray-scale histograms of frames I^{t-1} and I^t together with the KL distance between the color histograms of the separated foreground/background layers in I^{t-1}. We denote the two gray-scale histograms as h^{t-1} and h^t. Each histogram has eight bins and the values are normalized so $\Sigma h^{t-1}(i) = \Sigma h^t(i) = 1$. The KL distance between them is

$$\delta_{lum}^{KL} = \sum_{i=1}^{8} h^t(i) \log \frac{h^t(i)}{h^{t-1}(i)}. \quad (8)$$

The KL distance between the foreground and background color histograms as defined in Section 3.1 for frame I^{t-1} can be computed in a similar way. We denote their corresponding KL distance to be δ_{rgb}^{KL}.

The confidence of the color term is computed using δ_{lum}^{KL} and δ_{rgb}^{KL} as

$$\Re^c = \exp(-\frac{\delta_{lum}^{KL}}{\eta_{lum}^c}) \cdot (1 - \exp(-\frac{\delta_{rgb}^{KL}}{\eta_{rgb}^c})), \quad (9)$$

where η_{lum}^c and η_{rgb}^c are parameters that control the sharpness of the exponential functions. If δ_{lum}^{KL} is small, we assume there is no significant illumination changes or background color variation between image I^{t-1} and I^t, therefore the learned color histograms from I^{t-1} should be reliable. On the other hand, if δ_{rgb}^{KL} is small it implies the foreground and background layers have similar color distributions and accurate segmentation via color cue is difficult.

The confidence of the depth term is calculated based on the distance between the average depth values of the foreground and background layers in I^{t-1}. The distance can be approximated from the depth likelihood models defined in Section 3.2 as $\Delta\chi = |(\chi^f + \chi'^f) - (\chi^b + \chi'^b)|/2$. The confidence of the depth term is defined as

$$\Re^d = 1 - \exp(-\frac{\Delta\chi}{\eta^d}), \quad (10)$$

where η^d is a constant parameter. The confidence $\Re^d$ is small if the distance between the foreground and background layers is small. The weighting factors are then computed as $\lambda^c = \Re^c/(\Re^c + \Re^d)$ and $\lambda^d = \Re^d/(\Re^c + \Re^d)$.

We now provide the parameter setting for η_{lum}^c, η_{rgb}^c and η^d. According to our experiments, η_{lum}^c is the least sensitive parameter among the three and is set to 0.1 throughout our experiments. η_{rgb}^c and η^d require more tuning in practice. In our implementations η_{rgb}^c is chosen between 1.2 and 2.5 and η^d's range is from 45 to 60 (Note, $0 \leq \Delta\chi \leq 255$). We experientially found such parameter settings typically perform quite well.

4. Experimental Results

We test the TofCut algorithm on the ZCam as shown in Figure 1. The ZCam can produce synchronized 320×240 RGB color video and the same resolution depth maps at 30 frames per second. The color images and depth maps are internally aligned by the software. The proposed algorithm can be implemented fairly fast. Our current implementation can achieve 15 frames per second on a PC with 2.83GHz Intel Core(TM)2 CPU. Note that no machine specific optimization such as multi-threading technique is employed.

In this section we first introduce the data set used for our evaluation. Quantitative evaluation results are provided and further discussed in sections 4.2 and 4.3, respectively. In section 4.4 we test TofCut on a stereo video with depth cue from stereo matching. At last we compare TofCut against a commercial live foreground extraction routine from the 3DV Systems. We also suggest the readers to view the videos in our supplemental materials to verify the effectiveness of TofCut[1].

4.1. Data Sets

We have captured several color videos with additional depth information available using the Zcam. Currently, four sequences have their ground truth segmentation labeled manually, which allows quantitative evaluation to be performed. Sample images of the four video sequences used in this work are shown in Figure 2. The first WL sequence has 200 frames. This sequence contains rich foreground motion and the foreground/background color distributions are similar. Also the depth measurements of the dark hair region suffer from the intensity bias mentioned earlier. The second sequence MS has 400 frames. In this video we demonstrate the case with a moving camera. Note that both the background scene and global illumination are varying (although not significantly) over time during the camera's motion. Moving camera also produces some amount of camera shaking, which is not easy to handle for previous work. Both the third and the fourth sequences contain 300 frames. In MC, the light in the room was switched on and off to simulate global lighting variation and the background contains dynamic moving objects. The last one, CW sequence, is particular challenging for segmentation algorithms using depth cue because there is a person passing by the foreground layer and their relative distance is small according to the depth measurements.

Ground truth binary segmentation results were manually labeled on very fourth frame. Each pixel was labeled as background, foreground or unknown. The unknown band is two to three pixels in width and covers the mixed pixels along layer boundaries.

[1]The supplementary materials (11.2Mb) can be downloaded at: http://vis.uky.edu/~wangl/Research/media/Tofcut.zip

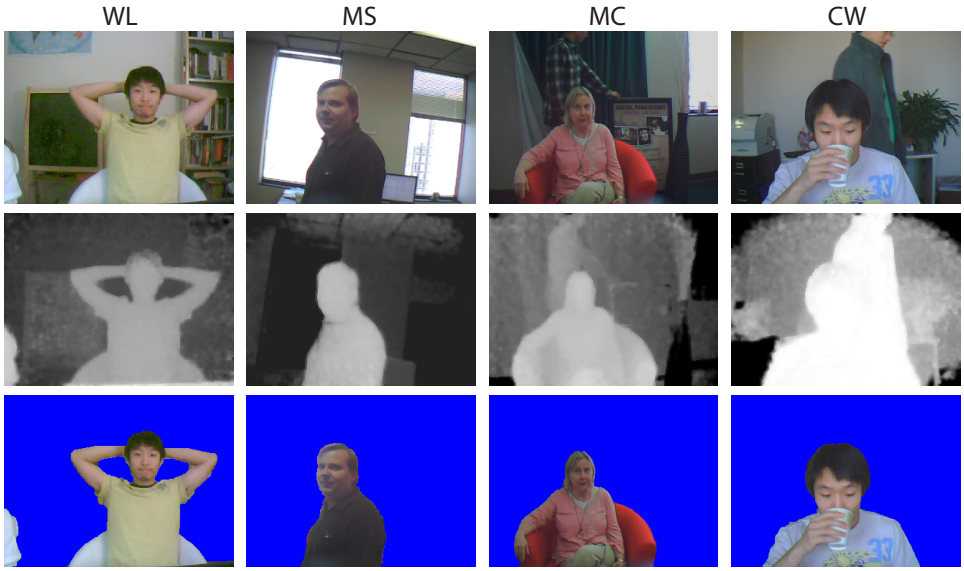

Figure 2. Data set used in this paper. The first two rows show some sample images and their corresponding depth maps returned by the TOF camera. The last row demonstrates our foreground extraction results using TofCut.

4.2. Quantitative Evaluation

Following [12], error rates are measured as a percentage of misclassified pixels w.r.t. the ground truth data. In our experiments, for each test sequence we quantitatively evaluate the segmentation accuracy of four different methods. Besides our TofCut algorithm that uses the adaptive weight fusion scheme, we also assess three different variants based on the MRF formulation in Section 2. First, segmentation algorithms that rely on either color or depth cue only are tested by setting λ^d or λ^c to zero, respectively. In order to validate the effectiveness of the adaptive weight fusion, we also compare TofCut against traditional equal weight method, *i.e*. set both λ^c and λ^d to be 0.5.

In Figure 3 we plot the error curves of the four methods w.r.t. to our test data. Percentage of misclassified pixels within known region is computed on all sequences, every fourth frame. In Figure 4, we further provide the mean segmentation error for each method. Note that the error statistics on both known region and the whole image area (unknown pixels are included when counting the total number of pixels) are shown in this table.

4.3. Result Analysis

The quantitative evaluation confirms that combining the depth and color cues in general achieves better accuracy than either using color or depth along. Furthermore, by determining λ^c and λ^d based on the discriminative capabilities of the color and depth cues, adaptive weight fusion outperforms equal weight fusion on most sequences especially on the challenging sequence CW.

By further investigating Figure 3 we can find that the color information is quite ambiguous for bi-layer segmentation. The depth information from the TOF camera seems to be a reliable cue and demonstrates good performances on the first three sequences. The equal weight fusion improves the depth only segmentation in general, however, it is worth noticing that it performs poorly for the CW sequence. Why fusing multiple cues can sometimes lead to worse results? We now look into those problematic frames to find the answer. As shown in Figures 3 and 5, near the 100th frame the background object starts being incorrectly labeled as foreground in the depth-based method because of the analogous depth distributions of the two layers. But for the fusion approach, since the color term still works reliably at that moment the segmentation remains correct. Near the 120th frame the depth ambiguity causes the background object to be misclassified as foreground for equal weight fusion. Around the 160th frame, although the background object is no longer seen by the camera, the error propagation (color cue learned from earlier false segmentation) causes the drifting artifacts. As a result, the equal weight fusion fails form frame 160 to frame 300. In comparison, by plotting the weighting factors as a function of time in Figure 6, we can see TofCut intelligently adjusts the importance of the two terms over time. When the background object moves closely to the foreground layer the weight of the depth term is decreased accordingly.

4.4. TofCut for Stereo Video

Although TofCut is designed for TOF camera-based foreground extraction, its general formulation makes no specific restriction on the depth acquisition method. In this experiment we replace the TOF camera with a pair of stereo cameras. The scene depth cue is computed using a dynamic

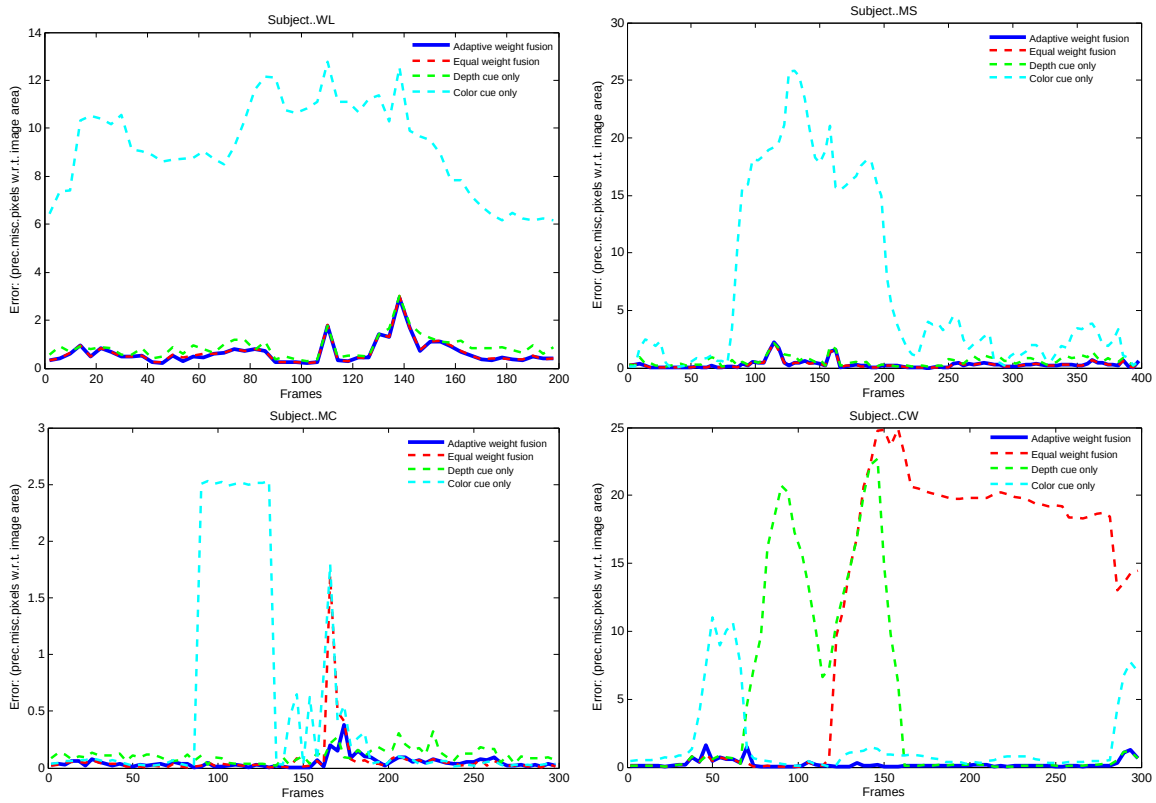

Figure 3. Evaluation on segmentation accuracy using ground truth data. Percentages of misclassified pixels in known region are computed on all four sequences, every fourth frame. Experimental results show that fusing color and depth cues outperforms using color or depth information along in general. Note that the quantitative evaluation also confirms that our adaptive weight fusion method performs more robustly than equal weight fusion on challenging sequences like MC and CW.

Mean error rate(%) / Data (# of frames) / Algorithm	WL (200)		MS (400)		MC (300)		CW (300)	
	Known	All	Known	All	Known	All	Known	All
Adaptive	**0.64**	**1.35**	**0.29**	**0.51**	**0.05**	**0.15**	**0.22**	**0.38**
Equal	0.66	1.37	0.29	0.51	0.07	0.16	11.54	11.68
Depth	0.88	1.68	0.59	0.92	0.11	0.26	4.41	4.62
Color	9.25	9.91	6.77	6.88	0.48	0.59	1.65	1.83

Figure 4. The mean segmentation error w.r.t. the known image region (known) and the whole image space (all) for different methods and test sequences. Again, this table demonstrates TofCut achieves better segmentation accuracy than the other three methods.

programming based real-time stereo algorithm as proposed in [16]. As shown in Figure 7, the camera baseline is small (about 6cm) to alleviate occlusion. We captured a stereo video using our setup and performed TofCut on this sequence. The video contains moving background, sudden illumination change and camera movement. Example disparity map and the extracted foreground layer with background replaced are shown. The full video and results can be found in our supplemental materials. As can be seen, despite depth information is from stereo other than a TOF camera, TofCut performs fairly well on this challenging video.

4.5. Comparison with Commercial Solution

Finally, we close our experimental evaluation with a comparison of TofCut against a commercial live foreground extraction routine released by the 3DV Systems. Since there is no way to perform off-line processing using that soft-

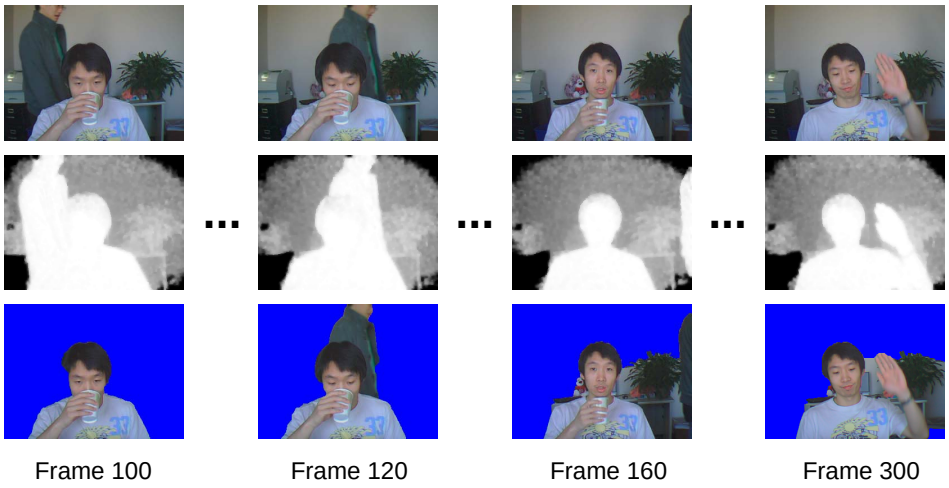

Figure 5. Sample frames demonstrating the error propagation of equal weight fusion. In comparison, adaptive weight fusion can avoid the drifting issue for this scenario. Full segmentation results from TofCut can be found in our supplemental materials.

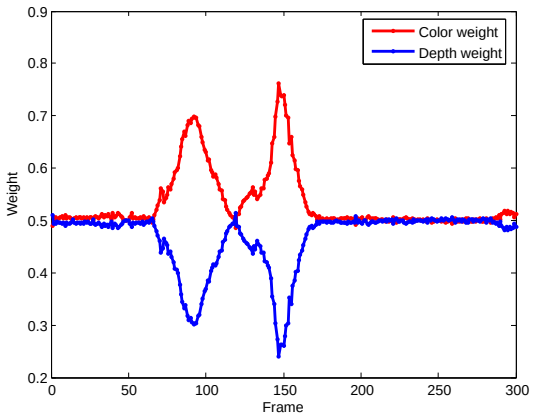

Figure 6. Plots of the color and depth weights as a function of time for the CW sequence. As can be seen the relative importance of the color term is increased when the background object is moving close to the foreground object (This figure is best viewed in color).

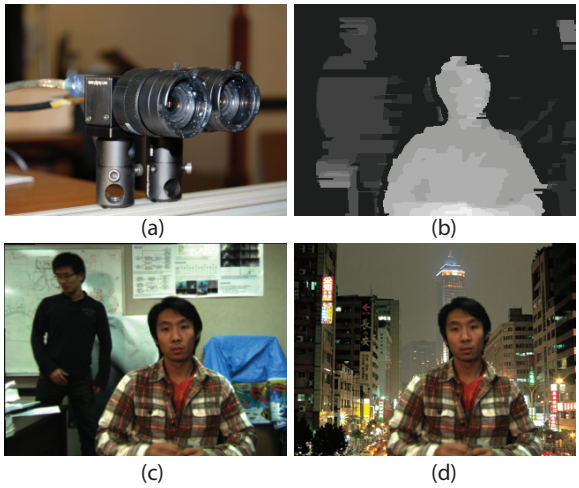

Figure 7. Apply TofCut for stereo video foreground extraction. (a) The stereo setup used in our experiment. (b) The disparity map computed using [16]. (c) The original color image. (d) Extracted foreground layer with background replaced by a new image.

ware we are not able to test their approach using our data set with ground truth. We instead let the commercial software perform segmentation on a scene similar to the one we have setup for CW. Note that parameters of the software are turned so it works at its best at the beginning. In Fig. 8 we shown the screen shot of the segmentation result from this commercial software. As can be seen, when the background is close to the foreground the software incorrectly treats the background object as foreground. In the second row we further provide our result and the corresponding color video frame. TofCut is able to compute correct segmentation by relying more on the color cue. For a side by side comparison of the software and TofCut please refer to our supplemental materials.

5. Conclusion and Future Work

We in this paper address the problem of robust real-time foreground extraction via a TOF camera and propose an effective solution named TofCut. TofCut combines color and depth cues into a unified framework and adjusts their relative importance adaptively to achieve improved robustness. Quantitative evaluation shows that TofCut operates well under a variety of challenging environments with dynamic backgrounds, camera movement and dramatic lighting variations.

There are several directions we would like to explore in the near future. For instance, from the system point of view, drifting is a serious concern for real-time segmenta-

Figure 8. The first row: screen shot of live foreground extraction from the a commercial software released by the 3DV Systems. Note that similar to the equal weight fusion the background is incorrectly estimated as foreground when the distance between the two layers is small. The second row: similar scene and the segmentation result using TofCut. Note the foreground image with blue background is flipped horizontally to be consistent with the software's live output.

tion that requires online learning. Although our adaptive weight method is able to reduce the chance of false labeling by better balancing the color and depth cues, there is no mechanism to retrieve the system from error accumulation and propagation. We plan to further investigate this issue and provide solutions to protect against error propagation. We also notice that one obvious visual problem in current segmentation results is the "flickering" artifacts on layer boundaries. We plan to improve segmentation consistency across different frames by constructing a *space-time* MRF model. In addition, our current algorithm only assigns pixels with binary label. In order to achieve visually pleasant effects, matting is required to provide cleaner boundaries.

Acknowledgements The authors would like to thank Mr. Mao Ye, Dr. Matt Steel and Dr. Melody Carswell for their help in data capture. This work is supported in part by US National Science Foundation grant IIS-0448185, and a grant from US Department of Homeland Security.

References

[1] 3dv systems. http://www.3dvsystems.com. 2

[2] Canesta inc. http://www.canesta.com/. 2

[3] Mesa imaging ag. http://www.mesa-imaging.ch/. 2

[4] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *Proc. of Europ. Conf. on Computer Vision*, 2004. 1, 3

[5] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proc. of Intl. Conf. on Computer Vision*, 2001. 2

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23, 2001. 2

[7] R. Crabb, C. Tracey, A. Puranik, and J. Davis. Real-time foreground segmentation via range and color imaging. In *Proc. of the IEEE Workshop on Time of Flight Camera based Computer Vision*, 2008. 2

[8] A. Criminisi, G. Cross, and V. Kolmogorov. Bilayer segmentation of live video. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. 1, 2, 3

[9] J. Davis and H. Gonzalesz-Banos. Enhanced shape recovery with shuttered pulses of light. In *Proc. of IEEE Workshop on Projector-Camera Systems*, 2003. 3

[10] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1999. 1

[11] M. Harville, G. Gordon, and J. Woodfill. Foreground segmentation using adaptive mixture models in color and depth. In *IEEE Workshop on Detection and Recognition of Events in Video*, 2001. 1

[12] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bilayer segmentation of binocular stereo video. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2005. 1, 3, 5

[13] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *Proc. of ACM SIGGRAPH*, 2004. 1

[14] J. Sun, J. Sun, S.-B. Kang, Z.-B. Xu, X. Tang, and H.-Y. Shum. Flash cut: Foreground extraction with flash and no-flash image pairs. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2007. 1, 2, 3

[15] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Background cut. In *Proc. of Europ. Conf. on Computer Vision*, 2006. 1, 3

[16] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister. High quality real-time stereo using adaptive cost aggregation and dynamic programming. In *Proc. of Intl. Symposium on 3D Data Processing, Visualization and Transmission*, 2006. 6, 7

[17] O. Wang, J. Finger, Q. Yang, J. Davis, and R. Yang. Automatic natural video matting with depth. In *Proc. of the Pacific Graphics*, 2007. 2

[18] P. Yin, A. Criminisi, J. Winn, and I. Essa. Tree-based classifiers for bilayer video segmentation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2007. 1

[19] T. Yu, C. Zhang, M. Cohen, Y. Rui, and Y. Wu. Monocular video foreground/background segmentation by tracking spatial-color gaussian mixture models. In *Proc. of the IEEE Workshop on Motion and Video Computing*, 2007. 1

[20] G. Zhang, J. Jia, W. Xiong, T.-T. Wong, P.-A. Heng, and H. Bao. Moving object extraction with a hand-held camera. In *Proc. of Intl. Conf. on Computer Vision*, 2007. 1

[21] J. Zhu, L. Wang, R. Yang, and J. Davis. Fusion of time-of-flight depth and stereo for high accuracy depth maps. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2008. 3