Real-time Foreground Segmentation via Range and Color Imaging

Ryan Crabb[†] rcrabb@soe.ucsc.edu
Colin Tracey[†] ctracey@ucsc.edu
Akshaya Puranik[††] apuranik@canesta.com
James Davis[†] davis@soe.ucsc.edu

[†]University of California at Santa Cruz
Santa Cruz, CA

[††] Canesta, Inc.
Sunnyvale, CA

Abstract

This paper describes a real-time method for foreground/background segmentation of a color video sequence based primarily on range data of a time-of-flight sensor.

This method uses depth information of a TOF-sensor paired with a high resolution color video camera to efficiently segment foreground from background in a two-step process. First a trimap is produced using only range data: areas are located in each frame that have a high probability of being background or foreground, respectively. Pixels which cannot be definitively classified as foreground or background, typically about 1-2% of the frame, are assigned alpha-matte values using a cross bilateral filtering, applied directly to an estimate of the alpha-matte.

1. Introduction

Background substitution is a regularly used effect in TV and video production, both professionally and with the at-home enthusiast. It's been an indispensable tool for the weatherman (via the blue-screen), and more recently becoming a popular feature in teleconferencing and internet chat.

The basic problem of background substitution is the segmentation of the foreground—those portions of interest from the original scene which we wish to keep—from the background. This problem is commonly worked out with the use of an alpha matte, which dictates the proportion of each displayed pixel that will be foreground and that which will be from the replacement background by assigning a value between 0 and 1 to each pixel. Typically, most pixels are either a 1 (all foreground) or 0 (all background), while the pixels at the borders of the foreground will have a value in between, allowing for a natural looking blending along the edges.

In television and film production, by far the most common technique for alpha matting is by way of a blue-screen, in which the action is filmed in front of a solid color (generally bright blue or green) which is easily identified and replaced. While this technique is both simple and effective, it does require a specially designed studio set and prohibits the blue or green hue from being used in the foreground. Similar techniques can allow for an arbitrarily colored background provided an image of the scene devoid of foreground. More sophisticated methods can even allow for unseen and potentially nonstatic backgrounds, however these 'natural matting' techniques are not performed in real time.

In this paper, we present a method of real-time background substitution based primarily on depth, for use with a time-of-flight depth sensor and paired color video camera, which can be performed against arbitrarily colored and non-static backgrounds. It requires only a depth thresholding plane, defining a distance from the camera plane in which objects are accepted as foreground. Given this dividing plane, along with a depth image and corresponding color image, a trimap is automatically generated, and using a cross bilateral filter [10], a complete alpha-matte is created for the frame.

2. Related Work

The problem of layer segmentation, background subtraction and/or substitution, or alpha- or natural-matting has been addressed by different fields with different methods, and of course for different purposes. Yet the underlying problem is very much related.

2.1. Matting and Background Replacement

As mentioned, the use of blue screens, or chroma-keying, is quite prevalent due its simplicity and was first innovated in the late 1930's—originally designed to work directly on film. The technique has carried on to and been enhanced by digital processing [1], which can operate on a per-pixel basis, allowing for such features as smooth blending and partial transparency.

The idea of replacing each constant color pixel is easily expanded to include any color at any pixel, so that rather than requiring the background to be entirely blue, it can take on any appearance, provided that the empty scene is known in advance. This method is employed in commercial products such as Apple's iChat [2]. However, ambiguities can arise whenever the foreground is similar

978-1-4244-2340-8/08/$25.00 ©2008 IEEE

in color to the expended background pixel, which can be more difficult to avoid than a single blue or green hue, given the arbitrary background requirements. Further, even minor changes in the background can potentially cause artifacts to appear, and a slight bump to the camera can disrupt the entire background model.

The general problem of background subtraction has been of interest to the computer vision community for some time for reasons outside of visual effects; for example, it is quite useful in tracking or to automatically detect unknown objects of interest. These techniques are designed to work in much less constrained circumstances, such as an arbitrarily colored background, or even a slowly changing scene. In [4] the background is modeled as a weighted combination of previous frames and pixels differing by more than a set threshold are labeled foreground. Elgammal et al. use a Gaussian distribution to model pixel values [3] and in [5] Stauffer and Grimson present the popular mixture of Gaussians model. These methods all can be performed easily in real-time and are adaptive, in varying degrees, to small changes in nonstatic backgrounds (such as trees and bushes) and sudden but persistent changes (an object set down or a camera bump). However, updating the background model can often lag and the segmentation will rarely be precise enough for natural blending.

Kolmogorov et al. describe a solution to the real-time background substitution problem using binocular stereo video equipment [6]. Their methods fuse depth-from-stereo information with color/contrast cues to perform segmentation; however, ambiguities in stereo matching do produce occasional artifacts.

2.2. Trimap Generation

Typical methods of natural matting first require the approximate location of segment edges, given by a trimap, which segments an image scene into foreground, background, or indeterminate. In [7], Wang et al. present a method to automatically generate the trimap based on depth cues from a TOF sensor. After upsampling the depth map to color image resolution by way of a cross bilateral filter, a depth threshold is applied, and the binary map is eroded and dilated. The differing pixels are the areas of the trimap to segmented using Bayesian or Poisson matting. We similarly use the depth data to generate a trimap, though in our method we estimate the unknown region before applying the bilateral filter.

3. The Substitution Method

The presented method of background substitution is designed around using a time-of-flight depth sensor paired with a RGB color video camera, such as the CanestaVision [11]. The cameras are registered such that each depth measurement can be projected onto the RGB plane, and typically the spatial resolution of the depth image is significantly less than that of the color image. Given a dividing plane, the scene can be segmented by only depth into a trimap. The indeterminate areas of the trimap are processed with a cross bilateral filter to assign an alpha-value to each color pixel along the edges, producing a natural looking blending.

3.1. Low Resolution Depth Projection and Segmentation

The depth and color cameras are calibrated and registered such that the relation between the fields of view is known. Still, because the camera centers are different, the correspondence of pixels in each sensor is not fixed, but rather dependent upon depth of objects in the scene. For each incoming pair of frames, each depth measurement is projected onto the corresponding color pixel or pixels (the low resolution depth measurements typically cover multiple color pixels). This can require a fixed, but nontrivial amount of computation time. This time can be reduced; a background depth model can be constructed given an initially empty scene, following the method of [3], and projection is limited to foreground depths and those areas immediately surrounding.

It should be noted that there are inherent issues when the depth and color measurements come from different sources. First, there is no guarantee that in the projection process every color pixel will be assigned a depth value, and in practice this is certainly not the case. Further, due to parallax, depth of some color pixels will not be seen by the TOF sensor, and other color pixel locations will be have two depth readings. In the latter case, the lesser depth value is assigned as it is obviously associated with what the color camera observes.

After the project process most pixels have associated depth values, some do not.

3.2. Trimap Generation

To separate foreground from background, a thresholding plane is defined by the user. Pixels are then each assigned probabilities of being foreground based on their depth measurement or lack thereof. The model for assigning probabilities can be arbitrarily complicated, but we use a simple approach of assigning pixels one of three parameterizable values: pixels with depth within threshold are high likelihood, outside of threshold are near zero likelihood, and pixels of unknown are assigned a low probability. The likelihood of unknown depths are weighted towards being background for three reasons: (1) parallax causes missing depth measurements for background only (foreground does not suffer this problem) and (2) further measurements are less accurate (inherent in TOF sensors), and are more likely to be projected

incorrectly into the color plane, and (3) if the background is too far away, there is simply no depth reading.

After each pixel is assigned a foreground probability based on its depth, we estimate the likelihood of it being foreground based on the surrounding pixels:

$$P_{FG}(C_i \mid Z_{j\in N_i}) \approx \frac{1}{K_i}\sum_{j\in N_i} w_{ij} P_{FG}(C_j \mid Z_j, Z_{thresh}),$$

where C_i is a color pixel, Z_i is the associated depth measurement, N_i is the neighborhood of the pixel i, w_{ij} is the weight of pixel j with respect to i, and K_i is a normalizing factor equal to the sum of all weights. With computational efficiency in mind, the neighborhood consists of a square window surrounding the pixel, and each pixel is weighted equally. In this way, we can use the integral image trick [8] [13] to compute the likelihood estimate for an arbitrary sized window in time linear with the size of the image.

Once each pixel is assigned a likelihood, a trimap is created. The trimap classifies each pixel into one of three categories: definitely foreground, definitely background, or uncertain. The term definite is used rather loosely for our purposes, in that the confidence of segmentation must only meet a defined threshold. However once designated as foreground or background in the trimap, the designation will not change, and the segmentation is absolute, i.e. alpha-values are 1 or 0, and that pixel is not blended. Those pixels falling in the undetermined portion of the trimap are assigned alpha-values through a bilateral filtering process. Fig. 2(d) demonstrates an example trimap, in which the undetermined area is about 3% of the frame.

3.3. Cross Bilateral Filter

When assigning each pixel its initial probability of being foreground, its depth is compared to the threshold, and it is assigned an alpha value of 1 or 0, which is recorded into what is called the 'sparse alpha-matte,' as in Fig. 2(c). These values are based on each pixel alone, rather than the neighborhood (as with the trimap). Pixels without depth measurements do not get an alpha-value (hence the name sparse—even though most pixels *do* have values) and are not included in bilateral filtering.

A cross bilateral filter is then applied to the sparse alpha-matte, using the color image as the guide for the range filter. The bilateral filter was introduced to the computer vision field by Tomasi and Manduchi [9] as a method of smoothing grayscale images; the idea is to preserve edges by taking a weighted average of local pixels and where the weight of each pixel in the filter is determined by its distance from the filtered pixel in both the grid lattice and range space. With cross bilateral filtering [10], the range weight comes from a different feature than the one being filtered over. In our case, we

Figure 1: (a) Foreground from original scene. (b) Background substituted image.

filter the alpha values, and base the weights on distance in the grid lattice and the color space.

The refined estimate for the alpha value A_i of each pixel is $A_i = \frac{1}{K_i}\sum_{j\in N,\exists\alpha_j} \alpha_j f(\| i-j \|) g(\| I_i - I_j \|)$, where α_j is the alpha-value from the sparse alpha-matte, f is the spatial filter kernel (in our case, a Gaussian centered at i), g is the range filter kernel (also a Gaussian), I is the color image, N is the neighborhood surrounding I (implemented here as a square window), and K is a normalizing factor, the sum of the product of filter weights defined as $K_J = \sum_{j\in N,\exists\alpha_j} f(\| i-j \|) g(\| I_i - I_j \|)$. The distance between colors is measured as a Euclidean the RGB color space, assuming a 256 value quantization in each channel. The size of the neighborhood window and sigma values for both Gaussian kernels are parameterizable by the user.

4. Experimental Results

The described method has been implemented in C++ and tested on an AMD Athlon 64 X2 Dual Core processor at 2 Ghz, using a CanestaVision camera. The TOF depth sensor has a resolution of 160×120 pixels, and the RGB color camera has a resolution of 640×480 pixels. We were able to run the algorithm as describe at rate of 10 frames per second with the following parameter settings. For the trimap generation, a window size of 13×13, foreground threshold of 95% and background of 85%, assuming measurements below the threshold are surely foreground, those above are surely background, and unmeasured pixels have a 25% chance of being background. As mentioned previously, these settings lead to about 2% of the pixels in each frame to require bilateral filtering. For the bilateral filter, we used a window size of 30 pixels, with a spatial

sigma of 30 pixels, and a color sigma of 12.

5. Conclusions, Limitations and Future Work

This paper outlined a method for real-time background substitution based on a time-of-flight sensor, which places few restrictions on background and foreground and merely requires a user define dividing plane.

Much of the processing time goes into the bilateral filter. This could time could be greatly reduced by using a quantized approximation of the bilateral filter, as described in [12]. Further subsampling of the spatial or color domain, or a smaller window for filtering could increase performance as well.

In this implementation the user is required to set a threshold value for the dividing plane. However, it seems quite reasonable that simple clustering methods such a k-means or mean shift approach could quickly estimate a reasonable dividing plane based on depth alone. Additionally, heuristic knowledge that the foreground is generally at the center of the field of view, or that the object of interest tends to move more than the background could also guide a completely automated segmentation scheme.

References

[1] A. Smith and J. Blinn. Blue screen matting. Proceedings of conference on Computer graphics and interactive techniques, pp. 259–268, 1996.

[2] http://www.apple.com/macosx/features/ichat.html

[3] A. Elgammal, D. Harwood, L. S. Davis, "*Non-parametric Model for Background Subtraction*", 6th European Conference on Computer Vision. Dublin, Ireland, June/July 2000.

[4] J. Heikkila and O. Silven: A real-time system for monitoring of cyclists and pedestrians in: Second IEEE Workshop on Visual Surveillance Fort Collins, Colorado (Jun. 1999) pp. 74–81.

[5] Chris Stauffer and W.E.L Grimson."Adaptive background mixture models for real-time tracking", In Proc. Conf. Comp. Vision Pattern Rec., Fort Colins, CO. June 1999.

[6] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother: "*Bi-layer* segmentation of binocular stereo video." In Proc. Conf. Comp. Vision Pattern Rec., San Diego, CA. Jun 2005.

[7] O. Wang, J. Finger, Q. Yang, J. Davis, R. Yang, "Automatic Natural Video Matting with Depth", Proceedings of the 15th Pacific Conference on Computer Graphics and Applications (Pacific Graphics), 2007.

[8] J. P. Lewis, ``Fast Template Matching", Vision Interface, pp. 120-123, 1995.

[9] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In ICCV, pp/ 839–846, 1998.

[10] J. Kopf, M. Cohen, D. Lischinski, M. Uyttendaele, "Joint Bilateral Upsampling," SIGGRAPH 2007.

[11] Canesta Inc. http://www.canesta.com

[12] S. Paris, F. Durand, A fast approximation of the bilateral filter using a signal processing approach in Proc. of Eur. Conf. on Comp. Vision, 2006.

[13] F. Tang, R. Crabb, and H. Tao, "Representing Images Using Nonorthogonal Harr-Like Bases," IEEE Trans. Pattern Analysis and Machine Intelligence vol. 29, no. 12, pp. 2120-2134, Dec. 2007

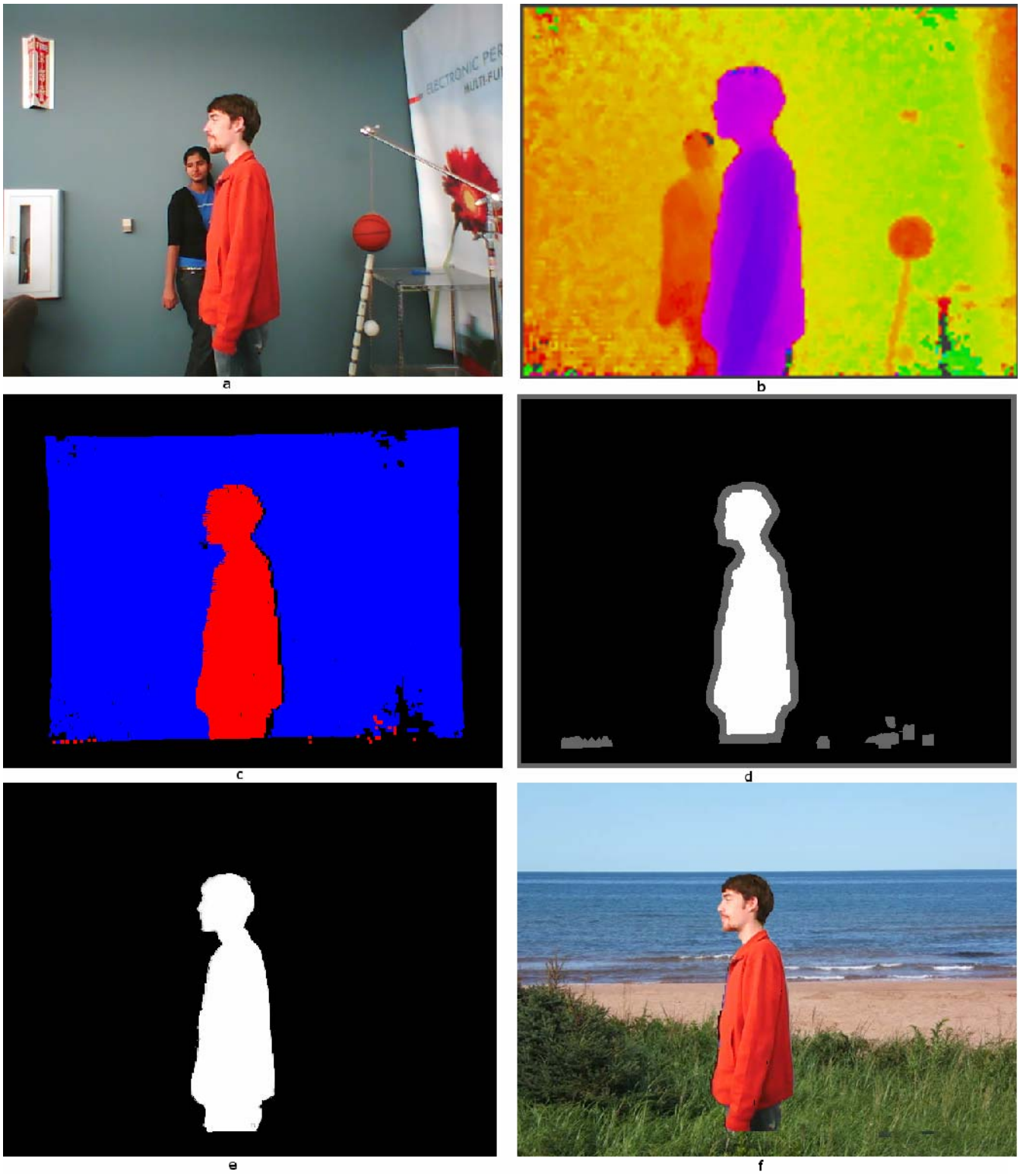

Figure 2: (a) The original scene from the RGB color frame. (b) The depth image from range sensor, at 16 times normal resolution. (c) Sparse alpha-matte, where background is represented by blue, foreground is red, and black are pixels with no depth value. (d) Trimap, grey is undetermined area. (e) Final alpha matting. (f) Foreground overlayed on beach background using alpha matte.