Video Background Replacement without A Blue Screen

Richard J. Qian and M. Ibrahim Sezan

Sharp Laboratories of America
5750 NW Pacific Rim Blvd.
Camas, WA 98607
Email: (qian, sezan)@sharplabs.com

Abstract

An algorithm is presented for replacing video background without a physical blue screen. The method requires pre-recording a background image of the scene without the presence of any foreground objects. Based on the color difference between the pixels in an input frame and their corresponding pixels in the background image, the algorithm computes a probability map which contains the likelihood for each pixel to be classified into the foreground or background. The probability map is further refined using anisotropic diffusion which reduces classification errors without introducing significant artifacts. Finally based on the refined probability map, a new video is synthesized by feathering the foreground pixels of the input frames onto a new background video or image. The algorithm requires moderate computation resources and has been implemented to run in real time. Experimental results are given in the paper.

1. Introduction

Most existing video background replacement methods use the so-called chroma-key technique [1] and they require a physical blue screen as background when recording the original video. Such a requirement is cumbersome in general and becomes practically prohibitive in the cases where a professional studio is not an option. For most consumer applications, for example, it is not practical to require a user to set up a physical screen every time when he or she wants to create a video sequence with a synthesized background.

What is desired, therefore, is a method which can replace video background without a physical blue screen. The development of such a method was reported by a few recent papers, e.g., [3] and [5]. Unfortunately the proposed method in [3] requires at least two cameras and camera calibration is also likely to be needed due to the nature of the method. The proposed method in [5] assumes that the color and texture characteristics of the foreground objects can be learned before the background replacement process. Moreover, since the method employs a multi-blob model to represent the foreground objects and the model is not precise at describing boundaries, it is unable to generate accurate segmentation results along the boundaries between the foreground objects and background. Our objective is to develop a video background replacement algorithm that (1) does not require a blue screen; (2) needs only one camera; (3) requires no camera calibration or prior knowledge about foreground objects; (4) is accurate in terms of distinguishing foreground pixels from background pixels; and (5) is computationally efficient and therefore can perform in real-time.

In Section 2, we first present an overview of our video background replacement algorithm. Then we describe each individual step in the algorithm. Emphasis will be given to the procedure which refines the foreground/background classification result using anisotropic diffusion. In Section 3, we describe our experiments and present examples of our results. We also describe the computational cost and memory requirement of our implementation. Finally in Section 4, we discuss a few aspects of our algorithm and point out some future work.

2. Color and Diffusion Based Background Replacement

Our algorithm locates and replaces the background in an input video based on the color difference between the pixels in each frame of an input video and the pixels in a pre-recorded background image. First, the algorithm acquires a background image of the scene without the presence of any foreground objects. Then the algorithm (1) takes an incoming frame from the input video as the current input image; (2) classifies the pixels in the input image into foreground pixels and background pixels; (3) refine the foreground/background classification result using anisotropic diffusion; (4) overlays the foreground pixels of the input image on a new background image by feathering; and (5) outputs the resulting image with the new background. Figure 1 shows a flow diagram of the proposed background replacement algorithm. Note that the new background can be selected as a uniformly blue

0-7803-5467-2/99/ $ 10.00 © 1999 IEEE

background. In that case, the algorithm generates a pseudo blue screen video sequence. Also note that the proposed algorithm may be directly used in replacing the background of a single still picture.

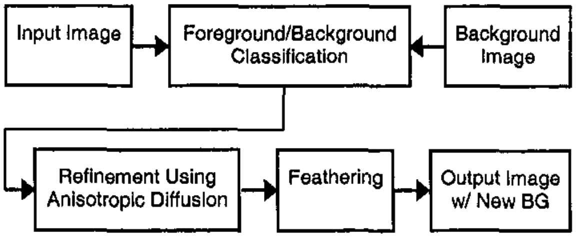


Figure 1. Diagram of the proposed video background replacement algorithm.

2.1. Foreground / background classification by color difference

Uncertainty may exist when determining whether in an input image a given pixel belongs to the foreground or background. To reflect this uncertainty, a probability function can be constructed to measure the likelihood for a pixel in the input image to belong to the foreground or background. In this paper, the color difference between a pixel in the input image and its corresponding pixel in the pre-recorded background image is used as the basis for defining the probability function. Let **p** denote a pixel in the input image, r and g denote its chromatic color components, i.e., $r = R / (R + G + B)$, $g = G / (R + G + B)$, and I denote its intensity. Let r', g' and I' denote the respective counter parts of the corresponding pixel of **p** in the pre-recorded background image. The probability function P is defined as follows:

$$P(\mathbf{p}_{x,y} \in FG) = \begin{cases} \Phi[\alpha(\mathbf{p}_{x,y})] & \text{if } I_{x,y} > \eta \\ \Phi[\beta(\mathbf{p}_{x,y})] & \text{else} \end{cases}$$

and

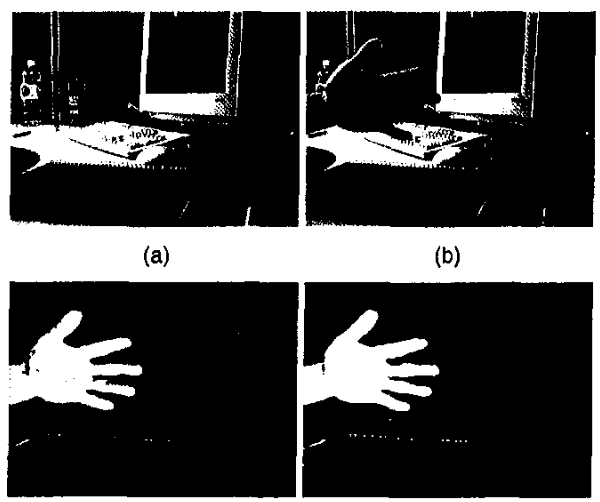
$$\Phi(u) = \min[\max[0.5 + sign(u) \cdot u^2,\ 0],\ 1]$$

$$\alpha(\mathbf{p}_{x,y}) = a \cdot \sqrt{(r_{x,y} - r'_{x,y})^2 + (g_{x,y} - g'_{x,y})^2} + b \cdot | I_{x,y} - I'_{x,y} | + d$$

$$\beta(\mathbf{p}_{x,y}) = f \cdot | I_{x,y} - I'_{x,y} | + h$$

where a, b, d, f, h, η are constants. The above formula uses the chromatic color components and intensity, instead of the raw RGB values. The purpose is to separate the chroma and intensity information, and therefore be able to compare them individually. The chroma information is most effective and reliable when the intensity is high, in which case more weight should be added to the chroma terms. When the intensity is low, the chroma information is usually noisy and unreliable, in which case the comparison should be done without the chroma terms.

Using the probability function, we can measure the likelihood for each pixel in the input image to be classified into the foreground or background and generate a probability map. In principle, the probability map can be binarized based on a prescribed threshold and the binary result will therefore segment the foreground objects from the background. In practice, the thresholding process will produce false classifications for both foreground pixels and background pixels due to the ambiguity existing in certain regions between the foreground objects and the background. Figure 2 presents an example in which Figure 2(a) shows the recorded background image, Figure 2 (b) shows an input image with the original background, Figure 2 (c) shows the computed probability map in which a brighter/darker pixel indicates a higher likelihood for it to belong to the foreground/background, and Figure 2(d) shows the binarized segmentation map. Note the existence of classification errors within both the foreground and background.



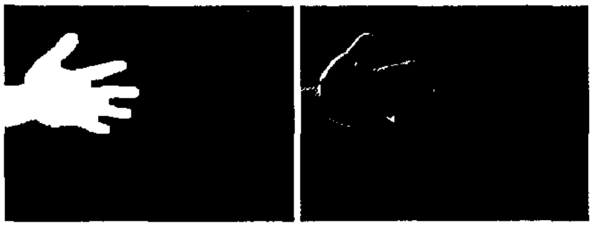
(a) (b)

(c) (d)

Figure 2. Foreground/background classification by color difference. (a) Recorded background image. (b) Input image. (c) Classification probability map. (d) Binarized classification map.

It is often necessary and also possible to refine the classification result by utilizing the context information in the input image. Morphological filtering [2], such as morphological opening and closing, is commonly used to eliminate isolated mis-classified pixels. However, morphological filtering may introduce artifacts.

Morphological closing, for example, may mistakenly fill actual holes or concave parts in a foreground object. Figure 3 presents such an example based on the same input from Figure 2. Figure 3(a) shows the classification map refined by morphological opening and closing and Figure 3(b) shows the input image with background replaced. Note that the morphological filtering reduced false classified pixels but also partially filled the gaps between the fingers by mistake.



(a) (b)

Figure 3. Classification refinement using morphological filtering. (a) Classification map refined by morphological filtering. (b) Background replacement result.

2.2. Classification refinement by anisotropic diffusion

We propose to use anisotropic diffusion [4] to refine the classification result. The anisotropic diffusion process is applied to the probability map and the gradient of intensity information in the input image is used to establish the existence of boundaries. The motivation of using anisotropic diffusion is to encourage smoothing within object boundaries and discourage smoothing across the boundaries.

Consider the anisotropic diffusion equation:

$$P_t = div[c(x,y,t)\nabla P] = c(x,y,t)\Delta P + \nabla c \cdot \nabla P$$

where $c(x, y, t)$ is the conduction coefficient, *div* denotes the divergence operator, and ∇ and Δ respectively denote the gradient and Laplacian operators, with respect to the space variables. The continuos diffusion equation may be discretized on a square lattice. Using a 4-nearest-neighbors discretization of the Laplacian operator, the equation becomes:

$$P_{x,y}^{t+1} = \lambda[c_N \cdot \nabla_N P + c_S \cdot \nabla_S P + c_E \cdot \nabla_E P + c_W \cdot \nabla_W P]_{x,y}^t + P_{x,y}^t$$

and

$$\nabla_N P_{x,y} = P_{x,y-1} - P_{x,y}, \quad \nabla_S P_{x,y} = P_{x,y+1} - P_{x,y},$$

$$\nabla_E P_{x,y} = P_{x+1,y} - P_{x,y}, \quad \nabla_W P_{x,y} = P_{x-1,y} - P_{x,y}$$

where $0 \le \lambda \le 1/4$ for numeric stability reason, *N*, *S*, *E*, *W* denote North, South, East and West, respectively. The conduction coefficients c_N, c_S, c_E, c_W may be computed as follows:

$$c_{N_{x,y}} = g(|\nabla_N I_{x,y}|), \qquad c_{S_{x,y}} = g(|\nabla_S I_{x,y}|),$$

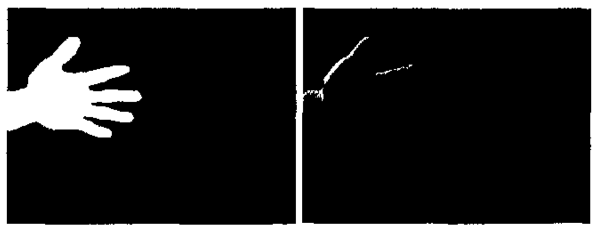
$$c_{E_{x,y}} = g(|\nabla_E I_{x,y}|), \qquad c_{W_{x,y}} = g(|\nabla_W I_{x,y}|)$$

and

$$g(|\nabla I|) = \frac{1}{1 + (|\nabla I|/K)^2}$$

where K is a constant and the subscripts are dropped for simplicity.

Using the same example in Figure 2, Figure 4(a) shows the classification map refined by anisotropic diffusion and Figure 4(b) shows the background replacement result based on the refined classification map. Note that the anisotropic diffusion reduced false classified pixels and did not fill the gaps between the fingers.



(a) (b)

Figure 4. Classification refinement using anisotropic diffusion. (a) Classification map refined by anisotropic diffusion. (b) Background replacement result.

2.3. Feathering

The refined probability map may be used as a guide to overlay the foreground pixels on a new background. To further reduce the artifacts due to misclassification between foreground and background especially around object boundaries, a feathering process is employed. At a given pixel location, the feathering process performs a weighted average over the pixel value of the input image and the pixel value of the new background image and the weights are determined based on the probability value from the probability map. At a given location (x, y) in the output image, the feathering formula is defined as follows:

$$R_{x,y}^{out} = P(\mathbf{p}_{x,y} \in FG) \cdot R_{x,y}^{in} + [1 - P(\mathbf{p}_{x,y} \in FG)] \cdot R_{x,y}^{newBG}$$

$$G_{x,y}^{out} = P(\mathbf{p}_{x,y} \in FG) \cdot G_{x,y}^{in} + [1 - P(\mathbf{p}_{x,y} \in FG)] \cdot G_{x,y}^{newBG}$$

$$B_{x,y}^{out} = P(\mathbf{p}_{x,y} \in FG) \cdot B_{x,y}^{in} + [1 - P(\mathbf{p}_{x,y} \in FG)] \cdot B_{x,y}^{newBG}.$$

3. Experimental Results

We have implemented the proposed video background replacement algorithm on a Sun workstation with a video camera and a video capture card. The algorithm generates visually satisfactory results when the foreground objects and the background do not share a large number of pixels which are similar in color and intensity. Figure 5 shows one good example of the background replacement results generated by our algorithm automatically. The algorithm failed when the foreground objects contain regions which coincide in color and intensity with the background. In the experiments, we also found that shadows and highlights often caused imperfect replacement results. Shadows and highlights hide the actual color information of the underneath pixels and make them less distinctive from grey areas commonly found in real-world surroundings. Figure 6 shows an example where shadow below the magazine caused replacement error in the result.

(a) (b)

Figure 5. Background replacement result. (a) Input image. (b) Output image with new background.

(a) (b)

Figure 6. Another background replacement result. (a) Input image. (b) Output image with new background.

The proposed video background replacement algorithm requires a moderate amount of computation and memory. Its computational cost is (556,800 additions + 288,000 multiplications) and (2,227,200 additions + 115,200 multiplications) per frame for frame resolutions of 160×120 pixels and 320×240 pixels, respectively. Its essential memory consumption is under 135K and 540K bytes for frame resolutions of 160×120 pixels and 320×240 pixels, respectively. The actual frame rate of our system on a 168MHz Sun workstation is 20 and 5 frames per second for frame resolutions of 160×120 pixels and 320×240 pixels, respectively. The above frame rates include the screen display time for both input and output video.

4. Discussion

The paper presents an algorithm which does not require a physical blue screen to perform automatic video background replacement. The algorithm classifies the pixels in an input image into foreground and background based on the color difference between the input image and a pre-recorded background image. The classification result is obtained by computing a probability function and the result is refined using anisotropic diffusion. The advantage of using anisotropic diffusion is to avoid generating artifacts such as filling actual holes in the foreground objects. There is no restriction on the foreground objects in terms of how many of them exist, whether they move or not, etc.. Due to the use of the chromatic color components which are normalized against the absolute color values, the algorithm is relatively robust to image brightness changes due to camera automatic gain control. When foreground objects share regions of similar color and intensity with background, the algorithm may produce errors in the background replacement result.

Future work includes (1) incorporating a statistical model to utilize temporal context information for refining the foreground/background classification result; and (2) removing the current restriction of requiring a stationary camera.

References

[1] D. J. Chaplin, "Chroma key method and apparatus." *U.S. Patent* 5,249,039, 1995.

[2] R. M. Haralick, S. R. Sternberg and X. Zhuang, "Image analysis using mathematical morphology." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532-550, 1987.

[3] Y. Ivanov, A. Bobick and J. Liu, "Fast lighting independent background subtraction." *M.I.T. Media Laboratory Perceptual Computing Section Technical Report*, no. 437, 1997.

[4] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, 1990.

[5] C. R. Wren, A. Azarbayeyani, T. Darrell and A. P. Pentland, "Pfinder: real-time tracking of the human body." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, 1997.