Background and Lighting Changes in Interview Footage

Robin Gaestel
UC Berkeley

Moeka Takagi
UC Berkeley

Abstract

In interview footage, it is often necessary to make editing cuts to create the desired result. However, cuts are not always natural, causing jumps in the flow of the video. Optical flow can mitigate some of these effects by creating a smooth transition over small changes such as facial expression. However, this method tends to fail in the presence of non-static backgrounds.

We present a solution to create seamless transitions in video cuts for all components in the shot. Our method proposes to operate on foreground and background elements separately, thus avoiding artefacts.

Keywords: Optical Flow, Video Matting, Video Textures

1 Introduction

Traditionally, Optical Flow [Horn and Schunck 1981] has been used to smooth over transitions in facial expression across cuts in interview footage. However, this is not a successful method when the background is not static. Optical flow tends to create noticable artefacts in the presence of background motion or lighting changes, due to sudden major changes which appear in the scene. Since there are no correspondance between the pixels in two background objects, this causes a noticable warping effect.

Figure 1: *In this example, optical flow fails due to the sudden appearance of the vehicle across the cut.*

We propose a method wherein foreground and background objects are operated on separately in order to create smooth transitions for the whole scene. Optical flow is used to good effect on the foreground to transition across subtle facial expression changes. This smooth transition is then composited with a Video Texture [Schödl et al. 2000] of the background. In order to obtain the foreground and background, we segment them using a form of video matting.

2 Related Work

There have methods proposed to address to combine foreground objects with novel backgrounds. However, these methods tend to depend on large amount of user input. For example, Interactive Video Cutout [Wang et al. 2005] presents the user with a temporal volume and provides tools for slicing through the volume to extract relevant objects to good effect. In contrast, we attempt to minimize the amount of user interaction by only requiring a rough trimap and a short clean background clip.

3 Methods

3.1 Video Matting

To separate the foreground objects from the background, we use a form of video matting to segment the scene.

The user is required to provide a rough initial trimap for the first frame to differentiate between the interview subject and the surrounding scene, but each successive trimap is determined automatically.

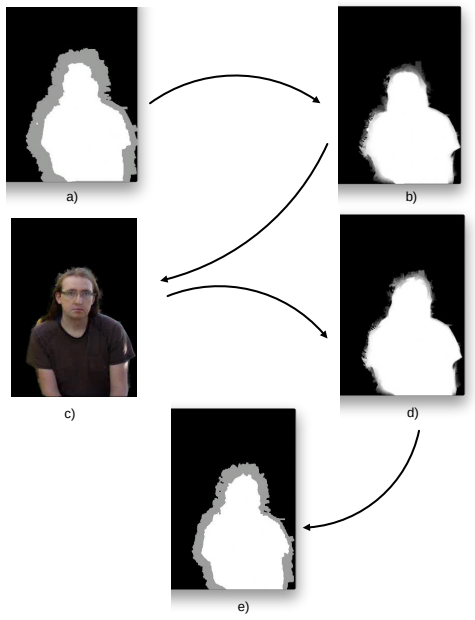

Figure 2: *Constructing a new trimap from the previous frame*

For each frame we perform the following process:

a) Given rough trimap for the frame,

b) We compute an alpha matte by using Learning Based Matting [Zheng and Kambhamettu 2009]

c) Using the computed alpha matte, we are then able to extract the foreground object.

d) We make the observation that between any neighboring frames in a video stream, the scene is usually very similar. Thus, the alpha matte computed for the previous frame will *almost* fit the next frame. Using this observation, we warp the computed alpha matte to conform to the next frame by computing the optical flow between the two successive frames. This allows the capture of the subject's motion between one frame to the next. Using this information, we can roughly warp the previous alpha matte to be a close fit to the next one.

e) While the warped alpha matte is usually sufficient for use in the next frame, over time propagation errors will accumulate from each warp. In order to mitigate this problem, we instead generate a new trimap from the warped alpha matte by extracting fully opaque areas as "certain" and partially transparent areas as "uncertain". This new trimap is then used to calculate an exact alpha matte for the next frame.

We found that the order of operations is important. Were we to simply warp the previous trimap, we begin to get artefacts as the trimap desyncs with the actual video. Computing a new trimap from the approximate alpha matte proved to me more robust against such errors.

Once the foreground and background have been separated, we then operate on them independently due to their differing interpolation requirements.

3.2 Foreground

To hide the cuts in the interview, transitions are computed over the foreground to mask slight changes in the interview subject's expression and posture.

We compute the optical flow over the segmented foreground frames before and after the cut. We then create the transition by interpolating between the frames. For each transition frame, a partial warp of the before frame towards after frame is calculated from the flow field.

We found that purely warping the initial frame to create transition frames could produce noticeable artifacts when there is a significant change in the subject's expression- such as the appearance of teeth when the subject smiles. This is caused by there being no corresponding pixels in the initial frame to be warped into the next. In order to address this problem, we compute the optical flow in *both* directions over the cut. We can then create more convincing intermediate frames by interpolating in both directions and averaging the results together.

3.3 Background

Instead of trying to address the problem of major background motion directly, we instead replace the background completely with a video texture. We do this by first taking a short video clip no longer than a minute of just the background to use as the "training video." This video should be taken in the same camera position as that of the interview footage with the interviewee, the foreground. Then, to create a clean background into the final video, we create a Video Texture with desired length corresponding to that of the resulting edited interview footage.

To implement Video Textures on the footage, we extract the video texture, preserve dynamics, rule out dead ends by anticipating the future, and sequence the video texture. To carry out the last step, there are two possible algorithms: random play and video loops. In our case, we use the Random Play algorithm to sequence the video texture because interview footage is not meant to be looped. Instead, all we need is to create *background* footage that never repeats exactly and thus looks natural.

3.3.1 Overview of our use of Video Textures:

1. Find the distances between all frames and store in a matrix.

$$D_{ij} = \|I_i - I_j\|_2$$

2. Convolve the distance matrix with a diagonal kernel to obtain the filtered distance matrix. We used a 4-tap filter with binomial weights.

$$D'_{ij} = \sum_{k=-m}^{m-1} w_k D_{i+k,j+k}$$

3. Use equation below to calculate the matrix containing the anticipated future cost of a transition from one frame to the next.

$$m_j = \min_k D''_{jk} D''_{ij} = (D'_{ij})^p + \alpha m_j$$

4. Use result from 3 in equation below to calculate the probability matrix.

$$P''_{ij} \propto \exp\left(-D''_{i+1,j}/\sigma\right)$$

5. Use Random Play to sequence video texture. Begin at any frame with a non-zero probability transition. After displaying this frame i, the next frame, j, is selected according to the probability matrix calculated in 4.

3.4 Composition

After the foreground transition frames are computed, they are composited with the new background in order to produce the final output.

$$C_i = \alpha F_i + (1 - \alpha) B_i$$

Here the original frame is multiplied by the alpha matte to separate the foreground. The background is multiplied by the inverse of the alpha matte to cut out a hole for the foreground. They are then added together to obtain he final frame.

4 Results

Since the foreground and background are operated on independently, there are no additional artifacts created by composition. However, good results require that matting and video textures be implemented well. In our example video, matting was extremely challenging due to large background movement occurring right behind the desired foreground. This lead to a breakdown of the matting process, and our result was significantly degraded.

We postulate that we would have greater success with a more subdued background.

5 Discussion

5.1 Foreground

We found that while our foreground segmentation worked well for relatively calm background scenes, it is not robust for backgrounds with high velocity motion. When a high velocity object passed behind the interview subject, the optical flow warping of the alpha matte can mistakenly pull away from the actual bounds of the subject due to the high magnitude of the warp.

Figure 3: *Violent motion in the background, such as this passing car, can result in the alpha matte getting overly warped such that it no longer fits the foreground subject.*

5.2 Background

While the Video Textures algorithm is sufficient in creating natural looking footage, choosing good parameters is often a mystery. It is often time-consuming to find the best match for parameters because they must be changed for different input videos in order to obtain desirable results.

Further, random play of video textures tends to produce unsatisfactory results due to the short look-ahead distance. Using the video loops algorithm would produce better clips because they are precomputed to the end.

5.3 Comparison

When attempting to use Adobe After Effects' Roto Brush to compare matting results, we found that a large amount of user input was necessary to get an outline comparable to our own method, and even then, the outline was very noisy. Further, while the matting outline was not corrupted by the background motion of the car, it had a tendency to desynchronize from the actual subject's movement after only a few frames.

We conclude that we unwittingly used an exceptionally difficult example of footage in our tests due to both extreme background motion and the similarity of foreground and background colors.

6 Future Work

One of the major weaknesses of our process is the requirement that the user provide an initial trimap. GrabCut [Rother et al. 2004] could be used acquire foreground objects with minimal user input. Even more attractive is the automatic acquisition of foreground objects by some statistical means.

Figure 4: *Roto brush propagation error*

Also, instead of requiring that the user provide a clean clip of the background to produce video textures from, a future implementation could first separate the foreground from the background, and then use video completion, such as Space-Time Video Completion [Wexler et al. 2004], to cover the hole left by the subject. The video texture could then be computed from this clean backdrop.

Acknowledgements

Thanks goes to C. Liu for Matlab Implementation of Optical Flow [Liu 2009], and to Yuanjie Zheng for Matlab Implementation of Learning Based Matting [Zheng and Kambhamettu 2009].

Thanks also to Tim Althoff for his assistance with video textures, and to Floraine Berthouzoz for guidance throughout the project.

References

HORN, B. K. P., AND SCHUNCK, B. G. 1981. Determining optical flow. *Artif. Intell. 17*, 1-3, 185–203.

LIU, C. 2009. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "grabcut": interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, SIGGRAPH '04, 309–314.

SCHÖDL, A., SZELISKI, R., SALESIN, D. H., AND ESSA, I. 2000. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '00, 489–498.

WANG, J., BHAT, P., COLBURN, R. A., AGRAWALA, M., AND COHEN, M. F. 2005. Interactive video cutout. In *ACM SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, SIGGRAPH '05, 585–594.

WEXLER, Y., SHECHTMAN, E., AND IRANI, M. 2004. Space-time video completion. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, I–120 – I–127 Vol.1.

ZHENG, Y., AND KAMBHAMETTU, C. 2009. Learning based digital matting. In *Computer Vision, 2009 IEEE 12th International Conference on*, 889 –896.