Materialized Tracer

Austin Schaumberg Jimmy Yuan Amr Hassaballah

December 19, 2016

HW-5: Term Project
CS 534: Computational Photography
Professor Chuck Dyer - Fall 2016

Web Site: Google Drive Repository

Abstract

This paper discusses the approach and implementation to a new video effect coined the “Materialized Tracer” effect. Within this paper we will describe our method in detail, including pseudocode and some of the general parameters used to produce a given tracer’s frequency and density. Commonly speaking, the technique is a fusion of foreground segmentation combined with a clever approach to image overlaying in order to produce the effect. Moreover, we will provide an analysis on our algorithm’s efficiency in addition to a variety of our project’s initial results.

Figure 0.1

Figure 0.2

Figures 0.1 and 0.2 are frames extracted from our Test 8, which depict a subject in motion with the matierialized tracer effect applied.

Introduction

Within the last quarter of the century, the general public’s access to high-resolution cameras and recording devices has increased dramatically. Given humanity’s newfound freedom to capture moments in our personal lives, spawns the desire to be creative. Thanks to steadily increasing improvements in technology, moments we capture can now be modified with the assistance of computational photography. Thereby, enabling us to produce and layer various effects from our devices onto those moments; transforming them into a modified perception of reality, which we then share with others through various mediums of social media or theater.

Our “Materialized Tracer” refers to the video effect where an object in motion (provided a static background), say a person walking down the street, is casually followed by a trail of copies of that same individual walking down the very same street. Depending on the parameters entered, the trail of copies (of the individual performing that same motion) will appear at varying densities and frequencies. The approach for this effect could have been implemented in many ways, however they must all begin with some type of segmentation, in order to separate the subject of interest from its surroundings. For this reason, our team decided to utilize “Foreground Segmentation” also referred to as “Background Subtraction” in combination with video processing and image overlaying methods.

Motivation

The inception of this project was originally inspired by the video effects used by David Dutton, in the music video “Midnight (Bassnectar Remix)” by Joker. The music video illustrates the desired effect we attempted to achieve, beginning at 00:55 (the link can be reached by clicking on Figure 1 or its caption and is set to begin at the noted timestamp). Unfortunately, our results do not produce such an accurate segmentation of the subject of interest due to the autonomous nature of our process, consequently our materialized tracer is not as dense and frequently contains noise in the form of pixels from the background slipped past ViBe's segmentation. One can compare professional results (Figure 1), to our results (Figure 2).

Figure 1 - Professional Video Effect:
JOKER - MIDNIGHT (BASSNECTAR REMIX)
Video directed by David Dutton
Dancing by Atomic Mari [5]

We attempted to reach out to David Dutton, but had little success in communication. As such, we were never able to ask whether the professional effects were developed manually (**), via a programmed algorithm or captured and edited using a camera with built-in segmentation capabilities similar to Microsoft’s Kinect 2. However, our code implements this autonomously given an input video and a set of density & frequency parameters. (The video corresponding to Fiugre 2 can be reached similarly to Figure 1) .

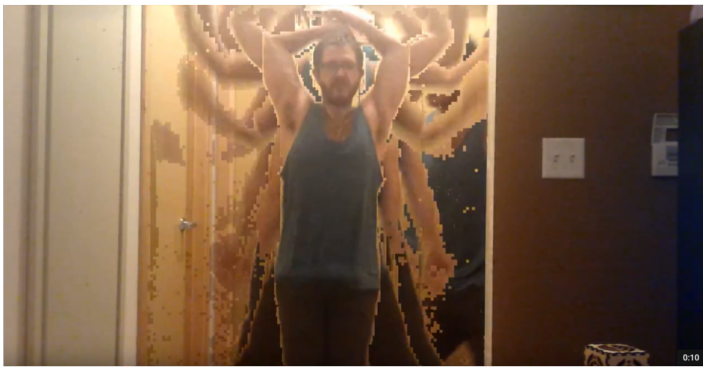


Figure 2 - Materialized Tracer Effect,
Test 5 - Frame 279 (ftp: 15, delay: 40)

** *A manual process of this nature was likely achieved through a process of manually segmenting frame by frame with a tool such as Adobe’s Photoshop or tools similar to Grab-Cut, then recompiling those frames onto one another to produce this effect.*

Problem Statement

To implement the materialized tracer our team needed to have overcome two key components: identify a feasible segmentation approach and video processing the input frame by frame with the segmented images. Both can be accomplished in a variety of ways. Yet, through countless iterations of trial and error, testing dated published approaches; we finally discovered a turnkey foreground segmentation algorithm, ViBe. All that remained to accomplish was to develop an effective algorithm to perform frame-by-frame comparisons, pseudo-transparent frames of the object in motion and compilation of those images back into a video file.

Theory of ViBe's Segmentation: Background Detection & Subtraction

After relentlessly testing numerous dated published source code and algorithms, the algorithm that produced optimal results was an application created by a team of individuals from the University of Liege, Belgium called ViBe [1]. The software solution uses a combination of traditional procedures referred to as Background Subtraction and a unique way to implement another set of procedures, namely dilation & erosion [2,3]. Background subtraction is implemented by some form of motion detection, either in a pixel-by-pixel bases [2] or an approach using four-dimensional vectors, with the fourth dimension being time [4]. Dilation expands the foreground of the image, while erosion expands the background [2]. Thus, each procedure is constantly keeping the other in check. ViBe implements this by constantly upgrading values at each pixel determined by the values at the pixel itself and its neighborhood as time progresses and then using those values to determine if the pixel is part of the background [3]. It differs from the traditional methods by not following the philosophy that the oldest values must be upgraded first [2,3].

Figure 3a

Figure 3b

Figure 3a and 3b [6]:
The image of a road with moving cars (from the baseline/highway directory of the "Change Detection" dataset) and the corresponding binary segmentation map (produced by ViBe). Foreground pixels are drawn in white.

Theory of Foreground Layering

The goal is to take the segmented video and integrate copies of it into the original video. Each segmented video has a transparent background and is delayed by a larger amount of time depending on its position in the sequence. The delays and the amount of copies stacked, or frequency, are parameters that can be controlled.

The transparency is achieved by manipulating the values in each frame's alpha channel at the desired pixels. However, when it comes to layering the segmented foregrounds with the original video, there are two schools of thought. In the traditional method (See Figure 5 below) the layering is started with the most delayed segmented video on the bottom and then layering less and less delayed videos on top until finally concluding with the original video. While our method implements the exact opposite approach (See Figure 4 below), here the bottom can be thought of as being "deeper into the scene being viewed" and the top as being "closer to the observer".

Figure 4: Our Method

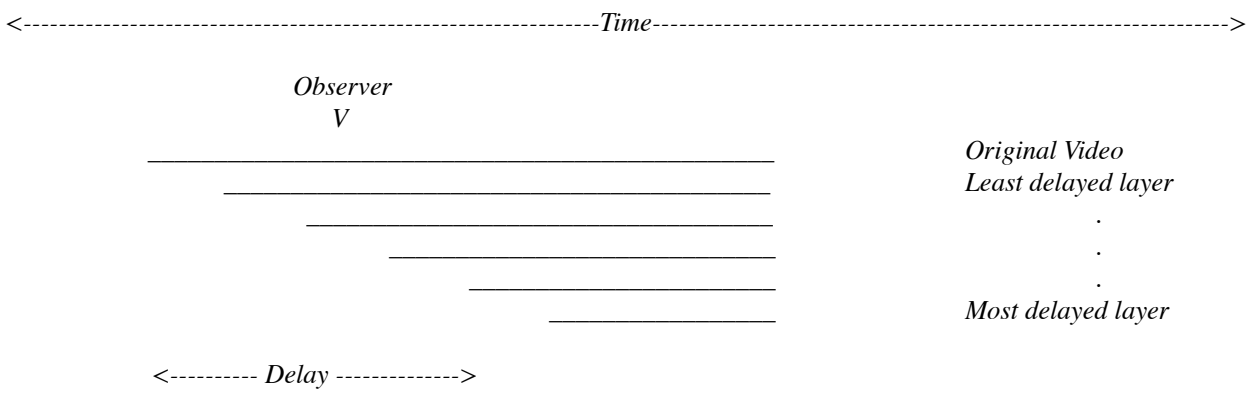


Figure 5: Traditional method

Implementation:

Segmentation

1. Capture input video (*A*) of subject in motion

2. Process initial video (*A*) through Vibe's Background Subtraction Algorithm to produce an output video (*B*) displaying the subject in motion outlined in the form of energy readings.

3. Iterate through inputs *A* and *B* frame-by-frame, For each corresponding $aF_1, \ldots, aF_n \subseteq A$ (and *B* respectively):

Using Input *B*'s set of frames, parse pixel by pixel, For each $p_1, \ldots, p_n \subseteq bF_n$:

Is the pixel black or white?

- If black, then the corresponding pixel in aF_n is set to "transparent" (by providing a negative value to its color channels, no alpha channel needed).
[We call this pseudotransparency]

- If white, then the corresponding pixel in aF_n retains its original value.

4. Resulting set of output frames are called *tFrames* ("tracer frames"): $tF_1, \ldots, tF_n$

Foreground Overlay

For each frame in our original video, we will create a new frame with tracers.

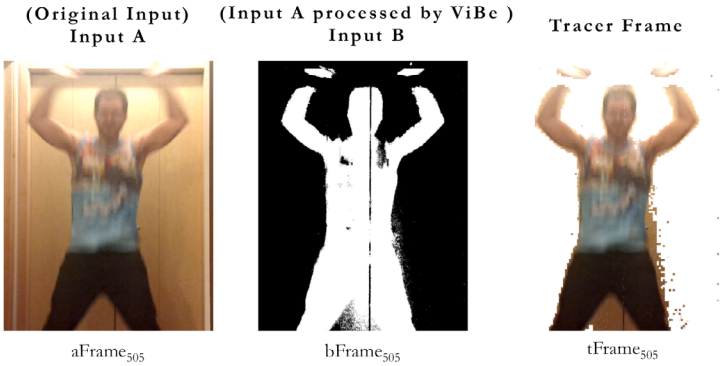
5. Based off of a user specified "frames per tracer" parameter (call it *fpt*), a new tracer will appear every *fpt* frames.

6. Each tracer simply starts as the original video, the difference being they will start with a delay based on the value *fpt*. Start as the image of the latest tracer (call it *IMG*), latest meaning the tracer that is the farthest behind in time relative to the original motion. Now, from latest to earliest, earliest being the original video, layer the tracer images on top of *IMG* so that the latest tracers are on top.*(*)*

7. Take all overlaid images created in steps 5 and 6, write them to a video file. This is our final output with the tracer effect applied.

(*) Oversimplification of step 6: Essentially, we can think of the images as a "stack", with the older tracer images on the bottom.

ViBe's segmentation steps are paraphrased as [step 2] (See Figure 6 below) the remaining steps were developed amongst our team and implemented by our primary developer Jimmy Yuan.



Results & Analysis:

This section will only refer to the theoretical results and implications; for those interested in output videos please consult the folders attached and the Google drive.

Findings

- Shadows cast from the subject of interest onto the 'static' background generates noisy data in the form of various rouge pixels surrounding the subject we wish to fully extract, causing a less accurate tracer in our final product.

- Input video must start with at least one frame portraying the background only for "reasonable" segmentation. The longer ViBe has to process the initial background, the more optimal the segmentation will be.

- Utilizing a tool like Microsoft's kinect would have been an optimal choice for future versions of this project, as it provides a very clean segmentation of the subject.

Input Parameters & Complexity

- Frequency: How many times a segmentation layer video is looped before being terminated.

- Density: The maximum number of active segmentation layers at any given time.

- Delay: How much each segmentation layer copy is delayed from the next layer.

Our Big-O complexity came out to $O(n^4)$. Furthermore, have found that the processing time is extremely sensitive to the frequency and density parameters. However, this is what was intuitively expected.

Summary of Process

- Record video with a static background and a dynamic foreground object(s).

- Segmentation through video processing with ViBe Algorithm

- Set frequency, density & delay parameters

- Implement segmented video transparency in MatLab via the pseudotransparency

- Layer segmented frames into original input video

- Output resulting video with applied materialized tracer effect

Closing Remarks

Special effects such as the one explored in this paper could be seen as pertinent to many various professions. Many music videos, advertisements, blockbuster movies and rendering software require "special effects". The ability to segment and create a materialized tracer in an autonomous fashion could save a given special effects editor time and save a given studio money.

Furthermore, they use manual processes with expensive specialized software to perform these functions, not dissimilar to those discussed in this document. Thus a large industry potential exists for the production and integration of such special effects.

Team Member Contributions

First and foremost, please keep in mind that these sections only highlight the major contributions of each partner. Most of the tasks described in one's section were also contributed to considerably by another and vice versa.

Austin Schaumberg
- Primary research and testing
- Established required starting point
- Wrote the initial abstract and presentation

Jimmy Yuan
- Primary developer of MatLab source code
- Camera operations for capturing inputs
- Supported algorithm research and testing

Amr Hassaballah
- Method and process dictation
- Project scheduling and communication
- Aided in research and implementation

Special Thanks

Special thanks to our instructor Chuck Dyer and his teaching assistant Qisi Wang for all of thier help throughout the semester.

Refrences

[1] http://www.telecom.ulg.ac.be/research/vibe/

[2] Howe, Nicholas R., Alexandra Deschamps, Computer Science, and Smith College. Better Foreground Segmentation Through Graph Cuts. Web.

[3] O. Barnich and M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. In IEEE Transactions on Image Processing, 20(6):1709-1724, June 2011.

[4] First groups presentation (Only other group that implemented video processing & segmentation)

[5] JOKER - MIDNIGHT (BASSNECTAR REMIX)
Video directed by David Dutton
Dancing by Atomic Mari
https://www.youtube.com/watch?v=7zZgy0vhiYU

[6] M. Van Droogenbroeck and O. Barnich. ViBe: A Disruptive Method for Background Subtraction. In T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, Background Modeling and Foreground Detection for Video Surveillance, chapter 7. Chapman and Hall/CRC, June 2014.

Project Links and Code

If for some reason the hyperlinks in this document were not functional here are various links to the following:

All Project Content:
https://drive.google.com/open?id=0B7CQ23Hqiuj-dVp1dVMzYjNvMGs

All Project Test Results:
https://drive.google.com/open?id=0B7CQ23Hqiuj-dlBZbjVFSEVUekU

MatLab Source code:
https://drive.google.com/open?id=0B7CQ23Hqiuj-dXFoeWZidndFY2s

ViBe .zip for .exe and Source Code:
https://drive.google.com/open?id=0B7CQ23Hqiuj-YXhobXdCN0lTNzg