Effect of Various Parameters in Flash Cut Foreground Extraction with Flash/Non-Flash Pairs

Lingxi Zhang
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
lingxiz@andrew.cmu.edu

Abstract

*This paper presents a partial implementation of the previous **flash cut foreground extraction with flash and no-flash image pairs** algorithm [1]. It only used the most essential foreground energy and background energy term to compute the labeling due to time and scope complexity of the original project. It also uses different methods of mean and standard deviation estimation to compute the probability attributes. This paper aims to discuss the effect of various parameters on final foreground extraction results, and as a side result, whether we can implement continuous distinction rather than binary distinction.*

1. Introduction

Image segmentation, which is the process of partitioning a digital image into multiple image segments, is one of the most important problems in the world of computer vision. Nevertheless, generating quality segmentation result from limited sources of images are still challenging. Many methods rely heavily on simply more images or additional information, such as motions, stereo, and infrared light, or the assumption of a static background. In a lot of situation, these factors are either costly to get and compute, or just impossible to obtain in some cases.

Luckily, we already have a well-established method from *Sun et al.* that use no additional factors other than the images themselves and only use two shots of images : the flash one and the non-flash one. Additionally, their method does not even require static setting of the background and the object. Relative small movement of the scene is fine instead of absolutely static scene.

2. Background

Existing foreground extraction framework only works when the foreground in the pictures is very far away from the background so that the flash effect is very different between foreground and background(essentially a binary distinction). However, in some cases the foreground is not so far from the background. As a result, the flash still has some influence over the part with deeper depth instead of zero influence in the previous assumption. As a result, the definition of background and foreground is not as clear as before. The question is, is there anyway that we can decice how much stuff will be in the background and foreground?

There are indeed some practical applications of this in real life. For instance, if Bob stands in front of Lincoln's statue(which is partially affected by the flash), and there are many skyscrapers in the far end. Traditional foreground extraction may only include Bob in the picture. But if we relax our distinction in the standard algorithm to some extent, we can also include Lincoln's statue.

3. Problem Formulation

A foreground/background segmentation can be formulated as a binary labeling problem. The answer of the problem are simply labels of 0(background) or 1(foreground) for all pixels in the images.

Let us denote the flash image I^f, the no-flash image I^n, the labeling for pixel p is $x_p \in \{0, 1\}$. The foreground layer is extracted by minimizing the following energy of an Markov Random Field(MRF):

$$E(X) = \sum_p E_d(x_p) + \alpha \sum_{p,q} E_s(x_p, x_q) \qquad (1)$$

where $E_d(x_p)$ is the data term for each pixel p, and $E_s(x_p, x_q)$ is the smoothness term associated with two adjacent pixels p and q. It is called "smoothness" because we penalize different labeling of adjacent pixels that have small intensity difference in the original image. For the sake of this project, we ignore this term and only focus on the data term as in that case the local optimal answer is also the global optimal answer, saving us from solving complex equations with gradient descent or other complex methods.

The original data term $E_d(x_p)$ models the flash effects on the foreground, the (motion compensated) background, and the color likelihood. It consists of three terms:

$$E_d(x_p) = \gamma_f E_f(x_p) + \gamma_f E_f(x_p) + E_c(x_p) \quad (2)$$

Again for the sake of complexity we emit the color term as well and focus on the first two terms.

3.1. The foreground term E_f

E_f is the foreground flash term, which tends to label the pixel with significant appearance change as foreground.

In the original paper, color histograms are used for the computation. We will use grayscale intensity histograms that simplifies the process.

Basically, the histogram is just made of a number of evenly-distributed bins. Each bin holds the count of pixels in the original image that has the intensity inside the range of the bin. Since in our experiment the image is large enough to provide efficient sources of bin elements, we will just use 256 bins for a 8-bit image, which means each pin will hold pixels that have a single intensity value.

Let us denote the corresponding bin for pixel p is k_p, and h_k is the count of pixels in bin k. Then the flash ratio for the flash and non-flash images is defined as

$$r_p^f = \max\{\frac{h_{k_p}^f - h_{k_p}^n}{h_{k_p}^f}, 0\}, r_p^n = \max\{\frac{h_{k_p}^n - h_{k_p}^f}{h_{k_p}^n}, 0\} \quad (3)$$

In other words, the flash ratio refers to how much positive change in pixel count from the flash classification to the non-flash classification for a single pixel p and vice versa. If the flash ratio is high, that means the pixel is more likely to belong to the foreground since the flash has more effect on the foreground.

We then define the energy term based on the flash ratio with a robust parameter δ as

$$E_f(x_p) = \begin{cases} 0 & x_p = 0 \\ \frac{1}{1-\delta}(\max\{r_p, \delta\} - \delta) & x_p = 1 \end{cases} \quad (4)$$

In our experiment we set the value of δ to 0.2.

3.2. The background term

E_m is the background flash term, which models the motion-compensated flash effect on the background. It tends to label the pixel with good matching and small appearance changes as background. This energy term considers both flash and motion cues.

Suppose we have a dense motion field $m = \{m(p)\}$, such that $p' = m(p)$ is simply the corresponding pixel in the flash image (after considering motion movement) compared to the original no-flash image. Since the image difference I_d is defined as

$$I_d = I_{p'}^f - I_p^n \quad (5)$$

Since the user is expected to capture the flash/no-flash images with distant background in quick succession, the appearance change of the background is expected to be small and uniform. It is thus reasonable to model the flash difference distribution of all background pixels as a **Gaussian distribution** $N(I_d|\mu, \sigma^2)$. Then, we can define the probability of a pixel p belonging to the background as

$$p_b(x) = \exp(-\sigma_b(I_d - \mu)^2) \quad (6)$$

where σ_b is the adjust standard deviation value so that the pixel with flash difference within the $\pm 3\sigma$ interal will be given a higher probability to be background. Finally, we can define the energy term as

$$E_m(x_p) = \begin{cases} 2\max\{p_b(x_p), 0.5\} - 1 & x_p = 1 \\ 0 & x_p = 0 \end{cases} \quad (7)$$

with this definition, $E_m(x_p)$ is normalized to be in the range $[0, 1]$.

3.3. Mean and Standard Deviation Estimation for the Background Probability

We use a different method from the original paper to estimate μ and σ. The original paper suggests a combination of making 1D histogram of flash difference and use a SIFT sparse feature matching to estimate the μ and σ more accurately. I simply calculate the mean from the distribution directly, and leave the standard deviation σ actually a tunnable variable in later experiments. (In the original paper they use the first local min bin after the mean and cutting off values of threshold to estimate the standard deviation).

The default setting for the σ value is 0.10.

3.4. Motion Estimation

We use the dense optical flow computation method from [2] and simply call *cv2.calcOpticalFlowFarneback()* from python opencv library.

Basically we compute a 2D flow vector $flow = \{(dx, dy)\}$ for each pixel such that

$$I_f(x, y) = I_n(x + dx, y + dy) \quad (8)$$

For more details please refer to the paper.

4. Methodlogy

We took pictures around 4-5 pm at my backyard with tripod and Nikon D3500 camera. Notice during the shots there are some non-ignorable movements of both camera and the scene object.

Figure 1. Original flash and no-flash pairs

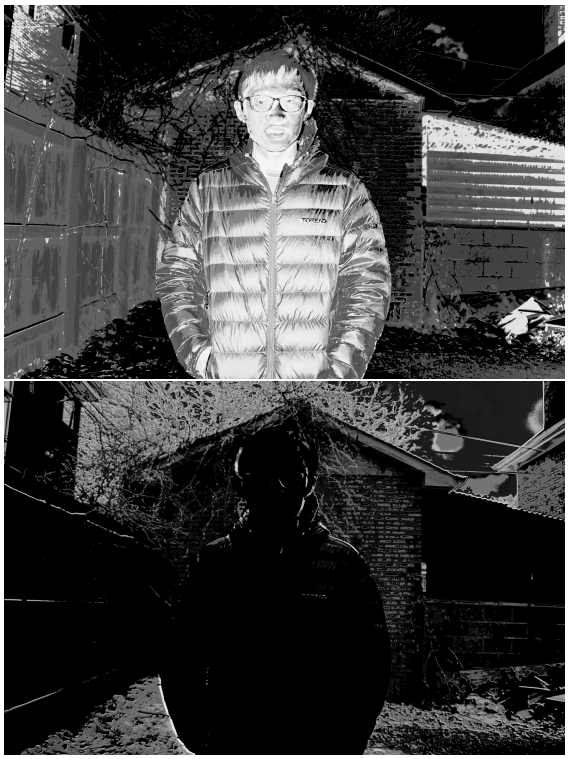

Figure 2. flash ratio visualization for flash and no-flash

5. Results

6. Conclusion

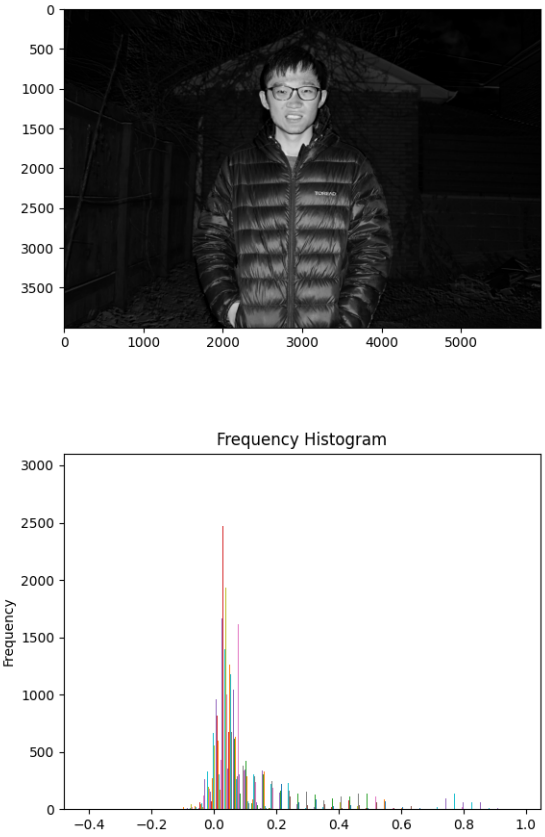

Figure 3. I_d visualization and distribution

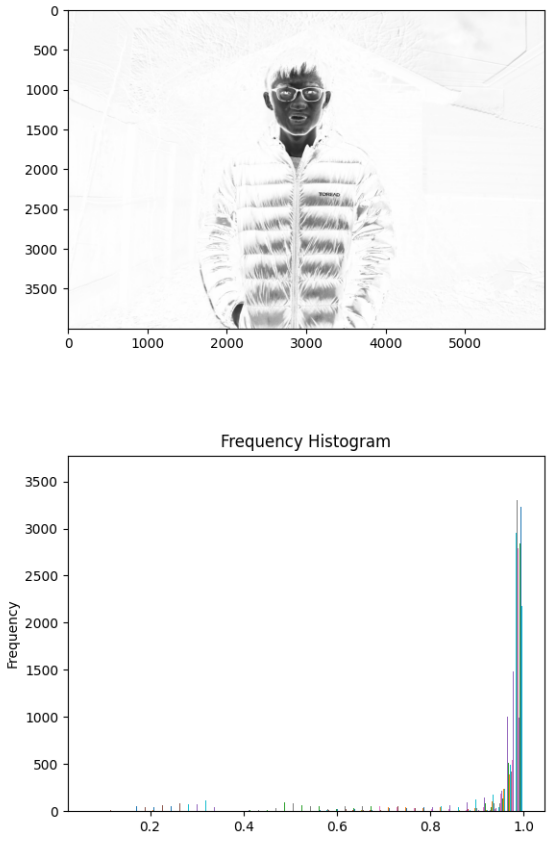

Figure 4. $p_b(x_p)$ visualization and distribution

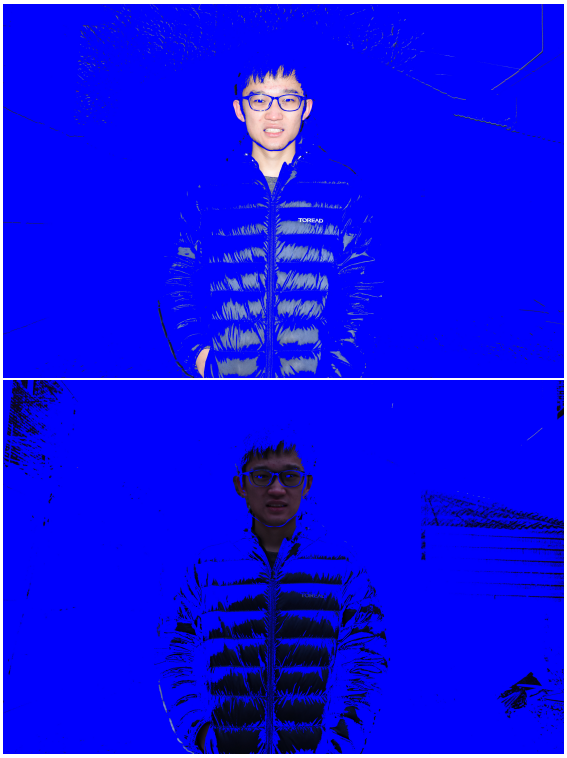

Figure 5. $p_b(x_p)$ visualization and distribution

7. Conclusion

7.1. Acknowledgement

7.2. Illustrations, graphs, and photographs

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]
                {myfile.eps}
```

References

[1] Sun et al. "flash cut: Foreground extraction with flash and no-flash image pairs", 2007. IEEE Conference on Computer Vision and Pattern Recognition, 2007, https://doi.org/10.1109/cvpr.2007.383080. 1

[2] Gunnar Farnebäck. "two-frame motion estimation based on polynomial expansion", 2003. Image Analysis, 2003, pp. 363–370. 2