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 from Sun et al[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):

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

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:

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

4. Methodology

5. Results

6. Conclusion

6.1. Acknowledgement

6.2. Illustrations, graphs, and photographs

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

References