Semi-automated Binary Segmentation with 3D MRFs

Yahong Zhu
School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
zhuy2@tcd.ie

***Abstract*—This paper presents a semi-automated binary segmentation approach with 3D MRFs for solving the matting problem, which separates foreground and background and generating alpha mattes. We first apply a Bayesian matting algorithm by by Chuang et al. [2001], which uses a maximum-likelihood criterion to estimate the optimal opacity[1]. We then utilizes 2-dimensional MRFs to achieve binary segmentation. Finally, a 3-dimensional MRFs with motion compensation is applied to yield a high-quality mattes of foreground in a video.**

***Index Terms*—Matting, binary segmentation, motion estimation, video processing**

I. Introduction

Video matting and compositing have become crucial areas in digital video post-production. In the matting, a foreground element is extracted from a background image by generating an alpha matte. Compositing is referring to combining the foreground with a new scene, which is represented by the compositing equation:

$$C = \alpha F + (1 - \alpha)B \tag{1}$$

where C, F, and B represents the composite, foreground, and background colors, and alpha is the opacity term used to blend foreground and background linearly. Automatic segmentation includes image segmentation and video segmentation which involves motion estimation. Automatic segmentation is important since it is broadly used in video post-production to enhance visual effects and various video processing algorithms. In this paper, we implement our algorithm based on a video where a person is against a monochromatic screen. The results show that 3D MRFs are better than 2D MRFs but with more cost of computational load.

II. Video Segmentation

Given an input image, the problem is to estimate a binary segmentation mask that is equal to 0 in the background and 1 in the foreground. Using Bayes' rule, we are allowed to take the probability of each pixel being foreground or background.

$$P(\alpha(x)|I(x)) = \frac{P(I(x)|\alpha(x))P(\alpha)}{P(I(.))} \tag{2}$$

It is shown in Equ.2, where P($\alpha(x)|I(x)$) is the likelihood of observing a particular composite image and background given the true foreground and alpha-matte value. P(α) is the prior term that is a constant and can be ignored. P(I($\cdot$)) is the evidence that is used as a normalizing factor which can also be ignored. Then Equ.2 can be written as follow:

$$P(\alpha|I) = \begin{cases} P(I|\alpha = 1), & \alpha = 1; \\ P(I|\alpha = 0), & \alpha = 0; . \end{cases} \tag{3}$$

While $\alpha = 1$, we consider the pixel at site x as foreground and background when $\alpha = 0$; If the image pixels are grey-scaled, we could say that the P(I|$\alpha = 0$) follows Gaussian distribution.A pixel is composed of 3 channel, so we instead use 3-channel Gaussian distribution. Hence we take the color values at every pixel site and measure the error energy by Equ.4:

$$E(\mathbf{x}) = \frac{(I_r - \bar{B}_r)^2}{2\sigma_r^2} + \frac{(I_g - \bar{B}_g)^2}{2\sigma_g^2} + \frac{(I_b - \bar{B}_b)^2}{2\sigma_b^2} \tag{4}$$

Where $\overline{B}$ and $\sigma^2$are the mean and variance of background color.

III. Optimisation

The algorithm is composed of three different modules:

- Apply the Bayesian matting approach to solve the maximum-likelihood of foreground, background, and alpha at pixels. Use its outputs as the binary mask and energy for MAP estimation algorithm[2]
- MAP estimation algorithm with 2D MRFs as a prior for the matte
- MAP estimation algorithm with 3D MRFs with motion compensation to solve 3D matte

A. Maximum-likelihood

We calculate the Maximum-likelihood on every pixel site based on the Bayesian matting approach in the previous chapter by Chuang et al. [2001]. We first extract the mean value and the variance of the background on our video frame as the parameters to our calculation. Then, utilizing Equ. 4,

we calculate the energy at each site. Finally, we set an energy threshold to obtain our binary segmentation mask.

B. MAP Estimation with 2D MRFs

The coarse trimap and estimated warped background for every frame are provided by the first module, as previously noted. In this module, we will then apply the Maximum a-Posteriori (MAP) Estimation algorithm to address the estimation for each pixel in the unknown area presented in the trimap. This algorithm is readily obtained by calculating the negative log-likelihood for each site and using 2D MRFs as a prior for the matte. The Equ.5 shows the principle of 2D MRFs method where q_k represents the neighboring pixel site. In our 2D MRF, we use 4 neighbors to implement that algorithm, i.e. a 3×3 filter to process images.

$$p(\alpha(x)|N_\alpha(x)) = \frac{1}{Z} exp - \Lambda \left[\sum_{k=1}^{4} \lambda_k |\alpha(x) \neq \alpha(x+q_k) \right] \quad (5)$$

C. MAP Estimation with 3D MRFs

In MAP estimation algorithm, we add motion compensation[3] to 3D MRFs to achieve video segmentation. Motion compensation is an algorithmic technique that allows for objects moving in a video to estimate a frame in a video given previous and/or future frames. In our approach, we use the next frame to obtain the estimation result readily by using several coding blocks in Nuke. Compared to solving 2D segmentation using 2D MRFs, this method add the temporal dimension which allows us to accomplish video segmentation with motion, the results are accurate and superior.

IV. Implementation

For the implementation that follows, we use NUKE and several blink scripts to conduct our experiment. We first implement the Maximum-likelihood approach to obtain the energy term and binary mask. Then, we use it as the initial binary mask and energy parameter in our MAP estimation algorithm with 2D MRFs. We do 5 iterations each for our 2D MRFs and 3D MRFs appoarches to reach a satisfying result.

A. Maximum-likelihood

We use a 68 frames video in our experiment and it is loaded in NUKE and linked to video processing blocks as the input data. The general structure of this approach in NUKE can be seen in Fig.1. In our blink script, we first pick the mean value of the background and store it as a constant. This constant contains 4 channels including the alpha channel and its values is [0.739345, 0.739345, 0.213008, 1]. We then transform our video from the Linear colorspace to the YCbCr channel for computational sake. The mean value of the background is subtracted from the original pixel values by a simple Merge block and results are squared to get the (I-$\overline{B}$)^2 term. As the variance is given we store it in a constant same as the mean and multiply r, g, b channels by 2 to get the denominator in the equation. Finally, the (I-$\overline{B}$)^2 term of each channel is divided by the $2\sigma^2$, summed together for calculating the error energy. We set the threshold to 60 the get the final result i.e. a binary mask of the ML approach. It is used as the input of the optimizing algorithms.

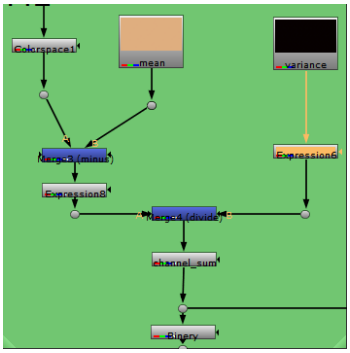

Fig. 1. Blink Scripts of Maximum-likelihood Algorithm

B. MAP Estimation with 2D MRFs

To implement MAP with the 2D MRFs algorithm, we take the binary mask generated from the Maximum-likelihood approach and the energy value at each pixel site as the input. Then, we can readily get the neighboring 4 pixels(i.e. up, down, left, and right pixel) by the position values of our current pixel. The input binary mask is processed by this filter which calculates the energy E0 and E1 at these 4 neighboring pixel sites by Equ. 5. Finally, we multiply the energy E0 and E1 by smooth term Lambda=20. Adding the energy value from Maximum-likelihood to E0 and alpha=25 to E1 to get the final energy values of E0 and E1. The final E0 and E1 values are compared to decide the pixel values at the current site. We iterate this process 5 times in our algorithm to reach the best performance. The snippets of blink scripts can be seen in the appendix. Fig.2 shows the implementation of this algorithm in NUKE.

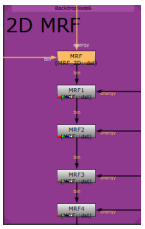

Fig. 2. Blink Scripts of MAP with 2D MRFs Algorithm

C. MAP Estimation with 3D MRFs

The basic principle of 3D MRFs is similar to what we do in 2D MRFs but with an additional temporal dimension. We readily estimate motion by the last frame and apply it to the motion compensation method in our blink scripts to elevate the performance of video segmentation. The binary mask and energy from Maximum-likelihood are the same, however, the same pixel from the last frame is also taken into account in this algorithm. This is accomplished by finding the same pixel site in the last frame, applying motion compensation to remove

the offset caused by the motion of the video, then as the third input in our 3D MRFs approach.This method provides a better solution to video segmentation compared to MAP with 2D MRFs since it involves motion estimation. The implementation of motion compensation is shown in Fig. 3. The snippets of this approach can also be seen in the appendix.

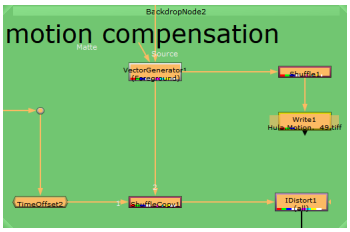

Fig. 3. Blink Scripts of Motion Compensation

V. Data

The data generated by the experiment are shown in Figures from Fig. 4 to Fig.10. Fig. 4 to Fig. 6 illustrates results in frame 47 to frame 49 from Maximum-likelihood, MAP with 2D MRFs, and MAP with 3D MRFs relatively. Fig. 7 to Fig. 9 shows the difference between each algorithm and manually segmented masks(i.e. ground-truth) in frames 47-49. Fig. 10 presents manually segmented masks from frame 47 to frame 49. By comparing those images, we now can easily see the performance of each approach.

In the next chapter, a simple measuring method is brought since we now have the ground-truth mask. A blink script was designed to show the difference between the two images. We put our generated results from every algorithm and ground-truth image into this script, the different pixel sites would be rendered as red color whereas the same pixel values would be green. Therefore, we could easily measure the quality of the output in each algorithm. Nevertheless, we are going to compare the red pixel values in the whole image to quantify the measurement.

VI. Results

This chapter goes over each of these results and compares results between each other for measuring the performance of our algorithms. The rest of this chapter goes over each of these results and compares results between each other for measuring the performance of our algorithms.

The numbers of total red pixels of each algorithm in our measuring method are given in Table 1. Generally, the MAP Estimation with 3D MRFs and motion compensation algorithm produce the best result, since it has the minimum number in each frame. Even without motion compensation, this algorithm shows slightly better results than the 2D MRFs algorithm and it is similar to the maximum-likelihood approach using the Bayesian matting.

Table 2 demonstrates the performance in every iteration on frame 47 which provides the coverage rate during 5 iterations of each algorithm. It can be seen that the MAP with 2D MRFs doesn't coverage throughout iterations. The MAP with 3D MRFs slightly converges to a point where the performance doesn't improve dramatically. Based on the observation, the number of iterations could be reduced to twice where the best results are shown.

TABLE I
Performance between each algorithm

Frame	ML	2D MRFs	3D MRFs	3D MRFs without motion compensation
47	0.00836	0.01296	0.00719	0.00819
48	0.01207	0.01439	0.00996	0.01308
49	0.02322	0.02055	0.02282	0.02436

TABLE II
Performance between each algorithm on every iteration

Iteration	2D MRFs	3D MRFs	3D MRFs without motion compensation
1	0.01289	0.00720	0.00739
2	0.01296	0.00715	0.00751
3	0.01296	0.00716	0.00771
4	0.01296	0.00716	0.00796
5	0.01296	0.00719	0.00819

VII. Conclusions

In this paper, we have presented a couple of algorithms for video segmentation that could extract foreground mattes with complicated silhouettes filmed in motion over a monochromatic screen. The main contribution of this paper is a framework that brings together various solutions of prior research, combining their strengths while addressing their imperfections. The result is a semi-automated binary segmentation method based on 3D MRFs that solves video segmentation on a frame-by-frame basis and allows for the extraction of mattes from foreground silhouettes.

From our experiments, we observe a few shortcomings of the algorithm. Even though it is possible to use our approach to extract mattes from foreground silhouettes, the results are not accurate and adequate for practical application in post-production. Spots are shown on the matte and the edges are not smooth enough to obtain a satisfactory composition result.

Inspired by Chuang et al. [4], in the future, we hope to develop a video segmentation algorithm that uses optical flow to take advantage of spatiotemporal coherence within the video. In their work, they introduce that the flow field serves as a guide for passing trimap labelings through the volume.Thus, they start at the first keyframe and flow its trimap forward in time. In their solution, they run flow in both directions — forward from one keyframe and backward from the next — and combine the observations according to a measure of per-pixel accuracy for each prospective flow.

References

[1] Yung-Yu Chuang, B. Curless, D. H. Salesin and R. Szeliski, "A Bayesian approach to digital matting," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. II-II.

Fig. 4. Sequence from 47-49 frames by Maximum-likelihood

Fig. 5. Sequence from 47-49 frames by MAP with 2D MRFs

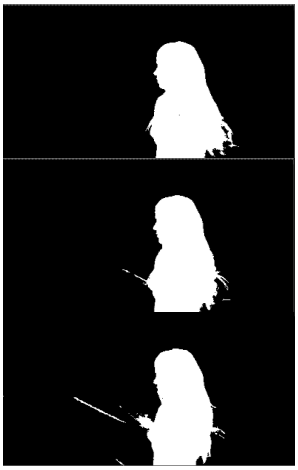

Fig. 6. Sequence from 47-49 frames by MAP with 3D MRFs

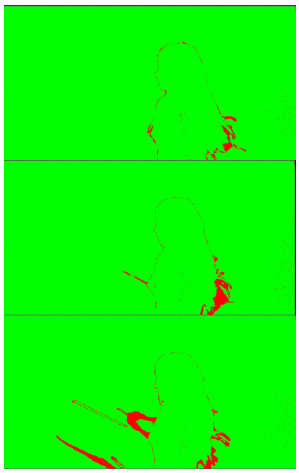

Fig. 7. Sequence from 47-49 frames by Maximum-likelihood compared with manually segmented masks

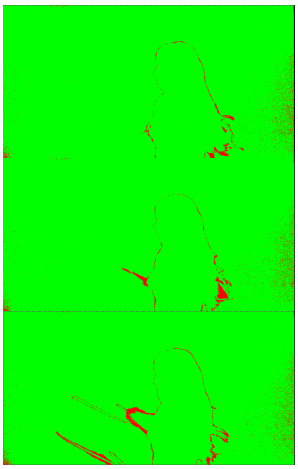

Fig. 8. Sequence from 47-49 frames by MAP with 2D MRFs compared with manually segmented masks

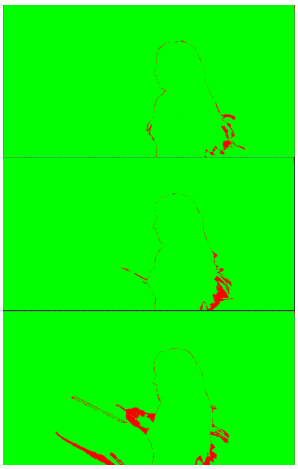

Fig. 9. Sequence from 47-49 frames by MAP with 3D MRFs compared with manually segmented masks

Fig. 10. Sequence from frame 47 to 49 of manually segmented masks

[2] R. Bassett and J. Deride, “Maximum a posteriori estimators as a limit of Bayes estimators,” Mathematical Programming, vol. 174, no. 1–2, pp. 129–144, Jan. 2018.

[3] M. E. Al-Mualla, C. N. Canagarajah, and D. R. Bull, “Multiple-Reference Motion Estimation Techniques,” Video Coding for Mobile Communications, pp. 141–155, 2002.

[4] Y.-Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski, “Video matting of complex scenes,” ACM Transactions on Graphics, vol. 21, no. 3, pp. 243–248, Jul. 2002.

[5] Y. -L. Huang, Y. -N. Liu and S. -Y. Chien, ”MRF-based true motion estimation using H.264 decoding information,” 2010 IEEE Workshop On Signal Processing Systems, 2010, pp. 99-104.

[6] A. Berman, A. Dadourian, and P. Vlahos, “Method for removing from an image the background surrounding a selected object,” 6,134,346, 2000, 2000.