Semi-automated binary segmentation with 3D MRFs

Submission for Assignment 1 EE5C01

Ruowei Li
Student ID: 22323197
Trinity College Dublin

***Abstract*—This assignment investigates the effectiveness of binary color keying in processing visual content and examines the factors that influence the matting effect, including performance differences between 2D, 3D MRF, and MRF with motion compensation. The assignment builds on existing literature on Bayesian matting and employs NUKE software to construct a binary color keying system. The system is used to process five frames of images, and multidimensional tests are conducted to evaluate its performance. The assignment effectively explores the influence of variables and factors and identifies improved methods for enhancing the matting effect. Overall, the assignment provides insights into the performance of binary color keying and offers practical recommendations for improving matting effects in visual processing.**

I. Introduction

Matting is one of the important technologies and processes in modern video and image intelligent processing. It refers to the use of matting technology to separate the foreground and background in a video or image, so that the segmented foreground has an ideal outline, then combine the foreground with other target backgrounds to form a new video or image. Matting is very important in the field of modern image and video processing, which allows developers to develop and edit visual content with creativity and precision to achieve the desired effect. By separating the foreground from the background, different foregrounds and backgrounds are combined to form effective creative visual design and development.

Matting, color keying, and compositing are important techniques and processes in the field of modern image and video processing. First of all, matting is realized through special technology such as color keying technology. Color keying separates the foreground and background of an image by using specific colors for subsequent processing. Secondly, matting is an important prerequisite and necessary preparation for compositing. The foreground image obtained in the matting process is combined with the target background image to realize the process of compositing. So the three are interrelated.

About the matting problem, many researchers have made significant contributions: Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski proposed a Bayesian approach to foreground matting in their paper "A Bayesian approach to digital matting" published in 2001. Ce Liu, Heung-Yeung Shum, and William T. Freeman proposed an extension to Bayesian matting called "Lazy Snapping" in their paper "Lazy Snapping" published in 2004.

Both binary matting and non-binary matting are techniques for separating the foreground and background of an image. The difference between the two is that binary matting divides all pixels in the image into foreground and background binary values (ie. 0 or 1), so that the pixels in the contour of the foreground object created obtain the value representing the foreground part, while All pixels outside the outline of the foreground object get a value representing the background. Non-binary matting is not a simple binary division. It assigns a transparency value combined with the foreground ratio and background ratio to all pixels in the image, so that the segmented image has a finer outline effect.

The particular problem posed in this assignment has a similar rationale to colour keying. Color keying is to use a specific color to represent the background, so that the foreground and background of the image are separated from each other in the later visual processing, so as to facilitate subsequent compositing processing. The key issue of this assignment is to divide the input image through binary segmentation technology, so that the foreground and background of the image correspond to white or black, so as to achieve the effect of foreground and background segmentation. To some extent, this is also a kind of color keying.

According to compositing equation below, the key problem in this assignment is to obtain the alpha, which is a binary variable between 0 and 1. The final alpha matting matrix is obtained through MAP, 2D MRF, 3D MRF and motion compensation in the matting process. The α value in the matrix corresponds to all the pixels of the source image, which determines the composition and outline of the foreground and background pixels.

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

II. Modeling Mathematically

Regarding the matting process, consider a pixel at site $\mathbf{x} = [h, k]$ in the observed image I_n in frame n, the main task is to base on the neighbor pixels of each pixel in the original image and The binary label of the neighbor pixel in the previous frame is estimated. Among them, a pixel of binary label 1 is foreground pixel, and the one of label 0 is the background pixel. We wish to get binary label α which maximizes the MAP distribution.

First, by estimating the mean value of the background image color, the Gaussian parameters are measured to obtain the background energy, and the appropriate foreground threshold parameters are set according to the image.

Subsequently, energy minimization is performed inside the MAP. We want each pixel to obtain a pixel binary label distribution that maximizes the probability, then we need to minimize the energy E of each pixel. The Gibbs Energy Function as MRF and MAP Estimate equation are shown below, and the pixel energy needs to be minimized according to the neighbor pixels of each pixel to obtain the maximum probability binary label α. Further detailed algorithms will be described in the next section.

$$p(\alpha(x) \mid N_\alpha(x)) = \frac{1}{Z} exp - \Lambda[\sum_{k=1}^{8} \lambda_k \mid \alpha(x) \neq \alpha(x+q_k) \mid] \tag{2}$$

$$E_l = \frac{(B_y - \bar{B_y})^2}{2\sigma_y^2} + \frac{(B_u - \bar{B_u})^2}{2\sigma_u^2} + \frac{(B_v - \bar{B_v})^2}{2\sigma_v^2} \tag{3}$$

$$E_s(0) = \sum_{k=1}^{8} V(0, \alpha(x+q_k)) \tag{4}$$

$$E_s(1) = \sum_{k=1}^{8} V(1, \alpha(x+q_k)) \tag{5}$$

III. Algorithm and Optimisation

This binary matting algorithm can be divided into three main parts according to different functions and the order of execution:

1. Maximum Likelihood (ML) part: measure Gaussian parameters such as background average color, set foreground judgment threshold, calculate background energy and initial maximum likelihood α, these parameters are used for subsequent MRF processing.

2. Markov Random Field (MRF) part: Obtain the background energy and initial α output by the ML part, combine with Maximum A Posteriori (MAP), and calculate the minimized energy within the range of each pixel's neighbor pixels to obtain the maximum probability The binary label α, and then implement binary matting on the original image. The part is mainly divided into 2D MRF (MRF in space), 3D MRF (MRF in time and space) and 3D MRF with Motion Compensation (3D MRF with Motion Compensation module).

3. Motion Compensated Bin part: Used to improve the accuracy of the alpha matte by taking into account the motion of the camera and/or the subject in the scene. Motion vectors are calculated between adjacent frames, and the foreground and background are shifted accordingly to align them. This results in a more accurate estimate of the alpha mask, leading to better results in the final compositing.

The main process of the algorithm is as follows: first, the background energy E_l0 and the maximum likelihood α are calculated through Gaussian parameters such as the color mean. Then, input E_l0, E_t and α to the MRF module for matting processing. In the MRF module, Iterated Conditional Modes (ICM) is used, which means choosing the variable which maximizes the local conditional density, and then doing the same at every site. The MAP algorithm in the MRF calculates the background and foreground energies $E0$ and $E1$ of all pixels and minimizes them, the energy calculation range is the surrounding eight pixels of the target pixel and the 3×3 pixels range centered on the pixel in the previous frame. and finally assigns the maximum possible α label to each pixel to obtain binary matting. The flow chart of the algorithm is shown in the fig 1.

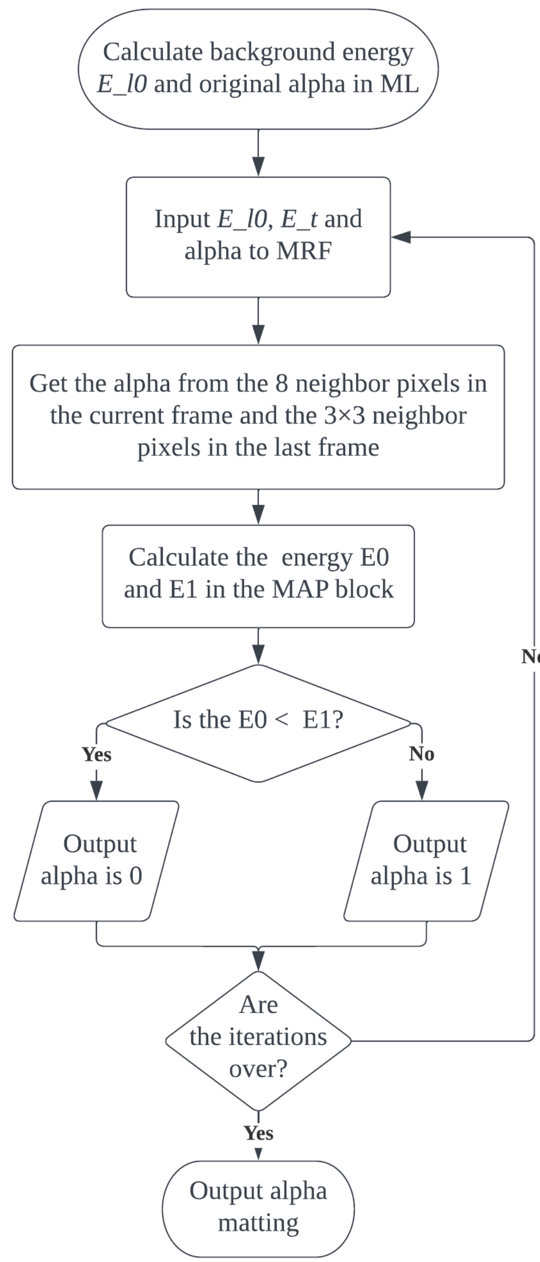

Fig. 1. Flow chart of the Algorithm

IV. NUKE IMPLEMENTATION

The above algorithm is implemented in NUKE. According to the algorithm structure and process, it can be divided into Maximum Likelihood (ML) block, Markov Random Field (MRF) (MRF) block and Motion Compensated BIn for motion compensation.

In the ML block, the original image is input in a linear manner, and the background energy and Maximum Likelihood alpha are calculated through the Background energy likelihood energy and ML estimate modules in turn, which are used as the output of the ML module.

In the MRF block, in order to test and evaluate the matting effect, it is divided into three branches: 2D MRF, 3D MRF and 3D MRF with Motion Compensation. Each branch is set with five iterations of MRF processing, resulting in more accurate alpha output. Each iteration is an MRF processing block, input background energy and initial or previous generation alpha. In 2D MRF, the neighborhood is set to 8 pixels around the target pixel; in 3D MRF, the neighborhood is set to 8 pixels in the current frame and a 3×3 block of 9 pixels in the previous frame. The pixel processing of the previous frame is realized through the offset module in NUKE.

In the Motion Compensated Bin block, use the offset module to obtain the image information of the previous frame, use the VectorGenerator to generate the motion vector, use shuffle to combine the image information of the previous frame and the motion vector, and the obtained output enters IDistort processing to obtain the previous frame with Motion Compensation A frame of alpha, which is fed into the 3D MRF with Motion Compensation block. The system diagram is shown in the fig 2.

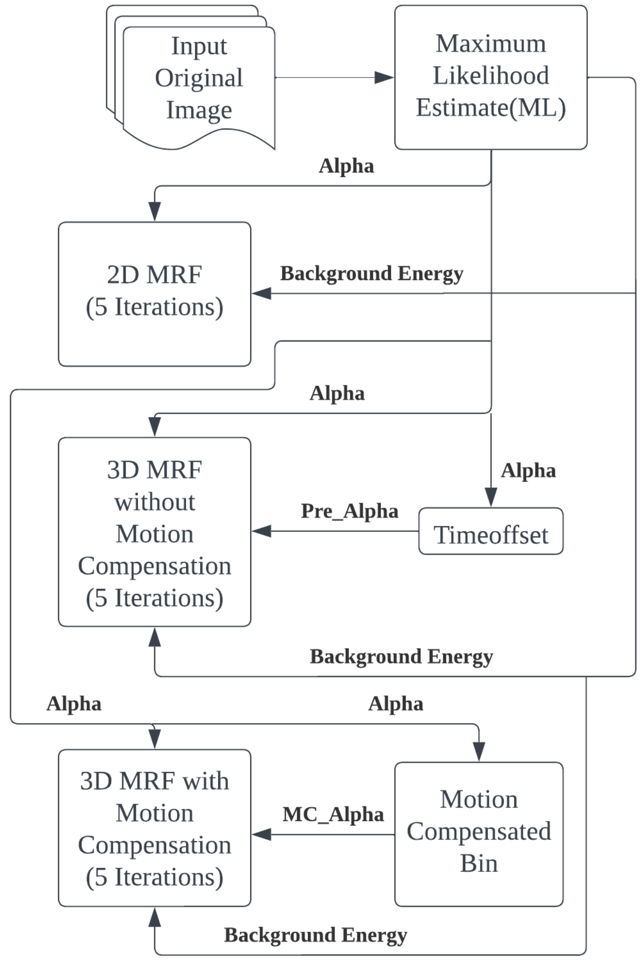

Fig. 2. System Diagram

V. EXPERIMENTS AND PERFORMANCE MEASURES

For the experiment of the results obtained by the matting algorithm, I mainly used the methods of calculating mse, calculating accuracy and calculating confusion matrix, for different algorithm blocks (2D MRFs, 3D MRFs and 3D MRFs with MC) and different iterations(2nd iteration and 5th iteration)are compared with each other, analyze experimental results and draw conclusions.

The experiment is mainly divided into four parts(The codes corresponding to each experiment are in brackets):

- (3D-2D) The output matting of 3D MRF and 2D MRF is evaluated and compared with the results, and the two parameters of mse and accuracy are compared and analyzed.
- (3D-ML) The evaluation experiment and comparison of the output matting of 3D MRF and single frame likelihood keying were carried out, and the two parameters of mse and accuracy were compared and analyzed.
- (3Dmc-3D) The 3D MRF with motion compensation and the output matting of 3D MRF without motion compensation were evaluated and compared, and compared and analyzed from the two parameters of mse and accuracy.
- (3Dmc-smo) Conduct a certain number of adjustment experiments on the Smoothness weight Lambda, and analyze the impact on 3D MRF with MC.

VI. RESULTS AND DISCUSSION

A. (3D-2D) Performance experiment between 3D MRF and 2D MRF

The experiment between 3D MRF and 2D MRF is carried out by calculating the mse and accuracy between the output matting and ground truth alpha respectively. The experimental results obtained are shown in the table I:

TABLE I
RESULTS OF EXPERIMENT 3D-2D

Experiment Name	**Comparing Metrics**		
	MSE	***Accuracy***	***TP Rate***
3D 5th iteration	0.018259	0.981741	0.909421
2D 5th iteration	0.019572	0.980428	0.925047

B. (3D-ML) Performance experiment between 3D MRF and single frame likelihood keying

The experiment between 3D MRF and single frame likelihood keying is carried out by calculating the mse and accuracy between the output matting and ground truth alpha respectively. The experimental results obtained are shown in the table II:

TABLE II
RESULTS OF EXPERIMENT 3D-ML

Experiment Name	**Comparing Metrics**		
	MSE	***Accuracy***	***TP Rate***
3D 5th iteration	0.018259	0.981741	0.909421
Single frame likelihood keying	0.022552	0.977448	0.930647

C. (3Dmc-3D) Performance experiment between 3D MRF with motion compensation and 3D MRF without motion compensation

The experiment between 3D MRF with motion compensation and 3D MRF without motion compensation is carried out by calculating the mse and accuracy between the output matting and ground truth alpha respectively. The experimental results obtained are shown in the table III:

TABLE III
RESULTS OF EXPERIMENT 3DMC-3D

Experiment Name	**Comparing Metrics**		
	MSE	***Accuracy***	***Subhead***
3D with MC 5th iteration	0.018163	0.981837	0.930478
3D no MC 5th iteration	0.018259	0.981741	0.909421

D. (3Dmc-smo) Performance experiment for a certain number of adjustment experiments on the Smoothness weight Lambda, and analyze the impact on 3D MRF with MC.

In the experimental group, the smoothing parameter Lambda was set to 20, 40 and 60, and the matting processing was performed to calculate the mse and accuracy respectively, and the obtained data were as follows:

TABLE IV
RESULTS OF EXPERIMENT 3DMC-SMO

Experiment Name	**Comparing Metrics**		
	MSE	***Accuracy***	***Subhead***
3D MC with Lambda 20	0.018163	0.981837	0.930478
3D MC with Lambda 40	0.018264	0.981736	0.932229
3D MC with Lambda 60	0.018338	0.981662	0.932667

VII. CONCLUSIONS

According to the above series of experiments and performance evaluation, we can conclude that the matting processing result with multi-frame image input is significantly improved than the processing result with only a single frame image input. 3D MRFs with motion compensation will also be better than 3D MRFs without motion compensation. The smoothing parameter also has a certain influence on the effect of matting, and the multi-layer iterative MRF processing has a positive effect on the effect of matting.

- According to the above performance experiments and factor analysis of binary matting supported by color keying, the following conclusions can be drawn: In different situations, the performance of video matting and image matting is directly different. Both have certain advantages and disadvantages. Video matting is not always better than image matting. Video matting has advantages over image matting when dealing with videos that contain motion. In video matting, the goal is to extract foreground objects from the background over a sequence of frames, taking into account any motion that occurs. Because video matting takes into account motion over time, it can often produce more accurate results than image matting when working with video. Image matting is usually more straightforward and easier to implement than video matting because it only needs to consider one image at a time. Image matting may be preferred in situations where foreground objects and background are relatively stationary and motion is not an important factor.
- According to the above-mentioned experiments and performance evaluation, my model is relatively general in dealing with poor motion estimation. The matting effect obtained by the pixel block in a relatively extreme motion state is relatively poor after directly passing through 3D MRFs without motion compensation, but After 3D MRFs with motion compensation, the effect of matting has been significantly improved. However, according to the above experiments, it can be seen that the matting effect still has a certain room for improvement compared with the ground truth matting.
- The influence and effect of MRF prior on the matting effect is significant. In the binary matting system I designed, MRF prior and MAP posterior complement each other and are indispensable. In the process of matting, the Bayesian formula is used, the goal of which is to estimate the probability that each pixel belongs to the foreground or background given the observed image. In this Bayesian formulation, the MRF prior is a common way to model the spatial dependence between adjacent pixels. The MRF prior assumes that the probability of a pixel belonging to a class (foreground or background) is influenced by the probabilities of its neighbors belonging to the same or a different class. Therefore, MRF priors can improve the accuracy and robustness of matting solutions, especially when images have complex textures or fine details. Without MRF priors, matting algorithms may produce jagged or inconsistent segmentations that do not reflect real object boundaries.
- According to the analysis of the above experiment (3DMC-smo) results, the three groups of smoothness hyperparameter are 20, 40 and 60 respectively. Obvi-

ously, when adjusting the smoothness hyperparameter, the output matting obtained by 3D MRFs with motion compensation is compared with the ground truth, and the obtained mse and accuracy are significantly changed. In my matting system, the change of the smoothness hyperparameter from 20 to 60 causes the mse of the matting result to rise continuously, and the accuracy continues to decline accordingly. Therefore, in this system, the most suitable smoothness hyperparameter should be smaller, around 20.

Compared with the complex and efficient DNN modeling method, the modeling method in this assignment is relatively simple and efficient in matting processing, and it is convenient. It is convenient for visual content editors to adjust parameters according to the situation and explore near-optimal parameter settings. However, the matting modeling method in DNN is usually very complicated. When unsatisfactory effects appear due to changes in processing source data, it is distressing to manually change the modeling structure or parameters.

VIII. Declaration

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar. I certify that this submission is my own work.

References

Appendix

A. Blink Script for 3D MRF

kernel MAP: ImageComputationKernel¡eComponentWise¿
Image¡eRead, eAccessPoint, eEdgeClamped¿ src$_b energy$;
Image¡eRead, eAccessRandom, eEdgeClamped¿ alpha;
Image¡eRead, eAccessRandom, eEdgeClamped¿ pre$_a lpha$;
Image¡eWrite¿ dst$_a lpha$;
void process(int2 pos)
float lambda = 20.0f;
// calculate background energy
float E$_l 0 = src_b energy()$;
// set E$_t = 5$
float E$_t = 5$;
// Detect the 8 neighbor pixels
float left = alpha(pos.x - 1, pos.y);
float top = alpha(pos.x, pos.y + 1);
float bottom = alpha(pos.x, pos.y - 1);
float right = alpha(pos.x + 1, pos.y);
float lefttop = alpha(pos.x - 1, pos.y + 1);
float righttop = alpha(pos.x + 1, pos.y + 1);
float leftbottom = alpha(pos.x - 1, pos.y - 1);
float rightbottom = alpha(pos.x + 1, pos.y - 1);
// Detect the 8 neighbor pixels in the previous frame
float pleft = pre$_a lpha(pos.x - 1, pos.y)$;
float ptop = pre$_a lpha(pos.x, pos.y + 1)$;
float pbottom = pre$_a lpha(pos.x, pos.y - 1)$;
float pright = pre$_a lpha(pos.x + 1, pos.y)$;
float plefttop = pre$_a lpha(pos.x - 1, pos.y + 1)$;
float prighttop = pre$_a lpha(pos.x + 1, pos.y + 1)$;
float pleftbottom = pre$_a lpha(pos.x - 1, pos.y - 1)$;
float prightbottom = pre$_a lpha(pos.x + 1, pos.y - 1)$;
float pcenter = pre$_a lpha(pos.x, pos.y)$;
//calculate Es0 spatial energy alpha = 0
float E$_s 0 = left + top + right + bottom + lefttop + righttop + leftbottom + rightbottom + pleft + ptop + pbottom + pright + plefttop + prighttop + pleftbottom + prightbottom + pcenter$;
//calculate Es1 spatial energy alpha= 1
float E$_s 1 = fabs(left - 1.0f) + fabs(top - 1.0f) + fabs(right - 1.0f) + fabs(bottom - 1.0f) + fabs(lefttop - 1.0f) + fabs(righttop - 1.0f) + fabs(leftbottom - 1.0f) + fabs(rightbottom - 1.0f) + fabs(plefttop - 1.0f) + fabs(pleftbottom - 1.0f) + fabs(prighttop - 1.0f) + fabs(prightbottom - 1.0f) + fabs(pcenter - 1.0f)$;
float E0 = E$_l 0 + lambda * E_s 0$;
float E1 = E$_t + lambda * E_s 1$;
// if E0¡E1 output 0, E0¿E1 output 1
if (E0 ¡ E1)
dst$_a lpha() = 0.0f$;
else
dst$_a lpha() = 1.0f$;
;