Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Yin and Yang Artistic Chaos
Bayesian Matting - Implementation

Computational Methods 5C22 Assignment - II

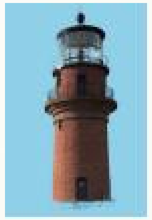
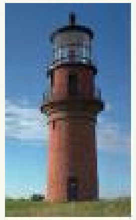
YUNING ZHU
ABHISHEK DUTTAGUPTA
CHAI JIE

Yeah, So What is Image Matting?

Concept & Math

➢ Brief introduction on Bayesian Matting

– Colour Segmentation and Matting is an important step for an image or video composting, which makes it very essential for movie post-production and other fields.

–   

(Input) (Alpha Matte) (Composite)

$$C = \alpha F + (1 - \alpha)B$$
$$C = Composite\ Image$$
$$F = Foreground\ of\ Image$$
$$B = Background\ of\ Image$$
$$\alpha = Alpha\ Matte\ of\ Image$$

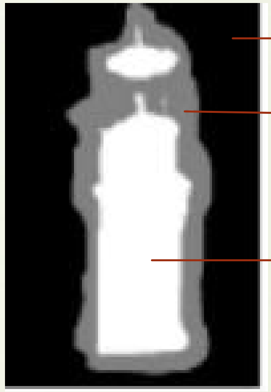
– Efficiently calculating α becomes essential, as with techniques described in Mishima, Knockout, and Difference Matting methods, it has been that the accuracy of hussy images is not that great.

– Similarly, using the Maximum likelihood process, in which we measure the Gaussian parameters of the background (mean and variance) and set some threshold in the energy distribution to confirm whether a site is black or not, doesn't give great accuracy.

– Bayesian Matting instead follows maximum posterior distribution and provides better results and improvements on the above. More details on the next page.

Concept & Math

- Bayesian Matting involves estimating the α values of the unknown region of the tri matte using the maximum posterior technique.
- Simplifying it, we initially create a tri matte of an image, marking the foreground, background, and the unknown region. Then using Bayes' rule for representing the result as the sum of log-likelihood, we select whether each of the pixels in the unknown region falls in the foreground or background, finally preparing the alpha matte.



The Math behind Bayesian Method

<u>***Step 1***</u> : - Using Bayes Rule, calculating the maximum likelihood of F,B and α given the Image.

$$\arg\max P(F, B, \alpha \mid C)$$
$$= \arg max\, P(C|F,B,\alpha)\frac{P(F,B,\alpha)}{P(C)}$$
$$= \arg max\, P(C|F,B,\alpha)P(F)P(B)\frac{P(\alpha)}{P(C)}\ (Considering\ P(F), P(B) \&\ P(\alpha)$$
$$independent\ of\ each\ other)$$

Concept & Math

– The Math behind Bayesian Method (contd.)

Step 2 :- Now given P(C) is constant

$$= \arg max\, P(C|F,B,\alpha)P(F)P(B)P(\alpha)$$

Step 3 :- Taking Logs

$$= \arg max \log P(C|F,B,\alpha) + \log P(F)P(B)P(\alpha)$$

$$= \arg max \log P(C|F,B,\alpha) + \log P(F) + logP(B) + logP(\alpha)$$

Step 4 :- Now, $\log P(C|F,B,\alpha)$ should follow or be consistent with the basic composting or matting equation [1] and considering Gaussian distribution, so

$$\log P(C|F,B,\alpha) = -\frac{||C-\alpha \mathrm{F} - (1-\alpha)B||^2}{\sigma^2} \text{ with } \sigma = standard\ deviation.$$

Concept & Math

– **The Math behind Bayesian Method (contd.)**

Step 5 :- Now, we need to find out the other probabilities. P(F) and P(B) can be obtained from the tri matte colour Gaussian distribution. Calculating the mean $\bar{F}$ and weighted variance $\sum F$, we obtain

$$\log P(F) = -\frac{(\mathrm{F}-\bar{F})^T(\mathrm{F}-\bar{F})}{2\sum F}$$

Step 6 :- Now $\log P(\alpha)$is assumed to be constant and F and B are estimated based on the below equation.

$$\begin{bmatrix} \Sigma_F^{-1} + I\alpha^2/\sigma_C^2 & I\alpha(1-\alpha)/\sigma_C^2 \\ I\alpha(1-\alpha)/\sigma_C^2 & \Sigma_B^{-1} + I(1-\alpha)^2/\sigma_C^2 \end{bmatrix} \begin{bmatrix} F \\ B \end{bmatrix} = \begin{bmatrix} \Sigma_F^{-1}\overline{F} + C\alpha/\sigma_C^2 \\ \Sigma_B^{-1}\overline{B} + C(1-\alpha)/\sigma_C^2 \end{bmatrix},$$

Step 7 : - Next, considering the F & B as constant, we can use the closed form solution for α $(where\ we\ try$ to maximise this function taking partial derivative with respect to α and setting them equal to 0).

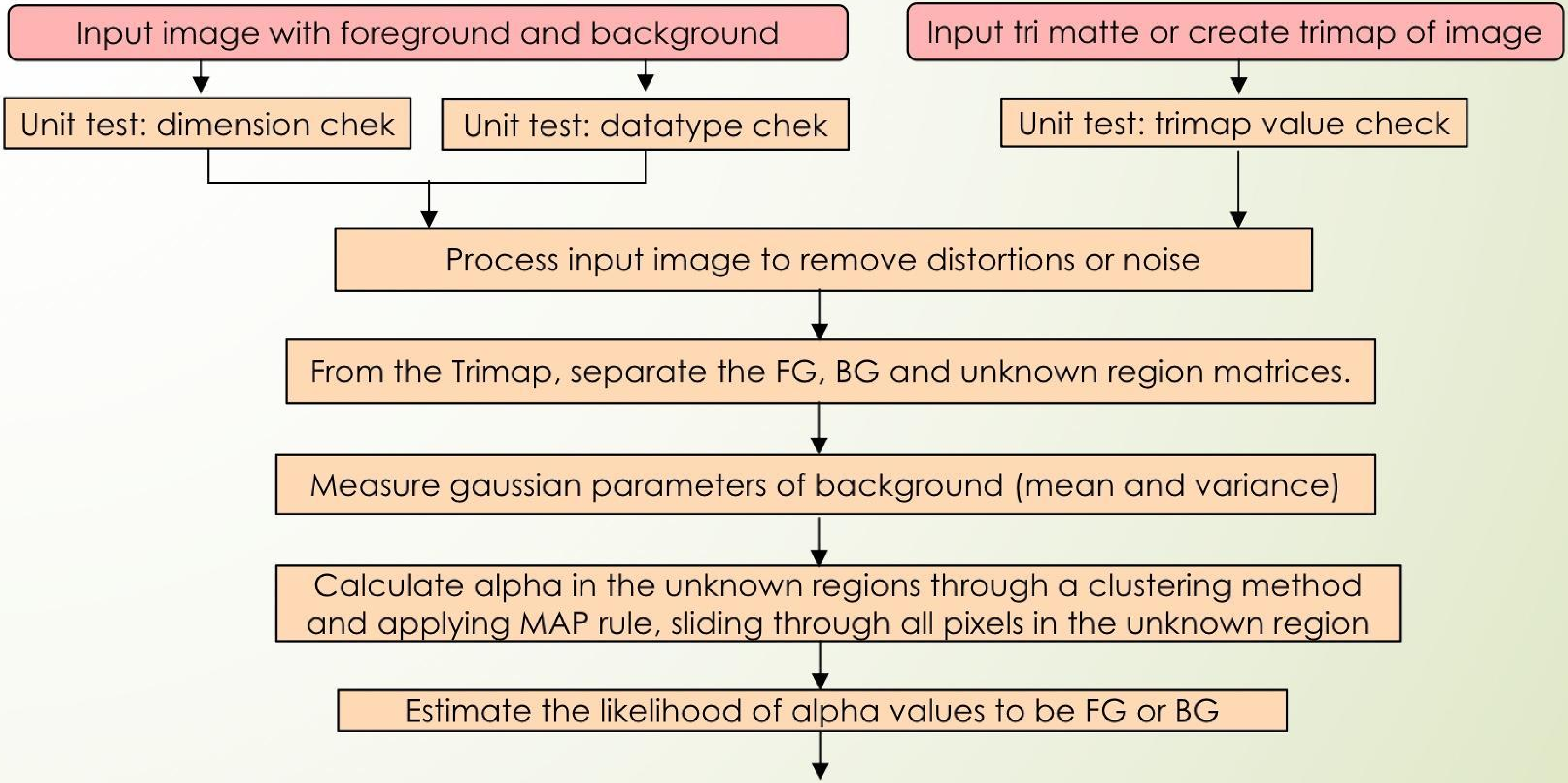
$$\mathrm{P}(\alpha) = \frac{(C-B)^T(\mathrm{F}-B)}{(F-B)(F-B)}$$

Step 8 : - Then using above α , F and B value's we estimate whether a pixel is F or B.

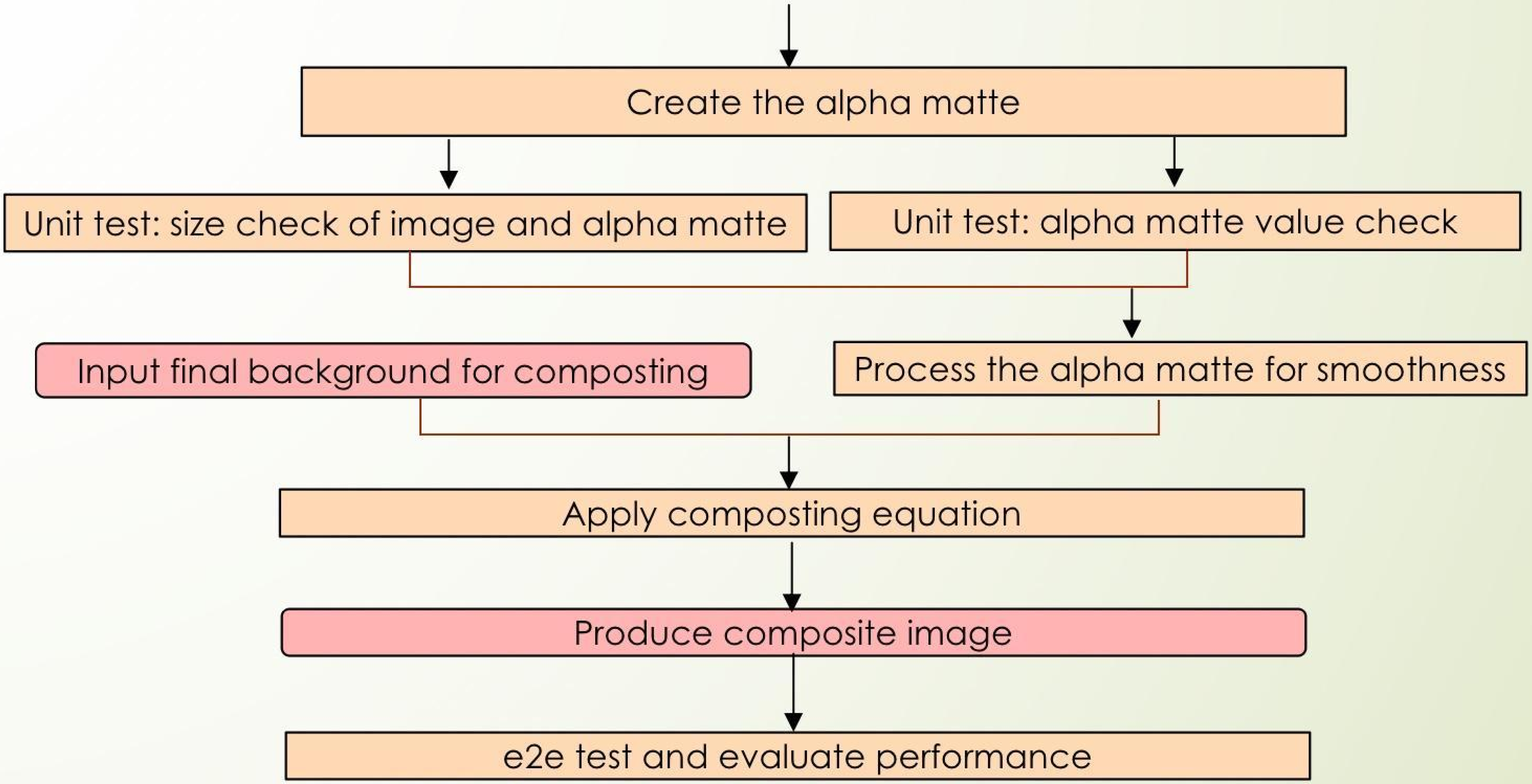
The Algorithm in simple terms

1. Begin by establishing the foreground and background values using the given tri-map.
2. Cluster the surrounding pixels of the foreground and background in the unknown tri-map regions through the application of established clustering techniques, based on the known pixel values.
3. Establish an initial estimate of the F and B values while keeping a constant.
4. Refine the estimate of a by repeating this process in a loop until the optimal values for all variables have been found.
5. Use the obtained a value to compute the final estimates of F and B for the unknown regions, resulting in the a matte.
6. This a matte can then be utilized to overlay various backgrounds onto the foreground image through the application of the composite equation.
7. The final result can be further improved by applying a smoothing technique to reduce noise and refine the a matte.

Flow Diagram of the Complete Process



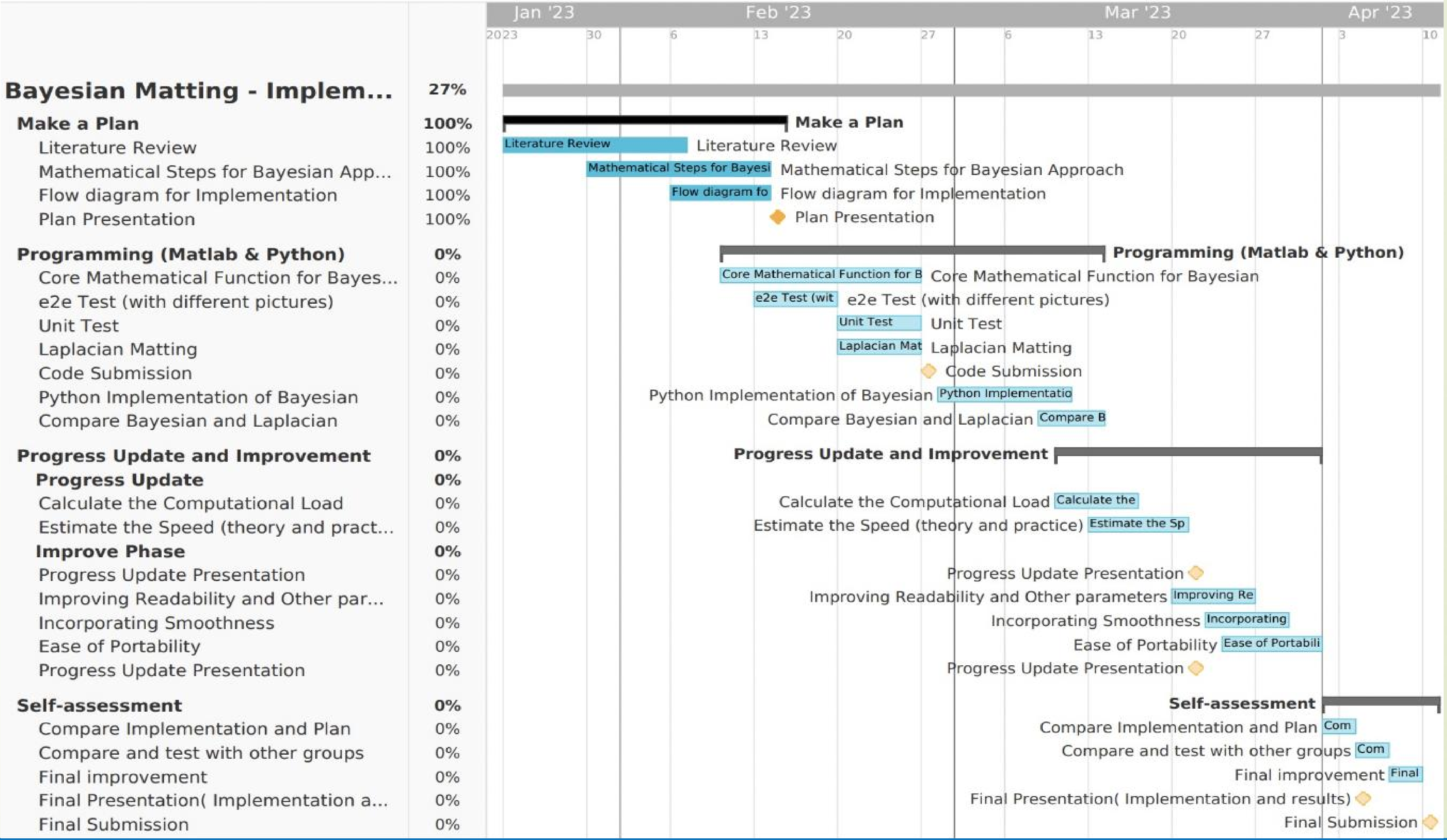
Flow Diagram of the Complete Process



Performance Evaluation

- Performance evaluation is done solely from images that can be obtained from http://www.alphamatting.com/
- Once we have the alpha matte of our code based on images from this website, we can compare them with the ground truth alpha matte images available on this website and evaluate our performance.
- The website provides us with performance evaluation systems namely SAD and MSE. Based on our image output and using any two of the above evaluation system, we will try to evaluate our performance.

Gantt Chart



Jan '23
Feb '23
Mar '23
Apr '23
2023
30
6
13
20
27
6
13
20
27
3
10
Bayesian Matting - Implem... 27%
Make a Plan 100%
Literature Review 100%
Mathematical Steps for Bayesian App... 100%
Flow diagram for Implementation 100%
Plan Presentation 100%
Programming (Matlab & Python) 0%
Core Mathematical Function for Bayes... 0%
e2e Test (with different pictures) 0%
Unit Test 0%
Laplacian Matting 0%
Code Submission 0%
Python Implementation of Bayesian 0%
Compare Bayesian and Laplacian 0%
Progress Update and Improvement 0%
Progress Update 0%
Calculate the Computational Load 0%
Estimate the Speed (theory and pract... 0%
Improve Phase 0%
Progress Update Presentation 0%
Improving Readability and Other par... 0%
Incorporating Smoothness 0%
Ease of Portability 0%
Progress Update Presentation 0%
Self-assessment 0%
Compare Implementation and Plan 0%
Compare and test with other groups 0%
Final improvement 0%
Final Presentation(Implementation a... 0%
Final Submission 0%
Make a Plan
Literature Review
Mathematical Steps for Bayesian Approach
Flow diagram for Implementation
Plan Presentation
Programming (Matlab & Python)
Core Mathematical Function for Bayesian
e2e Test (with different pictures)
Unit Test
Laplacian Matting
Code Submission
Python Implementation of Bayesian
Compare Bayesian and Laplacian
Progress Update and Improvement
Calculate the Computational Load
Estimate the Speed (theory and practice)
Progress Update Presentation
Improving Readability and Other parameters
Incorporating Smoothness
Ease of Portability
Progress Update Presentation
Self-assessment
Compare Implementation and Plan
Compare and test with other groups
Final improvement
Final Presentation(Implementation and results)
Final Submission

Timeline and Milestones

- Following Milestones are considered for completion of the total project with target dates.
- Milestone 1 : - Completion of Draft Code (Week 5).
- Milestone 2 : - Completion of e2e and unit tests Code (Week 6).
- Milestone 3 : - MATLAB code submission (End of Week 6).
- Milestone 4 : - Start of Python conversion of Complete Code (Week 7).
- Milestone 5 : - Completion of Python Code (Week 10).
- Milestone 6 : - Comparison of Python Code with other teams (Week 11).

Testing Proposal

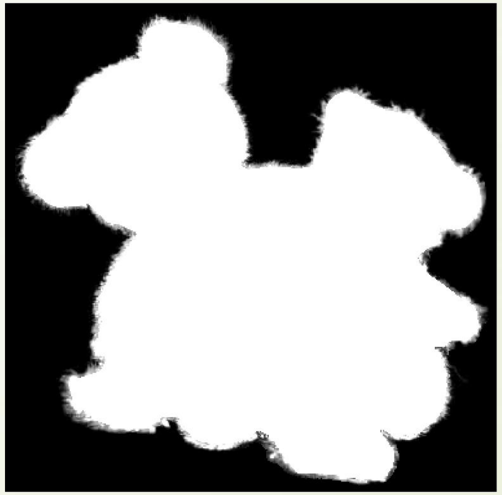
- End to End Testing

  Following end to end tests are planned.

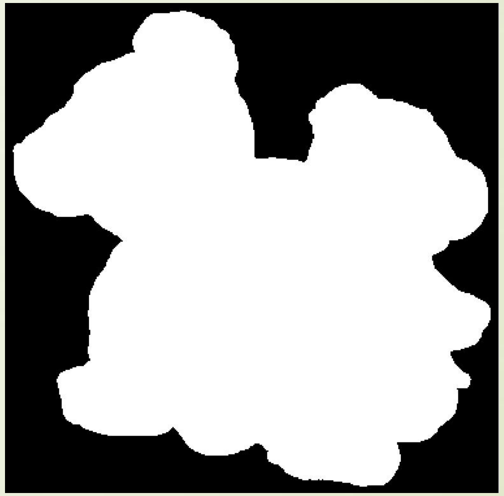
  - Alpha Matte Comparison with Matting Laplacian In-built Function.
    The alpha matte obtained from Bayesian Method (MAP) will be compared with the Alpha Matte obtained from the Matting Laplacian method using the MATLAB del2 function. Almost Similar or better results are expected.

  - Performance Evaluation (SAD, MSE.)

Input Image



Bayesian MAP

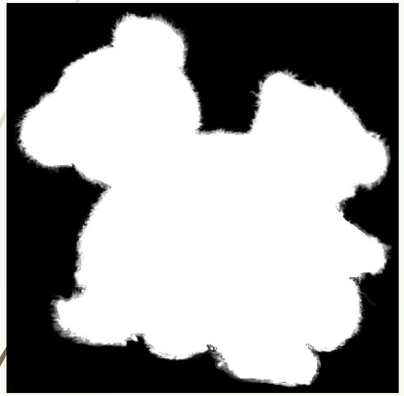


Inbuilt Laplacian

Testing Proposal

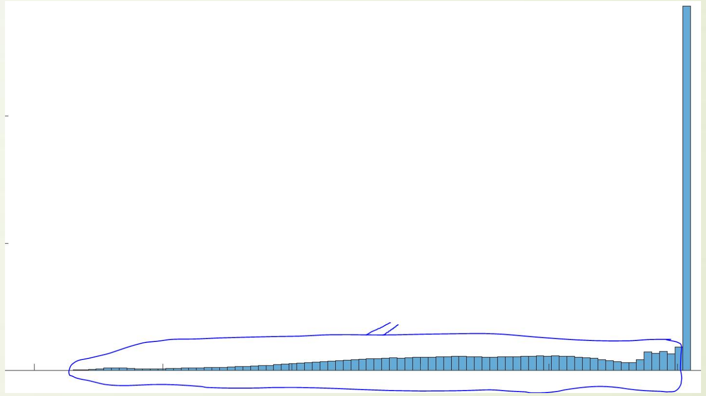
- End to End Testing

- Alpha Matte Comparison between MAP and ML methods.
  The alpha matte obtained from Bayesian Method (MAP) will be compared with the Alpha Matte obtained from the Maximum likelihood method. ML method estimates alpha matte with respect to the threshold set from a background. If a better Alpha Matte is obtained from MAP, it is considered a success.



MAP Bayesian

ML Method



Typical Histogram

- Foreground Colour distribution of Input and Composite Image.
  The Histogram distribution of the Input and Composite Image Foreground will be compared. The same histogram should prove the success of the matting process. The foreground should be more or less similar to the blue-circled image shown in a histogram.

Testing Proposal

- Unit Tests
- Following Unit tests are planned as of now.
  - Image Dimension Check.
    Dimension of the Input Image and Composite images are checked. Basically, a function will be implemented to check whether the image has three channels or not. For Example, if in the code, if c = 3, the test is passed.

```
% Load the input image and trimap
img = imread('input.jpg');
% Convert the image and trimap to double precision
img = double(img);
% Get the size of the image
[m, n, c] = size(img);
% Draft logic of Unit test
if (c == 3)
    printf("Dimension in correct");
else
    printf("Image is broken");
end
```

  - Size Check of the Input image and Alpha Matte Image.
    The image size of the Input and Alpha Matte should be equal, an size comparator unit test will be performed for checking this. For example, if m = h and n = k in the code aside, the test is passed.

```
% Load the input image and trimap
img = imread('input.jpg');
% Convert the image and trimap to double precision
img = double(img);
% Get the size of the image
[m, n, c] = size(img);
% Get the size of the alpha (after matting)
[h, k] = size(alpha);
% Draft logic of Unit test
if (m == h && n==k)
    printf("Input and Alpha Dimension's in correct");
else
    printf("Alpha Dimension is not correct");
end
```

Testing Proposal

- Unit Tests
  - Image Datatype check

    All image processing and computation of image will be done in double datatype as many computations can't be done in uint8 format. A check will be performed across all images converted to double format, whenever input is taken. A code snippet for it is shown aside.

```
% Load the input image and trimap
img = imread('input.jpg');
% Convert the image and trimap to double precision
img = double(img);
% Draft logic of Unit test
T = isa(img,'double');
if (T == 1)
    printf("Input image dat type is correct");
else
    printf("Image Data type needs change");
end
```

  - Alpha Matte Matrix value check.

    The Final alpha matte matrix is supposed to be of only 0's and 1's representing background and foreground. A check will be implemented to confirm whether any other values are not there. If the answer to check_U in the below snippet is 0 and check_1 > 0 and check_0 > 0, then it is correct (considering alpha as the final output alpha matte).

```
% Draft logic of Unit test
check_U = find(alpha~=0 && alpha ~= 1);
check_1 = find(alpha == 0);
check_0 = find(alpha == 1);
```

Roles and Responsibilities

YUNING	ABHISHEK	CHAI-JIE
DESIGN		
Flow Diagram Creation	Algorithm and Mathematical Implementations	e2e and Unit test Ideas
Assignment Plan and Timeline	Presentation Slide Preparation	Presentation Slide Preparation
CODING		
Input Image and Plotting Function	MAP Bayesian Implementation Function	Composite Image Creation Function
Tri Matte Function	Performance Evaluation Function	Image Smoothening / Filtering Function
TESTING		
e2e - Comparision of ML and MAP Alpha Matte's	e2e - Comparision of MAP and Matlab inbuilt Matting Laplacian	Unit Test - Image Dimension Check
Unit Test - Alpha Matte Matrix Value Check	e2e - Comparision of Input and composite image FG histogram	Unit Test - Size Check of Input image and alpha matte
		Unit Test - Image Data Type Check

Key Remembrance

- Since, this project is a group effort, following points below will be followed thoroughly to achieve the final completion.
  - Version Control:- A common GitHub directory will be made wherein every major step by any individual. The project files will be updated. Since it is a team collaboration, this becomes very important.
  - Linting:- Linting of the complete code becomes very important to improve readability, which will be made sure throughout.
  - All Unit tests and e2e tests are pre-code ideas and based on further progress of the project, more tests might be included or the current ones might be changed.

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Thank You