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

Bayesian Matting

KEITH DONOGHUE
JIANG MINSHUAI
SONG XUZHE
SHREYA UDAY KUMAR KODGI

Date: 14/2/2024

Contents

Roles

Task :	Name:	Cross Check
Code Dev.	Jiang	Shreya
Algorithm Dev.	Shreya	Jiang
Test	Keith	Song
	Song	Keith

Introduction

Timeline and milestones

Time	Task	week 1	week 2	week 3	week 4*	week 5	week 6*	week 7	week 8	week 9*	week 10	week 11*
week 1	Read paper of Bayesian and Laplacian Matting.											
week 2 & 3	Find open source code for reference.											
week 4	Decide roles of everyone in the group and make the initial design of the project.											
week 5	Try to demo the code in matlab and compare with Laplcian.											
	Test for read image and match image function.											
	Write the read image and match image function in Matlab.											
	Design the structure of E2E test.											
	Using APP Designer develop GUI											
week 6	Prepare for presentation 2											
	Doing unit test for the function of matting and output.											
	Write the matting image and output function in Matlab.											
week 7	Find a method to implement the alogrithm in python											
	Design the unit test for matting image and output function in python.											
	Write the read image and match image function in Python.											
	Implement the E2E test in python											
	Develop GUI by QT5.											
week 8	Focus on python implementation and e2e tests.											
	Doing unit test for matting image and output function.											
	Write the matting image and output function in Python.											
week 9	Cross check and update on the progress											
week 10	Finding Optimal Matting Algorithms and doing evaluation											
	Prepare for the final presentation											

Milestones

Week 4 : Finish the code design

Week 6 : Finish the MATLAB part

Week 9 : Finish the Python part

Week 11 : Finish the final presentation

SHREYA	
KEITH	
SONG	
JIANG	

Matting

Code structure

Batch input

Single input

Read image

Input:
Image path

Output:
Image data

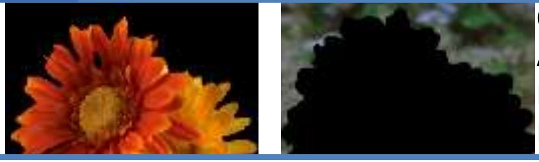
Match image

Input:
Image data

Output :
Foreground
Background
Unknown

Matting image

Input:
Foreground
Background
Unknown



Output :
Alpha matte
Foreground,
Background

Output and evaluation

Key takeaway: Use functions to read and match images. With the help of GUI, it can operate easily.

Bayesian matting

Algorithm

STEP 1 - Input Parameters: Image, Trimap, Variance, Inverse Covariance matrices of F and B, Mean values of F and B.

STEP 2 - Thresholding: Create arrays for foreground, background, and unknown regions.
Compute mean values and inverse covariance matrices.

STEP 3 - Initialize Variables: Create variables for F, B and unknown regions,
Initialize mean values, Compute inverse covariance matrices of F and B.

STEP 3 - Compute Alpha Matte: Iterate through unknown pixels, calculate alpha using the provided equation,
update colors and alpha matte.

$$\alpha = \frac{(C - B) \,.\, (F - B)}{\|F - B\|^2}$$

STEP 4 - Compute Foreground and Background Parameters: Compute mean and covariance matrices using updated alpha matte.

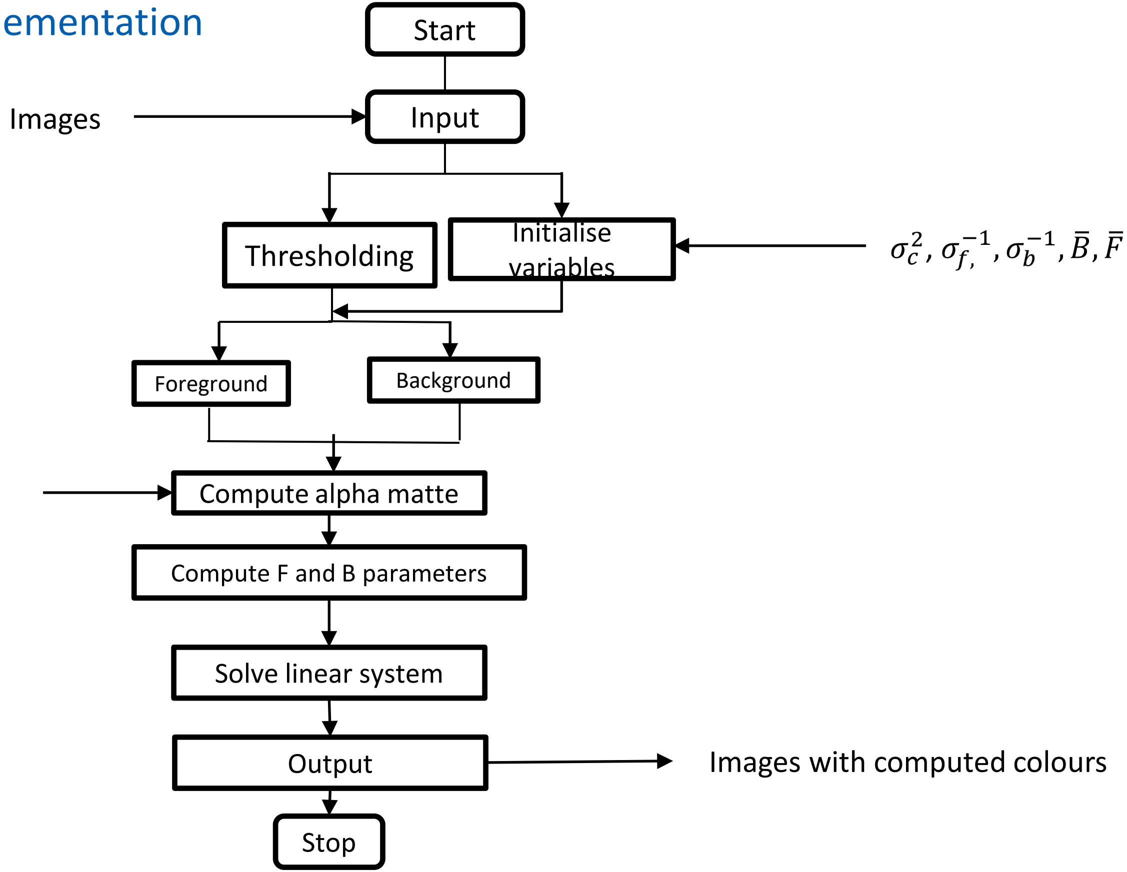
STEP 5 - Solve Linear System: Construct and solve a linear system of equations for foreground and background colors.

STEP 6 - Output: Display or save the resulting image with computed colors.

Bayesian matting

Flow diagram of implementation

$$\alpha = \frac{(C - B).(F - B)}{\|F - B\|^2}$$



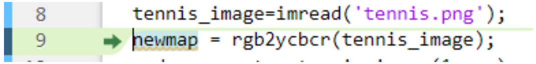
Functional Test

Guidance ISO/ IEC/IEEE 29119-3

- Objectives.

Test Methodology

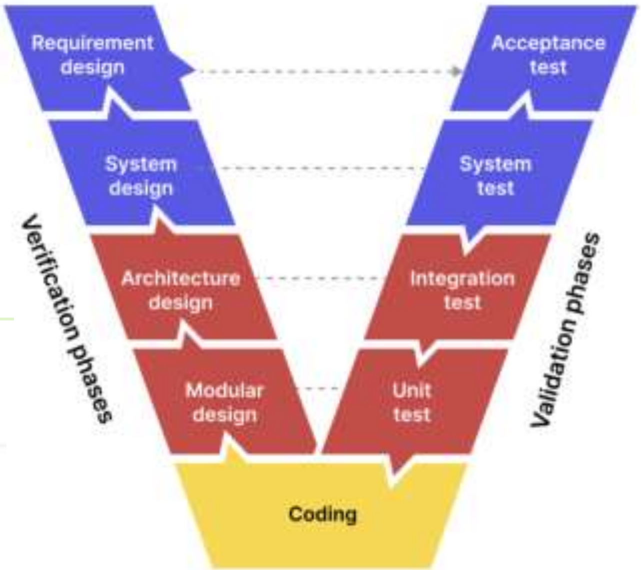
- Compare Output Against Expected Output.
- Unit Test.



- Assert Known Errors
- Sequential Execution. `print(“This is a Test")`

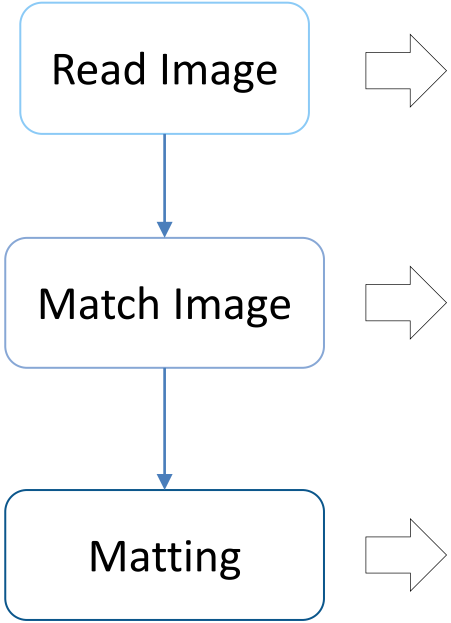
Reporting

- Risk Register.
- Test Completion Evaluation.



Unit test

matlab.unittest.TestCase



Resolution: Limitations on input images, Whether the input and output match.

Channel: 3-channel image， 1-channel trimap.

Correctness: According to the trimap, find if the images(foreground, background, unknown) are match correctly.

Position x, y: Checking that the positional values are within the resolution range.

Alpha: Checking that the Alpha within the range [0, 1].

Evaluation

SAD (Sum of Absolute Differences):

$$SAD = \sum_{i=1}^{N} |I_{\mathrm{r}} - I_{\mathrm{p}}|$$

MSE (Mean Squared Error):

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (I_{\mathrm{r}} - I_{\mathrm{p}})^2$$

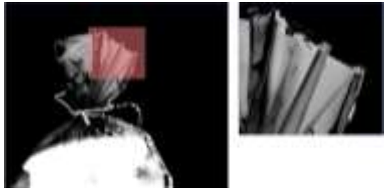
Gradient Error:

$$Grad. = \sum_{i=1}^{N} (\nabla\alpha_i - \nabla\alpha_i^*)^q$$

Connectivity Error:

$$Conn. = \sum_{i=1}^{N} (\varphi(\alpha_{\mathrm{i}} - \omega) - \varphi(\alpha_i^* - \omega))^p$$

$$\varphi(\alpha_{\mathrm{i}} - \omega) = 1 - (\lambda_i \cdot \delta(d_i \geq \theta) \cdot d_i)$$



Gradient error = 1.7

Gradient error = 3.1

Connectivity error = 4.1

Connectivity error = 2.8

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

Q & A
Thank You