CSCI 3290 Computational Photography

Assignment 3 – Image Matting

Total points: 100

Due Date: 11:59pm on Saturday, November 28th, 2015

I. Backgrounds

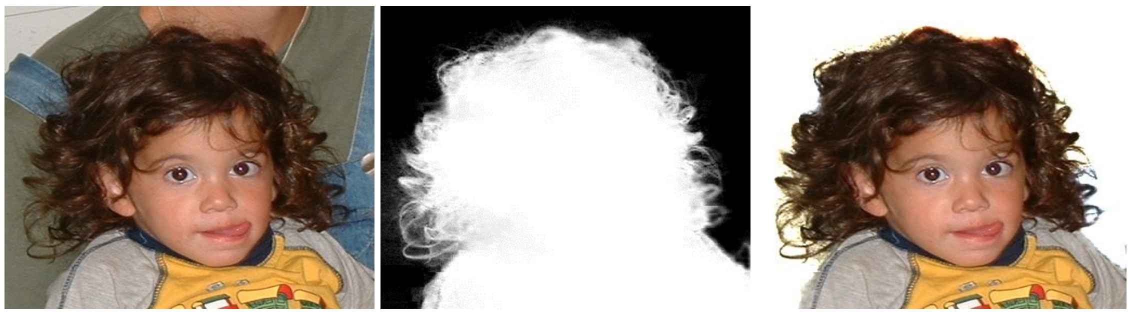


Figure 1. Image Matting

Image Matting – Matting refers to the problem of separating image into a foreground object image, a background image and an α matte. And α matte is the transparency channel of the foreground object. Matting is used to segment foreground and background in a soft way.

An example is shown in Figure 1. We use matting to segment the foreground child with fine details of hair, the α matte result is in the middle. And the segmented child is on the right.

Foreground and α can then be used to composite the foreground object into images with different backgrounds. If $\alpha_i = 1$ (or 0), then pixel i is called definitely foreground (background). Otherwise α_i indicates the confidence of pixel i being a foreground pixel.

Our task is to estimate α matte using different methods.

II. Details

In this project, we will implement different matting methods, including **Blue Screen Matting**, **Poisson Matting** and **Bayesian Matting**.

You will be given an input color image C and other additional conditions. The unknown latent foreground is denoted as F, and unknown background is as B. And we have: $C = \alpha F + (1 - \alpha)B$. You are required to implement a function to get α under different circumstances.

Part 1: Blue Screen Matting (refer to paper 'Blue Screen Matting' [1] for more details)

Blue screen matting refers to separating a foreground image from a background of constant, or almost constant color. Since background color is often blue, the problem, and its solution, is called *constant color matting* [1].

The goal is to estimate α under different assumptions.

Input is only one (or two) color image with known background color.

Solution 1: NO BLUE

Assumptions:

- Foreground contain no blue:

$$B_F = 0.$$

- Background is purely blue:

$$R_B = 0$$

$$G_B = 0$$

- So we have:

$$\mathrm{R} = \alpha R_F + (1-\alpha)R_B$$

$$\mathrm{G} = \alpha G_F + (1-\alpha)G_B$$

$$\mathrm{B} = B_B - \alpha B_B$$

Here (R_F, G_F, B_F) are red-green-blue channels of foreground image. (R_B, G_B, B_B) are 3 channel for background image. $(\mathrm{R}, \mathrm{G}, \mathrm{B})$ are 3 channels of input image. α is the transparency value for each pixel. $(\mathrm{R}, \mathrm{G}, \mathrm{B})$ are the only known values. All other variables are unknown.

So we have:

$$\alpha = 1 - \frac{B}{B_B}.$$

Hint: In the 'definite background' region, where $\alpha = 0$, $B_B = B$, you can find the value of B_B by selecting background blue-channel pixel value. Then calculate $\alpha = 1 - \frac{B}{B_B}$.

Solution 2: GREY OR FLESH

Assumptions:

- Foreground is known to be `gray', that is R_F $(or\ G_F)$ is related to B_F

$$R_F(\ or\ G_F) = a \cdot B_F + b$$

- Background is purely blue

$$R_B = 0$$

$$G_B = 0$$

- We loosen the condition to $R_F = B_F$ (**or** $G_F = B_F$). Then we have the following equations (we use $G_F = B_F$ for example here):

$$\mathrm{R} = \alpha R_F + (1-\alpha)R_B$$

$$\mathrm{G} = \alpha B_F + (1-\alpha)G_B$$

$$\mathrm{B} = \alpha B_F + (1-\alpha)B_B$$

So we can derive the formula: $\alpha = 1 - \frac{B-G}{B_B}$.

Hint: Value of B_B can be selected similar to Solution 1. For our given images, use $G_F = B_F$ may produce better results. Try and compare them in the report.

Solution 3: TRIANGULATION

Assumptions:

- Foreground is arbitrary
- Two Background with different shades are given:

$$B_{B1} = cB_B$$

$$B_{B2} = dB_B$$

- We have the following two equations ($[R_1, G_1, B_1]$ is the first input image, $[R_2, G_2, B_2]$ is the second):

$$[R_1, G_1, B_1] = [\alpha R_F, \alpha G_F, \alpha B_F + (1-\alpha)B_{B1}]$$

$$[R_2, G_2, B_2] = [\alpha R_F, \alpha G_F, \alpha B_F + (1-\alpha)B_{B2}]$$

Combining the two equations we get the following formula for α.

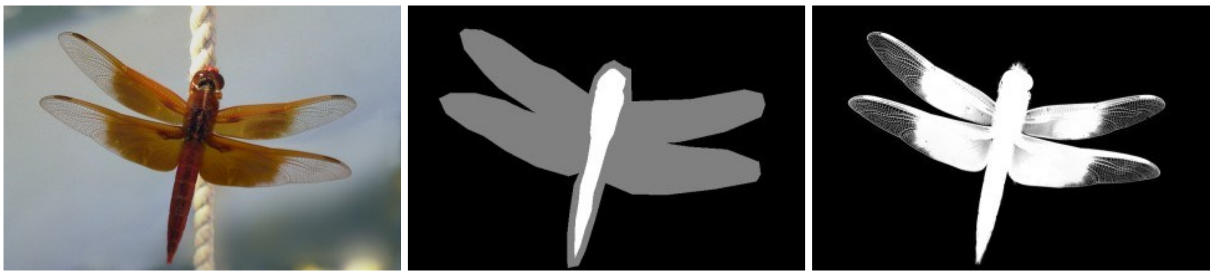
$$\alpha = 1 - \frac{B_1 - B_2}{B_{B1} - B_{B2}}$$

Hint: We can get B_{B1}, B_{B2} in the 'definite background' regions of input images.

Part 2: Poisson Matting (refer to paper 'Poisson Matting' [6] for more details)

In this method, image matting is solved via Poisson Equations.

Inputs are a color image (on the left) and a **trimap** (in the middle):



Here **trimap** is a three-value map: 0 for '**definite background**' region, 1 for '**definite foreground**' region and 0.5 for '**undecided**' regions. The final recovered alpha-matte is also shown on the right.

The original model is:

$$I = \alpha F + (1-\alpha)B$$

Where I is the input image, and F, B are foreground / background images. We take derivatives for α:

$$\nabla I = (F - B)\nabla\alpha + \alpha\nabla F + (1-\alpha)\nabla B$$

Here $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ are gradient operators for x and y directions. Therefore ∇I are x-gradient and y-gradient images of I. We assume $\alpha\nabla F + (1-\alpha)\nabla B$ is much more smaller than $(F - B)\nabla\alpha$, and get the approximation:

$$\nabla\alpha \approx \frac{1}{F - B}\nabla I$$

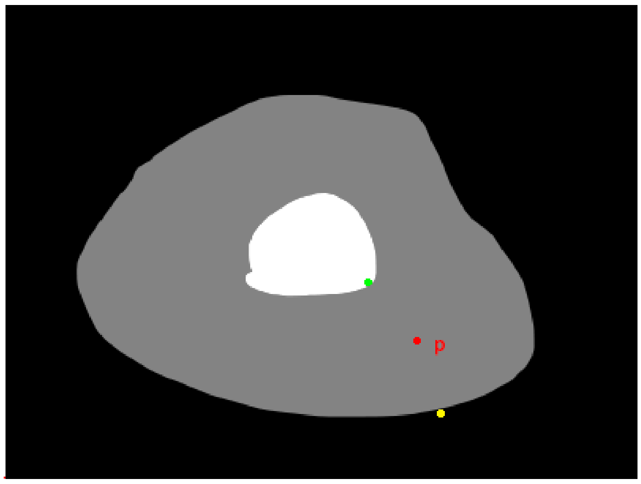
Now the problem changes to: solving for the best α with its gradients close to $\frac{1}{F-B}\nabla I$. This is exactly the definition of solving following Poisson equation:

$$\mathrm{argmin}_{\alpha} \sum_{\mathrm{p}\in\Omega} ||\nabla\alpha_p - \frac{1}{F_p - B_p}\nabla I_p||^2$$

Here Ω is the set for all 'undecided' pixels. And we define $\Omega_{\mathrm{F}}, \Omega_{\mathrm{B}}$ are sets for all pixels in foreground and background. The boundary conditions is: for boundary near Ω_{F}, $\alpha_p = 1$, and for boundary near Ω_{B}, $\alpha_p = 0$.

Step 1: (F-B) initialization

At the very beginning, we only need to know $(F_p - B_p)$ for each $p \in \Omega$. We already know: for $p \in \Omega_{\mathrm{F}}, F_p = I_p, B_p = 0$, and for $p \in \Omega_B, F_p = 0, B_p = I_p$.



Initially, for each pixel $p \in \Omega$, F_p and B_p are chosen by corresponding the nearest foreground pixel in Ω_{F} and background pixel in Ω_{B}.

For the above example of the red p in the 'undecided' region. The green pixel (in Ω_{F}) is the nearest to p, making $F_p = F_{green}$. And the yellow pixel (in Ω_{B}) is the nearest to p in Ω_{B}; therefore $B_p = B_{yellow}$.

Next, constructed $(F_p - B_p)$ image is smoothed by a Gaussian filter to suppress noise and inaccurate estimate of F_p and B_p.

Hint: For the Gaussian filter, use a small sigma and change this value in different tests to get the best results. You can start with 1, 2, or 3 (pixel values in range [0,1]), and gradually increase sigma.

Step 2: α reconstruction

With $(F_p - B_p)$ and input image, you then need to solve the Poisson equation using current $(F_p - B_p)$ and ∇I. Then you get a new α-matte image.

$$\operatorname{argmin}_{\alpha} \sum_{p \in \Omega} ||\nabla \alpha_p - \frac{1}{F_p - B_p} \nabla I_p||^2$$

We want to solve all α_p for $p \in \Omega$. Boundary conditions: for the boundary pixel in Ω_{F}, we have $\alpha_p = 1$; for the boundary pixel in Ω_{B}, we have $\alpha_p = 0$

Hint: We provide code for solving Poisson equations. Modify this code a bit to fit the above equation.

Step 3: F, B refinement

Given current updated α-matte image in Step 2, we now update F and B.

a) Update new F, B sets using the following rule. We first get two new set:

$$\Omega_F^+ = \{p \in \Omega \big| \alpha_p > 0.95, I_p \approx F_p\}$$

$$\Omega_B^+ = \{p \in \Omega | \alpha_p < 0.05, I_p \approx B_p\}$$

The conditions $\alpha_p > 0.95, I_p \approx F_p$ guarantee that pixels are mostly foreground. The same to background. Here I_p, F_p, B_p are color vectors at pixel p.

b) Update F_p, B_p for 'undecided' regions. For $p \in \Omega$, F_p equals to the color of the nearest pixels in $\Omega_F^+ \cup \Omega_F$ and B_p equals to the color of the nearest pixels $\Omega_B^+ \cup \Omega_B$. (similar to Step 1).

c) Apply Gaussian filter to smooth $(F_p - B_p)$

Go back to Step 2 and iterate Step 2 and Step 3 for a few passes until change in matting results is sufficiently small, or both Ω_F^+ and Ω_B^+ are empty.

Hint: $I_p \approx F_p$ means I_p is very close to current estimated F_p. To decide how 'close' it is, you can calculate the Euclidean distance between I_p and F_p. You may start from a small value (0.04 for example) and gradually increase it to get the best results.

Part 3: (Bonus) Bayesian Matting (refer to paper 'A Bayesian Approach to Digital Matting' [2] for more details)

Theory Part:

This method models both foreground and background color distributions as spatially varying **Gaussians**, and assumes a fractional blending of the foreground and background colors to produce the final output [2].

We define $\mathrm{F} = (R_F, G_F, B_F)$, background color $\mathrm{B} = (R_B, G_B, B_B)$ and α. The observed image $\mathrm{C} = (\mathrm{R}, \mathrm{G}, \mathrm{B})$. The target of this method is to estimate $\mathrm{F}, \mathrm{B}, \alpha$ given the observed image C.

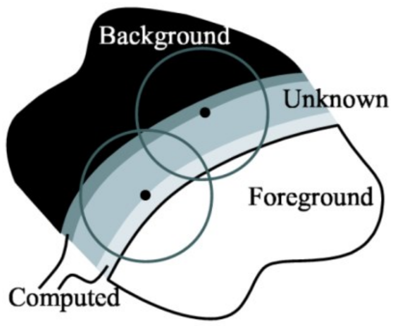


Figure 2: Three-part segmentation

This method builds F and B probability distributions from a each $N \times N$ neighborhood patch centered at each pixel. The matting problem is then formulated into Bayesian framework and solved iteratively.

In maximum a posterior (MAP) [3] estimation, we find the most likely estimates for F, B and α given the observed image C. We express the problem as

$$\begin{aligned} &\arg\max_{F,B,\alpha} P(F,B,\alpha \mid C) \\ &= \arg\max_{F,B,\alpha} P(C \mid F,B,\alpha)\, P(F)\, P(B)\, P(\alpha) \,/\, P(C) \\ &= \arg\max_{F,B,\alpha} L(C \mid F,B,\alpha) + L(F) + L(B) + L(\alpha)\,, \end{aligned}$$

Where $L(.)$ is the *log likelihood* [4]. The problem is now reduced to defining the log likelihoods function. $L(C|F,B,\alpha)$, $L(F)$, $L(B)$ and $L(\alpha)$. Figure 3 illustrates the distributions over F, B, α. We will detail the three terms in the following part：

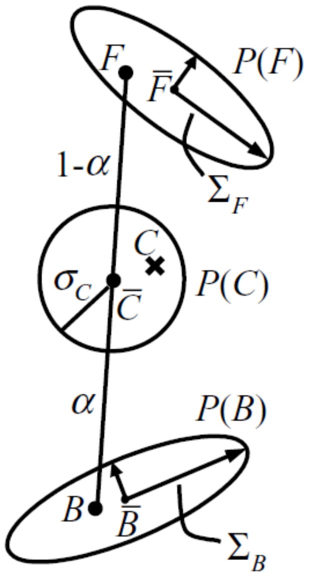


Figure 3: Gaussian distribution of F, B and C

Term 1: $L(C|F,B,\alpha)$

This models the likelihood of the unknown region pixels given the foreground model F, the background model B and the opacity α.

$$\mathrm{L(C|F,B,\alpha)} = -|C-\alpha F-(1-\alpha)B|^2/\sigma_c^2$$

This likelihood models error in the measurement of C corresponding to a Gaussian probability distribution centered at $C_0 = \alpha F + (1-\alpha)B$ with standard derivation σ_c.

Term 2. L(F)

We build the color probability distribution using the estimated colors within each pixels' N × N neighborhood. To make it robust, we weight the distribution of each nearby pixels i in N × N according to two separate factors:

1. First, we weight the pixel's contribution by α_i^2, which gives colors of more opaque pixels higher confidence.
2. Second, we use a spatial Gaussian falling off g_i with σ = 8 to stress the contribution of nearby pixels over those that are further away. That gives us $g_i = \exp(-\frac{x^2+y^2}{\sigma^2})$ where x and y is the row distance and column distance from pixel i.

Finally the combined weight $w_i = \alpha_i^2 g_i$.

Given a set of foreground colors and their corresponding weights, we first partition colors into several clusters using k-means.

You are required to implement one foreground and one background Gaussian component case.

We calculate the weighted mean:

$$\overline{F} = \frac{1}{W}\sum_{i\in N} w_i F_i$$

We calculate the weight covariance matrix:

$$\Sigma_F = \frac{1}{W}\sum_{i\in N} w_i (F_i-\overline{F})(F_i-\overline{F})^T$$

Notice:

$$W = \sum_{i\in N} w_i$$

Then the foreground likelihood is formulated as:

$$L(F) = -(F-\overline{F})^T \Sigma_F^{-1}(F-\overline{F})/2$$

Term 3: L(B)

For natural image, we use a similar term to that of the foreground setting $w_i = (1-\alpha_i)^2 g_i$. Substituting B for F in every term of the above equations leads to

$$\overline{B} = \frac{1}{W}\sum_{i\in N} W_i B_i$$

$$\Sigma_B = \frac{1}{W}\sum_{i\in N} \omega_i (B_i-\overline{B})(B_i-\overline{B})^T$$

$$L(B) = -(B-\overline{B})^T \Sigma_B^{-1}(B-\overline{B})/2$$

Optimization:

Given $L(C|F, B, \alpha), L(F), L(B)$ we solve this above maximization problem in an iterative manner. We break the above problem into two subproblems.

Subproblem1: Given α, solve F, B.

Let first order derivatives (with respect to F, B) of the overall likehood function to zero:

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

Subproblem2: Given F, B, solve α.

Now F, B are fixed, overall likehood function is a quadratic equation of α. We have closed-form solution:

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

Where the numerator is a dot product between two color difference vectors.

Implementation Details:

We solve for α in the unknown region pixels in an order from the boundary to the center of the unknown region.

- You can use a queue to keep all boundary pixels.
- Also you need to construct a binary image Q where pixel p is with $Q_p = 1$ when α_p have been computed. Initially, $Q_p = 1$ for $p \in \Omega_F \cup \Omega_B$.
- Fetch a pixel q from the boundary queue, process it, and remove it from the queue. Also add its neighboring pixels, which become new boundary pixels, into the queue. Set $Q_q = 1$.
- Iterate until all pixels are processed.

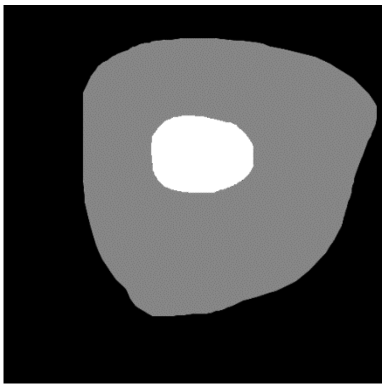


Figure 3(a): Original trimap

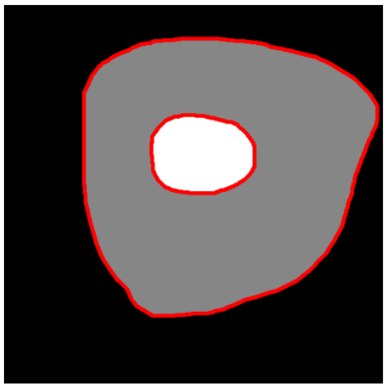


Figure 3(b): Iteration 1

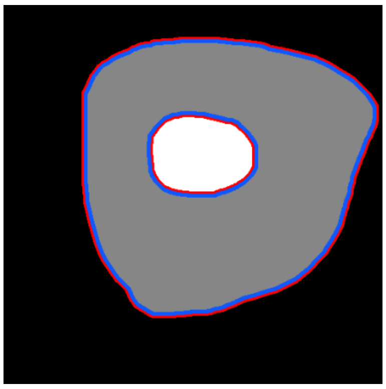


Figure 3(c): Iteration 2

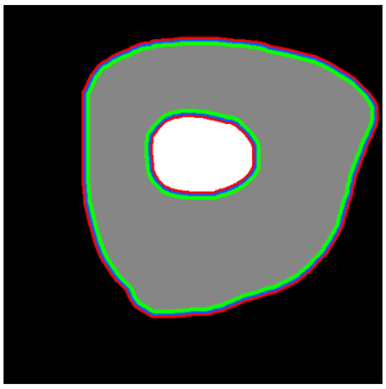


Figure 3(d): Iteration 3

Figure3: (a) ‘definite foreground’ (**white**), ‘definite background’ (**black**) and ‘undecided’ (**gray**). (b) The queue only contains boundary of the unknown part - the red circles in Figure 3(b). After the red-circle pixels are processed, inner boundaries shown in blue are in the queue, as shown in 3(c). 3(d) shows the green pixels which are in the third inner layer.

Step 1: Initialize α

The α value of the foreground region is set to be 1 and the α value of the background region is set to be 0.

In the ‘undecided’ region, only the boundary need to be initialized (red pixels in Figure3(b)). For each undecided α_p, we calculate the average value for all ‘decided’ α_j (pixel j is in the $\mathbf{n} \times \mathbf{n}$ neighbor region centered at pixel p, and α_j is either given or has been decided).

P is initialized with the boundary location of the unknown region (red pixel locations in Figure 3(b)).

Hint: typically, n is set to be **3**.

Step 2:Given α*, solve* F, B

Solve sub-problem 1 with the fixed α value.

Hint: typically, $\sigma_c = 5, \sigma = 8$ for image pixel values in range [0,255]. $N = 24$ for selecting around 500 nearest pixels. They need to be tuned to get better results.

Step 3:Given F, B*, solve* α

Solve sub-problem 2 to get α values at location in list P , with fixed F, B.

Step 4:Update matrix Q and list P

Repeatedly solving step2 and step 3 until convergence at locations maintained in P. Conventionally, 3 iterations are enough.

Now locations in P have been solved. We set $Q_p = 1$. And P is updated to the next boundary location.

Part 4: (Bonus part) Bayesian Matting with more than one foreground and background

In part 3, you are required to implement an algorithm to solve the Bayesian matting problem with only one foreground and background Gaussian cluster. But actually, natural images are very complex and thus cannot be simply modeled with one Gaussian model. We can resort to the mixture models instead. This means we have to reformulate L(F) and L(B). Following paper [2]. We first use k-means [5] , to pre-cluster the colors into different clusters. For each cluster, calculate the weighted mean and weighted covariance matrix as the above section for the foreground and background model. Then select the foreground and background pair which gives the maximum likelihood. The process is very similar to that of part 3.

Notice: this model, in contrast to the mixture Gaussian model, assumes that the observed color corresponds to exactly one pair of foreground and background distributions.

Hint: typically, k=3 is enough in most cases.

Packages we provide:

1. Specification of Assignment 3
2. MATLAB skeleton code: please refer to the package
3. Sample images for testing: input images needed for different part.

III. Requirements

Your submission should contain the following.

a) Codes including:

- `\blueScreen_matting\run.m` – To start your algorithm
- `\blueScreen_matting\noblue.m` – Solution 1
- `\blueScreen_matting\graymatt.m` – Solution 2
- `\blueScreen_matting\triangulationmatt.m` – Solution 3

- `\poisson_matting\run_pmating.m` - To start your algorithm
- `\poisson_matting\find_nearestFB.m` - To find nearest pixel in F/B
- `\poisson_matting\poisson_equ.m` - To solve Poisson equation

- `\bayesian_matting\run.m` - To start your algorithm
- `\bayesian_matting\bayesian_matting.m` - Solve using Bayesian matting

- Other codes you would like to add.

Remarks:

- You can use functions provided by MATLAB, its Toolboxes and libraries supplied by TAs (Unless identified elsewhere).

- Note you need to describe the algorithms in your **OWN** words in the report. If you use others' functions or codes, please clearly claim all of them in the report, and **SUBMIT** them together with your assignment. Otherwise, it might be considered as plagiarism.
- Your own work **MUST** be the majority of all your uploaded codes.

b) Reports:

- Make a report in **HTML**.
- **Describe** your algorithm and any decisions you made to write your algorithm a particular way. Show and discuss the **results** of your algorithm.
- Discussion on algorithms' **efficiency**.
- Highlight any **extra credit** you did.
- Ideas and algorithms from others **MUST** be clearly claimed. List papers or links as reference if needed.
- Feel free to add any other information you feel is relevant.

c) Hand-in

Create a folder '**<** ***your student ID*** **>-Asgn3**'. The folder must contain the following:

- **README.txt** - anything about the project that you want to tell the TAs, including brief introduction of the usage of the codes
- **code/** - directory containing all your code for this assignment
- **html/** - directory containing all your html report for this assignment (including images). Your web page should only display compressed images (e.g. **jpg** or **png**).
- **html/index.html** - home page for your results

Compressed the folder into <*your student ID*>**-Asgn3.zip** or <*your student ID*>**-Asgn3.rar**, and upload it to e-Learning system.

IV. Scoring & Extra Bonus

a) Score

- +40 pts: Blue screen matting in Part 1, at least 3 results each solution. (10 pts for each solution, and the remaining 10pts for results)
- +20 pts: Poisson matting in Part 2. (at least 2 successful results in report)
- +20 pts: Report
- +20 pts: Extra credits
- -5*n pts: Losing 5 points every time (after the first) you do not follow the instructions to prepare the uploading file format

b) Extra Credit

Options for extra credit:

- +20 pts: Implement Bayesian matting in Part 3.
- +10 pts: Extended version of Bayesian matting in Part 4.
- +10 pts: Use your own photos in Blue Screen Matting. (at least 2 cases)
- +10 pts: Use your own photos with your manually labelled trimap in Poison Matting. (at least 2 cases)
- +10 pts: Try your matting methods on benchmark website AlphaMatting and report your results using their metric. (at least 2 cases) http://www.alphamatting.com/eval_25.php

V. Other Remarks

- **Five** late days total, to be spent wisely
- 20% off from each extra late day
- The three assignments are to be completed **INDIVIDUALLY** on your own.
- You are encouraged to use methods in related academic papers.
- Absolutely **NO sharing or copying** code! **NO sharing or copying** reports! Offender will be given a failure grade and the case will be reported to the faculty.

VI. Reference

[1] http://research.microsoft.com/pubs/73833/p259-smith.pdf

[2] http://grail.cs.washington.edu/projects/digital-matting/papers/cvpr2001.pdf

[3] https://en.wikipedia.org/wiki/Maximum_a_posteriori_estimation

[4]https://en.wikipedia.org/wiki/Likelihood_function

[5] https://en.wikipedia.org/wiki/K-means_clustering

[6] http://www.cse.cuhk.edu.hk/leojia/all_project_webpages/Poisson%20matting/poisson_matting.html