Index

1. Abstract ......................................................................................................................... 3
2. Motivation and Introduction ......................................................................................... 4
   2-1. Motivation ................................................................................................................... 4
   2-2. Introduction ................................................................................................................ 4
       2-2-1. System Architecture ............................................................................................ 5
       2-2-2. Image Operators in Fantasy Illustrator ............................................................... 6
3. Method/Implementation ................................................................................................. 10
   3-1. Seam Carving for Content-aware Image Resizing (SIGGRAPH 2007) ................. 10
       3-1-1. Introduction of the Paper ..................................................................................... 10
       3-1-2. Method ................................................................................................................ 10
       3-1-3. Result ................................................................................................................... 12
   3-2. Poisson Image Editing (SIGGRAPH 2003) ............................................................... 14
       3-2-1. Introduction of the Paper ..................................................................................... 14
       3-2-2. Method ................................................................................................................ 15
       3-2-3. Result ................................................................................................................... 16
   3-3. Coordinates for Instant Image Cloning (SIGGRAPH 2009) .................................. 17
       3-3-1. Introduction of the Paper ..................................................................................... 17
       3-3-2. Method ................................................................................................................ 18
       3-3-3. Result ................................................................................................................... 21
   3-4. Object Removal by Exemplar-Based Inpainting (CVPR 2003) ......................... 23
       3-4-1. Introduction of the Paper ..................................................................................... 23
       3-4-2. Method ................................................................................................................ 24
       3-4-3. Result ................................................................................................................... 27
   3-5. Frequency-tuned Salient Region Detection (CVPR 2009) .................................... 29
1. Abstract

In this project, we propose a user-friendly image editing system, Fantasy Illustrator, in order to help people create a new work arbitrary by manipulating images from any collections user desired. In contrast with sophisticated operators Adobe Photoshop provided, our system classifies image editing operators into two critical operations: geometric based operation and tone based operation. There are five operators we implemented totally: image resizing, image cloning and image inpainting for geometric based operation; color transfer, and tone transfer for tone based operation. For an arbitrary image, user can modify it through geometric operators and tone operators, respectively. Moreover, we also provide simple and intuitive user interface with one-step operation to make user edit image to any aesthetic or mood they want easily.
2. Motivation and Introduction

2-1. Motivation

As the popularity of digital camera, digital images become part of our daily life. We create and interact with image all the time, and for various kind purposes. We take a picture, draw a sketch, recode and share life experience to friends with photo, or even take digital image as the material to create artwork. It seems digital image become more and more important than before. However, modifying or enhancing digital image content is difficult for most people. Users are required to have enough background knowledge and need tedious steps to obtain the ideal result. Even though famous image editing tools such as Adobe Photoshop, Adobe Lightroom, Apple Aperture provide integrated tools and allow precise and flexible control over the digital image, but their sophisticated user interface is too complicated to use. Consider increasingly frequent usage of digital images, reducing the user work is critical. That inspired us to propose an intuitive and powerful image editing system to help people edit Images more easily.

2-2. Introduction

In this project, we propose a simple, intuitive and user-friendly image editing system, Fantasy Illustrator, which provides several useful and powerful image editing operators and allow users of all skill level to easily modify image to certain aesthetic or style they want. As shown in Figure 2-2.1 our system, Fantasy Illustrator, work with familiar but more simple GUI than Adobe Photoshop. We simplify the process of editing image content. It presents a shallow learning curve to users of all skill levels. Moreover, our editing operators work in real time. It is really useful for user to edit image interactively.
Figure 2-2.1: The graphical user interface of Fantasy Illustrator. There are totally five operators in the top toolbar: three for geometric arrangement and two for tone based operation, respectively. And show the editing process in the main window.

2-2-1. System Architecture

Our work is related to image editing which has attracted great attention in recent Digital Image Processing. This technique take concern either global changes or local changes confined to a select region through the corrections, filters, deformations for geometry or color/intensity of image content [Patrick Pérez et al. 2003]. Based on different purposes, image editing operators can be broadly classified into many categories. However, there are two critical operators, i.e. geometric operators, such as geometric rearrangement and perspective correction, and tone operators, which includes color adjustments and contrast manipulation.

Due to above reason, our system classify editing operators into 2 types: Geometry and Tone operators; and implement several papers for each type operator, respectively. Our system architecture is shown in Figure 2-2.2.
As the figure 2-2.2 show, user first input any image. Then users can modify images arbitrarily through geometric based operators which include image resizing, image cloning and image inpainting, and tone based operators which include color transformation, and tone transformation.

2-2-2. Image Operators in Fantasy Illustrator

The goal of this project is to propose an interactive image editing system. We intend to allow users editing geometry and tone components in images. With the former operation, users can move objects into the desired image, non-homogeneously change the image resolution with minimum distortion of its nature structure or simple remove a object from image by using image cloning operator, image resizing operator and image inpainting operator, respectively; with latter operation, users are able either to transfer the color by or tone style of image by using color transfer and or transfer operator. So, in this project, we implemented several papers as show following.

**Geometric Image Editing**

(a) Image Resizing:
- Seam Carving for Content-aware Image Resizing (SIGGRAPH 2007)

(b) Image Cloning:
- Poisson Image Editing (SIGGRAPH 2003)
- Coordinates for Instant Image Cloning (SIGGRAPH 2009)

(c) Image Inpainting:
- Object Removal by Exemplar-Based Inpainting (CVPR 2003)

(d) Saliency Detection:
- Frequency-tuned Salient Region Detection (CVPR 2009)

**Color/Tone Image Editing**

(a) Color Transfer:

(b) Tone Transfer:
- Two-scale Tone Management for Photographic Look (SIGGRAPH 2006)

**User Interface**

Considering the simplicity and familiarity, we design the system user interface as traditional window layout. User can choose the interesting operation easily through click correspond button in top tool bar, then see the operation process and final result in the main window. Moreover, we optimize complex and tedious editing steps to almost full automatic one step operation and provide a tailor-made editing panel for each operator. The following figures are the print screen shot for our five operators.

![Image Resizing](image.png)

Figure 2-2.3: The user interface panel for image resizing operator
Figure 2-2.4: The user interface panel for image cloning operator

Figure 2-2.5: The user interface panel for image inpainting operator

Figure 2-2.6: The user interface panel for color transfer operator
In the following section, first, we will introduce the algorithms detail of our five main operators and system (Section 3). Demonstrate the experimental result of Fantasy Illustrator in Section 4. Finally, we will discuss more implement issues, result performance and limitation in Section 5 and display the final result.
3. Method/Implementation

3-1. Seam Carving for Content-aware Image Resizing (SIGGRAPH 2007)

3-1-1. Introduction of the Paper

Image resizing operation is very common in everyday tasks. Traditional homogeneous image resizing doesn’t consider image content, therefore it produces deformed result. This reference paper proposes a novel method called “seam carving” to resize an image according to the content in the image. By computing an energy function for each pixel, we can find an 8-connected path of pixel with the lowest energy from left to right, or from top to bottom. In other words, a seam represents the least noticeable path in the image. When repeating removing seams of the same direction, we can perform image resizing that minimizes the deformation or distortion. Figure 3-1.1 describes the concept of seam carving.

![Figure 3-1.1: Left: A vertical seam and a horizontal seam, each represents a connected path in the image. Right: By removing or adding a seam into the image, we can achieve content-aware image resizing.](image)

3-1-2. Method

Seam carving finds unnoticeable pixels and removes them to achieve content-aware image resizing. Thus the most important thing is to define which pixel is noticeable or unnoticeable. This reference paper uses gradient magnitude to compute energy of each pixel. The energy of each pixel is defined by the authors as following equation:

\[ E(I) = \left| \frac{\partial}{\partial x} I \right| + \left| \frac{\partial}{\partial y} I \right| \]

Let I be an \( n \times m \) image, we can define a vertical seam to be \( s^x \) and horizontal
seam to be \( s^x \). \( s^x \) and \( s^y \) are represented as follows:

\[
\begin{align*}
    s^x &= \{s^x_{i,j}\}_{j=1}^m = \{x(i), y(i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1 \\
    s^y &= \{s^y_{j,i}\}_{i=1}^n = \{y(j), x(j)\}_{j=1}^m, \text{ s.t. } \forall j, |y(j) - y(j-1)| \leq 1
\end{align*}
\]

When removing a vertical seam from an image, the remaining pixels which are at the right side of the seam are shifted left. Similarly, removing a horizontal seam from an image, the remaining pixels are shifted up. This causes the compensation for the missing path. Given the energy function \( E \) of each pixel, the energy of a seam is defined as

\[
E_s = \sum_{i=1}^n E(I(s_i))
\]

When resizing an image, we iteratively find the optimal seam which minimizes the cost function and remove it from the image. This can be represented as

\[
\text{arg min } E_s = \text{arg min } \sum_{i=1}^n E(I(s_i))
\]

We use dynamic programming to find the optimal seam. The method traverses an image from top to bottom, and computes the cumulative minimum energy \( M \) for all pixels:

\[
M(i, j) = E(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i, j-1, j+1))
\]

Thus the cost value of each seam can be found by \( M \) at the last row of the image. The overall algorithm can be written as follows:

---

**Algorithm: Seam Carving**

1: \textbf{for each} pixel \( p \in I \) \textbf{do}
2: \hspace{1em} Compute the energy \( E(I_p) \).
3: \textbf{end for}
4: \textbf{for each} pixel \( p \in I \) \textbf{do}
5: \hspace{1em} Compute the cumulated energy \( M(I_p) \).
6: \textbf{end for}
7: Sort all the seams \( S \) according to the cumulated energy of the last row pixels \( M(I_p) \).
8: \textbf{while} resolution of the image \( WH \) is larger than the target resolution \( WH' \) \textbf{do}
9: \hspace{1em} Remove a seam with the optimal energy and reduce the resolution by 1.
10: \textbf{end while}

---
3-1-3. Result

Figure 3-1.2 shows the seam analysis result of our test images. Each red line represents a seam in the image. When resizing these images, the algorithm removes these seams in order.

![Source Image](image1)

Source Image

![Vertical Seams](image2)

Vertical Seams

![Horizontal Seams](image3)

Horizontal Seams

Figure 3-1.2: Vertical and horizontal seams in our test images.

Figure 3-1.3 shows a number of Seam Carving results generated by our system. The results preserve content in the images when applying image resizing.
Figure 3-1.3: Horizontal image resizing using seam carving. Seam Carving images (1) to (3) refer to an increase of number of seams be removed.
3-2. Poisson Image Editing (SIGGRAPH 2003)

3-2-1. Introduction of the Paper

When editing an image, users may want to cut an object in an image and paste it into another image. Traditionally, this operation causes obvious tone and hue difference between pasted area and surrounding area (Figure 3-2.1). The result doesn’t satisfy us because it makes the image looks unnatural. The paper presents a generic interpolation method. By solving Poisson equation, we can achieve seamless cloning, which means the texture, the illumination, and the color of objects are modified so that it looks more natural in the target area (Figure 3-2.2).

![Source Image | Target Image | Result Image](image1)

Figure 3-2.1: Cut and paste the objects into another image.

![Source Image | Target Image | Seamless Clone](image2)

Figure 3-2.2: Seamless clone the objects into another image.
3-2-2. Method

The algorithm uses a guidance vector field to do interpolation. Figure 3-2.3 shows the concept of seamless image cloning. Let $S$ be the image domain, and let $\Omega$ be the closed unknown area with the unknown scalar function $f$. Boundary of the unknown area is noted as $\partial \Omega$. The scalar function of known area is defined as $f^*$. The vector field $v$ is defined over $\Omega$. With these definition, we try to interpolate $f$ from $f^*$. It can be defined as the solution of the following problem:

$$\min_{f} \int_{\Omega} |\nabla f - v|^2 \quad \text{with} \quad f|_{\partial \Omega} = f^*|_{\partial \Omega}$$

where $\nabla = \left[ \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right]$. The optimal solution of the minimization problem is the solution of Poisson Equation with Dirichlet boundary conditions:

$$\Delta f = \text{div} v \quad \text{over} \quad \Omega, \quad \text{with} \quad f|_{\partial \Omega} = f^*|_{\partial \Omega}$$

where $\text{div} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ is the divergence of the vector field.

This is the method of Poisson editing of images. If we want to deal with a color image, three Poisson equations of three channels RGB are solved independently. The above equation is the form of continuous domain. We transform the minimization problem into discrete domain. For an pixel, $\Delta f$ and $\text{div} v$ can be computed using the following equations:

$$\Delta f \equiv (f_{x+1,y} - 2f_{x,y} + f_{x-1,y}) + (f_{x,y+1} - 2f_{x,y} + f_{x,y-1})$$

$$= f_{x+1,y} + f_{x-1,y} + f_{x,y+1} + f_{x,y-1} - 4f_{x,y}$$

$$\text{div} v \equiv G_x(x,y) - G_x(x-1,y) + G_y(x,y) - G_y(x,y-1)$$
and remember we want to solve $\Delta f = \text{div}\mathbf{v}$. Thus we have to solve a very large sparse linear system $AX = B$. The dimension of matrix $A$ is $NM \times NM$. This is represented as follows:

$$
\begin{vmatrix}
1 & 1 & -4 & 1 & 1 \\
1 & 1 & -4 & 1 & 1 \\
1 & 1 & -4 & 1 & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\end{vmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
b_1 \\
b_2 \\
\vdots \\
\end{bmatrix}
$$

Finally, matrix $X$ represent the solution of Poisson image editing.

### 3-2-3. Result

![Object Image](image1)
![Target Image](image2)
![Result Image](image3)

Figure 3-2.4: Results of Poisson image cloning.

Figure 3-2.4 shows the result of Poisson seamless image cloning. With the technique of solving Poisson equation, the texture and color of object area is modified according to the differential of gradient information of the target image. This make the image look more natural.
3-3. Coordinates for Instant Image Cloning (SIGGRAPH 2009)

3-3-1. Introduction of the Paper

Seamless cloning, which allows user to insert a source image patch into a target image without aliasing, is a useful and fundamental technique in image editing. As the popularity of digital image, it has received significant attention in recent years. The typical method is to solve a Dirichlet boundary condition by Poisson equation, which smoothly interpolates the disparity between the source boundary and the target. The state of the art can obtain convincing result; however, needs a considerable time to solve a large linear interpolation system. Therefore, this paper proposes a novel and alternative approach based on Mean-Value Coordinates (MVC). Rather than solving the time-consuming system, the method uses the weighted combination of source boundary pixels to interpolate the interior area. The coordinate-based method has the advantages of fast speed, small memory footprint and ease of parallelizability. The efficient work can achieve the real-time cloning over large regions; meanwhile, obtains the satisfying result almost the same as Poisson cloning, as shown in Figure 3-3.1.

![Figure 3-3.1: Seamless cloning of source image patch (a) into the target image (b) by two cloning methods, (c) Poisson cloning and (d) MVC cloning via transfinite interpolation, without solving a](image-url)
large linear linear system.

3-3-2. Method

The algorithm includes three important parts, the main Mean-Value Coordinates cloning and two optimizations, which are described below.

(1) Mean-Value Seamless Cloning

In Poisson cloning, proved by Perez et al. [2003], solving the Poisson equation is equal to solving the Laplace equation with the Dirichlet boundary condition on the boundary discrepancies between the source and the target. To achieve it, Poisson cloning constructs a harmonic (or membrane) interpolant to smoothly spread the differences along the boundary into the inward area. Therefore, the key purpose of this paper is to find a harmonic-like interpolant directly without solving the linear system.

To produce such interpolant, some researches in the transfinite interpolant fields have proposed the solutions based on generalized barycentric coordinates. One important method is Floater’s Mean-Value Coordinates (MVC) [Floater 2003], which use a simple closed-form formula to imitates the mean-value properties of harmonic functions. Due to its nice properties and speed, the literature uses MVC to replace the solving of Poisson equation, and the designed algorithm is as follows.

First, the user chooses the source image patch $P_s \subset S$ from the source image $S$ and decide the target region $P_t \subset T$ on the target image $T$. Their boundaries are $\partial P_s$ and $\partial P_t$, and $g : S \to R, f^* : T \to R$ would be their corresponding image intensities. Then, the target function is $f : P_t \to R$, and the Laplace equation transferred from Poisson equation would be:

\[ \Delta \tilde{f} = 0, \text{ w/dirichlet boundary conditions } \tilde{f}_{\partial P_t} = f^* - g. \]

The final cloning outcome is $f = g + \tilde{f}$. 
Next, we can use MVC to construct the similar smooth interpolant \( r \) to alternate the original harmonic function \( \tilde{f} \). Consider a point \( x \in P \) with the 2D polygonal boundary curve (with counter-clockwise order) \( \partial P = (p_0, p_1, \ldots, p_m = p_0), p_i \in \mathbb{R}^2 \). Its mean-value coordinates are given by

\[
\lambda_i(x) = \frac{w_i}{\sum_{j=0}^{m-1} w_j}, i = 0, 1, \ldots, m - 1,
\]

\[
w_i = \frac{\tan(\alpha_{i-1} / 2) + \tan(\alpha_i / 2)}{\|p_i - x\|}
\]

where \( \alpha_i \) is the angle of \( \angle p_i, x, p_{i+1} \), as shown in Figure 3-3.2.

![Figure 3-3.2: The illustration of angle for mean-value coordinates.](image)

Then, the mean-value interpolant at point \( x \) is

\[
r(x) = \sum_{i=0}^{m-1} \lambda_i(x)(f^* - g)(p_i)
\]

And the cloning result is given by

\[
f = g + r
\]

This unoptimized procedure is shown in Algorithm 1 (provided by the authors).
Since the interpolant become smoother away from the boundary, lesser sample points are actually necessary toward the center. Therefore, we can use the CGAL [Cgal 2007] library to generate an adaptive mesh over the source patch $P_s$. As shown in Figure 3-3.3, the adaptive mesh uses more triangles near the boundary. Once the mesh is available, only the mesh vertices need to be computed for their mean-value coordinates and interpolant result. The other pixels inside a triangle will be obtained by linear interpolations of the three vertices. This optimization step can reduces great computation time on line 4 and 13 in Algorithm 1.

(2) Adaptive Mesh

Figure 3-3.3: The adaptive mesh constructed over the cloned region. The red dots on the boundary are the boundary points selected by adaptive hierarchical boundary sampling for the mesh vertex marked in blue.
(3) Hierarchical Boundary Sampling

Instead of using all the boundary points, a further speedup is achieved by hierarchically sampling on the boundary. The idea is inspired by previous researches on adaptive hierarchical, such as [Carrier et al. 1988] and [Hanrahan et al. 1991]. Since the mean-value weights decrease quickly with distance, an approximation can be obtained by sampling the boundary with the density inversely proportion to distance, as shown in Figure 3.2.

In the building phase, we first set the sequence of all boundary points as the finest level. Then, each coarser level only preserves half points uniformly from the previous finer level. The process stops when the coarsest level contains points less than a predefined constant.

Next, in the traverse phase, the process is started from the top (coarsest) level down. When a mesh vertex \( x \) traverse to the \( i \)th point in the \( k \)th level, three conditions will be examined:

\[
\begin{align*}
&\|x - p_i^k\| > \varepsilon_{\text{dist}} \\
&\angle p_{i-s}^k, x, p_i^k < \varepsilon_{\text{ang}} \\
&\angle p_i^k, x, p_{i+s}^k < \varepsilon_{\text{ang}}
\end{align*}
\]

where \( s \) is the index step between successive points at that level. If all three conditions hold, that means the point \( p_i^k \) is sufficient representative for \( x \) and no further refinement is necessary around \( p_i^k \). Otherwise, denser sampling is required to obtain better approximation. Hence, three points \( p_{i-s/2}^{k+1} \), \( p_i^{k+1} \) and \( p_{i+s/2}^{k+1} \) in the finer level will be traversed.

This optimization can significantly reduce the number of used boundary points, and save considerable computation time on line 4 and 13 in Algorithm 1. Besides, due to the adaptive properties, it can preserve the visual quality well also.

3-3-3. Result

Figure 3.4 shows the result of MVC image cloning. Due to the nice properties of MVC, it can achieve similar and convincing results as Poisson cloning. Besides,
since we have implemented both optimizations, as shown in Figure 3-3.4 (b), we can obtain our results almost real-time (a minor delay may be influenced by the original structure of GUI software).

(a) Target image

(b) Source patch with triangular mesh

(c) Result image

Figure 3-3.4: MVC results.
3-4. Object Removal by Exemplar-Based Inpainting (CVPR 2003)

Also published in:
Region Filling and Object Removal by Exemplar-Based Image Inpainting (IEEE Transactions on Image Processing 2004)

3-4-1. Introduction of the Paper

Image inpainting, as an important operator in image editing, helps user to remove objects from an image and fills the holes in a nature way. The size of object and the plausibility of filling are the main challenges. Typically, the problem can be solved by two classes of algorithm: (i) “texture synthesis”, which analyze the “texture” – repeated two-dimensional patterns and generate large regions by sample textures, and (ii) “inpainting” techniques, which focus on the linear “structures” and fill small image gaps. The linear structures are some one-dimensional patterns, such as lines and object contours.

The paper proposes a novel exemplar-based method which combines the strengths of both approaches into a single, efficient algorithm. The exemplar-based texture synthesis contains both essential texture and structure information; however, is highly depended on the filling order. Thus, the literature introduces a best-first algorithm which uses the priority term involving both kinds of information. This algorithm is effective for removing both large objects and thin scratches, as shown in Figure 3-4.1, and can achieve the efficiency by the block-based sampling method.
Figure 3-4.1 Apply the proposed method on removing a large object from (a) and achieve convincing result (b).

3-4-2. Method

We followed the paper’s algorithm described below.

First, the user selects the target region $\Omega$ to be removed, and the rest areas of the entire image will be the source region $\Phi = I - \Omega$. Next, as all exemplar-based texture synthesis, user needs to specify a proper size of the template window $\Psi$, which is suggested to be slightly larger than the largest texture pattern in the source image.

Then, in the initialization, due to the property of perceptual uniformity, we transfer the color space into CIELab (the unfilled pixels is defined as “empty”), and calculate the confidence value of each pixel, which represents the confidence in that pixel. The confidence value $C(p)$ of a point $p$ is set to be 0, $\forall p \in \Omega$, and 1, $\forall p \in I - \Omega$. After the setting, the algorithm iterates the following three steps until filling all the pixels.

1) Compute Patch Priorities

Before filling, we needs to select a filling patch for this iteration. Hence, we
first compute the priority of each patch on the boundary of removed region, also referred as the “fill front” $\partial \Omega$. Then, the patch with highest priority is chosen by the best-first strategy.

The priority $P(p)$ of a patch $\Psi_p$ centred at point $p \in \partial \Omega$ is defined as the product of the confidence term $C(p)$ and the data term $D(p)$:

$$P(p) = C(p) \cdot D(p),$$

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I - \Omega)} C(q)}{|\Psi_p|}, \quad D(p) = \frac{||\nabla I_p^\perp \cdot n_p||}{\alpha}$$

where $|\Psi_p|$ is the area of patch, $\alpha$ is the normalization factor, $n_p$ is a unit norm vector to the fill front on $p$ and $\nabla I_p^\perp$ is the orthogonal vector to the gradient of $p$. The corresponding illustration is shown in Figure 3-4.2.

The confidence term may be thought as the measurement of the reliability surrounding $p$. The points with more pixels from source region will be filled earlier, and vice versa. Therefore, this term can approximately enforce the desired filling order, from out layer inward into the centre.

The data term is an estimation of the strength of isophotes hitting the fill front $\partial \Omega$ on point $p$. In other words, this factor represents the gradient at $p$ (information of linear structure) and whether the gradient is orthogonal to the front at $p$. This term is the fundamental of the algorithm since it can encourage the linear structures, like lines, to be synthesized towards the “right” direction first. Therefore, the broken lines and contours can tend to connect, and the structure of image can be maintained.
Figure 3-4.2: Illustration of the priority evaluation. For a target region $\Omega$ on image $I$, the patch $\Psi_p$ is centred at point $p \in \partial \Omega$. $n_p$ is the norm of the fill front $\partial \Omega$ on $p$, and $\nabla I_p^\perp$ is the isophote (direction and intensity) at $p$.

(2) Propagating Texture and Structure Information

Once the patch $\Psi_{\hat{p}}$ with highest priority is found, we can use some suitable data from source region to fill it. Unlike the traditional diffusion method, which leads to blur especially for large removed region, we propagate image texture by filling the patch with pixels from source region. Therefore, we search the whole source region for the patch most similar to $\Psi_{\hat{p}}$ by

$$\Psi_{\hat{q}} = \arg \min_{\Psi_q \in \Phi} d(\Psi_{\hat{p}}, \Psi_q),$$

where the distance $d$ between two patches is defined as the sum of squared differences (SSD) over the already filled points.

After found the most similar patch, the source exemplar $\Psi_{\hat{q}}$, the value of each unfilled pixel $p' \mid p' \in \Psi_{\hat{p}} \cap \Omega$ is copied directly from the corresponding pixel in $\Psi_{\hat{q}}$.

(3) Updating confidence values
After filling the patch $\Psi_{\hat{p}}$, we should update the confidence value $C(p)$ in it as follows:

$$C(p) = C(\hat{p}) \quad \forall p \in \Phi_{\hat{p}} \cap \Omega.$$  

The update of decayed confidence can achieve our desired filling order, from the boundary towards centre.

Algorithm 2 shows the pseudo-code of the algorithm, where the superscript $t$ represents current iteration.

**Algorithm 2. Inpainting algorithm.**

- Extract the manually selected initial front $\delta \Omega^0$.
- Repeat until done:
  1a. Identify the fill front $\delta \Omega^t$. If $\Omega^t = \emptyset$, exit.
  1b. Compute priorities $P(p) \quad \forall p \in \delta \Omega^t$.
  2a. Find the patch $\Psi_p$ with the maximum priority, i.e., $\hat{p} = \arg \max_{p \in \delta \Omega^t} P(p)$.
  2b. Find the exemplar $\Psi_{\hat{q}} \in \Phi$ that minimizes $d(\Psi_p, \Psi_{\hat{q}})$.
  2c. Copy image data from $\Psi_{\hat{q}}$ to $\Psi_p \quad \forall p \in \Psi_p \cap \Omega$.
  3. Update $C(p) \quad \forall p \in \Psi_p \cap \Omega$

3-4-3. Result

Figure 3-4.3 shows the inpainting results of the proposed method, which look natural. Despite of the large size of removed region, our results still looks sharp without any diffusion blur since we directly use the texture found in the source region. As both the texture and structure information are use, the exemplar-based texture synthesis can achieve great structural continuation in the synthetic image.
Figure 3-4.3: The inpainting procedure and results.
3-5. Frequency-tuned Salient Region Detection (CVPR 2009)

3-5-1. Introduction of the Paper

Detection of visually salient regions is a useful and fundamental technique in Image processing, which can assist many applications such as image resizing, object segmentation and recognition. This paper proposes an efficient algorithm which produces full resolution saliency maps with obvious salient objects, as shown in Figure 3-5.1. Compared with previous methods, the boundaries of these objects retain significantly more frequency content from the original image. This frequency-tuned approach, inspired by the biological concept of center-surround contrast, estimate the contrast by color and luminance features, which makes four advantages over previous work: uniformly highlighted saliency map with well-defined boundaries, full resolution, ease of implementation and computational efficiency.

![Figure 3-5.1 The original images and their saliency maps obtained by our method.](image)

3-5-2. Method

The paper’s algorithm is described below.

For an input image $I$, we first apply the Gaussian blurred filter, a 3x3 or a 5x5 separable binomial kernel, on $I$ and obtain the filtered result $I_{\mu\nu}$. This process can eliminate fine texture details, noise and coding artifacts. Next, we transfer
$I_{w_h}$ into CIELab color space, and evaluate its arithmetic mean image vector $I_\mu$.

Then, the saliency map $S$ can be formulated as:

$$S(x, y) = \|I_\mu - I_{w_h}(x, y)\|$$

where $\| . \|$ is the $L_2$ norm, also known as Euclidean distance. This algorithm can achieve all the following five requirements for salient region detection:

- The largest salient object is stressed.
- The whole salient region is uniformly highlighted.
- The boundaries of salient objects are well-defined.
- The high frequency parts are removed, such as texture, noise, and artifacts.
- Produce a full resolution saliency map efficiently.

### 3-5-3. Result

Figure 3-5.2 shows some examples of our saliency maps. The salient objects in our maps are obvious and uniformly highlighted with well-defined boundaries. The maps can be produced instantly in full resolution, which may assist other applications significantly, such as image resizing.
Figure 3-5.1: Saliency map of the images.

3-6-1. Introduction of the Paper

One of the most important elements of image is color; therefore, the most common and effective task of image processing is color manipulation. The literature introduced an easy and general method to transfer one image’s characteristics to another. The algorithm utilizes simple statistical analysis and applies plain operations on a suitable color space. After the experiments, the most proper color space they found is \( l\alpha\beta \), which minimizes the correlation between each color channels. The concept is intuitive; nevertheless, receive significant results, as shown in Figure 3-6.1.

![Source image](image1.jpg) ![Target image](image2.jpg) ![Result image](image3.jpg)

(a) Source image  (b) Target image  (c) Result image

Figure 3-6.1: The color transfer result.

3-6-2. Method

This paper includes two important parts: statistics and color correction, analysis
of suitable color space.

The first part is the main algorithm of this literature. We first transfer both the source and target images into \( \alpha \beta \) color space. Next, compute the mean and standard deviation of both images, \( (\bar{l}_s, \bar{\alpha}_s, \bar{\beta}_s, \sigma'_{l}, \sigma'_{\alpha}, \sigma'_{\beta}) \) and \( (\bar{l}_t, \bar{\alpha}_t, \bar{\beta}_t, \sigma'_{l}, \sigma'_{\alpha}, \sigma'_{\beta}) \). Then, we subtract the mean values from source image:

\[
\begin{align*}
l^* &= l_s - \bar{l}_s \\
\alpha'^* &= \alpha_s - \bar{\alpha}_s \\
\beta'^* &= \beta_s - \bar{\beta}_s
\end{align*}
\]

After that, scale the new values by the factor of standard deviation ratio:

\[
\begin{align*}
l' &= \frac{\sigma'_{l}}{\sigma_s} l^* \\
\alpha' &= \frac{\sigma'_{\alpha}}{\sigma_s} \alpha^* \\
\beta' &= \frac{\sigma'_{\beta}}{\sigma_s} \beta^*
\end{align*}
\]

Finally, instead of adding original mean values, we add the mean of target image into the result data.

\[
\begin{align*}
l_{res} &= l' - \bar{l}_t \\
\alpha_{res} &= \alpha' - \bar{\alpha}_t \\
\beta_{res} &= \beta' - \bar{\beta}_t
\end{align*}
\]

This transfer will let the source image have the similar color representation as the target image due to the same statistics properties, mean and standard deviation.

As for the suitable color space, the paper analyzes three representative spaces, RGB, \( \alpha \beta \) (similar to the standard CIELab) and CIECAM97s (similar to \( \alpha \beta \)). After comparing the correlations between each color channels, the authors find that RGB are almost complete correlated between all pairs of channels, \( \alpha \beta \) has the similar result with CIECAM97s but more compressed. Thus, the most suitable color space with decorrelated channels they found is \( \alpha \beta \).
3-6-3. Result

Since the $l\alpha\beta$ color space is similar with the standard CIELab, we have examined both results, as shown in Figure 3-6.2 and 3-6.3. The further discussion about these two color space and different ways of normalization will be presented in Section 5-4.
Figure 3-6.2: Color transfer from a normal building photo into a poster style.
Figure 3-6.3: Color transfer from a cold style into a sunset style.
3-7. Two-scale Tone Management for Photographic Look (SIGGRAPH 2006)

3-7-1. Introduction of the Paper

Most recent tone manipulation technique has been dedicated to tone mapping for display high-dynamic-range image, or image styling in Non-Photorealistic Rendering research domain. However, traditional tone manipulation and analogy approaches in NPR have limitation: previous approaches can’t represent tone transfer among the spatial variation of detail effectively. So, in this paper authors propose a new non-linear tone manipulation framework, i.e. two-scale tone management, for modify image to certain mood or aesthetic.

Two-scale tone management is based on non-linear decomposition of an image into base layer and detail layer. After two scale decomposition, we transfer global tone balance in base layer and control local tone balance based on the spatial variation of detail, respectively. Finally, we reproduce image by combine both modified layer result. Moreover, we can also use this framework to make image to change original photographic look to another’s style.

3-7-2. Method

![Figure 3-7.1: The overview pipeline for two-scale tone management](image)

As the framework of Figure 3-7.1 shows, this approach includes four steps:

1) **Two scale decomposition**: in this step, we want to decompose original input image into base layer and detail layer. The former controls the overall tone distribution of image and the latter describes the detail content, respectively.
Step 1.1 Working in logarithmic domain
At first, we transform to logarithmic domain because contrast is a multiplicative effect.

Step 1.2 Bilateral filter decomposition
We use the bilateral filter to get base layer B and detail layer D from original input image:

\[ B = bf(I) \quad \text{and} \quad D = I - B \]

Where I, B and D are work on logarithmic domain and bilateral filter of image I at pixel p is defined by

\[ bf(I)_p = \frac{1}{k} \sum_{q \in I} g_{\sigma_s}(|p - q|) g_{\sigma_r}(|I_p - I_q|)I_q \]

where \( \sigma_s \) specifies the spatial neighborhood scale, \( \sigma_r \) control the influences of intensity difference and k is the normalized term:

\[ k = \sum_{q \in I} g_{\sigma_s}(|p - q|) g_{\sigma_r}(|I_p - I_q|) \]

Due to experiment, authors recommend \( \sigma_s = \min(\text{width, height})/16 \) and \( \sigma_r \), which differentiates important edges from detail, estimate by the edge amplitude of input:

\[ \sigma_r = p_{90}(\|\nabla I\|) \]

where \( p_n(I) \) is the intensity value such that n% of the values of I are under it.

Step 1.3 Poisson correction
In order to avoid halo effect, we have to do Poisson correction that prevents gradient reversal and preserves detail. In other words, we build a gradient filed

\[ \mathbf{v} = (x_v, y_v) \]

Where

\[ x_v = \begin{cases} 0 & \text{if } (\frac{\partial l}{\partial x}) \text{ and } (\frac{\partial d}{\partial x}) \text{ have the different sign} \\ \frac{\partial l}{\partial x} & \text{if } |\frac{\partial d}{\partial x}| > |\frac{\partial l}{\partial x}| \\ \frac{\partial D}{\partial x} & \text{others} \end{cases} \]

and \( y_v \) is defined in similar way. Then we solve Poisson equation

\[ \frac{\partial l}{\partial t} = \Delta l - div(v) \]

In order to correct the detail layer D. In our implementation, we solve linear system of Poisson equation by using TAUCS Library with large sparse matrix.
And finally, we update the base layer according as $B = I - D$.

(2) **Histogram matching in base layer:** In this step, we perform histogram matching to transfer the model base layer $B_M$ onto $B_I$ such that the original input image have similar overall tone distribution as model image.

(3) **Detail management based on the frequency analysis**

**Step 3.1 Computing textureness**
In order to enhance tone style based on the spatial detail variation of image content, in this paper, authors define the *textureness* to describe the high frequency (detail) variation of image. The *textureness* term at pixel $p$, $T(I)_p$, is defined by cross bilateral filter

$$T(I)_p = \frac{1}{k} \sum_{q \in |H|} g_{\sigma_s}(||p - q||) \ g_{\sigma_r}(|I_p - I_q|)|H_q|$$

where $k$ is the normalized term defined as

$$k = \sum_{q \in I} g_{\sigma_s}(||p - q||) \ g_{\sigma_r}(|I_p - I_q|)$$

and $H$ is a high pass vision of input image with cutoff $\sigma_s$. Besides, authors also recommend to use same $\sigma_r$ value with bilateral decomposition, but set value of $\sigma_s$ 8 times larger.
Figure 3-7.2: The textureness map. (a) Input image (b) textureness map for input image (c) textureness map for base layer (d) textureness map for detail layer
Step 3.2 Textureness transfer

At first, we calculate the textureness map for input $I$, model $M$, modified base layer $B'$ and detail layer of input $D$, respectively. Secondly, we enforce the histogram of $T(M)$ onto $T(I)$ by using histogram transfer. To avoid halo effect, we calculate the composing ratio $\rho_p$ at pixel $p$

$$\rho_p = \max \left(0, \frac{T_p' - T(B'_p)}{T(D)_p} \right)$$

Where $\rho_p$ is restricted to use the non-negative to prevent gradient removal.

(4) Re-composition: In this step, we linear recombine the result image

$$\text{Result} = B' + \rho D$$

(5) Detail preservation: The previous result $\text{Result} = B' + \rho D$ may have problem in over saturated highlight and shadow, in other words, we may generate a high dynamic range image in the previous work. So, authors purpose 2 steps to preserve the image detail.

![Figure 3-7.3](image.png)

Figure 3-7.3: The re-composition result compare with result after detail preservation. (a) The re-composition result (b) The result after detail preservation

3-7-3. Result

Because of the high global contrast and variable amount of texture as shown in model image, we get final result image which has more high overall contrast. Moreover, the result also presents level of local contrast clearly and shows more detail information of image content after two-scale tone management.
Figure 3-7.4: The result for two-scale tone management
3-8. Integrated User Interface

3-8-1. Developing Graphical User Interface with QT Library

In order to integrate all image editing operators, we develop a graphical user interface. The user interface is the basic component of Fantasy Illustrator. To develop the user interface, we have surveyed a number of C++ libraries that support our goal, for example, MFC, Smartwin++, wxWidgets, FLTK…etc. Finally we choose QT library. The advantage of QT library over other libraries is as follows:

(1) QT library is a cross platform library. Developing with QT enables our system run on different platforms.
(2) GUI developed with QT library has good appearance.
(3) QT utilizes the characteristic. The signal/slot mechanism of QT is very outstanding. With this mechanism, we can easily perform event listening task.

The most important mechanism of QT is signal/slot concept. QObject sends a signal to environment. When another QObject receives the signal, it can do some tasks. To connect two objects via signal/slot mechanism, we use the following C++ code:

```cpp
QObject::connect(QObject *sender, char *signal, QObject *receiver, char *method)
```

The signal and slot declaration is as follows:

```cpp
class MyObject : public QObject
{
  Q_OBJECT
  signal:
    void exampleSignal();
  public slots:
    void exampleSlot();
};
```

A signal function is taken as an object to be send by a QObject. When a signal and a slot are connected, the task that is defined in the slot will be performed. We can use the following code to send a signal:

```cpp
emit exampleSignal();
```

The above is the core event listening mechanism of QT library.

3-8-2. Interaction
In the section, we will demonstrate our system and show the operating flow for each operator, respectively.

**Geometric-editing operators**

(1) Image Resizing

First, user chooses the target image as shown in Figure 3-8.1 and clicks the image resizing button in the tool bar. Then system will display editing panel with some option as like Figure 3-8.2(a).

After start the resizing process, there would have been a number of red line on the screen. A red line means a seam, and seams appear in order to represent the relative least noticeable path which can be omitted. The results of horizontal and vertical oriented resizing are shown in Figure 3-8.3 and Figure 3-8.4, respectively.
Figure 3-8.2: The editing panel for image resizing operator. (a) The user can choose resize direction they want. (b) Top: resizing along vertical direction. Bottom: resizing along the horizontal direction.

Figure 3-8.3: Horizontal oriented resizing

Figure 3-8.4: Vertical oriented resizing
(2) Image Cloning

First, user chooses the target image as shown in Figure 3-8.5 and clicks the image cloning button in the tool bar. Secondly, our system will show editing panel and user can specify the cloned region wanted to paste onto the target image by circle a closed region (Figure 3-8.6(a)(b)). Then, our system will calculate the triangular mesh like Figure 3-8.7 in real time. Thirdly, user can drag cloned region over target image arbitrarily to choose the pasted destination. Finally, we would get a seamless cloning result image.

![Figure 3-8.5: Choose the target image](image1)

![Figure 3-8.6: Choose the source image region. (a) Read an object image. (b) Draw a closed object area. (c) Generate the triangular mesh to accelerate the computation.](image2)
Figure 3-8.7: Choose the target image

(a)

(b)

Figure 3-8.8: The results image of image cloning
(3) Image Inpainting

The whole image inpainting process is shown as Figure 3-8.9. At first, user load a image (Figure 3-8.9 (a)). Secondly, user specifies the region want to remove (Figure 3-8.9 (b)). The inpainting algorithm fills unknown region from the region boundary to the inside as shown in Figure 3-8.9 (c)(d)(e). Finally, we will get the result image with no hole as Figure 3-8.9 (f).

Figure 3-8.9: Image inpainting
**Color/Tone-editing operators**

(1) Color Transfer

First user loads the image as shown in Figure 3-8.10 (a). Then user also need to specify the model image (Figure 3-8.10 (b)). After color transfer operator, we get the new color style image at finally.

![Color Transfer](image.png)

Figure 3-8.10: Color transfer. (a) Input image (b) Model image M1 (c) Result image transferred from model image M1 (d) Model image M2 (e) Result image transferred from model image M2

(2) Tone Transfer

First user loads the image as shown in Figure 3-8.11 (a). Then user also need to specify the model image (Figure 3-8.11 (b)). After color transfer operator, we get the new tone balance style image at finally.
Figure 3-8.11: Tone transfer operation. (a)Input image (b) Model image (c) Result
4. Experimental Result

Many results have been shown in previous sections. Here we show the step-by-step use of operators to produce a desired image.

<table>
<thead>
<tr>
<th><img src="image1.png" alt="Source Image" /></th>
<th>This is the source image.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image2.png" alt="Inpainting" /></td>
<td>We apply image inpainting operator to remove the butterfly.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Airplane" /></td>
<td>We paste an airplane into the image.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Color Transfer" /></td>
<td>Finally we apply color transfer to transfer the color in the image. The small image is taken as the model image.</td>
</tr>
</tbody>
</table>
5. Discussion and Conclusion

5-1. Image Resizing

Seam carving algorithm performs good in many case. However, it is possible that the algorithm destroys the structure in the image. Figure 5-1.1 shows the result that seam carving destroys structure in the image. Because seam carving considers only the local energy and removes the seams without considering a large area of structural object. Thus the final result is not that good. To prevent this problem, a method presented in SIGGRAPH 2008 named “Optimized Scale-and-Stretch for Image Resizing” can produce better results.

![Figure 5-1.1: Seam carving destroys structure in the image.](image)

Besides, we have found that a good saliency map guides the algorithm to produce better results. Thus we implement a CVPR 2009 paper (see Section 3-5) to compute a robust saliency map. With good saliency evaluating algorithm, image resizing operator can achieve better results.

5-2. Image Cloning

We first analyze the performance of MVC cloning and Poisson cloning. After comparing variant examples, we can conclude that although the interpolants are absolutely different (MVC is merely an approximation), the cloned results are both satisfying and hard to distinguish visually. Even if we could find the minor difference, it is still hard to tell which one is better and to discriminate which cloning method it used only by the result images. The difference between these two membranes tends to be more apparent in the concave regions, and become smaller in the convex shapes, such as rectangle and polygon. The reason is that, since MVC decides the sampling density of boundary points by distance, it may find many “wrong” boundary points with small distance. This kind of error will be obvious when the concave shape located in the scene with extreme gradients and...
texture, like the example shows in Figure 5-2.1. However, for a more general scene with less drastic variations, the result becomes comparable to Poisson cloning in visual quality.

Next, we will discuss the efficiency of the two cloning methods. MVC cloning can achieve real-time interactive performance on CPU, no mention to GPU. On the other hand, Poisson cloning must use much more time on CPU, and even it can achieve real-time by GPU recently, it will take substantial memory footprint, and once the footprint exceed the available capacity, the performance drops. Besides, solving Poisson equation on GPU is a much more complex task than MVC cloning, and it may produce the noticeable flicker while user moves the cloned region.
5-3. Image Inpainting

5-3-1. Comparison with Traditional Diffusion Method

Compared with the traditional image diffusion inpainting, as shown in Figure 5-3.1 and 5-3.2, our algorithm offers two advantages:

First, the diffusion will produce considerable blur in the target region and loss the entire high-frequency texture information. On contrast, our result automatically propagates the linear structure and produces the clear and remarkable result by plausible textures from source region. The proposed method can achieve satisfying result over most area; nevertheless, some patches may not be filled perfectly. Although this shortage may look naturally sometimes, it can be improved by further process of inpainting.

Second, since our method does not have such diffusion step, it can save more computation time.

(a) The original image.  
(b) The selection of target region.
Figure 5-3.1: The inpainting result on an aerial photograph.

(a) The original image.  (b) The selection of target region.

(c) The result of diffusion method.  (d) The result by proposed method.
5-4. Color Transfer

5-4-1. Different Color Space and Normalization Method

Figure 5-4.1 and Figure 5-4.2 show the results of the same color transfer algorithm on different color spaces, \( l\alpha\beta \) and CIELab, and with different normalization methods.

Comparing the two color spaces, although the results are similar in some degree, we can see that the CIELab can achieve better results than \( l\alpha\beta \). The reason may be that since CIELab become the widely used standard, it could obtain better properties of decorrelation and perceptual uniform than \( l\alpha\beta \).

As for normalization, three important notes should be considered beforehand. First, since we only have the domain of CIELab color space, \( 0 \leq L \leq 100, -127 \leq \alpha \leq 127, -127 \leq b \leq 127 \), the normalization can be applied only on this (The authors do not provide the domain of \( l\alpha\beta \)). Second, the normalization should be used on the CIELab not on RGB, since the procedure of color transfer is all on CIELab. Finally, three normalization conditions are
considered in our experiments, “without normalization”, “common” normalization” and “improved normalization”. The common one calculates the range directly by the maximum and minimum of each channel, and rescales it to the corresponding domain range. The improved one takes the similar process, but calculates the range with the examination of the domain range. If the range does not exceed the domain range, it should not be modified.

According to the paper’s results, “without normalization” result, which clips the over-ranged values, will be the closest one since it better preserves the original statistical distribution. Nevertheless, it may produce some over-exposed (under-exposed) results due to those over-ranged values. On the other hand, the normalization methods can solve the above over-ranged problem, but meanwhile change the statistical distribution and sometimes produce unexpected results. The improved one works much better than the common one; however, still cannot totally follow the color representation of target image. To combine the advantages of “without normalization” and “improved normalization”, the HDR method may be a proper solution for future work.
(e) Result in CIELab space with improved normalization.

(f) Result in $l\alpha\beta$ space without normalization.

Figure 5-4.1: Color transfer comparison for an example of normal light variation.
Figure 5-4.2: Color transfer comparison for an example of extreme light variation.
5-5. Tone Transfer
5-5-1. Comparison with Linear Tone Manipulation

Compare with the histogram matching which is the traditional solution for transferring an intensity distribution, it is obvious that two-scale tone management represent better tone transfer among the spatial variation of detail effectively. That is because histogram matching linear transfer original image histogram curve to certain shape we want. So, even histogram matching can make good result in overall tone balance, however, it can’t make input image has distinct local contrast as model image shown.

Figure 5-5.1: (Left) Input image (Right)Model image

Figure 5-5.2: The result image after histogram matching
5-5-2. Normalization for Display and High Dynamic Range Problem

Consider most digital images are in a discrete intensity range from 0 to 255, it’s critical to find a continuous domain in order to enhance and differentiate the contrast in overall and local balance. However, even two-scale tone management does a good work, there is an important problem, i.e. the result image usually is in the high dynamic range. That means we need to recover high dynamic range image through detail preservation procedure in gradient domain, reconstruct image by solving Poisson equation and to the tone mapping. Otherwise, we may get the overexposure or underexposure result like as Figure 5-5.5 and Figure 5-5.6.
In order to get better result than previous work, this paper modifies every pixel value based on textureness, i.e. the texture map describes the spatial detail variation. By using the textureness map, the result image show clear detail texture with obvious local contrast as the model image shown, for example, the smooth region that cloud and complex texture exist among the tree, respectively.

5-5.3. Textureness Map
6. Division of Labor

As described in previous sections, we actually do a lot of tasks in the final project. In this project, all tasks are equally divided and distributed to three group members. Totally we implement seven international conference and journal papers. Moreover, we develop an integrated user interface, which visualizes the information that is related to the algorithms and allows users to interact with images. The division of labor is described in detail in the following paragraphs.

Writing Proposal

All group members put lots of efforts discussing the original idea, “Fantasy Illustrator”. 翁郁婷(Wong-Yu Ting) surveyed many useful papers and was responsible for Motivation section. 羅聖傑(Sheng-Jie Luo) was responsible for Approach section and designing system architecture. 張明旭(Ming-Hsu Chang) surveyed many useful papers and was responsible for Expected Result section and correcting the proposal document. All group members performed proof reading together.

System Development and Implementation

Fantasy Illustrator is a very large scope project. There are five image editing operators. All the operators interact with the user interface to show some information and results. The integration step of implementation is not easy. 翁郁婷(Yu-Ting Wong) implemented the tone transfer operator, and cooperated on image resizing operator and graphical user interface design and development. 張明旭(Ming-Hsu Chang) was responsible for image inpainting operator and color transfer operator. He also cooperated on image cloning operator and implemented image saliency algorithm. 羅聖傑(Sheng-Jie Luo) handled image resizing operator and cooperated on image cloning operator. He also integrated every part of the codes into final version of Fantasy Illustrator. The user interface component architecture and appearance was developed and designed by 羅聖傑(Sheng-Jie Luo) and 翁郁婷(Yu-Ting Wong). Optimization and memory management of the C++ code is done by 張明旭(Ming-Hsu Chang). Overall, all group members have done a lots of tasks in the project and help each other when facing problem. All group members think that the team work is great. We learned a lot from the project.
Demo and Presentation

During presentation, 翁郁婷(Yu-Ting Wong) was responsible for opening, motivation of the project and the system architecture parts. 張明旭(Ming-Hsu Chang) presented the algorithm of the five operators in detail. 羅聖傑(Sheng-Jie Luo) demonstrated final version of Fantasy Illustrator and showed all the result to audiences. All group members think the presentation experience is great.

7. Reference

7-1. Implemented Papers in the Project


SOONMIN BAE , SYLVAIN PARIS , Frédo DURAND. 2006. Two-scale tone
management for photographic look, ACM Transactions on Graphics (TOG), v.25 n.3. (Proc. of SIGGRAPH 2006)

7-2. Other Resources


OpenCV
http://sourceforge.net/projects/opencvlibrary/

QT Library
http://qt.nokia.com/products

TAUCS, a Library of Sparse Linear System
http://www.tau.ac.il/~stoledo/taucs/