

# Design and Implementation of Personal Image Processing Assistant

Zengyu Cai <sup>1</sup>, Xuhui Wang <sup>1</sup>, Jianwei Zhang <sup>2 +</sup>, and Chunfeng Du <sup>1</sup>

<sup>1</sup> School of Computer and Communication Engineering Zhengzhou University of Light Industry, Zhengzhou, China

<sup>2</sup> School of Software Engineering College Zhengzhou University of Light Industry, Zhengzhou, China

**Abstract.** As an important subject area, image processing has a wide range of applications, such as military, scientific research and our daily life, etc. However, the disadvantage of existing image processing software makes it impossible to meet the needs of the public. The paper designs an image processing software that can run on Windows system. It takes C# as the programming language and uses Visual Studio as the development environment. The image processing assistant realizes the basic processing of the image such as image cropping and zooming, color filtering and reversing, the adjustment of transparency and contrast, etc. Results of test show that the function and performance of the image processing assistant meet the expected requirements, and can meet the daily needs of users.

**Keywords:** image processing, image zooming, color reversing, full-featured.

## 1. Introduction

Nowadays, image is the material and spiritual requirement, has become an indispensable part of people's life and work [1]. It can record some good moments such as personal portrait, wedding photography, etc. However, the original image obtained through the camera can't meet the needs of people so that it needs to be perfected with the corresponding image processing software. Currently, Photoshop and Mtxiuxiu are the most common image processing software in China and abroad [2]. Among them, Photoshop is recognized as the most professional image processing software, which has powerful editing capability and comprehensive analytical force on the color of the image, but high cost, high difficulty and large memory result in Photoshop is difficult to become the mainstream of the public image processing software. Mtxiuxiu is the most popular image processing software in the domestic market. It is easy to operate and the interface is simple and friendly so that it is popular with the public. Mtxiuxiu advocates one-stop operation so that it can enable users to achieve a series of functions with simple operation. And it can be used in the mobile intelligent terminal, so users can use it anytime and anywhere. However, the limited function results in it doesn't satisfy the user's needs very well.

The personal image processing assistant designed and developed in this paper is an image processing software that can run on Windows operating system. It mainly includes five function modules: file management, image zooming, image editing, image processing and skin replacement. The personal image processing assistant mainly realizes image opening and saving, image scaling, image cropping and rotating, image graying, color filtering and reversing, the adjustment of brightness, contrast and transparency. In addition, the skin can be replaced, etc.

## 2. System Design

The system design mainly includes interface design and function design [3]. It is based on the analysis of the function of image processing system and users' demands.

---

<sup>+</sup> Corresponding author. Tel: +13603829696; fax: +0371-63556921.  
E-mail address: ing@zzuli.edu.cn.

## 2.1. Interface design

In order to ensure the personal image processing assistant is easy to operate [4], its interface is in the form of multi-document (MDI). According to the function and the interface standard of Windows operating system, the main menu includes file, edit, view, insert, picture, color, special effects and screenshots, etc. At the same time, the corresponding dialog box is set up to ensure good interactivity.

The interface is friendly, beautiful and generous, it also has strong practicability. There are many parameter setting dialog boxes applied to determine the parameter. Users can easily compare the image processing effect of different parameters to understand the role of these parameters in image processing by entering different parameters. Friendly interface can ensure good interactivity. And the system will pop up the corresponding prompt box when a user makes a mistake, such as in the process of image graying, if the image input is a grayscale image, the system will pop up a warning box and shows “please input a color image”. Thus, users can use the software without hindrance based on the system prompts.

## 2.2. Function design

The personal image processing assistant is an image processing system, which mainly realizes the image processing. According to the analysis of the function of an image processing system and users’ needs [5], its functions mainly include image scaling, image cropping, image rotating, image graying, color filtering and reversing, the adjustment of brightness, contrast and transparency, etc. In addition, the skin of system operation interface can be replaced to bring good experience to the user. The overall function of the personal image processing assistant is shown in Fig 1.

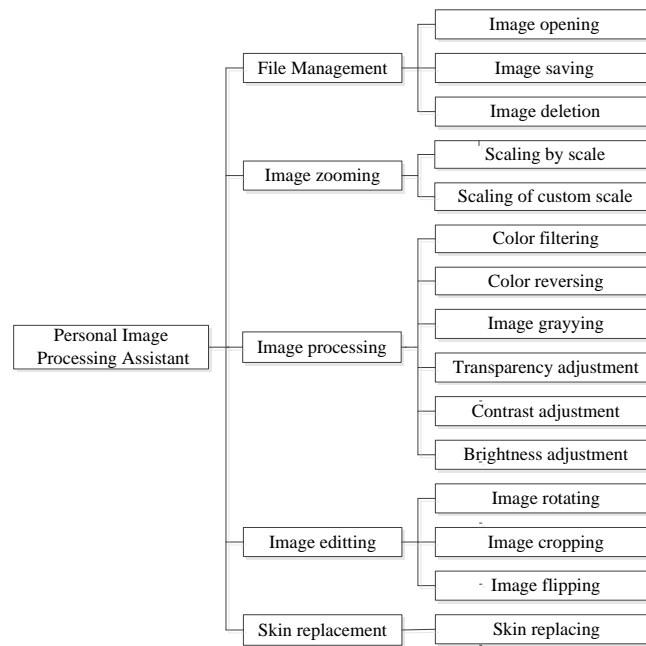


Fig. 1: The overall function.

As can be seen from Fig 1, the image processing system mainly consists of five functional modules: file management, image zooming, image editing, image processing and skin replacement. The functions of these modules can be used together to realize the basic processing of images.

Among them, file management module mainly includes image opening, deletion and saving, and can specify the location when saving an image. Image zooming module can scale the image, including scaling by scale and scaling of custom scale. Scaling by scale is based on the scale provided by the system, such as 50%, 150%, 300%, etc. Custom zooming is to zoom the image based on the scale defined by users. Image editing module mainly realizes color filtering and reversing, image graying, the adjustment of transparency, contrast and brightness, etc. Image processing mainly includes image rotating, cropping and flipping, etc. In addition, the skin of system operation interface can be replaced to bring good experience to the user [6-7].

## 3. Function Implementation



### 3.1. File management

File management module mainly includes image opening, deletion and saving, and can specify the location when saving an image [8]. The *Bitmap* class is defined in the process of opening an image. It can read the path of the image, get the image from the path and display it in the operating interface. In the *Bitmap* class, the method named `public int outWidth` is used to obtain the width, the method named `public int outHeight` is used to obtain the height.

### 3.2. Image zooming

Image zooming module mainly includes scaling by scale and scaling of custom scale. Scaling by scale is enlarge the coordinates of pixel points on a certain scale, and get the corresponding new coordinates. What's more, the *interpolation* method is applied to avoid the image being blurred. The key code to implement this function is shown in Table I.

TABLE I: THE KEY CODE OF IMAGE ZOOMING

---

```
private void menuItemZoom50_Click(object sender EventArgs e)
{
    zoomFactor == 0.5;
    cZoom.Checked = false;
    menuItemZoom50.Checked = true;
    cZoom = menuItemZoom50;
    this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width * zoomFactor),
    Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
    this.Invalidate();
}
```

---

### 3.3. Brightness adjustment

Each pixel has its own three components of RGB, and each component has 256(0 ~ 255) brightness levels [9]. Among them, 0 represents the darkest, or black, and 255 is the brightest, which is white. The adjustment of brightness is realized through adjusting the brightness of each pixel in the image.

A parameter named *nBrightness* is set in the function, its value ranges from -255 to 255, which needs users to input through the dialog box. When system get this value, it will be added to the original value of RGB of each pixel point, resulting in the change in brightness. In addition, in order to ensure the newly obtained RGB values are within the normal range, the corresponding control mechanism is also added in the system. If the value of RGB newly acquired is out of the normal range, the system will give a prompt and require users to input again.

### 3.4. Image graying

At present, there are two ways to realize graying [10]. One is to calculate the average of three components of RGB of each pixel point and then return the average to each pixel point. The other is to turn an image into a grayscale image according to YUV color space. The corresponding relationship between brightness Y and RGB components can be shown as equation (1) according to the relationship between YUV and RGB(Y represents the brightness level):

$$Y=0.3R+0.59G+0.11B \quad (1)$$

The first method is applied to realize image graying here. However, because the eyes are very sensitive, the grayscale image obtained by means of the average method is not ideal. After many debugging, the three components of RGB are set to 298, 586, 115, which can get the ideal grayscale image.

### 3.5. Color reversing

The color reversing is realized mainly through an algorithm named *Bitmap*. First, the RGB value of each pixel point is reduced by 255, and then the result is given to the RGB of each pixel point. All the pixel points in the image are processed with the same method until the full image is processed. The key code to implement this function is shown in Table II.

The unsafe code block is the core part of realizing this function, and its principle can be expressed as equation (2):

$$p[0]=(\text{byte})(255-p[0]) \quad (2)$$

First, get the data pointer inside the image and then set the offset. Since each pixel point contains three components of RGB, the  $nWidth$  should be set to  $b.Width*3$  to process each pixel point three times. Then, process all the pixel points with two nested loops. Finally, the image resource is released through  $b.UnlockBits(bmData)$  and returns true.

TABLE II: THE KEY CODE OF COLOR REVERSING

---

```

public static bool Invert(Bitmap b)
{
    BitmapData bmData = b.LockBits(new Rectangle(0,0,b.Eidth, b.Height), ImageLockMode.ReadWrite,
    PixelFormat24bppRgb);
    int stride = bmData.Stride;
    System.IntPtr Scan0 = bmData.Scan0;
    unsafe
    {
        Byte * p (byte *) (void *) Scan0;
        Int nOffset = stride - b.Width*3;
        Int nWidth = b.Width * 3;
        For (int y=0; y<b.Height; ++y)
        {
            For (int x=0;x<nWidth; ++x)
            {
                p[0] = (byte)(255-p[0]);
                ++p;
            }
            p += nOffset;
        }
    }
    b.UnlockBits(bmData);
    return true;
}

```

---

### 3.6. Image rotating

The rotation function [11] is realized through the mutual transformation between the origin of the image and the origin of Cartesian coordinates. An image is rotated around its center point, but the origin of the image is in its upper left corner, thus, before rotating the image, the origin should be moved to center and flip the Y-axis.

Assuming the width of the image is  $W$ , the height is  $H$ , the coordinate of one pixel point is  $(x_0, y_0)$ , the coordinate of the origin transformed is  $(x_1, y_1)$ . The transformation relationship is shown as equation (3):

$$\begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} = \begin{bmatrix} x_0 & y_0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ -0.5W & 0.5H & 1 \end{bmatrix} \quad (3)$$

Assuming the rotation angle is  $\theta$ , and the coordinate of origin rotated with the rotation matrix is  $(x_2, y_2)$ . The rotation relationship is shown as equation (4):

$$\begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Assuming the width of the image rotated is  $W^*$ , the height is  $H^*$ . Then the position of the origin can be changed back to the upper left corner according to equation (5), its coordinate is  $(x_3, y_3)$ .

$$\begin{bmatrix} x_3 & y_3 & 1 \end{bmatrix} = \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0.5W^* & 0.5H^* & 1 \end{bmatrix} \quad (5)$$

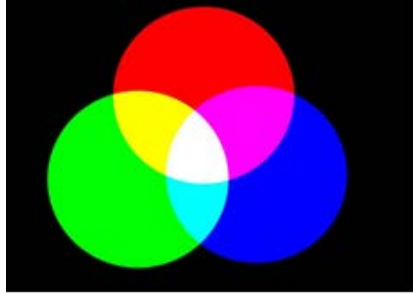
## 4. Function Test

Testing, the final step in the software development process, is regarded as the most important part. Only through testing, can we find the flaw in the software design, so that the software run more stably and reliably through debugging. Testing is to verify that the design meets the requirements, whether it is inconsistent with the requirements, so that a better plan is proposed.

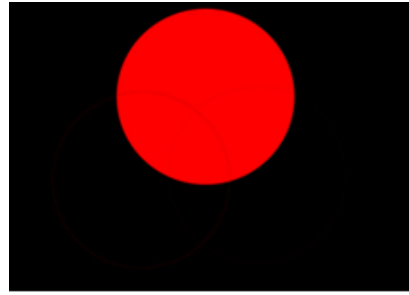
Personal image processing assistant is an image processing software, its main function is to process image, mainly realizes image opening and saving, image scaling, image cropping, image rotating, image graying, color filtering and reversing, the adjustment of brightness, contrast and transparency, etc.

### 4.1. Color filtering

Color filtering is realized by adjusting the brightness of the different colors. A color will be filtered out if its brightness level is adjusted to 0. To preserve red, filter blue and green for example, the effect is shown in Fig 2.



(a) The original image



(b) The effect of filtering blue and green

Fig. 2: Color filtering.

### 4.2. Contrast adjustment

The contrast adjustment is realized by adjusting the difference between pixel points. The larger the difference is, the more obvious the contrast of the image will be. Taking 56 as the difference for example here, the effect is shown in Fig 3.



(a) The original image



(b) The effect of taking 56 as the difference

Fig. 3: The contrast adjustment.

## 5. Conclusion

This paper designs an image processing software in C#. It can run on Windows operating system and realize the basic processing of image, such as image graying, color filtering and reversing, the adjustment of brightness, contrast and transparency, etc. Compared to the existing software, the personal image processing assistant is easy to operate and has complete functions, its interface is beautiful and generous. At the same time, this software supports the vast majority of image formats so that it can well meet users' needs.

## 6. Acknowledgment

This work is supported by Key Technologies R & D Program of He'nan Province (172102210059), National Natural Science Foundation of China (No.61672471 and No.61502436) and Plan For Scientific Innovation Talent of Henan Province (184200510010).

## 7. Reference

- [1] Jian L I, Tsing hua Y U, Zheng Y. Development of digital image processing web expeimental platform based on matlab[J]. Experimental Technology & Management, 2006.
- [2] Huang Z, Jiang S, Liu J. Realization of digital image processing virtual laboratory based on Web[J]. Experimental Technology & Management, 2010.
- [3] Yao N, Guo Y, Yu J. Development and realization of system of digital image processing based on MATLAB platform[J]. Journal of Digital Information Management, 2014, 12.
- [4] Cha W. The Application of Cognitive Interface Design Methodology for a Digitalized Main Control Room in Nuclear Power Plant[C]// Analysis, Design, and Evaluation of Human-Machine Systems. 2013, pp. 410-414.
- [5] Shen X, Maa P Y. A camera and image processing system for floc size distributions of suspended particles[J]. Marine Geology, 2016, vol. 376, pp. 132-146.
- [6] M. Gorgoń. Parallel performance of the fine-grain pipeline FPGA image processing system[J]. Opto-Electronics Review, 2012, vol. 20, pp. 153-158.
- [7] Mertes J G, Marranghello N, Pereira A S. Real-time Module for Digital Image Processing Developed on a FPGA[J]. IFAC Proceedings Volumes, 2013, vol. 46, pp. 405-410.
- [8] Zhang Y D, Zhu S M, Li L. Development of Digital Image Processing System Based on MATLAB[J]. Advanced Materials Research, 2014, pp. 971-973:1594-1597.
- [9] Wang X. Research and Realization of the Image Processing System Based on the Android Platform[J]. Lecture Notes in Electrical Engineering, 2014.
- [10] Zhang X, Wang X. Novel Survey on the Color-Image Graying Algorithm[C]// IEEE International Conference on Computer and Information Technology. IEEE, 2017, pp. 750-753.
- [11] Kuo H J, Liu Y C, Cheng Y C. Image processing system and method of improving human face recognition[J]. 2016.