Changing the ratio of an image intelligently according to its contents: an image processing tool based on pixel weight and face detection

By: Han Qi

Tianjin Nankai High School, Tianjin, China

Aug. 30th, 2017

Contents

Abstract

Objective: To meet the need for changing the ratios of pictures in different situations, this paper presents an image processing tool which can intelligently resize the pictures according to their contents. Compared with the traditional methods which pull and push the pictures directly or cut the pictures, this program can avoid making the important contents distorted or lost.

Methods and Results: The program gave every pixel in an image different weight to indicate how important it was. Then, with the help of dynamic programming, the program deleted the seam containing the smallest sum of weights, which meant the program just deleted some useless pixels in order to reserve the important contents while changing the ratio of pictures. There were also two improvements in the final program. One was face detection, which was a supplement to the pixel weight, for the faces were an exceptional situation for the pixel weight. The other one was customization, which allowed users to choose which parts of the image they specifically wanted to reserve and then the seams found by my program would avoid this part.

Conclusion: This program can be applied in photo printing, graphic designing, screen displaying, and other practical applications in our daily life. The paper has three innovations: in science, pixel weight and face detection are innovations of algorithm; in technology, customization function can meet users' more special needs; in application, the program is a useful tool suitable for various fields.

Keywords: pixel weight; face detection; dynamic programming; customization; image resizing

1 Introduction

Nowadays, there are a number of situations in which we have to change the ratio of an image. For instance, due to the difference of ratio of the printing papers and the ratio of the pictures in our phones, when we want to print a picture out, we have to change the ratio of this picture. Also, nowadays there more and more kinds of electronic devices appearing, but the screen ratios of these devices are totally different. If we want to realize full-screen displaying on all of these devices, we have to change the ratios of picture. Full-screen displaying is very important: not only does it make the whole screen into use without any "waste", but also it is what the designers focus on, such as the design of iPhone X (which is famous for its full-screen displaying). Changing the ratio of an image is also a necessary and important work for designers. Because they have to make the pictures fit in their design, they cannot avoid changing the ratio of pictures.

Fig. 1.1 Photos of devices with different screen ratios

But, I find that until now there is not any good way to change the ratio of an image. Someone pushes or pulls the picture directly, but as a result, this method will make the contents deformed and have very bad influence on the visual effect. The other people may cut the picture, but as a result, this way make the picture lose some important contents and also it influences the meaning that this picture wants to convey.

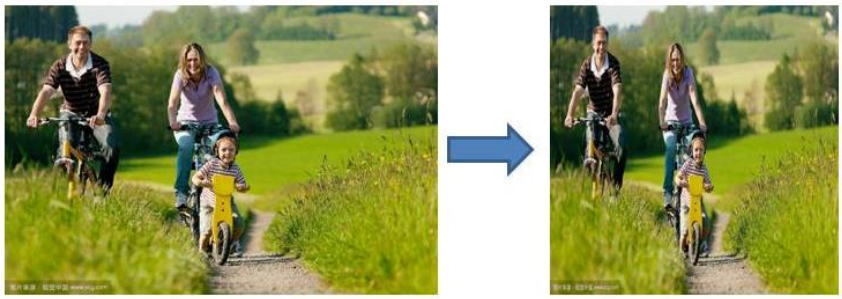

Fig. 1.2 Picture resized by compressing directly

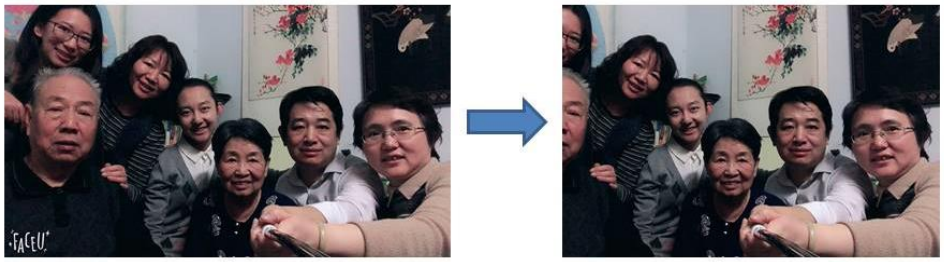

Fig. 1.3 picture resized by cutting

So, I want to write a program which can change the ratio of an image intelligently according to the contents of this image. In other words, after being processed by my program, this image's ratio is changed but its major contents are not influenced.

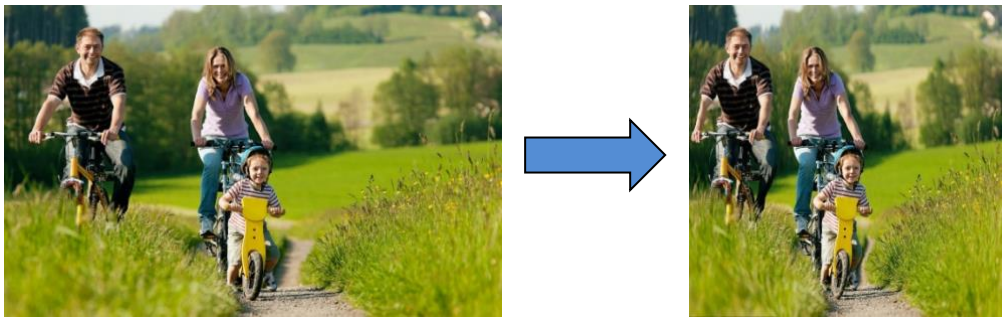

Fig. 1.4 Picture resized according to its contents by this program

2 Model & Theory

2.1 Pixel Weight

The first step in this program is to find out which pixels are important and which pixels are useless. According to humans' visual characteristics, humans always view the contents which are very different from the contents around them as important part, for those contents are "highlighted" when people see the picture at the first sight. On the contrary, if a part of the picture is very similar to the contents around it, such as the grasses, sky, this part is not very pivotal, for there are so many similar contents around. In other words, if we just delete this part, people even cannot detect the loss of this part. Based on the above discussion, to judge whether a pixel is important or not, I will find out this pixel's changing rate with the pixels around it. And I use pixel weight to indicate the changing rate of each pixel. If this pixel is very different from the pixels around it, it will have large pixel weight. On the contrary, it this pixel is very similar to the pixels around it, it will small pixel weight.

The definition of the pixel weight: $E(P) = \left|\frac{\partial}{\partial x}P\right| + \left|\frac{\partial}{\partial y}P\right|$.

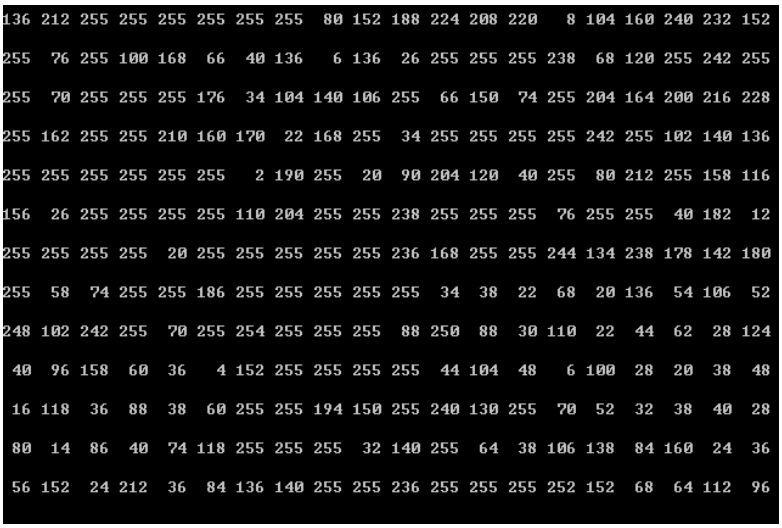

Fig. 2.1 A demonstration of the pixel weight of each pixel calculated by our program

2.2 Dynamic Programming

We can delete same number of pixels from each row to narrow the width of image while keeping the rectangular shape. We should minimize the sum of pixel weight of pixels removed, thus maximize the amount of useful information of the remaining parts. But only to do so will destroy the continuity of picture. And only remove pixels of one column whose sum is the smallest at a time will also lead to breakage. So I remove pixels by deleting seams. The definition for seam is like this: there must be one pixel per line and the pixels in contiguous lines must be consecutive. I find that dynamic programming is a very good algorithm to find such a seam containg smallest sum of pixel weight.

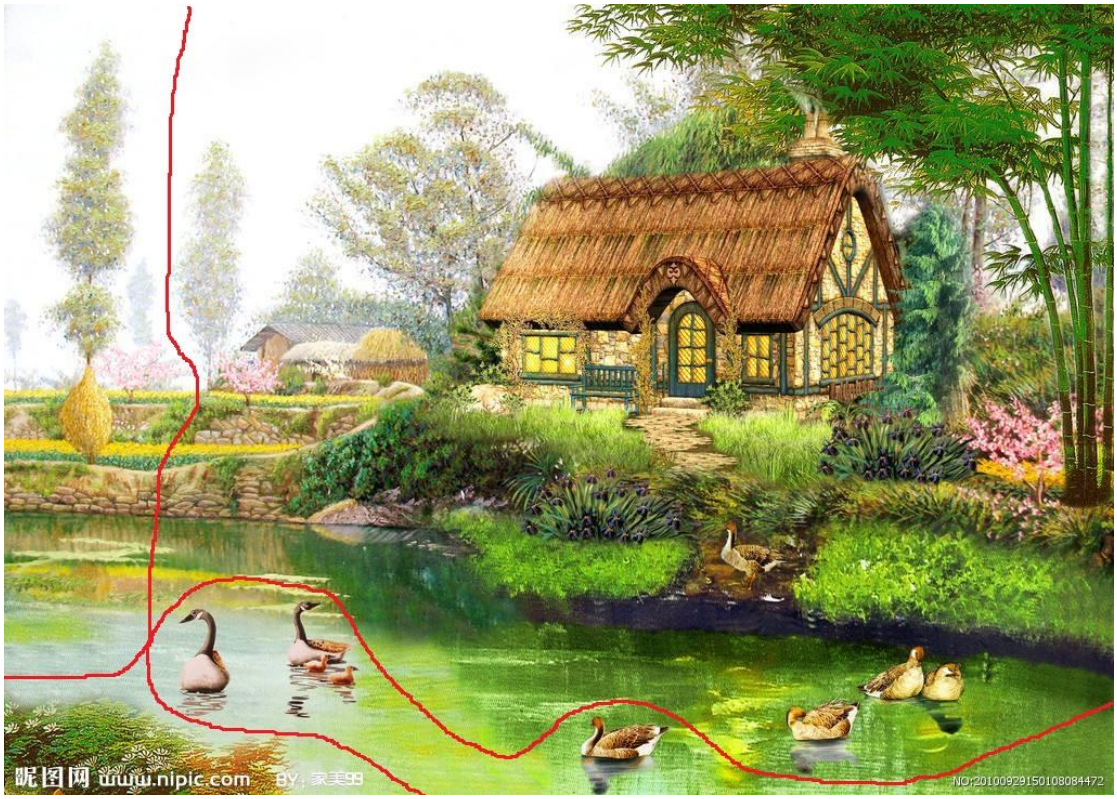

Fig. 2.2 Vertical seam and horizontal seam

2.2.1 Initialization

For the first row, the smallest sum of pixel weight for each pixel if the pixel's own pixel weight. So M[1][j] is w[1][j] and path[i][j] is j.

2.2.2 Procedure

Then, from the second row, I will calculate each pixel's smallest sum of pixel weight from the first column to the last column. According to the idea of dynamic programming, the smallest sum of pixel weight at pixel M(i,j) is the result of adding the minimun of {M(i-1,j)， M(i-1,j-1)， M(i-1,j+1)} to the pixel weight of M(i,j).

$$M(i,j) = w(i,j) + min(M(i-1,j-1), M(i-1,j), M(i-1,j+1))$$

This program will repeat the above operations and the program will also record the path by which we can get the smallest sum of pixel weight. After comparing the smallest sum of pixel weight at each pixel in the last row, we can find out the seam containing smallest sum of pixel weight.

2.2.3 Conclusion

There are a number of advantages of using dynamic programming to find out the seam which meets our requirement. For example, Fig.2.3a shows the correct seam that we want to delete. But in Fig.2.3b, although the sum of the pixel weight in this seam is smaller than that of Fig.2.3a, if I delete this seam, I will cause the picture discontinuous. But with dynamic programming, I will avoid the error. In Fig.2.3c, it shows a seam which does not contain the smallest sum of pixel weight. If I delete this seam, I will make the contents lose so much useful information. With dynamic programming, I will also avoid this error. In conclusion, dynamic programming is the most appropriate algorithms to find a seam which is suitable to delete.

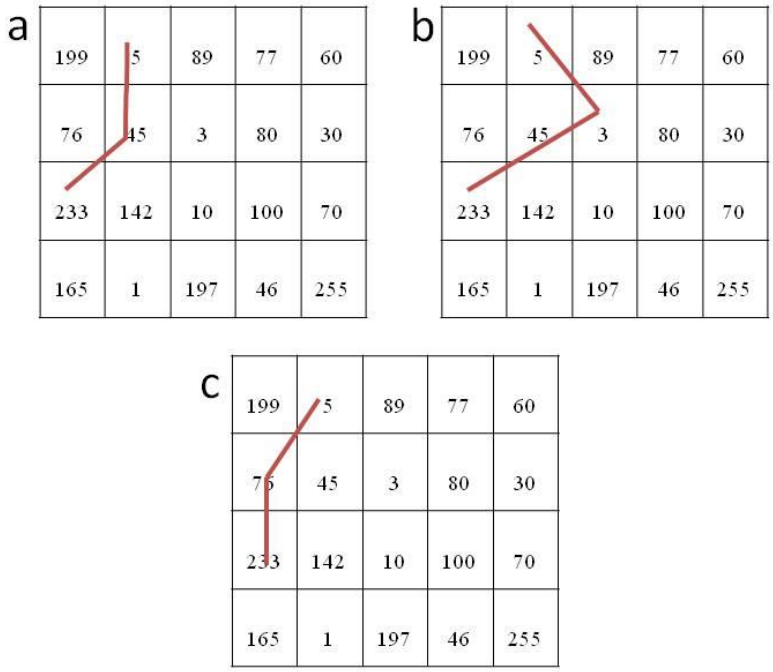

Fig. 2.3 correct and incorrect seams

2.3 Improvement I-face detection

Face detection function will help to better figure out the important pixels in the pictures which contain human faces. Human faces are an exceptional situation which the basic program cannot deal with. Because human faces are very smooth (almost all of the face is in the same color), according to the definition of the pixel weight, this program will give the pixels on faces very small weights, which will cause that some part of the faces will be deleted. This will have very bad influence on the processing effect because faces are very pivotal part in the pictures. After adding face detection function to the program, this program can solve this problem automatically.

The improved program with pixel weight and face detection can find out all of the important pixels and reserve them. So the improved program can resize all kinds of pictures intelligently according to the pictures' contents.

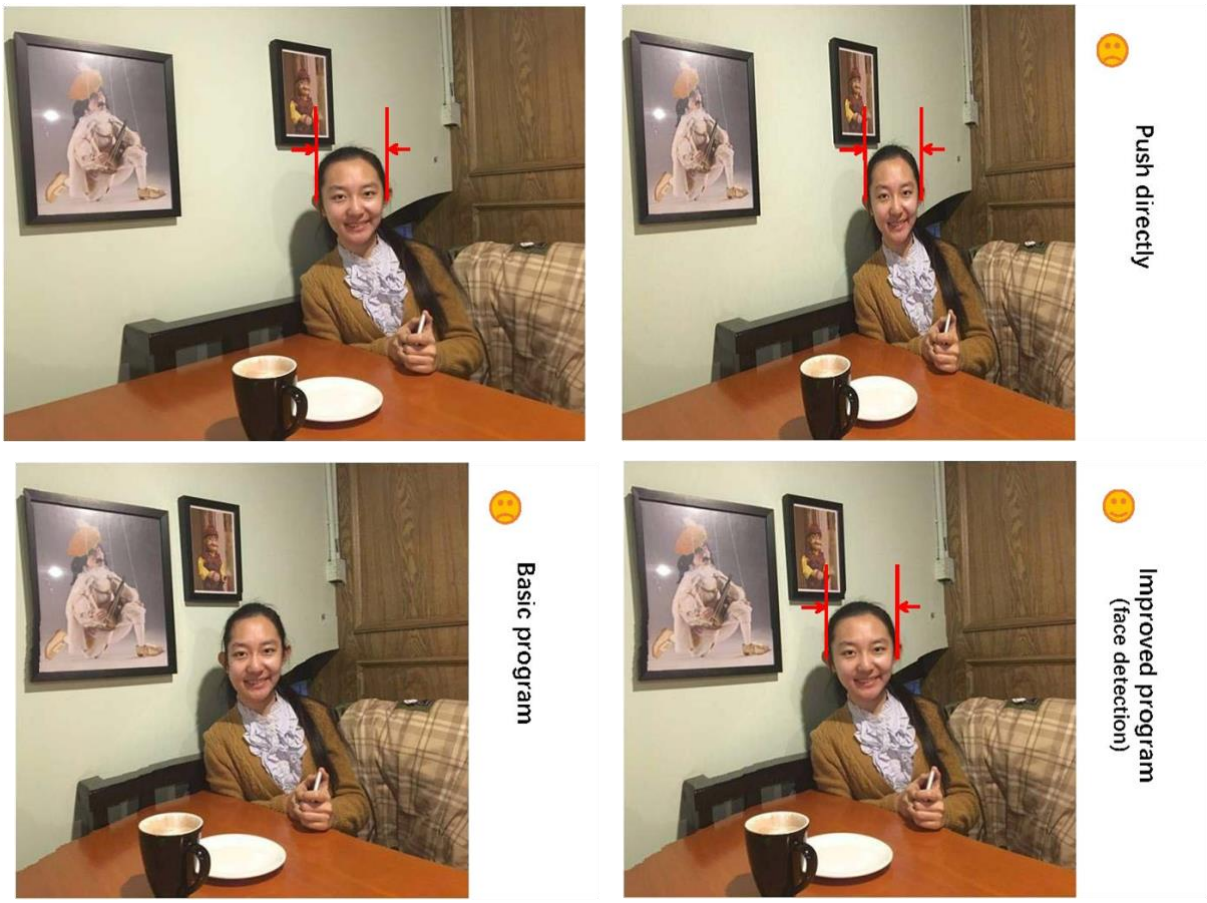

Fig. 2.4 Comparison of the pictures processed by traditional method (pushing directly), basic program and improved program with face detection

2.4 Improvement II-customization

To meet users' more special needs about processing the pictures, the improved program also has customization function. The customization function allows users to choose which part they specifically want to reserve. Then, the program will avoid this part when finding seams to delete. For example, in common situations, grasses are not very important picture contents, since there are so much grasses and people will not care if some part of the grasses is deleted. But if someone wants to emphasize the enormous area of the grasses when processing the picture, with customization function, he can circle the grasses and reserve all of the grasses when resize the picture. The customization function provides users the opportunity to interact with the computers, which makes this program more useful and intelligent.

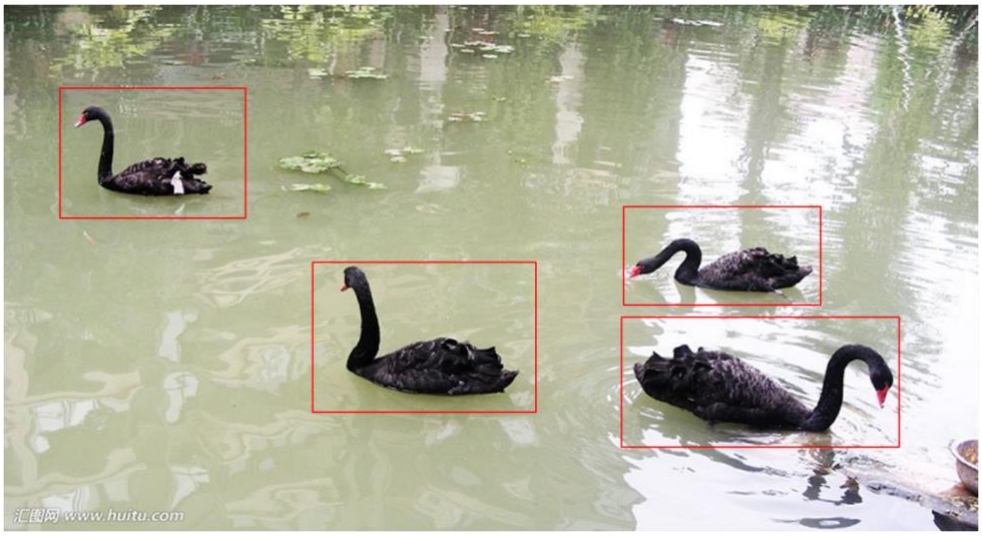

Fig. 2.5 Exhibition of how users can choose the part that they specifically want to reserve

3 Outcome

3.1 The pictures processed by basic program

Fig. 3.1 Successful sample I

(Original picture: waiting for being rescaled; Push directly: being scaled directly; Basic program: the result image of this program)

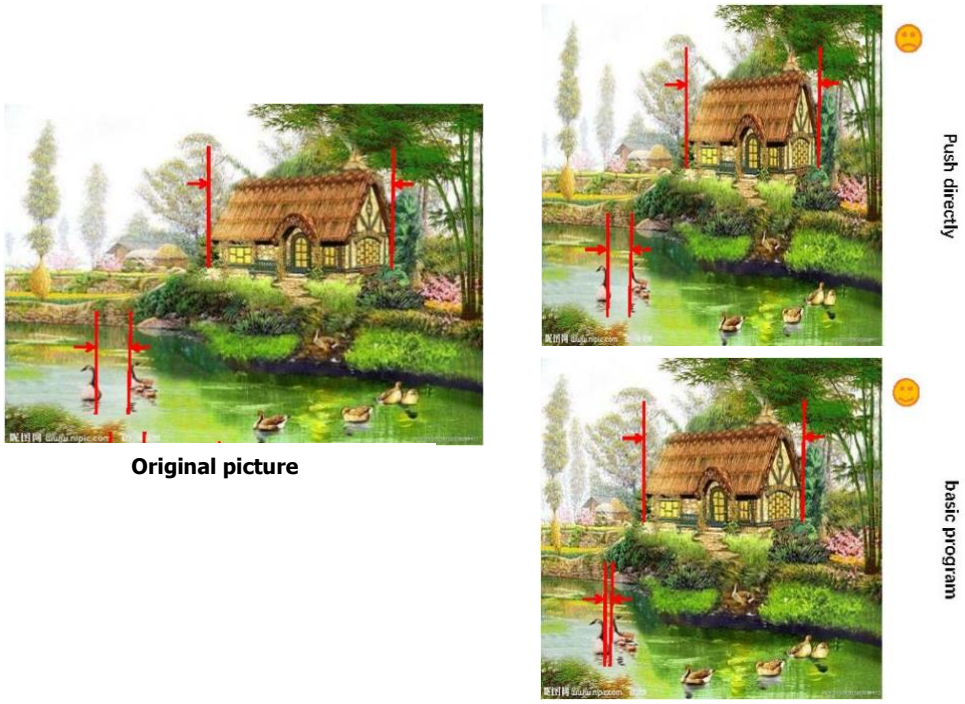

Fig. 3.2 Successful sample II

(Original picture: waiting for being rescaled; Push directly: being scaled directly; Basic program: the result image of this program)

3.2 Pictures processed by improved program

Fig. 3.3 Successful sample III

(Original picture: waiting for being rescaled; Push directly: being scaled directly; Basic program: the result image of the basic program which is not perfect; Improved program: the result image of the improved program which is perfect)

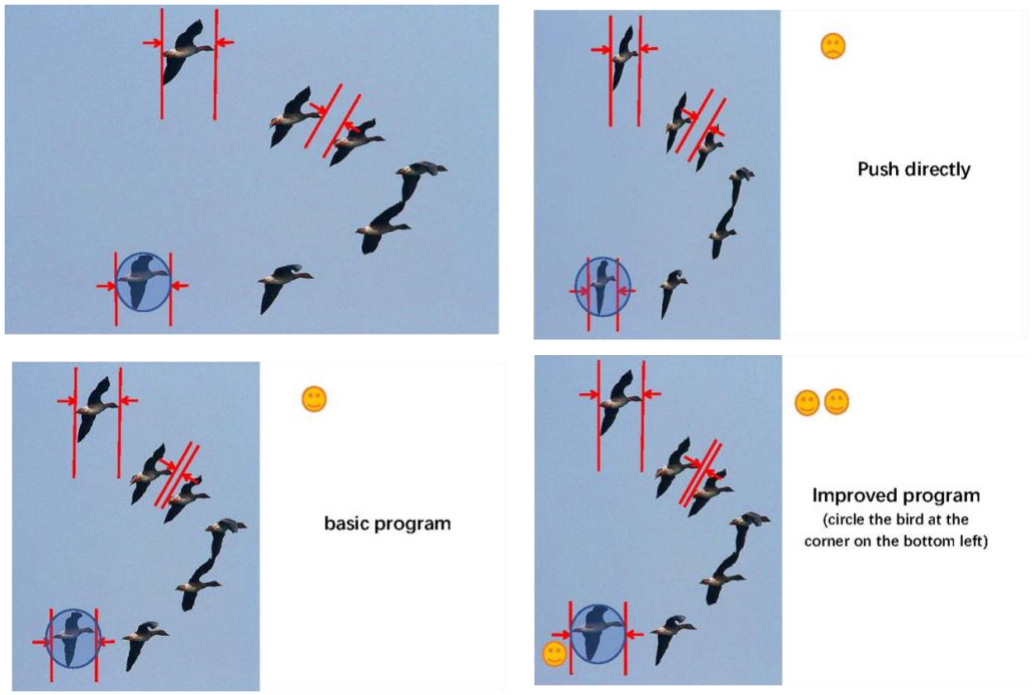

Fig. 3.4 Successful sample IV

(Original picture: waiting for being rescaled; Push directly: being scaled directly; Basic program: the result image of the basic program which is not perfect; Improved program: the result image of the improved program which is perfect)

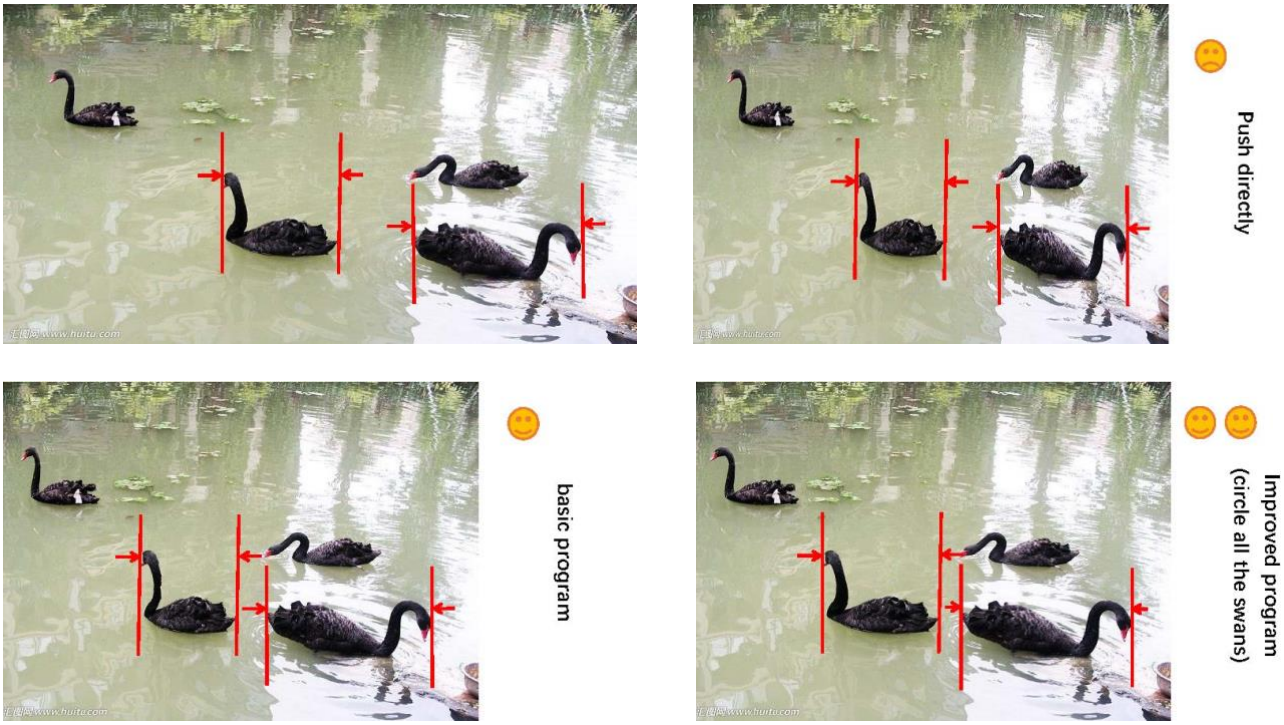

Fig. 3.5 Successful sample V

(Original picture: waiting for being rescaled; Push directly: being scaled directly; Basic program: the result image of the basic program which is not perfect; Improved program: the result image of the improved program which is perfect)

Fig. 3.6 Successful sample VI

(Original picture: waiting for being rescaled; Push directly: being scaled directly; Improved program: the result image of the improved program which is perfect)

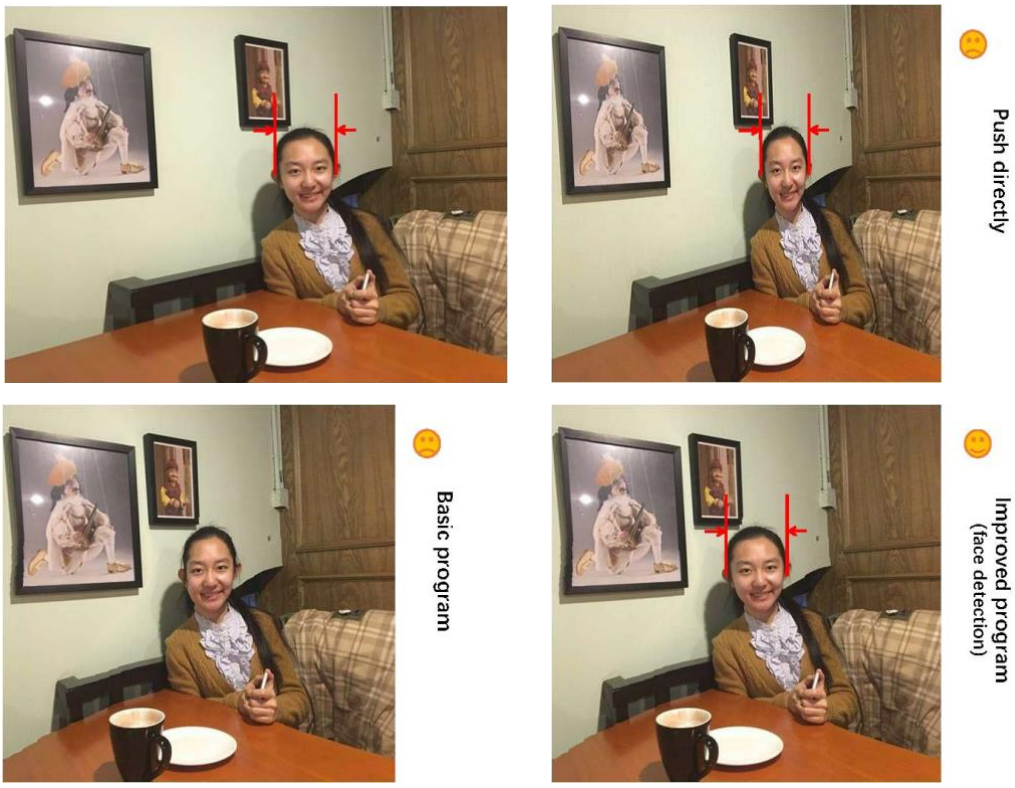

Fig. 3.7 Successful sample VII

(Original picture: waiting for being rescaled; Push directly: being scaled directly; Basic program: the result image of the basic program which is not perfect; Improved program: the result image of the improved program which is perfect)

4 Applications

4.1 Photo Printing

Now, more and more people are equipped with a printer at home, in the supermarket is also very easy to buy like paper, but when some photos taken with mobile phone print out, you will find the phone screen length and width ratio and our traditional Like paper, the aspect ratio varies greatly. This is because the mobile phone developers now for the appearance of mobile phone appearance, the screen size has lost the standardization, has not meet our traditional photos a few inches limit. But this is the case, the print out of the photo or leave a blank space, or photos will be due to changes in the aspect ratio of the deformation, the problem has become a lot of family print photos of a major distress. But our program can help users to achieve a good print.

Sheet. 4.1 Mobile Phones

Model	Resolution	Approximate Ratio
iPhone 4 Series	960*640	3:2
iPhone 5 Series	1136*640	7:4
iPhone 6 & 6S	1334*750	3:2
iPhone 6+ & 6S+	1920*1080	9:5
Others	1280*720 or 2560 * 1440 or …	9:5

Sheet.4.2 Standard Size of Printing Paper (In China)

	Size (cm)	Approximate Ratio
5 cun	8.9*12.7	5:7
6 cun	10.2*15.2	5:7
7 cun	12.7*17.8	5:7

Nowadays, there is a kind of machines which are so welcomed in many shopping malls-the free printers. But the users will be somehow disappointed that the photos

printed out will always have white margins (like the picture showed below). That's because the different size of photos in our phones. If our program can be applied to this kind of machines, we are sure that these machines will be more popular.

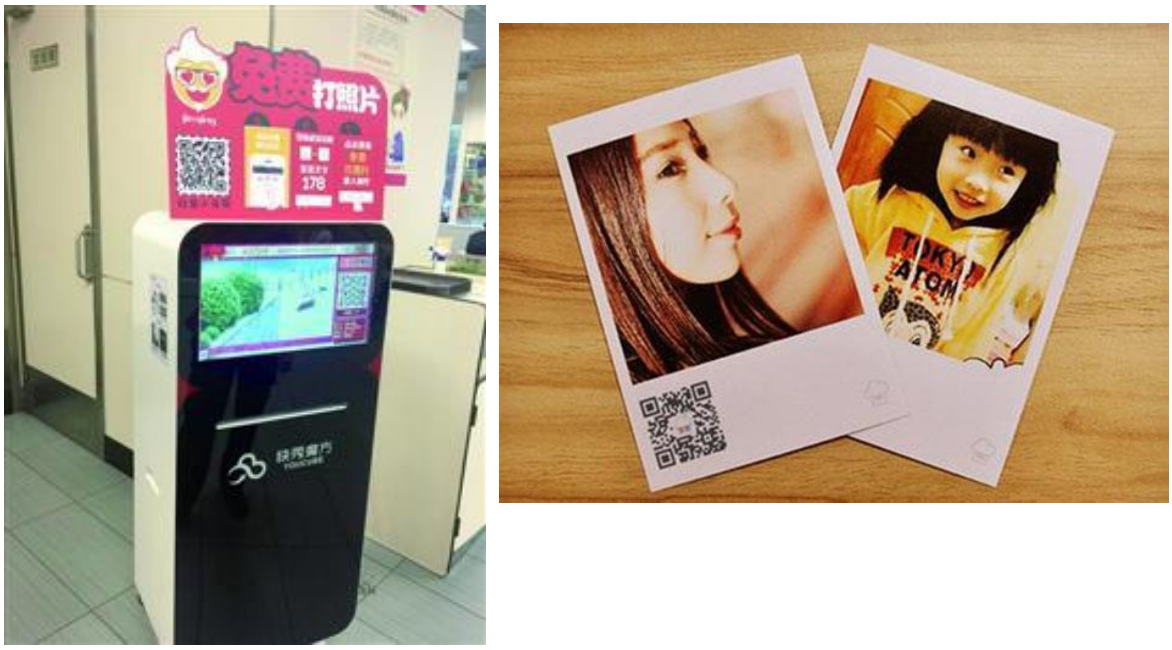

Fig. 4.1 Photo printer and its products

Moreover, many people now want to be their photos as WeChat, QQ picture, or print them out, installed in a small pendant as a decoration. In these cases, people usually want to convert the photo into a square, if you take a direct drawing method, will affect the picture content of the aesthetics, so our program will be based on image content to cut, to meet people's ideal for photo scaling requirements.

4.2 Picture resizing on different devices

With more and more kinds of technology devised, we can use various devices to take pictures or recording videos. But due to the different sizes of the devices' screens, it's hard to make full screen without distorting the image content when displaying the photos or video on other devices. For example, in our daily life, many families would like to buy wide-screen TVs. But they will find that many objects show by their TVs seem to be "fatter", which damages the visual effect and even go against the basic aim of wide-screen TV-to have better visual effect. If our program can be applied to these situations, these problems will be solved.

4.3 Website making and the similar fields.

Having made surveys among those whose jobs are associated with pictures, such as the website makers or even some people who want to add pictures to their papers, essays or presentation, we find that many people are annoyed by a problem: usually the sizes of the original pictures doesn't meet their need. For example, to a website maker, he will get pictures as the resources of the website from various sources, such as the Internet, photographer's work and so on. But usually the sizes of these pictures doesn't fit his design of the website's composition (for instance, he reserves a square space for the picture but actually he gets a rectangular one). He has to resize the pictures with the conventional method only to get a picture with malformation, which has a bad effect on the visual effect of the website. To a broader context, many people have met the problem of the deformity of picture content due to the resizing with traditional method. 7 of 10 people, according to our survey, said that they always add pictures to their papers or presentations and they are looking forward to an app which can change the size of pictures optionally while reserving the original visual effect. If our program can be made as an app, it will solve many problems and have a broad use.

5 Conclusions:

The essence of resizing pictures (here, I use the situation of compressing the pictures as an example) is to delete some part of the picture so that the width of the picture will decrease. Different from the traditional method which compress the picture directly and make every part of the picture thinner than before, this program can resize the picture intelligently. This program will only delete the useless part of this picture and reserve the important part. This will improve the visual effect of the processed picture. The pixel weight is used to define the extent of importance for most of the pixels. For the exceptional situation-the pictures with human faces, the face detection function is added to help better define the weights of each pixels. Also, this program has customization function will can meet users more special needs. The combination of pixel weight, dynamic programming, face detection and customization can perfectly guarantee that the program change the ratio of an image at the same time the important contents are all reserved. Pixel weight and face detection are two major tools in my program to find important pixels automatically, which is a core part of my program. Customization is a very interesting and useful function of my program, for in fact different users may have different command about how to process the image and many people actually want to interact with the computers to tell them their special needs about how to process the images.

References

1. Richard S. Computer Vision: Algorithms and Applications. Springer, 2010

2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. Massachusetts Institute of Technology, 3rd, 2009

3. 毛星云，冷雪飞等. OpenCV3 编程入门. 电子工业出版社，2015

4. Avidan S, Shamir A. Seam Carving for Content-Aware Image Resizing[J]. ACM Transactions on Graphics, 2007, 26 (3) :10

5. Shen J, Wang D, Li X. Depth-aware image seam carving[J]. Transactions on cybernetics, 2013,43(5):1453-1461

6. Zhao D, Wang B. Optimized content-aware image resizing with merging and improved importance diffusion[J]. Journal of harbin institute of technology (new series), 2015,22(2):67-73

7.Boykov Y, Jolly M. Interactive graph cuts for optimal boundary & region segmentation of objects inn-dim-ages. International Conference on Computer, 2001,ICCV(1):105–112.

8. Suh B, Ling H, Bederson BB, Jacobs, DW. Automatic thumb nail cropping and its effectiveness. InUIST’03: Proceeding soft the16th annual ACM symposium on User interface software and technology, ACMPress, 2003, NewYork, NY, USA, 95–104

9. Liu H, Xie X, Ma W, Zhang H. Automatic browsing of large pictures on mobile devices. Proceeding soft the eleventh ACM international conference on Multimedia, 2003:148–155

10. Belongie S, Malik J, Puzicha J. Shape matching and object recognition using shape contexts[J]. IEEE Transactions on pattern analysis and machine intelligence, 2002(24):509-522

11. Zhang GX, Cheng MM, Hu SM, Martin RR. A shape-preserving approach to image resizing[J]. Computer Graphics Forum (Pacific Graphics). 2009,28(7):1897-1906