摘要

利用單一影像做到超解析度影像重建(image super-resolution reconstruction)是近幾年來在電腦視覺、圖學領域上熱門的研究領域。在科技發展快速的時代，許多行動裝置提供的低解析度影像已不符合大眾需求，因此如何利用一些現有的低解析度影像分析其資訊進而做到無失真、可信並快速的超解析度影像放大成為一大課題。

這篇論文中我們提出一個新的方法，結合影像增強、相似區塊填補和利用邊緣資訊內插法去重建出一張粗略放大影像。利用影像增強，我們可以保留更多邊緣資訊避免過度平滑並且讓它看起來更接近原始圖。接著利用邊緣區塊相似性去填補讓邊緣連接處較平順，最後再用邊緣資訊像是梯度、向量...等內插，使整張圖概略放大。最後在利用顏色、距離、梯度、梯度值去做精鍊，在合適的時間內保留原始圖片特徵做到少失真的超解析度影像重建。

Abstract

Single image super-resolution has recently become a hot topic in computer vision and computer graphics communities. This technology had been applied on various media devices in our daily life. The problem how to enlarge images without artifacts and in real time is the core of super-resolution we need to solve. We propose a new method combining image enhancement and interpolation according to edge gradient information. We enhance the original input image at first. This can avoid oversmooth and let the image looked more like the ground truth image. Then we use edge direction and color information to interpolate the unknown pixels, this way can help retain the edge structures roughly. After rough image interpolation we refine the images making it accurate. This method preserves the original features and the nature texture and the whole system can be executed in adaptive time.

Acknoledgement

時光飛逝，在交大學習兩年的日子轉眼間就過了。在交大有共同努力修課的同學、有陣容強大的系排、有充滿歡笑和搞笑的撞球社，更有親切幽默的指導教授。於此我誠摯的感謝我的指導教授林奕成教授，老師耐心的引導我找尋碩論的方向，細心的教導我報告的技巧和專業的知識讓我更有自信迎接人生下一個挑戰。在老師幽默的帶領下，讓 meeting 氣氛和樂融融，也鼓勵我們多多發問。無論在待人處世或是在專業領域我都在老師身上學到了許多，也成長了許多，我想這是我在交大最大的收穫。

我也要感謝我的父母親，感謝我的姐姐，在我迷惑時為我指引方向，在我看不懂 paper 時和我共同奮戰，一起玩樂、一起熬夜，永遠在背後支持我，是我最大的力量泉源。

感謝我的同學映萱、星寒和世佑在課業上常常幫助我這少一根筋的同學，時常忘東忘西，多虧有你們的提醒我才不至於忘記繳學費、學分費、沒印審定書...等等，讓我們在未來的道路上一起加油吧！

Table of Contents

List of Figures

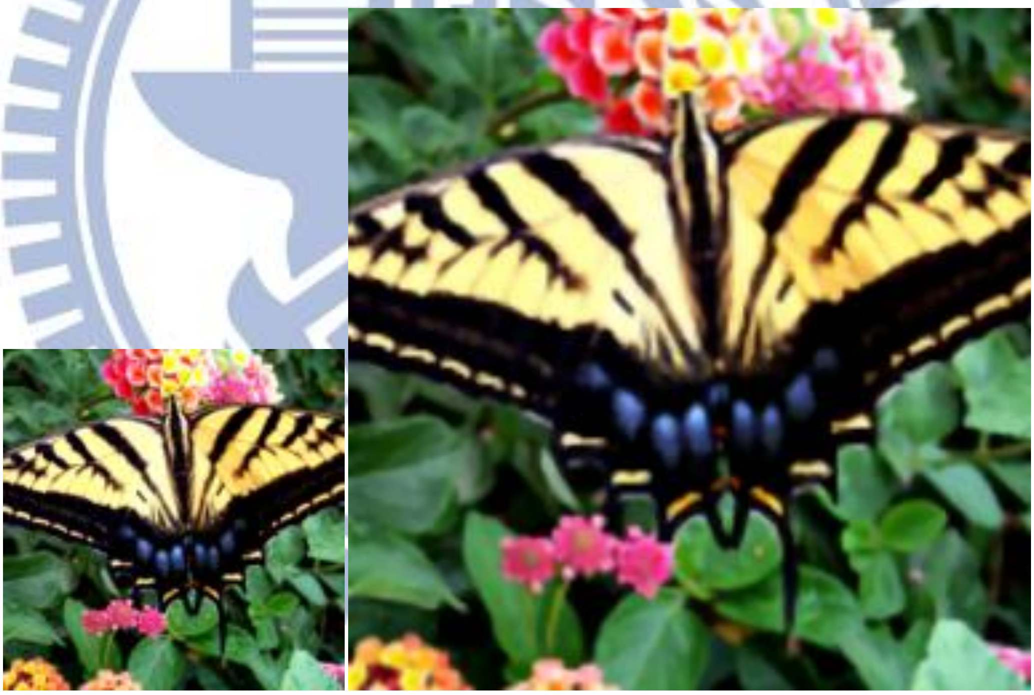
List of Tables

Chapter1

Introduction

1.1 Image Super-resolution

With the progress of science and technology, more and more digital devices can display high quality images but there still exist a lot of devices like mobile phones or surveillance cameras capturing only low-resolution images. Furthermore, image upscaling is applied to many devices that we usually use like digital TV, video telephone, media players, car safety view video system, and so on in our daily life. So, the image super-resolution, or image upscaling, becomes a big challenge and important research field.



(a) (b)

(c)

Figure 1.1: Example of image super-resolution. (a) Original image. (b) Upscale image by input image information to predict unknown pixels. (c) The ground truth image.

The task of image super-resolution is using the information of low-resolution image to predicting the high-resolution image. There are multitudinous unknown pixels needed to be predicted. Many researches addressed this topic and we summarize them into three categories**:** pixel interpolation, example-based super-resolution and edge adaptive methods.

The ways using pixel replication, nearest neighbor, bilinear or bicubic interpolation belong to the first category-pixel interpolation. Although, they have light computational complexity compared to the others but they also have some visual artifacts like ringing, jagged edges, pixelization and oversmoothing.

To avoid these artifacts and to preserve nature texture, several researches proposed example-based super-resolution. In this way, the systems need a huge database including various kinds of images. And then, it searches the database to find one patch which is the most similar to the target patch. Beside on the retrieved patch, this kind of methods acquire the high frequency information and refine the upscaled image. However, how to construct a proper database including sufficiently representative set of examples is another problem. Then other researchers proposed multiframe super-resolution, finding the similar patches in a single image at different scales. This way only need single image and get satisfactory performances. The edge adaptive methods like new edge-direction interpolation (NEDI) or improved NEDI (iNEDI) present impressive appearance but of high computational complexity.

1.2 Motivation

The motivation of the thesis is that we want to solve the artifact such as ringing, jagged edges, and over smoothing produced by the image super-resolution.

Artifact is always a big problem when we upscale images. How to make edges sharp without jaggy and keep flat area smooth simultaneously and how to estimate the original structure, useful features and the nature texture are difficult challenges.

In this paper, we present a system to generate less artifact results. We add high frequency information on input image before coarse image interpolation. Once the original image becomes more distinct, the image upscaled by interpolation will looked more like the real image. Then we proposed a new method to refine pixels value, where neighbors' Euclidian distance, color and gradient information are mapped for the influences of neighbors. Our method can reduce artifact significantly and execute in the adaptive time.

1.3 Thesis Organization

The remaining chapters of the paper are organized as follows. In chapter 2, we introduce the related works. Chapter 3 gives the system overview of our method and introduces how we enhance the original image, fill the similar pair and interpolate the pixels value considering edge direction. In chapter 4, we show how to refine the pixels value. Chapter 5 presents the result of this technique. Finally, the conclusion and future works are presented in chapter 6.

Chapter2

Related Works

Image super resolution has been studied for a very long time. Early, many researchers used interpolation methods for predicting unknown pixels. The values of the new unknown pixels were computed by functions such as linear combination of the original image pixels near the predicted pixel position. The nearest neighbor, bilinear, and bicubic interpolation are familiar to user. Linear interpolation methods are good at computation time. They are efficient, but cause artifacts such as jaggies, ringing, blurring and aliasing. Then some studies proposed using weighted interpolation, such as Li et al. [2001]. They proposed a noniterative scheme where the interpolation is based on the local edge orientation. Su et al. [2004] chose three out of the four nearest pixels for linear interpolation. These methods allow reducing the ringing effects and obtains sharper edges. However, these methods still cannot make the upscaled image "nature". Then Asuni et al. [2008] improved new edge-direction interpolation (NEDI) by finding the adaptive region to estimate the covariance of the low resolution image. This scheme has high quality performance and preserves edge features and fewer artifacts, but it requires intensive computation.

Inspired by nature image statistics, random MarKov field models are used to define probability density of upscaled image. Many researches maximize the model to get the scaled output images. These methods can be classified to two classes : one is based on parametric image model and the other one is not, such as example-based models.

In the work of Freeman et al. [2002], they proposed example-based super-resolution. This method needs a database of example patches. Each patch has two parts. One is the low frequency band; the other is the residual high frequency band. They enlarged input image by interpolation, and then searched the best-matched patches from database and added the high frequency information to the upscaled image. This method relied on a complete database. So choosing which images to construct the database is an essential problem itself.

Besides example based methods, recently, several parametric image models were proposed for super resolution. These methods described and modeled various image features that have dependency at different scales. As Fattal [2007], they modeled the relationship of edge descriptors which are extracted from the input image and the upscaled one. In Sun et al. [2008], a prior for the reconstructed edge profile is proposed. These methods are faster than their example-based counterparts, and can reproduce sharp edge with less noise. Nevertheless, the results are somewhat unrealistic because they are made of generic edges which often separate color plateaus.

Recently, Freedman et al. [2010] proposed a new method extending the example-based super-resolution frameworks. They used “local self-similarity” assumption searching matched patches in extremely local regions. This way replaced the old researches which extracted patches from whole external example database or the whole input image; so it can reduce searching time. Because local self-similarity assumption works well at small scaling factors, they also proposed non-dyadic filter banks in their implementation. The method obtains sharp edges and works efficiently, but it is not effective at handling fine-detail regions such as animal’s fur. In Giachetti et al. [2011], they proposed a new method, iterative curvature-based interpolation, based on filling holes and iterative correcting the values of predicted pixels by minimizing the constraint function.

Chapter3

Coarse Image Upscaling By Interpolation

3.1 System Overview

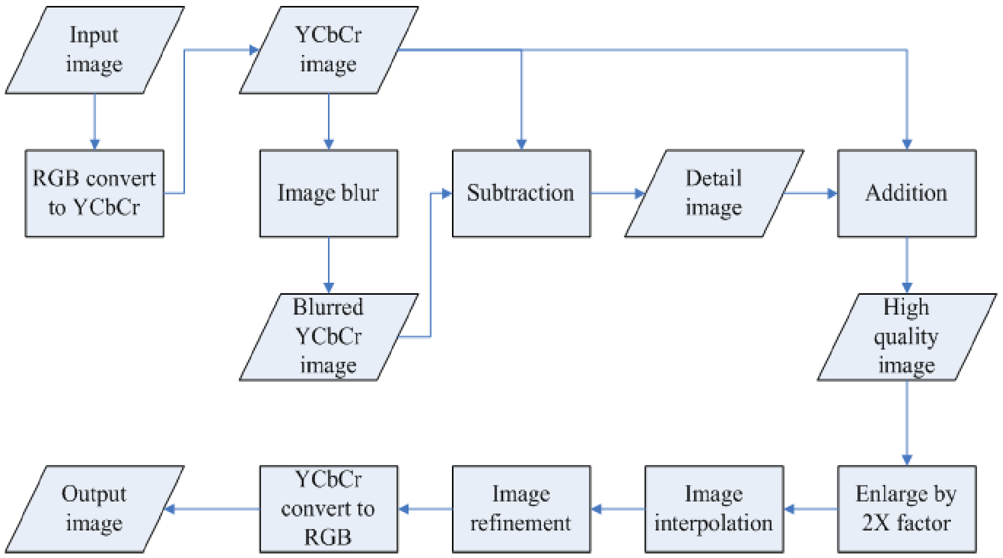


Figure 3.1: System flowchart.

Figure 3.1 depicts the flowchart of our system. We divide the flowchart into two parts: 1) image enhancement and coarse image linear interpolation according to edge direction; 2) image refinement using distance, chrominance and luminance to counting weight.

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

(a)

(b)

(c)

(d)

Figure 3.2: (a) Gaussian filter. (b) Blurred image by Gaussian filter. (c) Down sampling. (d) Original image.

Figure 3.2 shows the comparison between low resolution images downed sample with Gaussian filter and high resolution images. We find out that the low resolution images always loss some high frequency information comparing to the real images. Every real image downscaled must lose some information. So using these images to be the original input source can only interpolate the upscaled images without some important information. This phenomenon can explain that the chrominance, luminance and contrast of the image are not enough in the human eyes. To solve this problem, we first enhance the input image. This can help us to interpolate upscaled image with enough chrominance and luminance in human eyes. Coarse image interpolation may make some wrong pixels value, so we rarefy it to get adaptive pixels value.

3.2 RGB to Y'CbCr Color Space

In this paper, we use Y'CbCr color space to process images. Using Y'CbCr is better than RGB color space due to the two reasons. First, human eyes are sensitive to luminance, so we can focus on adjusting luminance channel while keeping the chrominance distribution identical to original images. It can reduce the calculating time. Second, using Y' channel to process gradient information can effectively decrease the errors making by using RGB three channels to calculate gradient map dividedly.

Y'CbCr is a way of encoding RGB information. Y' is the luma component and Cb and Cr are the blue difference and red difference chrominance components.

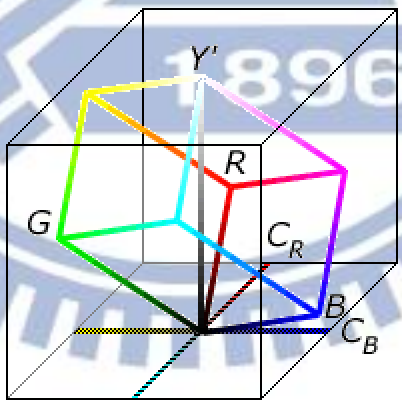


Figure 3.3: The relationship between RGB color space and Y'CbCr color space

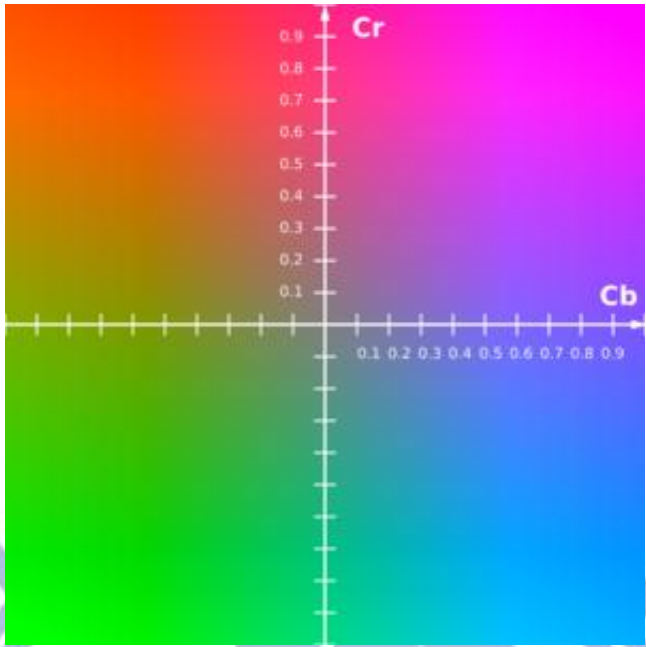


Figure 3.4: The $C_b C_r$ plant at constant luma Y'=0.5

$$Y' = 0.299 \times R + 0.587 \times G + 0.114 \times B \tag{3.1}$$

$$C_b = -0.169 \times R - 0.331 \times G + 0.5 \times B + 128 \tag{3.2}$$

$$C_r = 0.5 \times R - 0.4186 \times G - 0.0813 \times B + 128 \tag{3.3}$$

Formula (3.1) (3.2) (3.3) are the translation of RGB to Y'CbCr formula. R, G, B mean red, green and blue channel values.

$$R = Y' + 1.402 \times (C_r - 128) \tag{3.4}$$

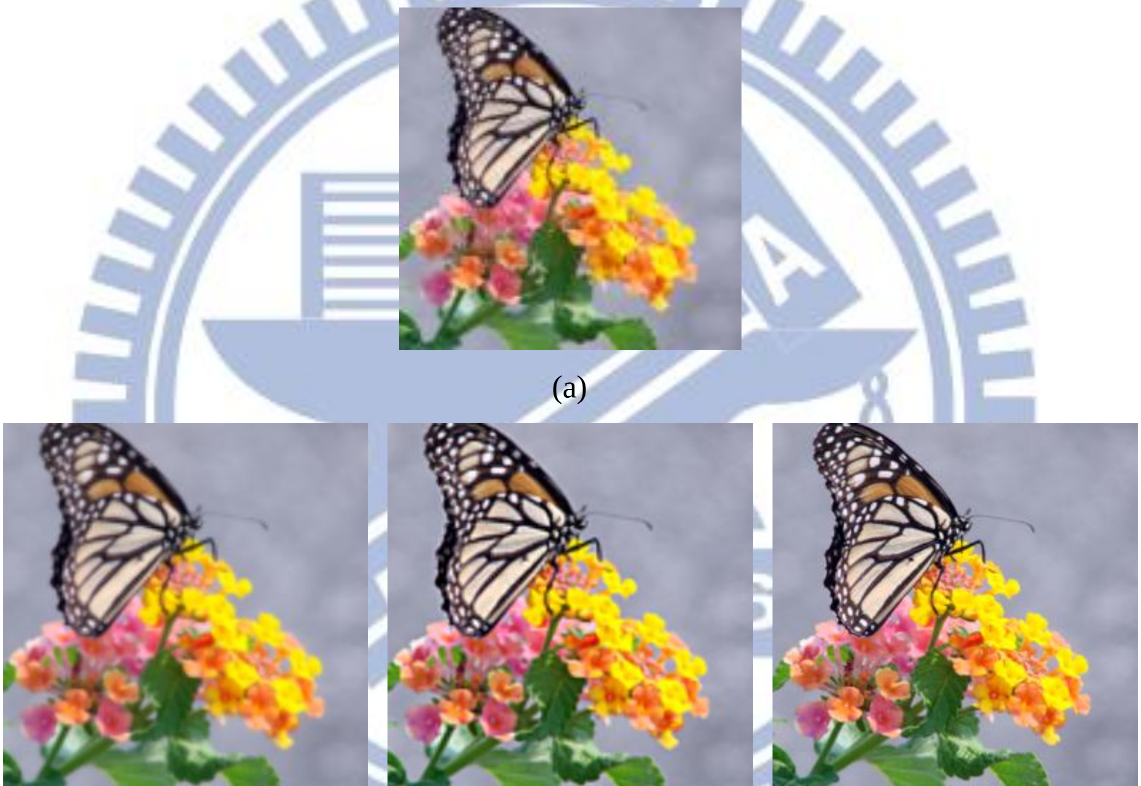
$$G = Y' - 0.3441 \times (C_b - 128) - 0.7141 \times (C_r - 128) \tag{3.5}$$

$$B = Y' + 1.772 \times (C_b - 128) \tag{3.6}$$

Formula (3.4) (3.5) (3.6) are the translation of Y'CbCr to RGB formula.

3.3 Input Image Enhancement

Figure 3.4 (a) is original input image and (b) is the ground truth image (we called the real image in this paper). It depicts that the large-scale image has more details than low resolution image. And, the high resolution image is more colorful and the contrast is obvious. Figure 3.5 (b) is an image enlarged by interpolation method using the original input image. (c) is an image upscaled by enhanced original image. We observed that the image using enhanced input image looked more like real image than the other one, and the image magnified from original input image has less luminance and contrast.



(a) (b) (c) (d)

Figure 3.5: The difference of luminance, chrominance and contrast between upscaled images by interpolation and real image. (a) Original image. (b) High resolution image without image enhancement. (c) High resolution image with enhancement on original image. (d) Ground truth image.

In this part, we blur the input image with Gaussian blur filter at first.

$$\mathrm{I_b} = \mathrm{I_o} * \mathrm{M_G} \tag{3.7}$$

$$M_G = \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

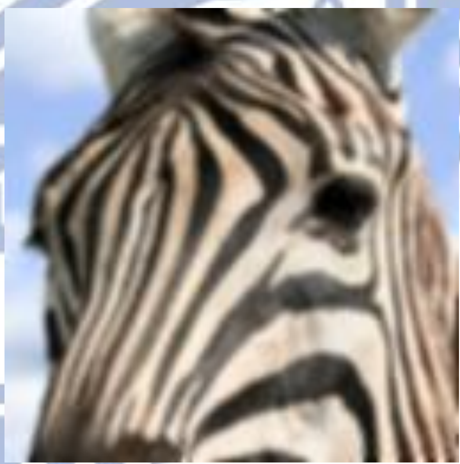
Where I_b is the blurred image, I_o is the original image, $\ast$ operator means convolution and MG is the 3x3 Gaussian blurred mask. After this step, we use original image to subtract the blurred image to getting high frequency information.

$$I_d = I_o - I_b \quad (3.8)$$

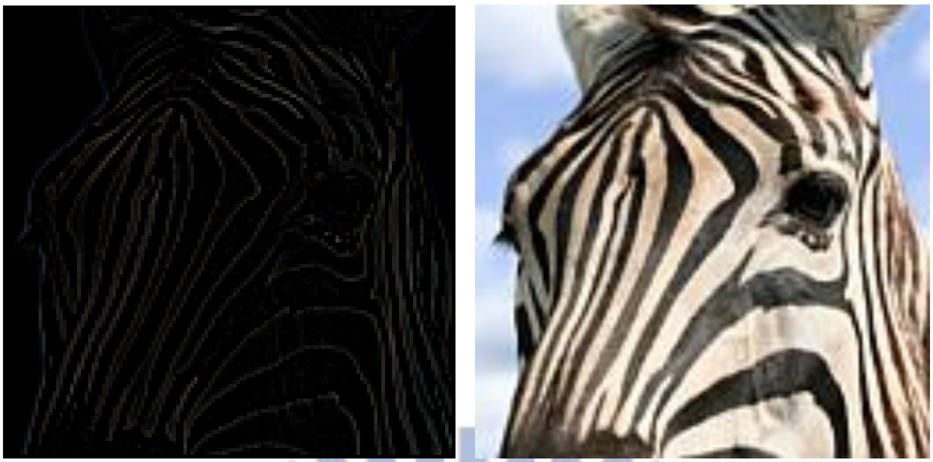
$$I_e = I_o + I_d \quad (3.9)$$

Where I_d is the detail image and we can use it to enhance original image. I_d preserves high frequency information like edges information. Then we add I_d to original image to improve I_o. I_e is the enhanced image. Figure 3.6 depicts the original image (a), blurred image using Gaussian blurred filter (b) and the detail image (c). (d) is the enhanced image.

(a)



(b)

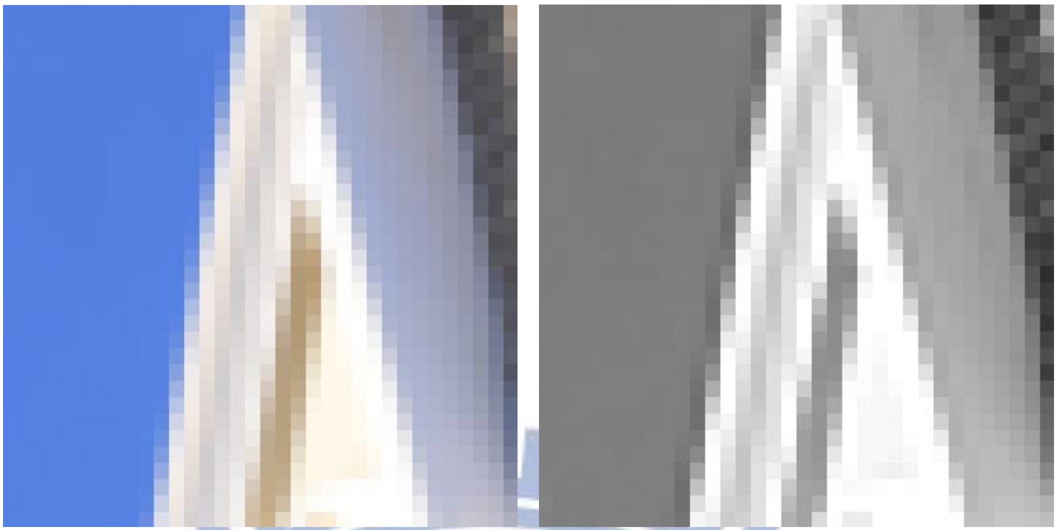


(c) (d)

Figure 3.6: (a) Original input image. (b) Blurred original image. (c) High frequency information. (d) Improved image.

3.4 Similar Triplet Filling

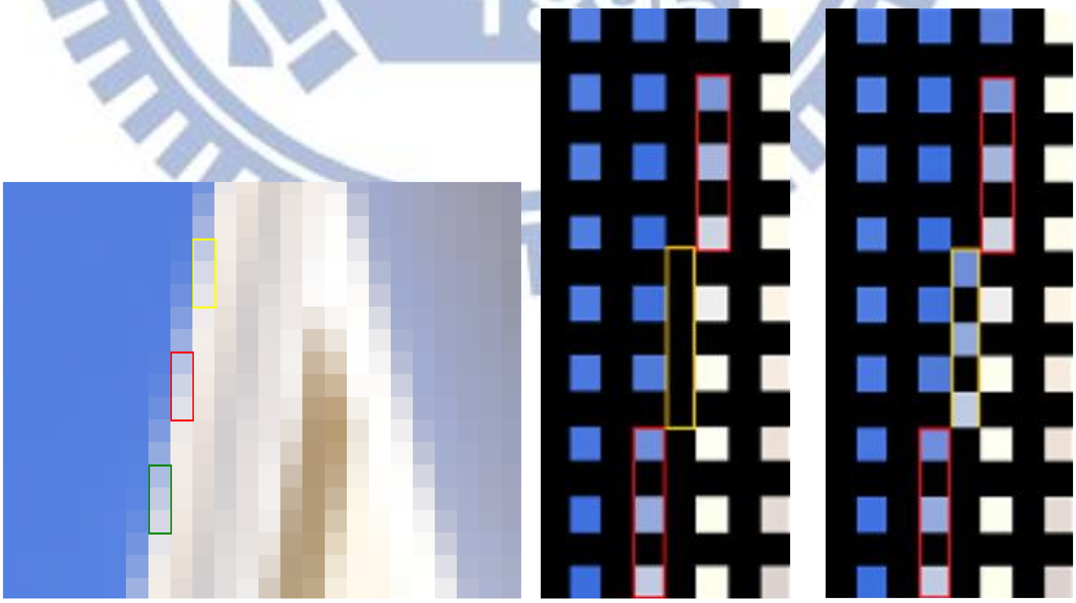
(a)



(b) (c)

Figure 3.7: (a) Original image (b) The detail of edges. (c) The luminance of (b)

The luminance and color usually be strictly increasing or strictly decreasing when there exist edges. In figure 3.7 we show the color and luminance detail on edges. Using this phenomenon we fill the unknown pixels by searching the most similar triplet. Similar triplets filling include two steps. At first we find out all the strictly increasing or decreasing in three pixels. Then search the most similar pattern and fill the pixel between them with their value. (Figure 3.8)



(a) (b) (c)

Figure 3.8: (a) The red grid find two similar triplet. (b) Counting the location

between two matched triplets. (c) Filling value.

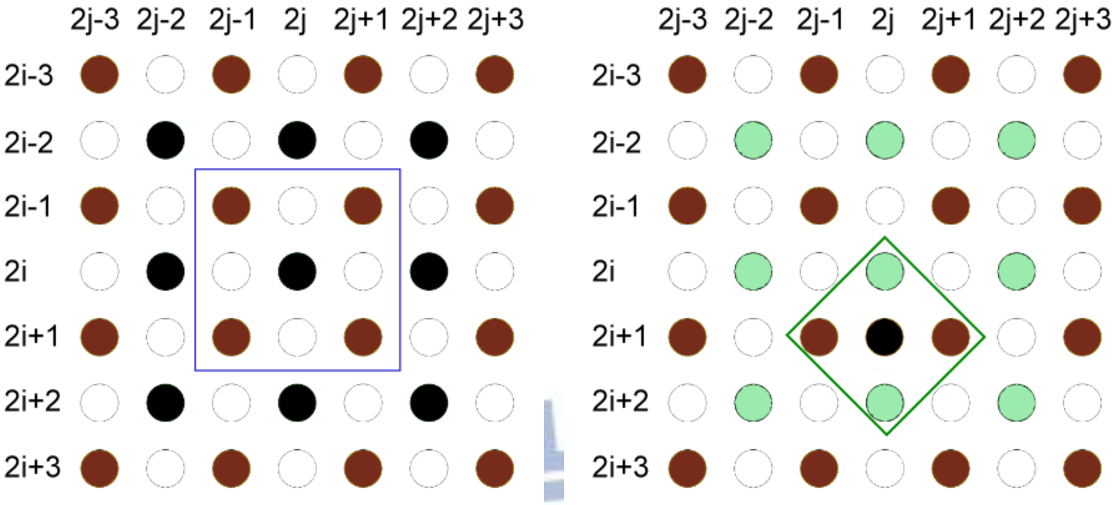
We do similar triplet filling in horizontal, vertical and diagonal direction. Figure 3.9 show the result.

Figure 3.9: The result of similar triplet filling.

3.5 Rough Image Interpolation

Rough image interpolation includes two steps. As we can see in the figure 3.10 (a), we use four diagonal neighbors to fill the unknown pixels (2i , 2j) at first. Then use four nearest neighbors to interpolate unknown pixel (2i+1 , 2j) in figure 3.7 (b). After these two steps, the whole pending pixels are filled.

In order to keep sharp edges, we should know the direction of the edge crossing the unknown pixel is near to which diagonal direction and calculate the weights of these two directions. Then we multiple the weights and the two pixel values on the diagonal to predict unknown values.



(a) (b)

Figure 3.10: Tow steps interpolation using four neighbors. The black pixels are unknown pixels, brown pixels are the enhanced original image and the green pixels in (b) are predicted by first steps pixel values. (a) Use four diagonal neighbors' value to fill the black pixel. (b) Use four nearest neighbors in vertical and horizontal direction to predict the black pixel.

$$\mathrm{gradient}(2i,2j) = \frac{[\,I(2i-1,2j+1)-I(2i+1,2j-1)]}{I(2i-1,2j-1)-I(2i+1,2j+1)} \tag{3.10}$$

$$\theta(2i,2j) = \tan^{-1}\mathrm{gradient}(2i,2j) + \frac{\pi}{2} \tag{3.11}$$

$$I(2i,2j) = w_1 \times \left(\frac{I(2i-1,2j+1)+I(2i+1,2j-1)}{2}\right) + (1-w_1) \times \left(\frac{I(2i-1,2j-1)-I(2i+1,2j+1)}{2}\right) \tag{3.12}$$

$$w_1 = \frac{\frac{\pi}{2}}{\left|(\theta(2i,2j)-\frac{\pi}{2})\right|} \tag{3.13}$$

Formula (3.10) count the gradient value and $\theta(2i,2j)$ is the edge direction. W_1 is the weight of this edge direction. We count the θ value is near $\frac{\pi}{2}$ or far from it. Using the reciprocal of it to be the weight then calculate $I(2i,2j)$ by formula (3.12).

Figure 3.11 shows the result of rough image interpolation using original image without image enhancement.

Figure 3.11: Enlarged image with 2x factor by linear interpolation.

We observe that the image in figure 3.11 avoid jagged edges but still is oversmooth crossing edges. Figure 3.12 display the difference between using enhanced image and original image to be the source image. Obviously, (c) has sharper edges than (b).

(a)

(b) (c)

Figure 3.12: Comparison of image interpolation after enhancement. (a) Original image. (b) Image upscaled without enhancement. (c) Image upscaled with enhancement.

However, we observe the enhanced upscaled images carefully and discover that there are some unnatural pixels on edges. It may appear light ringing effect seriously. (see figure 3.13)

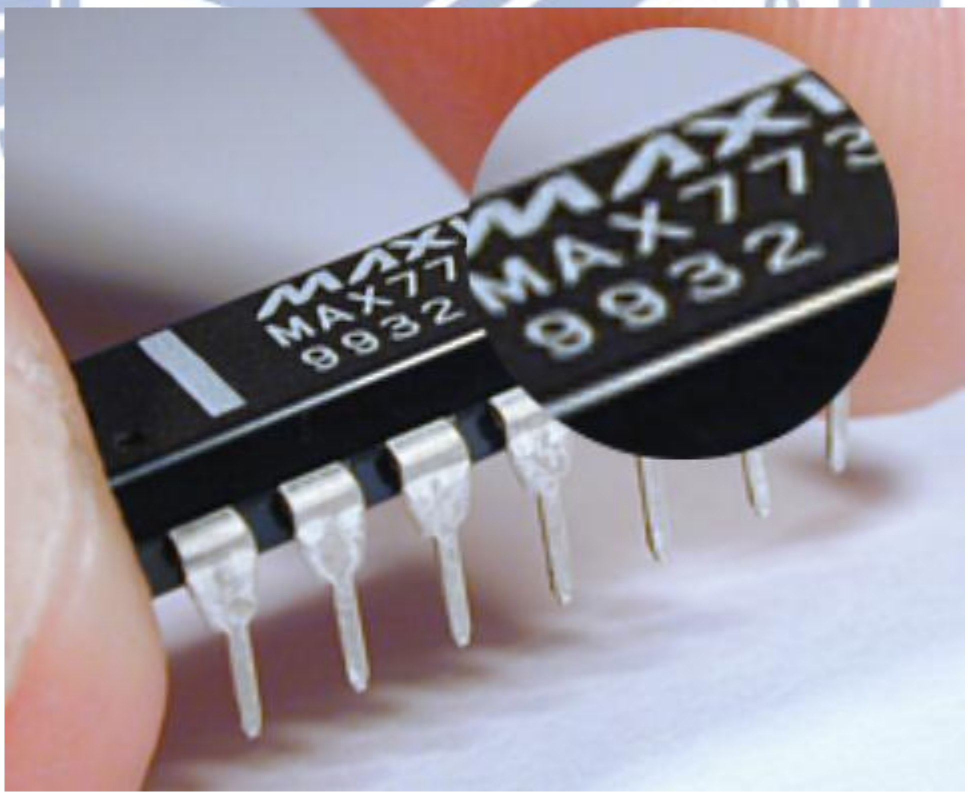


Figure 3.13: It exists some slight artifacts when we enhance the sourse image.

Chapter4

Image Refinement

4.1 Image Refinement

In last chapter, we discuss the effect caused by enhancement. It obtains some artifacts like ringing and jagging, so the final results in chapter3 still need to be revised. In this chapter, we present how to refine the naturalness of images.

The core of this method is setting 5x5 neighbors from a central pixel and itself trusty weights. We propose using distance and color difference and edge gradient information to count the weights.

$$\mathrm{I_r}(c) = \sum_{\mathrm{n}\in\Omega(\mathrm{c})} \mathrm{W}(c,n) \times \mathrm{I}(c) \tag{4.1}$$

,where I_r is the refined image and $\Omega(c)$ is a set of neighbors from pixel c. In this paper we use 5x5 neighbors to calculate weights. W(c,n) is the weight indexed by central pixel c with neighbor pixel n. The weight W consists of four components: distant term, color term , gradient term and magnitude term.

$$\mathrm{W(c,n)} = \exp\left(\frac{-\,\mathrm{t}\times\mathrm{D}\times\mathrm{C}\times\mathrm{G}}{\mathrm{M}}\right) \tag{4.2}$$

Where t is a normalization constant. D term is the Euclidian distance term defined as

$$\mathrm{D} = \sqrt{(\mathrm{c_x}-\mathrm{n_x})^2 + \left(\mathrm{c_y}-\mathrm{n_y}\right)^2} \tag{4.3}$$

C term is the color difference term between n and c defined as

$$\mathrm{C} = (\mathrm{I(c)} - \mathrm{I(n)})^2 \tag{4.4}$$

G term is the gradient term to measuring θ between vector $\overrightarrow{\mathrm{nc}}$ and the gradient vector $\vec{\mathrm{g}}$. If θ is small enough, we believe the pixels on this direction are right values.

$$\cos\theta = \frac{\overrightarrow{\mathrm{nc}}\cdot\vec{\mathrm{g}}}{|\overrightarrow{\mathrm{nc}}||\vec{\mathrm{g}}|} \tag{4.5}$$

$$\mathrm{G} = \begin{cases} 0.5, & |\cos\theta| < 0.08 \\ 1, & \text{else} \end{cases} \tag{4.6}$$

$$\mathrm{M} = \frac{1}{\mathrm{K}}\sum_{\mathrm{n}\in\Omega(\mathrm{c})} \|\overrightarrow{\mathrm{g_n}}\| \tag{4.7}$$

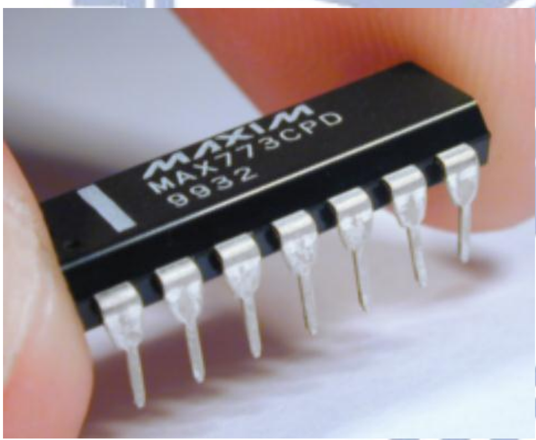
M term is the sum of neighbors' gradient magnitudes. M term increases when the

5x5 area is not smooth and decreases when it is flat. When M term increased the weight term D, C and G will be more sensitive. We can find accurate neighbors to give them major weights.

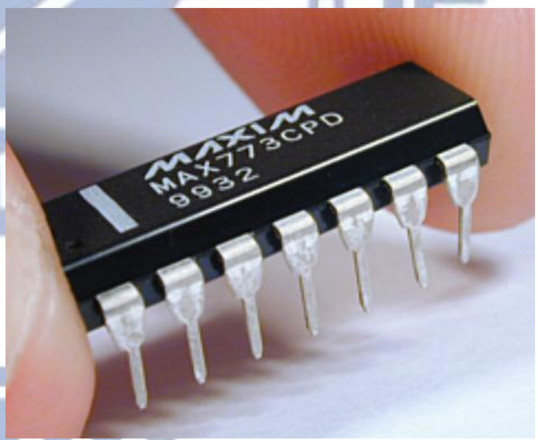
After modify all pixel values, the results are shown in figure 4.1

Figure 4.1: Result of image refinement.

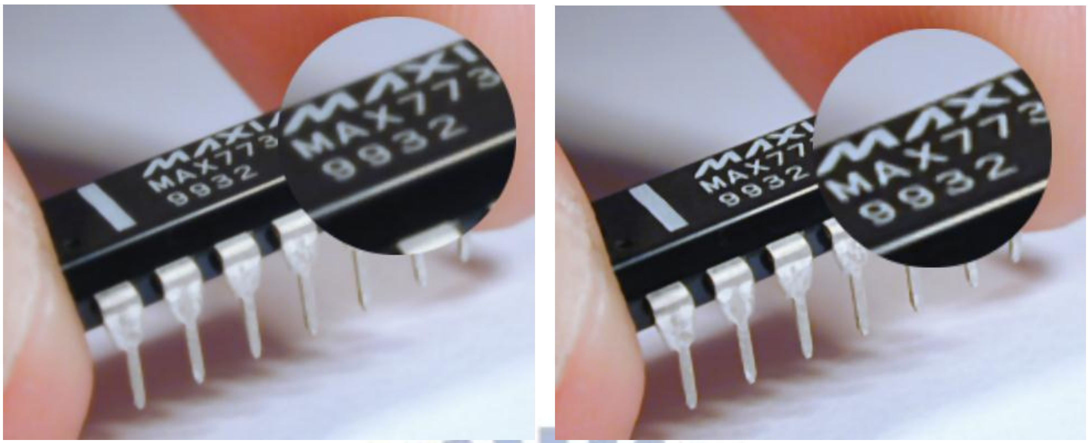
We show comparison of the result with/without image enhancement preprocessing in figure 4.2. The final image (d) has more details than (c).



(a)



(b)



(c) (d)

Figure 4.2: Comparison. (a) The image enlarged by linear interpolation. (b) The image upscaled after enhance the source image. (c) Result of refining (a). (d) Result of refining (b).

Chapter5

Experimental Results and Discussion

5.1 Results

In this section, we show the experimental results of our system. We test our method on a variety of images. Figure 5.1 shows the results using our method with different factors of magnification. Figure 5.2 shows the results with different input images.

(a)

(b)

(c)

Figure 5.1: (a) Input image. (b) Image magnified by 2x factor with our method. (c) Image upscaled by 4x factor with our method.

(a) (b) (c)


(d)


(e)


(f)


(g)


(h)


(i)

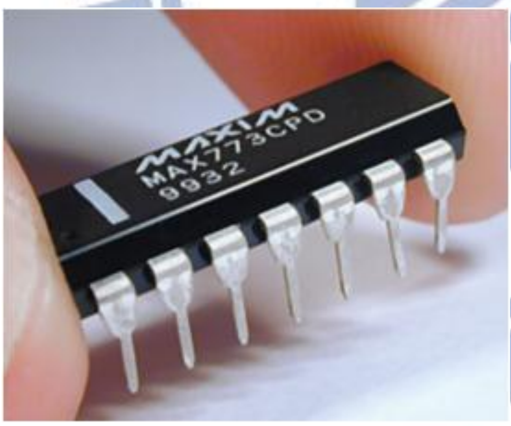

(j)

(k)

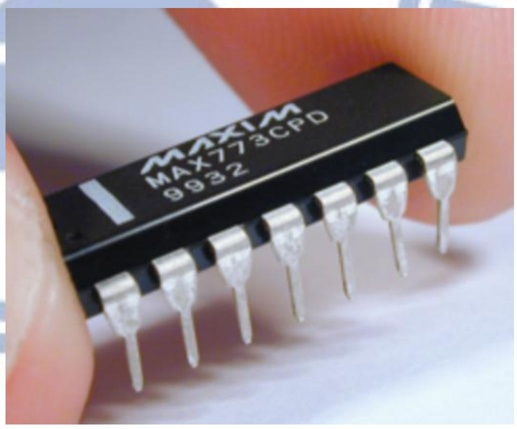
(l)

Figure 5.2: The different input images upscaled with 2X factor. (a) (b) (c) (d) (e) (f) are the original Images and (g) (h) (i) (j) (k) (l) are enlarged image of them.

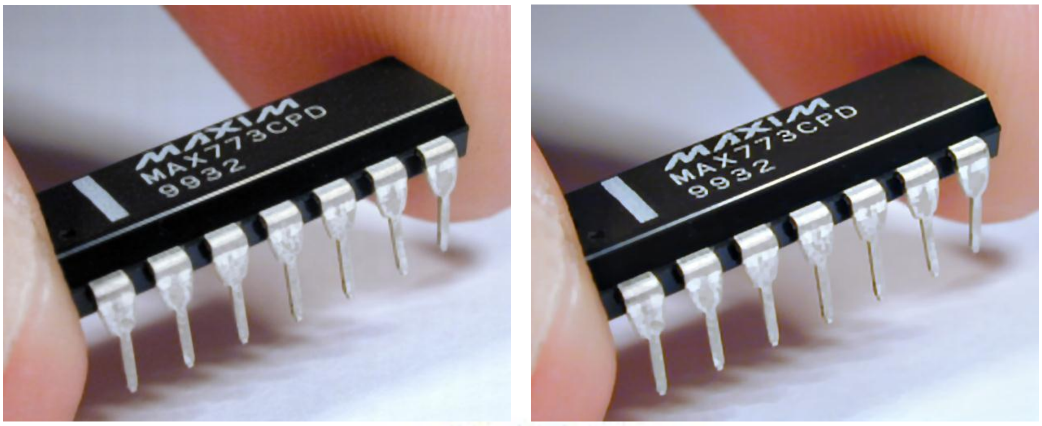
In figure 5.3, figure 5.4 and figure 5.5 show the comparison of different methods. Then figure 5.6 shows the highlights of the results in figure 5.4 and table 5.1 display the PSNR and MSSIM values of each methods.



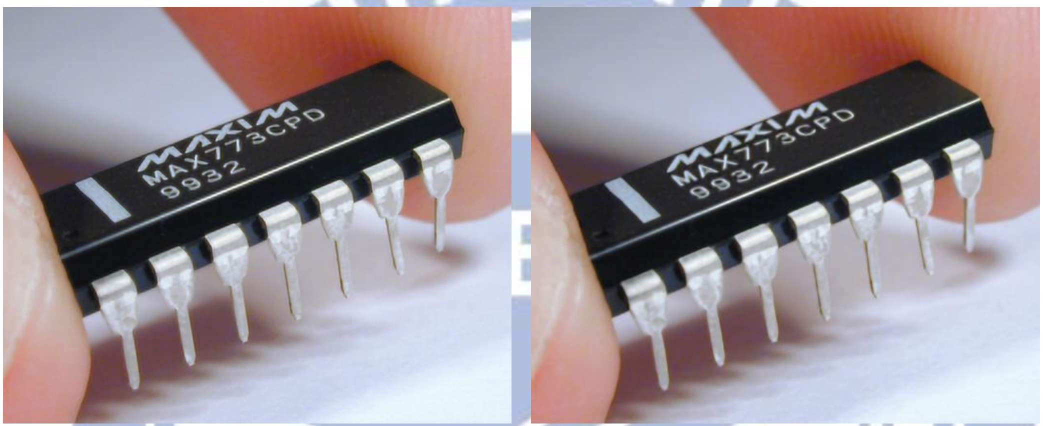
(a)



(b)



(c) (d)



(e) (f)

Figure 5.3: Chip images with 4X factor. Super resolution comparison of different methods.(a) Our method.(b) Bicubic. (c) [Fattal 2007]. (d) [Shan 2008]. (e) [Glasner 2009]. (f) [Freedman 2010].

(a)

(b)

(c)

(d)

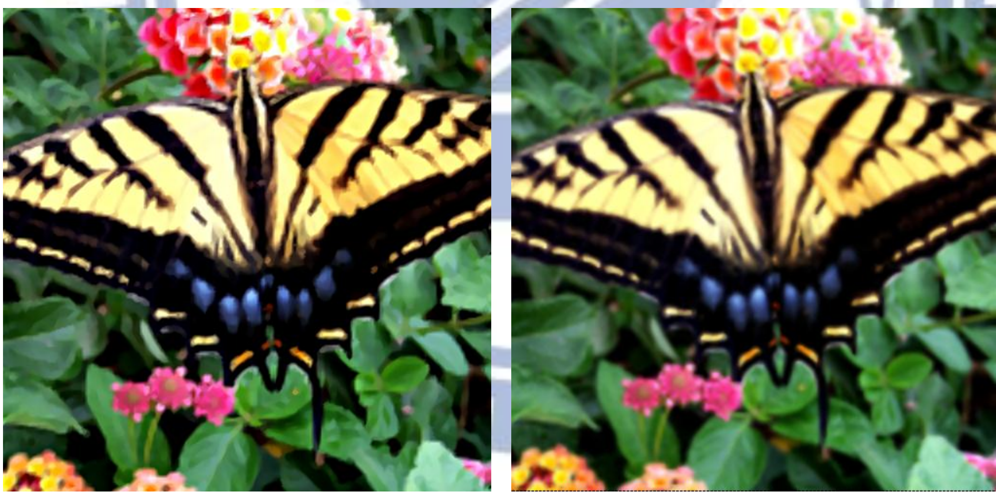
(e)

(f)

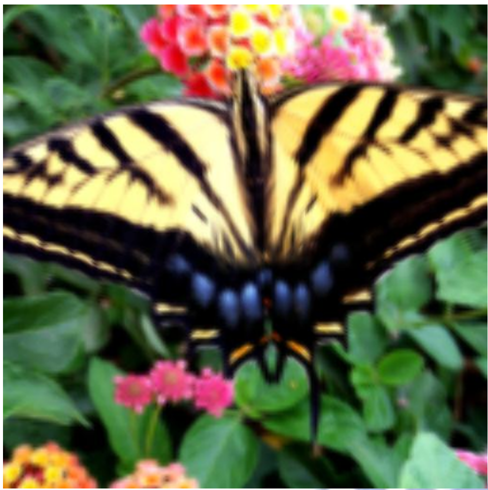
(g) (h)

(i) (j)

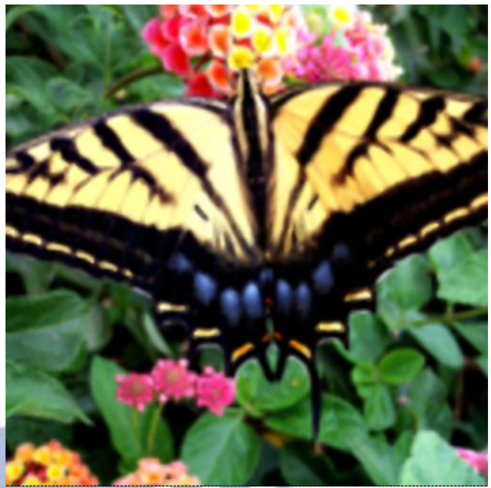
Figure 5.4: Child images with 4X factor. Comparison of different methods.(a) Our method.(b) Bicubic. (c) [Fattal 2007]. (d) [Glasner 2009]. (e) [Shan 2008]. (f) [Freedman 2010]. (g) [HaCohen 2010]. (h) [Yang 2010]. (i) [Tai 2010]. (j) [Giachetti 2011].



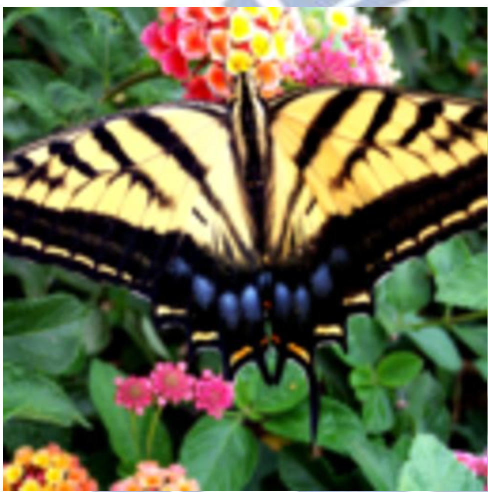
(a) (b)



(c)



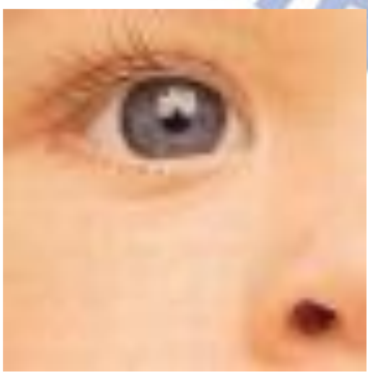
(d)



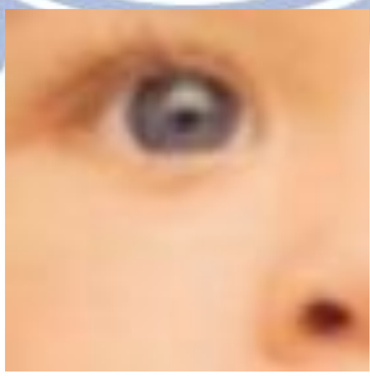
(e)

(f)

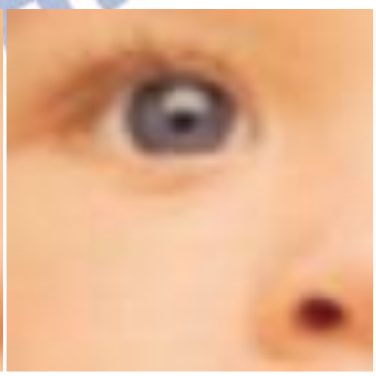
Figure 5.5: (a) Our method. (b) ICBI. (c) NEDI. (d) iNEDI. (e) Bicubic. (f) Ground truth.



(a)



(b)



(c)

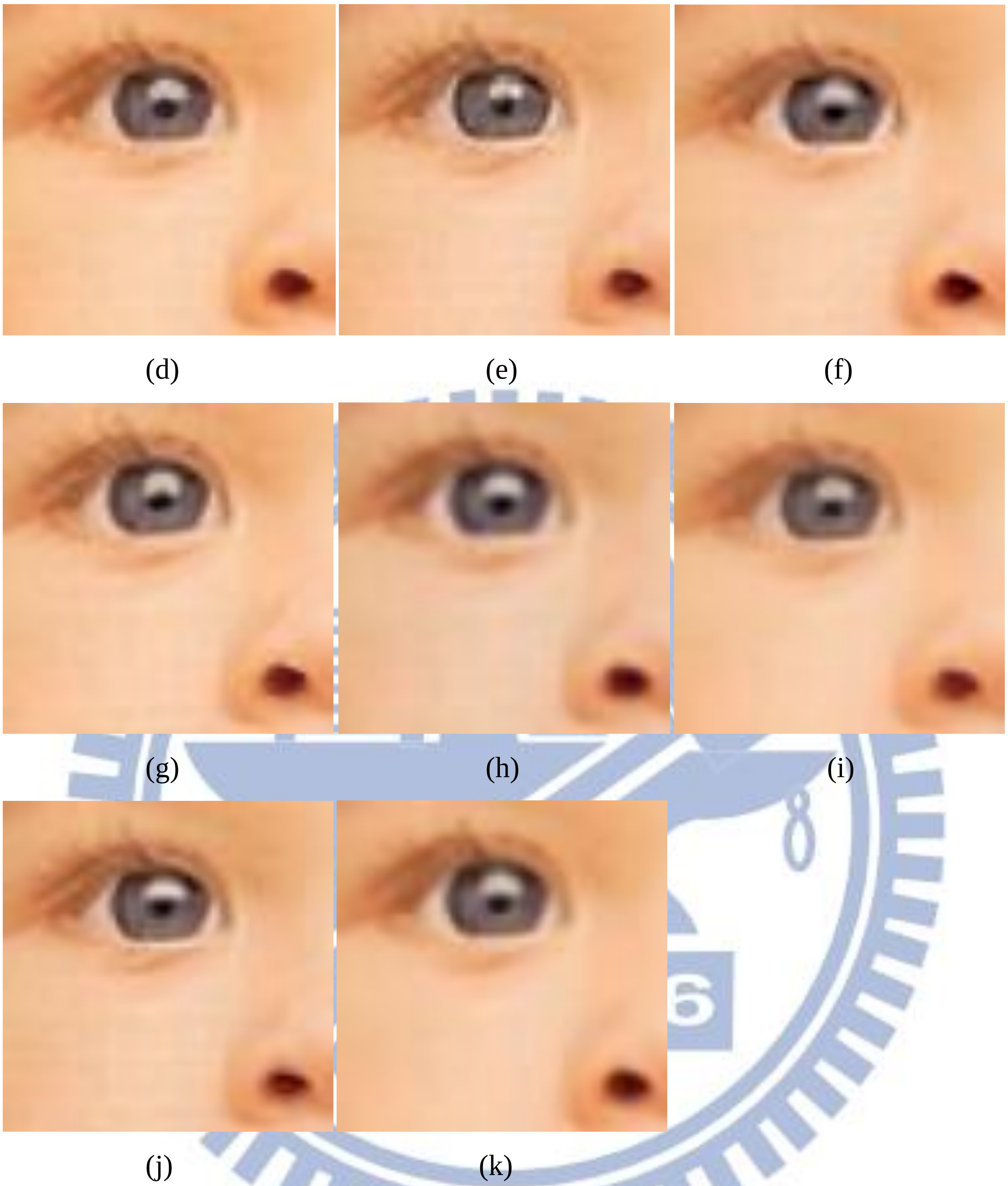


Figure 5.6: The highlight results of child image. (a) Ground truth. (b) [Giachetti 2011]. (c) Bicubic. (d) [Fattal 2007]. (e) [Glasner 2009]. (f) [HaCohen 2010]. (g) [Yang 2010]. (h)[Tai 2010]. (i) [Shan 2008]. (j) [Freedman 2010]. (k) Our method.

	Our	**Giachetti**	**Bicubic**	**Fattal**	**Glasner**
PSNR	32.47	31.07	30.36	29.64	30.63
MSSIM	0.893	0.848	0.819	0.824	0.807

	HaCohen	Yang	Tai	Shan	Freedman
PSNR	31.74	32.57	31.68	29.88	32.25
MSSIM	0.818	0.913	0.889	0.796	0.905

Table 5.1: PSNR and MSSIM values of each method.

5.2 Discussion

In chapter 5.2 we discuss each method and their advantages and disadvantages. In the work of Giachetti et al. [2011], they proposed a new method, iterative curvature-based interpolation, based on filling holes and iterative correcting the values of predicted pixels by minimizing the constraint function. This method consider the edge information, but the results shown in figure 5.4 (j) are still blur.

Glasner et al. [2009] proposed an approach combining classical multi-image super-resolution and Example-Based super-resolution. Their method use only input image without any extra database or prior example images. However, this method has serious ring effect shown in figure 5.3 (e) and figure 5.6 (e) depicts that the edge of nose is a straight line not curve.

Freedman et al. [2010] used "local self-similarity" assumption searching matched patches in extremely local regions. The method obtains sharp edges and works efficiently, but it is not effective at handling circle and fine-detail regions such as animal's fur. In figure 5.6 (j), the baby's eye is approximate hexagon not round.

Shan et al. [2008] proposed a method minimizing the energy function using feedback-control framework. This method is excellence in computation time but the quality of result images is not excellent enough. Figure 5.3 (d) and figure 5.4 (e) show the results. They are not sharp and exist alias.

In the work of HaCohen et al.[2010], they assumed that an image is a union of regions which can be describe by distinct texture models. They combined patch-based super-resolution and texture synthesis. This method can generate the missing high-pass band making the results realism in the reconstructed textures. In figure 5.4 (g) shows the result of this method and we can find out that it contains more details than each other result. However, this approach need example texture database and the computation time is approximately thirty minutes upscaling an 128x128 image with 4 factors.

Yang et al. [2010] proposed a method that use self-similarities and generate image pairs from the input image pyramid of one single frame. This patch pairs are clustered for training a dictionary used to construct super resolution image. Comparing with other results, the result shown in figure 5.3 (h) has fewer artifacts and the PSNR and MSSIM values are the best. However, although our PSNR and MSSIM are lower than it our computation time is extremely fewer than it.

In the work of Tai et al. [2010], they proposed a method that reconstruct significant edges in the original image using gradient profile prior and take the missing detail from a user-supplied example texture or image. The result in figure 5.4 (i) shows that it is excellent at dealing with circle such as the baby's nostril or eyeball but it has serious ring effect at baby face's outline and edges are not sharp enough.

Our results is shown in figure 5.3 (a), figure 5.4 (a), figure 5.5 (a) and figure 5.6 (k). We perform excellent at dealing edges and circle. Our average execution time is about 16.73 seconds upscaling 128x128 image with 4X factor and our PSNR and MSSIM values are higher than most of methods and nearly equal to the others.

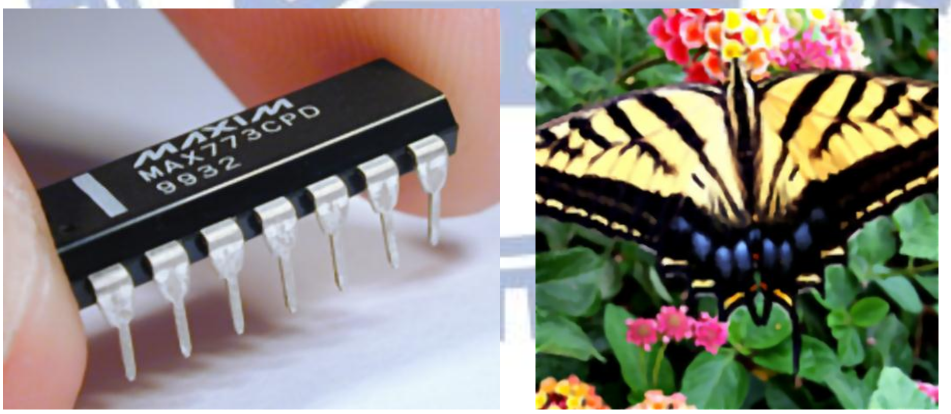
Chapter6

Conclusion and Future Works

In this paper, we propose a new method to do image super-resolution. This system includes two main parts. The first part is coarse image upscaling by interpolation described in chapter3, and the other part is image refinement described in chapter4.

The coarse image super-resolution includes two steps. We first enhance the original input image to get one details included image. Then we do similar triplet filling and use four neighbors pixels value to interpolate a coarse upscaled image. This image may include several wrong pixels value and the edges may have some ringing effect.

Another main point is image refinement. Once we obtained a rough magnified image, we renew every pixel value. We count each weights of the neighbors between center pixel, in this paper we calculate 5x5 neighbors, according the Euclidian distance, color difference and gradient information. We can see some the results in chapter5 and figure 6.1. Our method can keep original structure and features with fewer artifacts.



(a) (b)

Figure 6.1: (a) Chip magnified with 2x factor. (b) Butterfly magnified with 4x factor.

In the future, there are still some parts that can be improved in our system. How to more accurately predict the pixels value by interpolation and more efficiently reduce calculation time are both important works.

Reference

[1] ASUNI, N. and GIACHETTI, A. 2008. "Accuracy improvements and artifacts removal in edge based image interpolation," in Proc. 3rd Int. Conf. Computer. Vis. Theory Appl. (VISAPP), pp. 58–65.

[2] FATTAL, R. 2007. "Image upsampling via imposed edge statistics." ACM Trans. Graph. 26, 3, 95.

[3] FREEMAN, W. T., JONES, T. R., AND PASZTOR, E. C. 2002. "Examplebased super-resolution." IEEE Comput. Graph. Appl. 22, 2, 56–65.

[4] FREEDMAN, G. and FATTAL, R. 2011. "Image and video upscaling from local self-examples." ACM Trans. Graph. 30, 2, Article 12.

[5] GIACHETTI, A. AND ASUNI, N. 2011. "Real-Time Artifact-Free Image Upscaling," IEEE Image Process., vol. 20, no. 10, pp. 2760-2768.

[6] LI, X. AND ORCHARD, M. T. 2001. "New edge-directed interpolation." IEEE Trans. Image Processing 10, 10, 1521–1527.

[7] SU, D. AND WILLIS, P. 2004. "Image interpolation by pixel-level datadependent triangulation." Computer Graphics Forum 23, 2, 189–202.

[8] SUN, J., XU, Z., AND SHUM, H.-Y. 2008. "Image super-resolution using gradient profile prior." In Computer Vision and Pattern Recognition, CVPR 2008. IEEE Conference on. Pages 1–8.

[9] LI, X. AND ORCHARD, M.,T. 2001. "New edge-directed interpolation," IEEE Trans. Image Process., vol. 10, no. 10, pp. 1521–1527.

[10] GIACHETTI, A. AND ASUNI, N. 2008. "Fast artifact free image interpolation," In

Proc. BMVC 2008

[11] SUN, J., XU, Z. AND SHUM, H. 2008. “Image super-resolution using gradient profile prior.” In CVPR08, pages 1–8, 2008.

[12] GIACHETTI, A. AND ASUNI, N. 2008. “Accuracy improvements and artifacts removal in edge based image interpolation.” In Proc. 3rd Int. Conf. Computer Vision Theory and Applicatoins (VISAPP’08), 2008.

[13] CHEN, M., HUANG, C. AND LEE, W. 2005. “A fast edge-oriented algorithm for image interpolation.” Image and Vision Computing, 23:791-798

[14] KIM, K. I. AND KWON, Y. 2008. “Example-based learning for single-image super-resolution.” In Proceedings of the 30th DAGM symp. on Patt. Rec., pages 456–465

[15] OUWERKERK, J. 2006. “Image super-resolution survey.” Image and Vision Computing, 24:1039–1052

[16] HACOHEN, Y., FATTAL, R. AND LISCHINSKI, D. 2010. “Image Upsampling via Texture Hallucination” In Computational Photography ICCP 2010. IEEE International Conference on. pages 1-8

[17] YU-WING, T., SHUAICHENG, L., BROWN, M., S. AND LIN, S. 2010.” Super Resolution using Edge Prior and Single Image Detail Synthesis” In Computer Vision and Pattern Recognition CVPR 2010. IEEE Conference on, pages 2400-2407

[18] CHIH-YUAN, Y., JIA-BIN, H. AND MING-HSUAN, Y. 2010. ” Exploiting Self-Similarities for Single Frame Super-Resolution” In Asian Conference on Computer Vision ACCV 2010, pages 497-510

[19] GLASNER, D., BAGON, S. AND LRANI, M. 2009. “Super-Resolution From a

Single Image." In Computer Vision ICCV, 2009 IEEE International Conference on , pages 349-356