Report of Final Project
Introduction to MATLAB
Southern University of Science and Technology
May 28, 2020, Shenzhen, Guangdong, CHINA

Introduction to MATLAB-ME112

IMAGE SUPER-RESOLUTION AND FRAME INTERPOLATION

Xudong Han[1]

[1]Southern University of Science and Technology, Shenzhen, China

ABSTRACT

Super resolution (SR) is a method to improve the resolution of the original image, which is the reconstruction process of high resolution (HR) image using low resolution (LR) image. Frame interpolation or motion-compensated frame interpolation (MCFI) is a form of video processing in which intermediate animation frames are generated between existing ones by means of interpolation, in an attempt to make animation more fluid.

Keywords: super-resolution, frame interpolation

1. INTRODUCTION

When dealing with the video of LEGO ® BUGATTI Chiron 3D Modeling, a CAD and Engineering Drawing course project, I wonder if it is possible to conduct high-fidelity amplification of video images, namely super-resolution reconstruction, and frame insertion compensation to make the motion smoother. After consulting the literature, I determined two algorithms for image super-resolution, namely bicubic interpolation and super-resolution convolutional neural network (SRCNN), and one algorithm for frame interpolation, namely phase-based video frame interpolation (PBVI). All three algorithms can run completely only depend on CPU platform.

2. METHODS

The project used three algorithms. And in this part, I will show the basic algorithm thought.

2.1 Bicubic Interpolation

Bicubic interpolation is an extension of cubic interpolation for interpolating data points on a two-dimensional regular grid. The interpolated surface is smoother than corresponding surfaces obtained by bilinear interpolation or nearest-neighbor interpolation. In contrast to bilinear interpolation, which only takes 4 pixels (2×2) into account, bicubic interpolation considers 16 pixels (4×4). Images resampled with bicubic interpolation are smoother and have fewer interpolation artifacts.

Bicubic interpolation needs to applying a convolution with the following kernel as Equation (1), and the plot of it shown in Figure 1.

$$W(x) = \begin{cases} 1 - 2.5|x|^2 + 1.5|x|^3, & |w| < 1 \\ 2 - 4|x| + 2.5|x|^2 - 0.5|x|^3, & 1 \le |w| < 2 \\ 0, & |w| \ge 2 \end{cases} \quad (1)$$

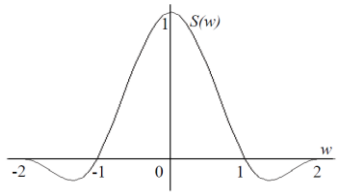

FIGURE 1: BICUBIC INTERPOLATION KERNEL

Bicubic interpolation function is shown in Equation (2), in which $f(i,j)$ express the value at (i,j) in Figure 2.

$$f(i+u, j+v) = \boldsymbol{ABC} \quad (2)$$

where

$$\boldsymbol{A} = [W(1+u)\ W(u)\ W(1-u)\ W(2-u)] \quad (3)$$

$$\boldsymbol{B} = \begin{bmatrix} f(i-1,j-2) & f(i,j-2) & f(i+1,j-2) & f(i+2,j-2) \\ f(i-1,j-1) & f(i,j-1) & f(i+1,j-1) & f(i+2,j-1) \\ f(i-1,j) & f(i,j) & f(i+1,j) & f(i+2,j) \\ f(i-1,j+1) & f(i,j+1) & f(i+1,j+1) & f(i+2,j+1) \end{bmatrix} \quad (4)$$

$$\boldsymbol{C} = [W(1+v)\ W(v)\ W(1-v)\ W(2-v)]^T \quad (5)$$

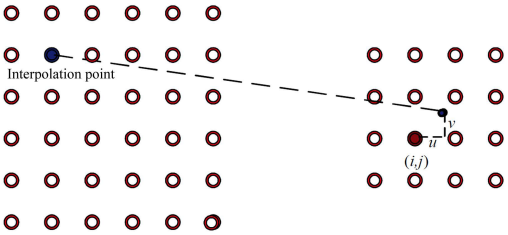

FIGURE 2: BICUBIC INTERPOLATION MAPPING

When bicubic interpolation algorithm is used for image super-resolution, the values of the intermediate points of each original coordinate are calculated through the above formula to form a map, and then the image amplification can be completed. This method is simple and fast, but has limitations in image quality.

2.2 Super-resolution Convolutional Neural Network

Super-resolution convolutional neural network (SRCNN) is another model for image super-resolution.

We first upscale it to the desired size using bicubic interpolation, which is the only pre-processing we perform3. Denote the interpolated image as $\boldsymbol{Y}$. Our goal is to recover from $\boldsymbol{Y}$ an image $F(\boldsymbol{Y})$ which is as similar as possible to the ground truth high-resolution image $\boldsymbol{X}$. For the ease of presentation, we still call $\boldsymbol{Y}$ a "low-resolution" image, although it has the same size as $\boldsymbol{X}$. We wish to learn a mapping F, which conceptually consists of three operations:

1. Patch extraction and representation: this operation extracts (overlapping) patches from the low-resolution image $\boldsymbol{Y}$ and represents each patch as a high-dimensional vector. These vectors comprise a set of feature maps, of which the number equals to the dimensionality of the vectors.

2. Non-linear mapping: this operation nonlinearly maps each high-dimensional vector onto another high-dimensional vector. Each mapped vector is conceptually the representation of a high-resolution patch. These vectors comprise another set of feature maps.

3. Reconstruction: this operation aggregates the above high-resolution patch-wise representations to generate the final high-resolution image. This image is expected to be similar to the ground truth $\boldsymbol{X}$.

An overview of the network is depicted in Figure 3.

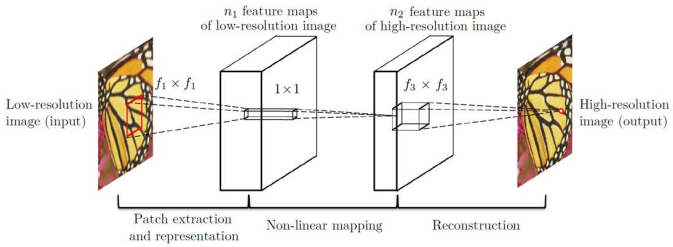

Figure 3: OVERVIEW OF NETWORK

This SRCNN is quite similar to sparse-coding-based method. The latter method first cuts out small patches from the original image and preprocess (normalization). The cutting is dense, meaning there is overlap between the patches. Then sparse parameters are obtained by encoding with a low-resolution dictionary. All the sparse parameters were used to reconstruct with high -resolution dictionary. Finally, all the small patches are spliced together, and the overlapping parts are spliced with weights.

Compared with sparse coding, the SRCNN model is end-to-end, which is convenient for optimization. Moreover, the least squares solution is not required, so the computation speed is faster.

Model training is another important work. The model used is trained by ImageNet datasets. Some patches are randomly selected from the high-resolution image, then down-sampled and up-sampled and we get a low-resolution image which are used as the input. The original high-resolution image is taken as the target, and the per-pixel loss is used as the optimization target.

It is possible to add more convolutional layers to get better quality image. But this can increase the complexity of the model significantly and it will need more time and data to train, which will lose the efficiency as the quality promotion is not obvious.

2.3 Phase-based Video Frame Interpolation

Phase-based approaches build on the insight that the motion of certain signals can be represented as phase-shift. We first explain the basic concepts, from 1D to 2D case.

Consider a one-dimensional sinusoidal function shown in Figure 4 which is defined as $y = Asin(\omega x - \phi)$, where A is the amplitude, ω the angular frequency and ϕ the phase. A translation of this function can be described by modifying the phase, e.g. by subtracting $\pi/4$ in our example. The phase shiftϕ_{shift}, which corresponds to the actual spatial displacement between two translated functions, is defined as the phase difference ϕ_{diff} between the two phases of the curves scaled by ω:

$$\phi_{shift} = \frac{\phi_{diff}}{\omega} \tag{6}$$

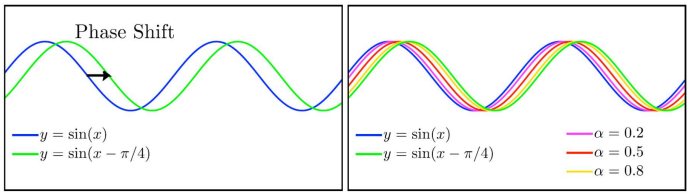

Figure 4: THE TRANSLATION OF TWO SINUSOIDAL FUNCTIONS (LEFT) CAN BE INTERPOLATED ACCORDING TO EQUATION (2) (RIGHT)

Let us now modify the phase difference according to a weight $\alpha \in (0,1)$ that describes an intermediate position between the functions:

$$y = Asin(\omega x - \alpha\phi_{diff}) = Asin(\omega(x - \alpha\phi_{shift})) \tag{7}$$

The resulting functions then correspond to intermediate sinusoids representing the translational motion, see Figure 5.

This idea can be extended to general functions $f(x)$ translated by a displacement function $\delta(x)$. The shifted function $f(x + \delta(t))$ can be represented in the Fourier domain as a sum of complex sinusoids over all frequencies ω:

$$f(x + \delta(t)) = \sum_{\omega=-\infty}^{\omega=\infty} R_\omega(x, t) \tag{8}$$

where each sinusoid represents one band $R_\omega(x,t) = A_\omega e^{i\omega(x+\delta(t))}$. The corresponding phase $\phi_\omega = \omega(x+\delta(t))$ can be directly modified with respect to α, leading to modified bands

$$\hat{R}_\omega(x,t) = A_\omega e^{i\omega(x+\alpha\delta(t))} \tag{9}$$

The in-between functions are then obtained by integrating the modified bands in accordance to Equation (8).

For two dimensional functions one can separate the sinusoids into bands not only according to the frequency ω, but also according to spatial orientation θ, using e.g. the complex-valued steerable pyramid.

The complex-valued response $R_{\omega,\theta}$ obtained by applying the steerable filters $\Psi_{\omega,\theta}$ to an image I can be written as:

$$R_{\omega,\theta}(x,y) = (I * \Psi_{\omega,\theta})(x,y) \tag{10}$$
$$= A_{\omega,\theta}(x,y)e^{i\phi_{\omega,\theta}(x,y)} \tag{11}$$
$$= C_{\omega,\theta}(x,y) + iS_{\omega,\theta}(x,y) \tag{12}$$

where $C_{\omega,\theta}$ is the cosine part, representing the even-symmetric filter response, and $S_{\omega,\theta}$ is the sine part, representing the odd-symmetric filter response. From this we can compute the amplitude $A_{\omega,\theta}(x,y) = \sqrt{C_{\omega,\theta}(x,y)^2 + S_{\omega,\theta}(x,y)^2}$, and the phase $\phi_{\omega,\theta}(x,y) = arctan(S_{\omega,\theta}(x,y)/C_{\omega,\theta}(x,y))$.

Based on the assumption that small motion is encoded in the phase shift, interpolating it requires the computation of the phase difference ϕ_{diff} between the phases of the two input frames as

$$\phi_{diff} = atan2(sin(\phi_1 - \phi_2), cos(\phi_1 - \phi_2)) \tag{13}$$

where atan2 is the four-quadrant inverse tangent. This approach results in angular values between $[-\pi,\pi]$, which correspond to the smaller angular difference between the two input phases. It additionally determines the limit of motion that can be represented, which is bounded by

$$\left|\phi_{shift}\right| = \frac{|\phi_{diff}|}{\omega} \leq \frac{\pi}{\omega} \tag{14}$$

where $\omega = 2\pi v$, and v being the spatial frequency.

In the multi-scale pyramid, each level represents a particular band of spatial frequencies $v \in [v_{min}, v_{max}]$. Assuming v_{max} corresponds to the highest representable frequency on that level, then a phase difference of π corresponds exactly to a shift of one pixel. While this is a reasonable shift for low frequency content represented on the coarser pyramid levels, it is too limiting for high frequency content to achieve realistic interpolation results in the presence of larger motions. We need to create a bounding model for shift correction.

Our approach is based on the assumption that the phase difference between two resolution levels does not differ arbitrarily, i.e. phase differences between levels can be used as a confidence measure that quantifies whether the computed phase shift is reliable.

The actual shift correction depends on the difference between two levels, which serves as our confidence estimate

$$\varphi = atan2\left(sin\left(\phi_{diff}^{l} - \lambda\phi_{diff}^{l+1}\right), cos\left(\phi_{diff}^{l} - \lambda\phi_{diff}^{l+1}\right)\right) \tag{15}$$

where the phase value of the coarser level is scaled according to an arbitrary pyramid scale factor $\lambda > 1$ to get a scale-independent estimate. If $|\varphi| > \pi/2$, we apply shift correction and obtain the corrected phase difference as $\tilde{\phi}_{diff}^{l} = \lambda\phi_{diff}^{l+1}$.

While our shift correction allows larger motions to be modeled, there is still a limit to the motion that can be represented without introducing blurring artifacts. We therefore propose an additional enhancement, which limits the admissible phase shifts to well representable motions. We limit the phase difference by a constant ϕ_{limit}. If the phase value is above this limit, $\left|\phi_{diff}^{l}\right| > \phi_{limit}$, the phase value from the next coarser level is used as the corrected phase difference, $\tilde{\phi}_{diff}^{l} = \lambda\phi_{diff}^{l+1}$.

We define ϕ_{limit} depending on the current level l, the total number of levels L, and the scale factor λ as

$$\phi_{limit} = \tau\pi\lambda^{L-l} \tag{16}$$

where the parameter $\tau \in (0,1)$ determines the percentage of limitation. On the coarsest level we set the corrected phase difference to zero if its magnitude exceeds ϕ_{limit}.

We now explain how to compute a smooth interpolation between phases $\phi_1$and ϕ_2. Due to the shift correction, $\phi_1 + \tilde{\phi}_{diff}$ is no longer guaranteed to match ϕ_2, or any equivalent multiple $\phi_2 \pm \gamma 2\pi$, where $\gamma \in \mathbb{N}_0$. In order for the resulting images to be smoothly interpolated, we must preserve the original computed phases ϕ_1 and ϕ_2, up to the 2π ambiguity, while still respecting the shift corrected phase difference $\tilde{\phi}_{diff}$. We do this by searching for a phase difference $\tilde{\phi}_{diff}$ that is $\pm\gamma 2\pi$ the original phase difference ϕ_{diff} from Equation (13) while being as close as possible to $\tilde{\phi}_{diff}$, i.e.,

$$\tilde{\phi}_{diff} = \phi_{diff} + \gamma^* 2\pi \tag{17}$$

where γ^* is determined as

$$\gamma^* = argmin\left\{\left(\tilde{\phi}_{diff} - \left(\phi_{diff} + \gamma 2\pi\right)\right)^2\right\} \tag{18}$$

Due to the adjustment we can now compute the phase ϕ_α of the interpolated images based on the phase of one input frame and a fraction of the final phase differences $\tilde{\phi}_{diff}$ as

$$\phi_\alpha = \phi_1 + \alpha\tilde{\phi}_{diff} \tag{19}$$

Subtracting $\alpha\tilde{\phi}_{diff}$ from ϕ_2 would give the same result.

To reconstruct the interpolated images, we not only need to interpolate the phase, but also the low-pass residual and the

amplitude. To obtain a smoother transition, we propose to linearly blend the amplitude as well as the low frequency residual.

In combination, extensions proposed above allow for smooth interpolation between the two images. Additionally, we add back the high-pass residual of the closer input frame to retain as much high frequency content as possible.

3. RESULTS AND DISCUSSION

With the help of the above algorithm, a MATLAB application with a graphical user interface (GUI) was developed. It has three tabs, including image super-resolution, video super-resolution and video interpolation.

In image super-resolution tab, There are three buttons, including “Opening Image”, “Reset”, and “Start”. The model and amplification preset selection boxes are designed. And there is also an information bar to show the current program state. For image super-resolution processing, we just need to click “Opening Image” button, select one image, choose model and amplification preset, click “Start” button and wait a moment for it done.

In video super-resolution tab, the interface is almost the same and the operation is also similar.

In video interpolation tab, the only difference is that the amplification preset selection box is changed into frame rate selection box. Users should choose one to determine the number of frames interpolated between two frames.

Now I show the image super-resolution by both two algorithms. Table 1 shows the image quality evaluation by peak signal to noise ratio (PSNR). It can be defined by Mean Square Error (MSE).

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\|I(i,j) - K(i,j)\|^2 \quad (20)$$

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX^2}{MSE}\right) = 20 \cdot log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right) \quad (21)$$

In most of the time, we only need to consider unit8 image, so that the maximum value of pixel $MAX = 2^8 - 1$.

If PSNR of one image is larger than others, it could be thought as a better quality. PSNR is an objective evaluation and could be similar to evaluating by men most of the time.

Table 1: IMAGE SUPER-RESOLUTIOON BY BICUBIC AND SRCNN EVALUATED BY PSNR AND RUNNING TIME

Images	Bicubic		SRCNN	
	PSNR	Time(s)	PSNR	Time(s)
baby	39.5448	6.79	46.7783	35.39
bird	37.3693	1.79	47.5034	12.99
building	34.9086	14.37	39.8667	78.54
butterfly	28.9857	1.83	37.2927	10.23
head	38.4557	1.89	42.4426	9.54
house	26.4476	13.87	32.8809	91.24
lamp	36.6462	14.87	47.7337	79.36
people	38.0381	3.81	43.4831	21.64

We find that though bicubic interpolation was faster than SRCNN, the latter method provided better quality images evaluated by PSNR. And we can also go through Figure 5~12 to see whether SRCNN is better than bicubic interpolation on image quality judging by men eyes.

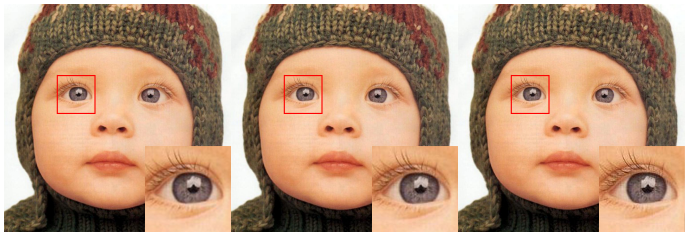

Figure 5: TEST IMAGE BABY.JPG (LEFT), BICUBIC INTERPOLATION IMAGE (MEDIUM), SRCNN IMAGE (RIGHT)

Figure 6: TEST IMAGE BIRD.JPG (LEFT), BICUBIC INTERPOLATION IMAGE (MEDIUM), SRCNN IMAGE (RIGHT)

Figure 7: TEST IMAGE BIULDING.JPG (LEFT), BICUBIC INTERPOLATION IMAGE (MEDIUM), SRCNN IMAGE (RIGHT)

Figure 8: TEST IMAGE BUTTERFLY.JPG (LEFT), BICUBIC INTERPOLATION IMAGE (MEDIUM), SRCNN IMAGE (RIGHT)

Figure 9: TEST IMAGE HEAD.JPG (LEFT), BICUBIC INTERPOLATION IMAGE (MEDIUM), SRCNN IMAGE (RIGHT)

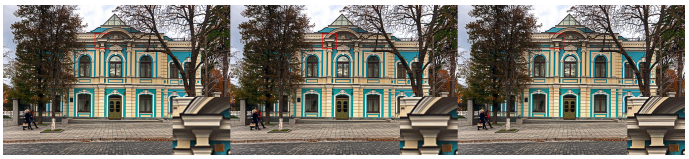

Figure 10: TEST IMAGE HOUSE.JPG (LEFT), BICUBIC INTERPOLATION IMAGE (MEDIUM), SRCNN IMAGE (RIGHT)

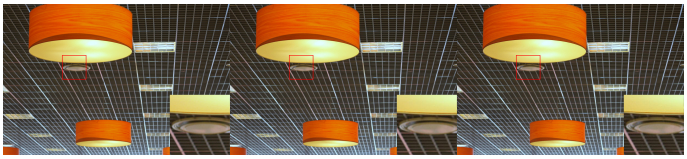

Figure 11: TEST IMAGE LAMP.JPG (LEFT), BICUBIC INTERPOLATION IMAGE (MEDIUM), SRCNN IMAGE (RIGHT)

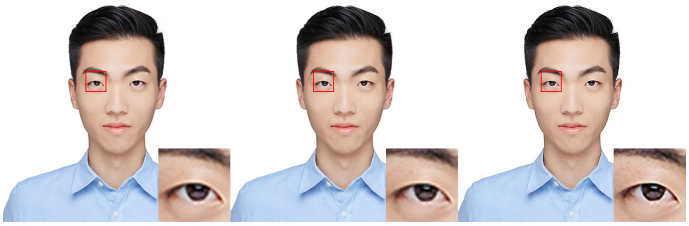

Figure 12: TEST IMAGE PEOPLE.JPG (LEFT), BICUBIC INTERPOLATION IMAGE (MEDIUM), SRCNN IMAGE (RIGHT)

Now we can obviously declare that SRCNN could make higher quality super-resolution than bicubic interpolation. This depend on not only neural network training, but also the three times' convolution.

For the video super-resolution, it is similar to the image super-resolution. We know that video is just a combination of images (frames). We only need to disassemble the video into images, reconstruct and assemble them to get a super-resolution video. I used a 11s video for test. The result could be found as Video_SRCNN_540p_to_1080p.mp4. The quality was judged by the mean of PSNR of every frame.

For the frame interpolation, I used a 15s video for test. The video was disassembled into images. Then depending on the frame rate target given by user, the program built up all the frames between each two original neighbor frames. Finally, combining all the frames gave the result we want. Video_Origin_30fps.mp4 is the original video and Video_PBVI_60fps.mp4 is the result. Especially when the motion is fast, the former video with low frame rate has discontinuous vision. But after frame interpolation, the frame rate is improved and the latter video looks smoother.

4. CONCLUSION

The project realized the image super-resolution based on bicubic interpolation and SRCNN, and also realized the frame interpolation based on phase-based method. And a friendly GUI was created. The project achieved the expected target with practicability. Further, it can be optimized in terms of computing speed, training set, and neural network, so as to provide a better experience.

REFERENCES

[1] Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image Super-Resolution Using Deep Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2), 295-307.

[2] Meyer, S., Wang, O., Zimmer, H., Grosse, M., & Sorkinehornung, A. (2015). Phase-based frame interpolation for video. computer vision and pattern recognition.