Fast Image/Video Upsampling

Qi Shan[1] Zhaorong Li[1] Jiaya Jia[1] Chi-Keung Tang[2]

[1]The Chinese University of Hong Kong [2]The Hong Kong University of Science and Technology

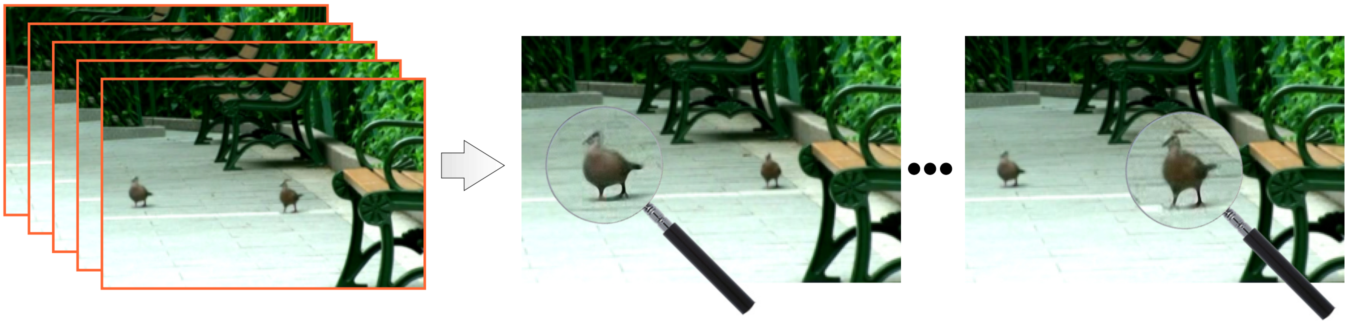


Figure 1: *Local video upsampling. A magnifying effect is simulated by upsampling local regions using our algorithm.*

Abstract

We propose a simple but effective upsampling method for automatically enhancing the image/video resolution, while preserving the essential structural information. The main advantage of our method lies in a feedback-control framework which faithfully recovers the high-resolution image information from the input data, *without* imposing additional local structure constraints learned from other examples. This makes our method independent of the quality and number of the selected examples, which are issues typical of learning-based algorithms, while producing high-quality results without observable unsightly artifacts. Another advantage is that our method naturally extends to video upsampling, where the temporal coherence is maintained automatically. Finally, our method runs very fast. We demonstrate the effectiveness of our algorithm by experimenting with different image/video data.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.4.3 [Image Processing and Computer Vision]: Enhancement—Sharpening and deblurring

Keywords: image/video upsampling, image/video enhancement, image deconvolution

1 Introduction

The operation which estimates a fine-resolution image/video from a coarse-resolution input is often referred to as *image/video upsampling*. This is a fundamentally important imaging research topic, where the main purpose is to recover sharp edges and textures, while suppressing pixel blocking (known as jaggies) and other visual artifacts. For videos, there is an additional requirement – that is, to maintain the natural temporal coherence and to avoid flickering among the upsampled frames during playback. Image/video upsampling finds many applications in computer vision and graphics, such as resizing, surveillance, and texture mapping. It is vital for all image browsing and video play-back softwares, where the zoom-in function is almost standard.

Single image/video upsampling corresponds well to the situation where multiple inputs are not available. As the number of the unknown pixels to be inferred is much more than the size of the input data, this problem can be very challenging. For instance, if we upsample an image by a factor of 4, on average, one pixel in the input image corresponds to 16 unknown pixels. The simplest techniques for image/video resizing are those based on interpolation or linear filtering, including the bilinear, bicubic, and Lanczos algorithms. These methods run fast and are easy to implement. However, they inevitably produce blurry results when applied to upsampling an image/video with a large factor. Example-based methods [Freeman et al. 2002] aim to learn appropriate high-frequency information from a set of training images. These methods are usually much more computationally expensive, and depend critically on the quality of the available examples.

In this paper, we describe a new single-image (or video frame) upsampling algorithm. Our algorithm is based on an image formation process [Baker and Kanade 2000] that models how a coarse-level image is produced from its clear finer-resolution version. Our goal is to simulate a reasonable inversion of this image formation process to infer an upsampled image that is not only free of observable artifacts but also faithful to the low-resolution input. To this end, we develop a feedback-control loop to automatically measure whether the output from the system satisfies the imaging model through a de-convolution and re-convolution scheme. This scheme provides an important clue on how the image estimate deviates from the unknown ground-truth. We then refine our estimates by reinforcing the pixel information from the input image/video. Consequently, this approach yields an image upsampling algorithm that is simple and efficient, wherein the majority of the processing time is spent only on a handful of Fast Fourier Transform (FFT) operations.

More importantly, when our method is applied to video in a frame-by-frame manner, the temporal coherence can be naturally preserved without any additional or special treatment. This is because our method provides a good natural-image estimate for the inverted image formation process. It does not rely on imposing strong lo-

ACM Reference Format
Shan, Q., Li, Z., Jia, J., Tang, C. 2008. Fast Image/Video Upsampling. *ACM Trans. Graph. 27*, 5, Article 153 (December 2008), 7 pages. DOI = 10.1145/1409060.1409106 http://doi.acm.org/10.1145/1409060.1409106.

cal constraints (such as using exemplar patches or gradient information) that are widely employed in image upsampling for reconstructing sharp edges. Figure 1 demonstrates one example that simulates the magnifying-glass effect during video playback. Readers are referred to our supplementary video for the complete sequences.

2 Related Work

Multi-image super-resolution methods attempt to reconstruct a single high-resolution image from several low-resolution inputs with sub-pixel displacements. Typical approaches [Baker and Kanade 2000; Tipping and Bishop 2002] formulate the reconstruction process with an image formation model. Our method is also designed using this model. But it only takes a single image input and allows for upsampling with a large factor.

Single-image upsampling includes a wide spectrum of work. The simplest methods are those based on interpolations, such as the bilinear, bicubic, and cubic spline algorithms [Keys 1981]. These methods usually run very fast, but generate excessively blurred results. Learning- or example-based approaches are also in the main stream of single image upsampling. Hertzmann *et al.* [2001] proposed a general "image analogies" framework to synthesize the detailed structures from the exemplar image pairs. This framework was improved in [Freeman et al. 2002], where an "example-based" image super-resolution strategy is used to learn patch priors and store them in a database. The inference of high-resolution information is achieved using a recognition approach. Example-based approaches are often time-consuming, because finding all compatible patches usually requires expensive computation, such as optimizing a Bayesian network. Tappen et al. [2003] proposed training a group of linear interpolators from examples. These interpolators are used to compute high resolution patches. An algorithm that resembles belief-propagation is then employed to select one high resolution patch from the candidates for each low resolution pixel.

Different from the above methods that utilize the patch information, Fattal [2007] and Sun *et al.* [2008] employed the statistical edge information to help constraining the image upsampling problem. Another notable upsampling work includes the "solution upsampling" proposed by Kopf *et al.* [2007], which aims to upsample a solution obtained in an image plane. The solution is first calculated at a low resolution, which is then upsampled using joint bilateral filtering. In [Avidan and Shamir 2007], a "content-aware" image resizing algorithm was proposed. Rather than resizing an image by scaling, this method carves out or inserts to the image "seams", that is, the pixels of less importance.

For video upsampling, reconstruction based methods [Schultz and Stevenson 1996; Patti et al. 1997; Zhao and Sawhney 2002] require computing the motion between neighboring frames when applied to video. The accuracy of motion estimation at each pixel, which determines the pixel displacement, significantly influences the quality of the results. In [Irani and Peleg 1993], a back-projection method was proposed to iteratively minimize the reconstruction error. It has been observed that grid and ringing artifacts, especially along sharp edges, can be produced. Example-based image super-resolution cannot be directly extended to video due to the unacceptable "flicker" artifact. To handle this problem, two types of temporal priors were introduced in [Bishop et al. 2003] to enforce inter-frame smoothness. Kong *et al.* [2006] proposed using still pictures captured by an auxiliary high-resolution camera to guide the upsampling. All these methods largely depend on the number and quality of the examples for a specific factor. Bhat *et al.* [2007] proposed a framework using a certain number of static high-quality images of the same scene to facilitate various video enhancement applications, including video super-resolution. This method uses a multi-view stereo algorithm for depth estimation. Different from these methods, our upsampling framework can handle natural videos without any camera or object motion assumptions, as our method is designed in tandem with the image formation model and considers natural image statistics. The input videos to our system can contain different types of moving objects, and be taken in different lighting conditions.

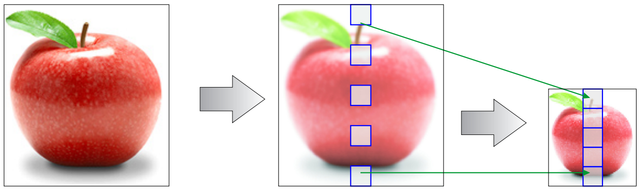


Figure 2: *The image formation process can be separated into two steps: linear filtering and pixel decimation.*

3 The Image Formation Process

The imaging process can be modeled as an integral over the spatial lighting field. Considering a continuous spatial domain S and denoting the received light at location $\mathbf{x}$ as $S(\mathbf{x})$, the captured image I can be regarded as a degradation process of S subject to the camera's Point Spread Function (PSF), which is modeled as

$$I(\mathbf{y}) \quad = \quad \int_{\mathbf{x}\in C(\mathbf{y})} \omega(\mathbf{y}-\mathbf{x})S(\mathbf{x})\,d\mathbf{x}, \tag{1}$$

where $I(\mathbf{y})$ is the pixel value at $\mathbf{y}$ in the captured image. ω denotes a PSF which is commonly approximated by a Gaussian filter, and $C(\mathbf{y})$ is the corresponding region of the PSF at $\mathbf{y}$.

As shown in [Baker and Kanade 2000], by discretizing S into a high-resolution image H and letting f denote the discretized PSF ω, the discrete version of the low-resolution image $I(\mathbf{y})$, denoted by $L(\mathbf{y})$, can be computed as a weighted sum of pixel values in the corresponding region in the high-resolution image H. $L(\mathbf{y})$ is thus expressed as

$$L(\mathbf{y}) \quad = \quad \sum_{\mathbf{x}\in C'(\mathbf{y})} f(\mathbf{y}-\mathbf{x})H(\mathbf{x}),$$

where $C'(\mathbf{y})$ is a set of pixels in the high-resolution image that correspond to $L(\mathbf{y})$. This degradation process from H to L can be further expressed as

$$L = (f \otimes H) \downarrow^{d}, \tag{2}$$

where $\downarrow^{d}$ is a decimating (subsampling) operator with factor d. This equation can be split into two steps, describing a convolution followed by a decimation process:

$$\tilde{H} = f \otimes H, \quad L = \tilde{H} \downarrow^{d}, \tag{3}$$

where $\tilde{H}$ is the linearly filtered high-resolution image. If these two steps can be approximately inverted, that is, by first upsampling the image and then deconvolving it using the linear filter, a reliable upsampling process can be yielded. The non-blind deconvolution problem has been studied in [Yuan et al. 2008; Shan et al. 2008], where robust solvers using optimization are available and can be directly deployed in our system. The remaining decimation process accounts for most information loss, because only one pixel is kept for every $n = d \times d$ pixels in the fine-resolution image. This results in a severely ill-posed problem to recover $\tilde{H}$ from L.

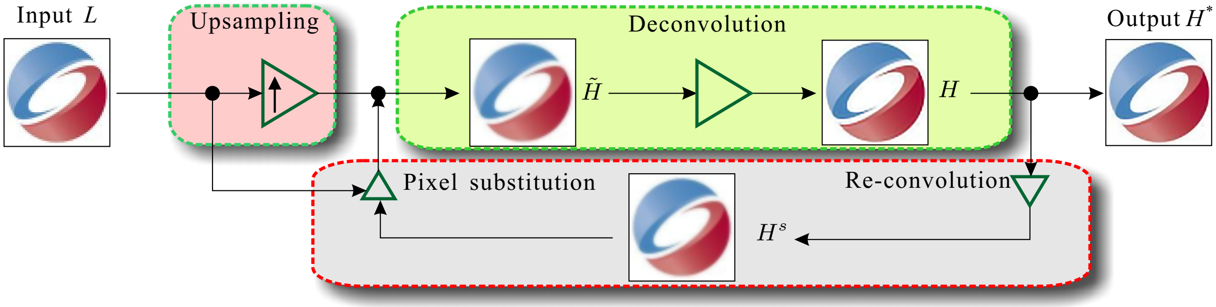


Figure 3: *The system framework. The input image L is initially upsampled as illustrated in the pink box. The green and gray boxes in the middle respectively represent the steps for deconvolution and the feedback-control upsampling. The output is a finer-resolution image H^*.*

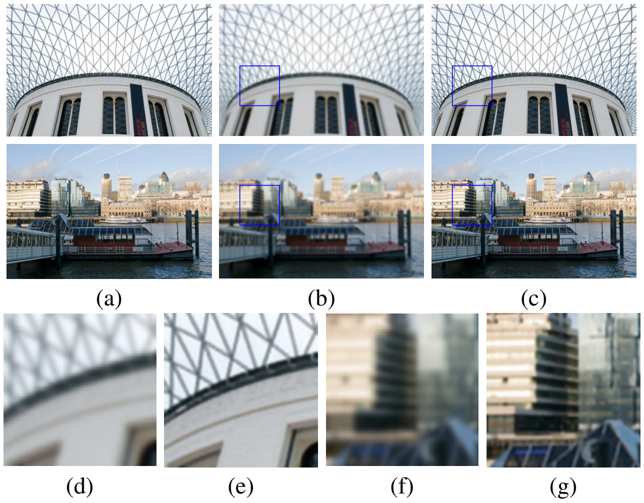


Figure 4: *The effect of deconvolution. (a) The input images. (b) The blurred images generated from (a) by convolving a Gaussian kernel of size 11×11 and with variance 4. (c) The deconvolved image using our method. Their PSNRs with respect to the ground truth images in (a) are as high as 35.6166 (upper image) and 36.2453 (lower image). (d)–(g) Close-ups of (b) and (c).*

4 Feedback-Control Upsampling

Our image upsampling framework is shown in Figure 3. It adopts an iterative scheme, consisting of an image deconvolution step (shown in light green) to remove the effect of linear filtering, and a feedback-control process (shown in light gray with red-frame) to constrain the image upsampling. The input to our system is a low-resolution image or video frame L, which is initially upsampled to the desired resolution by a fast bicubic interpolation (shown in pink). This initially upsampled $\tilde{H}^{(0)}$ is iteratively improved in the feedback-control loops, which progressively reduces image blurriness and recovers natural structures.

One important step in our framework is to compute $\tilde{H}$ from the input L in Eq. (3). The challenge underlying this step is that much pixel information is lost in a regular-grid basis during decimation. However, we observe that although the pixels discarded during downsampling are unknown, it is still possible to measure whether the replenished pixel values in $\tilde{H}$ are reasonable, which is instrumental in refining our estimates. The feedback-control upsampling consists of three steps: deconvolution, re-convolution, and pixel substitution. They are detailed as follows.

[Deconvolution] Given a $\tilde{H}^{(k)}$ that is obtained in iteration k, where $k \geq 0$, the deconvolution process estimates a $H^{(k)}$ in Eq. (3) given a linearly-filtered image $\tilde{H}$. This step is illustrated in the green box in Figure 3. $\|f \otimes H - \tilde{H}\|$ can be written as a multiplication of two matrices: a convolution matrix W representing $\otimes$, and an image vector. Then, minimizing $\|f \otimes H - \tilde{H}\|_2^2$ becomes solving $WH = L$. As W does not always have an exact inverse and is easily influenced by noise, the direct minimization of $\|f \otimes H - \tilde{H}\|_2^2$ is not well-conditioned. In this paper, we adopt a non-blind deconvolution method similar to that described in [Shan et al. 2008] to estimate the deconvolved image. This process is described in our supplementary file. We briefly explain the main steps here.

We first add a prior term to make the deconvolution well-posed. Recent research [Huang and Mumford 1999; Fergus et al. 2006] has shown that natural image gradients generally follow a heavy-tailed distribution, indicating that most pixels have small gradient magnitudes whereas salient structures are present. For image/video upsampling, as we have a coarse-resolution image/video, the global gradient density distribution can be approximated. We thus introduce a gradient density distribution prior, $\Phi(x)$, defined in the same form as the global prior proposed in [Shan et al. 2008] to make the solution unique. Basically, $\Phi(x)$ uses two piece-wise continuous functions to fit the gradient density distribution from the input image/video frames.

By incorporating such a prior into the deconvolution process, we are subject to minimize the following energy function:

$$E(H) \propto \|f \otimes H - \tilde{H}\|_2^2 + \lambda_1(\|\Phi(\partial_x H)\|_1 + \|\Phi(\partial_y H)\|_1), \quad (4)$$

where $\partial_x H$ and $\partial_y H$ respectively denote the values of the x- and y-direction gradients, and λ_1 is a weight. It can be rapidly estimated through a variable-substitution scheme, where the main step is to apply Fast Fourier Transforms (FFTs) to compute the optimal values. As each FFT operation can be executed rapidly, the overall deconvolution process is efficient. Note that directly minimizing $E(H)$ using gradient descent, or matrix manipulations such as computing the inverses, can be much more time-consuming. We will analyze our running time in the next section.

To verify the effectiveness of the deconvolution, we show two examples in Figure 4 which compare our deconvolved image with the ground truth. The Peak Signal-to-Noise Ratios (PSNRs) are very high, showing that this method faithfully deconvolves the images without introducing other ringing artifacts.

[Re-convolution] Based on the above analysis, we observe that if the reconstructed filtered image $\tilde{H}^{(k)}$ is similar to the ground truth, the convolution model is satisfied. More specifically, if we re-filter image $H^{(k)}$ output from the above deconvolution process, we should be able to get an image pretty close to $\tilde{H}^{(k)}$:

$$H^s = (\tilde{H} \otimes^{-1} f) \otimes f \approx \tilde{H},$$

where $\otimes^{-1}$ denotes the deconvolution operator. Otherwise, $\tilde{H}$ must contain incorrectly-filled pixel values, as it does not satisfy the image formation model. In this case, we need to refine $\tilde{H}$ by reinforcing the information obtained from L.

[Pixel substitution] In the pixel decimation process, a high-resolution image H is subsampled by a factor of d to get L, meaning that only one pixel is kept in L for every $d \times d$ samples. Based on it, in our control loop, we replace the values of the corresponding pixels in $H^{s(k)}$ by those from L, in order to reinforce the low-resolution image information to be preserved in the upsampled result. The detail of this process is as follows. If the upsampling factor is n, we replace pixel $(n \times i + 1, n \times j + 1)$ in the high resolution image H^s with those in image L with coordinate (i, j) for each i and j. When $n = 3$, about one-ninth of the image pixels are replaced. We show one illustration of the pixel substitution process with the upsampling factor of 3 in Figure 5.

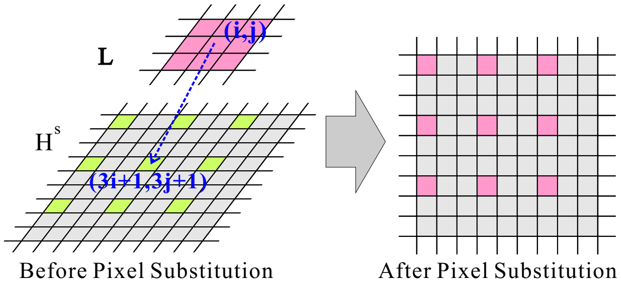


Figure 5: *Illustration of the pixel substitution process.*

The benefit of using pixel substitution is twofold – that is, to achieve the inverse of the decimating process (Eq. (3)), and to approximate H^s as a Gaussian-filtered image with the feedback loop. More specifically, (i) pixel substitution simulates a uniform pixel decimation process in the imaging model to form L, where one pixel is sampled for every $n \times n$ pixels in H^s; (ii) in iterations, the substituted color information is propagated to the neighboring missing pixels in H^s with the deconvolution and re-convolution loop, which eventually makes H^s a Gaussian-filtered image.

Then the pixel-substituted image $\tilde{H}$ is passed to the next iteration for deconvolution and re-convolution, where the filled-in pixel values from L are propagated to others to refine the high-resolution estimate. In Figure 6, we plot a set of error curves for the natural image examples shown in this paper. They are used to illustrate how the reconstruction errors are reduced iteratively in our algorithm. The error of the k-th iteration is computed as $err^{(k)} = \|(f \otimes H^{(k)}) \downarrow^d - L\|_2^2$ normalized by the initial error $err^{(0)}$. So all curves start from 1. This figure shows that our feedback-control loop effectively reduces the estimation errors and converges in only a few (less than 5) iterations. In the result section, we show that our method can upsample an image with a factor as large as 4 or 8, and meanwhile preserving natural structures.

Although our feedback-control loop looks somewhat similar to the back-projection mechanism in [Irani and Peleg 1993], where the enforcement of the reconstruction constraint using the input image is achieved via an iterative refinement scheme, there are two major differences. First, the back-projection algorithm deconvolves an error map that may contain unpredictable structures. In contrast, our algorithm substitutes the pixel values in $\tilde{H}$ by the low-resolution counterparts. So our algorithm only deals with natural images, whose statistical properties have been well studied and understood. Second, we use a state-of-the-art deconvolution operator in lieu of the "back-projection" filter. It is known that the latter can produce ringing artifacts along strong edges. These two improvements in our algorithm together lead to the fast convergence and visually-appealing upsampled results, even with a large factor.

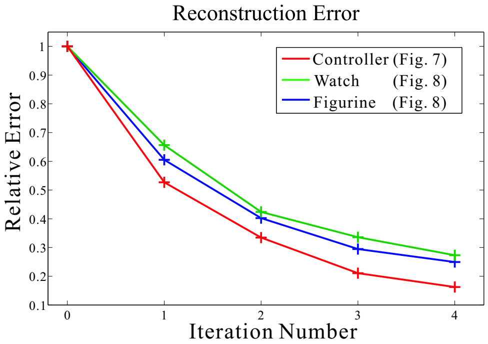


Figure 6: *The reconstruction error curves for the images shown in Figures 7 and 8.*

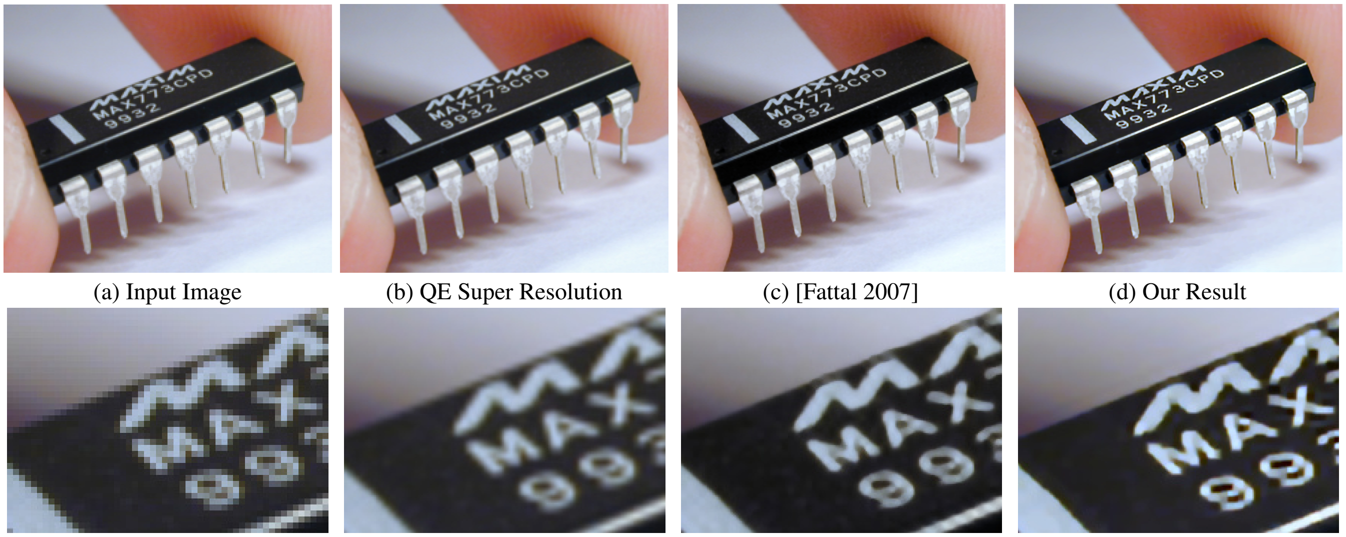
5 Analysis

Image formation model Our algorithm is based on the image formation model. But it is not restricted to only upsampling a coarse-resolution images obtained from such a process (through filtering and decimation). In experiments, our method works well for images/frames obtained by other methods such as Lanczos filtering and bicubic interpolation. Our method is also robust against additive image/video noise, which is handled in the deconvolution step as the deconvolution errors.

Video temporal coherence Although we do not explicitly impose any temporal coherence constraint, when we apply our upsampling algorithm to video clips by processing them frame-by-frame, the upsampled video results do not suffer from any observable temporal artifacts such as flickering. This is because, in our feedback-control loop, the pixel values in the high-resolution frame $\tilde{H}$ are evenly substituted by the corresponding pixel values in the input image L. So the upsampled video frames faithfully preserve the details observed in the input video. Moreover, the deconvolution process evenly distributes the remaining errors to the entire image, therefore causing less biases in reconstructing video structures locally. We tested a variety of video clips captured with different camera and object motions, and show our upsampled video results in the supplementary video.

Analysis of computation time The computation time required to produce a single high-resolution image is dependent of the image size and the iterations included in the feedback loop. In our algorithm, most computational effort is put in computing Fast Fourier Transforms (FFTs). In each iteration, for the deconvolution step, we perform FFTs $3 * 4 + 1$ times as described in our supplementary file. In the re-convolution step, we perform another 2 FFT operations. Finally, considering that our algorithm iterates 3 times to get $H^* = H^{(3)}$ using the feedback-control loop, we need to perform deconvolution and re-convolution 4 times and 3 times respectively. Therefore, a total of 58 FFT operations is performed. In our non-parallel implementation, we employ the library "fftw" [Frigo and Johnson 2006]. With the frame size fixed for a given video, it is initialized with flag "FFTW_EXHAUSTIVE".

If the input to the library only contains real numbers, such as the



(a) Input Image (b) QE Super Resolution (c) [Fattal 2007] (d) Our Result

(e) (f) (g) (h)

Figure 7: *Image upsampled with a factor of 4. (a) An input image from [Fattal 2007] magnified with a factor of 4 using the nearest-neighbor interpolation. (b) The result generated by a software named "QE Super Resolution" [QELabs 2005]. (c) The result generated by Fattal [2007]. (d) Our result. (e)-(h) Close-ups of (a)-(d).*

pixel values in an image, each FFT operation takes about 4–6 milliseconds. However, in this case, only half of the DFT spectrum of the data is computed. In our implementation, we use the input with complex numbers, each of which is expand from a real pixel value. The FFT operation takes about 10–20 milliseconds for an image of resolution 800×600. Taking account of other computational overhead, it takes in total about 700 milliseconds to generate a high-quality upsampled frame. Obviously, there is much room for acceleration, such as making use of programmable graphics hardware to speed up FFTs. For video upsampling, because we can distribute the upsampling tasks of different frames to multiple CPUs, much higher processing rate can be achieved.

Other implementation details The size of the Gaussian kernel used in our algorithm for the deconvolution step is fixed at 13×13. We found that a reasonable variation from 11×11 to 19×19 in this value does not significantly influence the results, although we also note that a large kernel would lead to loss of high-frequency information, whereas an exceedingly small kernel does not sufficiently suppress the aliasing problem. The deviation of the Gaussian kernel is set empirically, depending on the value of the upsampling factor: 1.05 for a factor of 2, 1.5 for a factor of 4, and 2.02 for a factor of 8.

Our iterative image-upsampling algorithm accepts a grayscale image as an input. For color images, we first convert the RGB to YUV color space, and then upsample only the Y channel. The UV channels are then upsampled to a higher resolution simply by a fast bicubic interpolation: here, our method exploits the human's relative insensitivity to small disagreement in chromaticity.

6 Results

Image upsampling While one main focus of this paper is to deploy our algorithm in video upsampling, we first show several image upsampling results. In Figure 7, we show one image example where the input appeared in [Fattal 2007]. We show our upsampling result with a factor of 4. It is comparable in quality to the results from other state-of-the-art image upsampling algorithms, such as a software named "QE Super Resolution" [QELabs 2005] and the edge-statistics-based method [Fattal 2007]. Two additional image examples are shown in Figure 8, for which we set the upsampling factors respectively as 8 and 4. Our results also exhibit natural structures. The small characters in the watch example are reconstructed correctly.

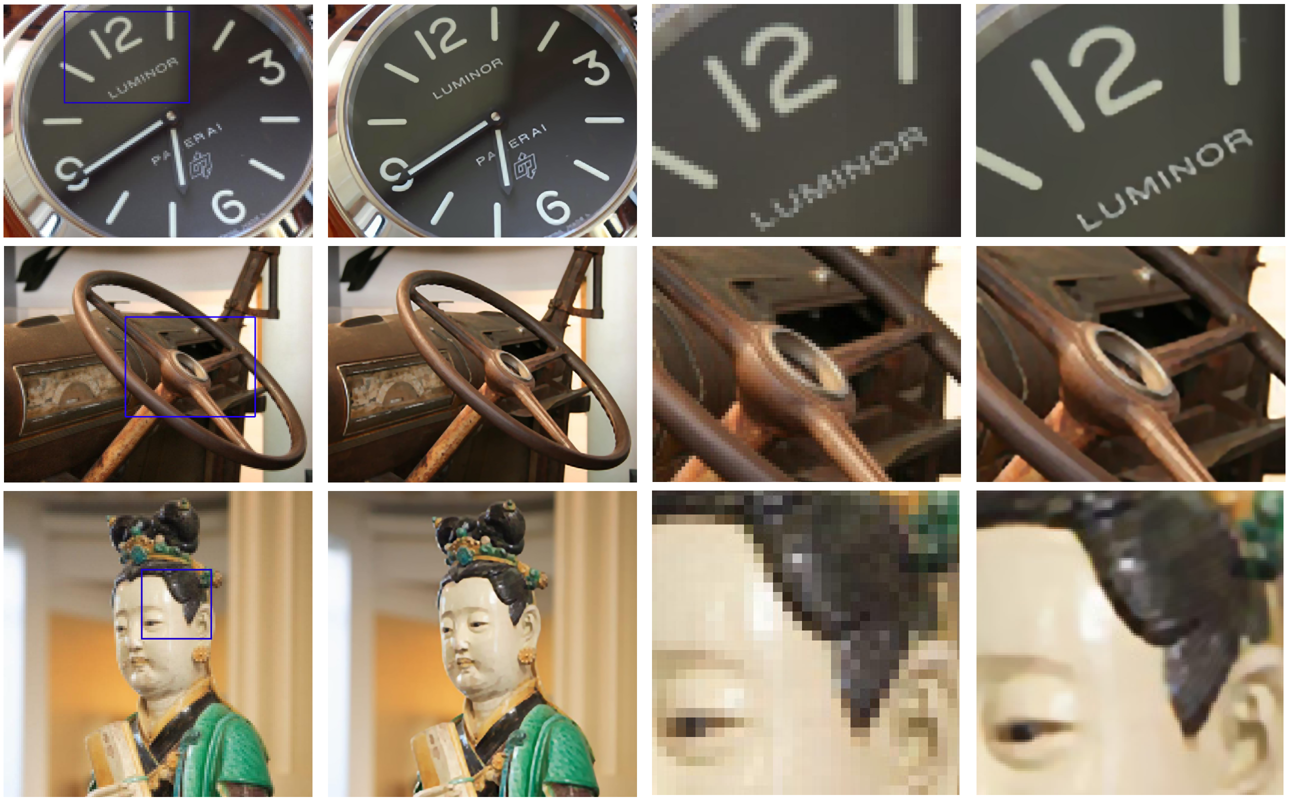
Video upsampling We show two video examples in this paper. Readers are referred to our supplementary video for additional examples and evaluating their quality, in particular the temporal coherence maintained by our method. In Figure 9, we show three frames extracted from a "bird" sequence. They are upsampled with a factor of 4. We compare our results with those generated by other methods. Observe the difference in edge sharpness and texture clearness. In Figure 10, we show a local upsampling result focusing on a swimming fish. The user can modify the upsampling factor and the interested region on-the-fly.

7 Concluding Remarks

High-definition (HD) display devices, such as HDTV, is getting more popular and eventually more affordable. However, to obtain HD contents, special capture devices are generally required. With numerous videos captured and movies made before any HD standard ever coming into existence, an efficient upsampling algorithm is in great demand for enhancing these image/video contents.

In this paper, we have proposed a simple and yet effective image/video upsampling algorithm to reconstruct images/frames of a higher resolution. Our algorithm consists of a feedback loop to control the upsampled image quality, by enforcing the relevant information propagated from the input low-resolution images. The main operation in our algorithm consists of a handful of FFT operations, which makes our algorithm easy to be encoded in an embedded system. We have demonstrated that our algorithm can be used to upsample a video in a frame-by-frame manner while automatically preserving temporal consistency. This makes our algorithm amenable to parallel computation, where different frames can be upsampled concurrently at multiple CPUs or programmable graphics hardware to further accelerate the upsampling process.

Because our method does not introduce new information except that



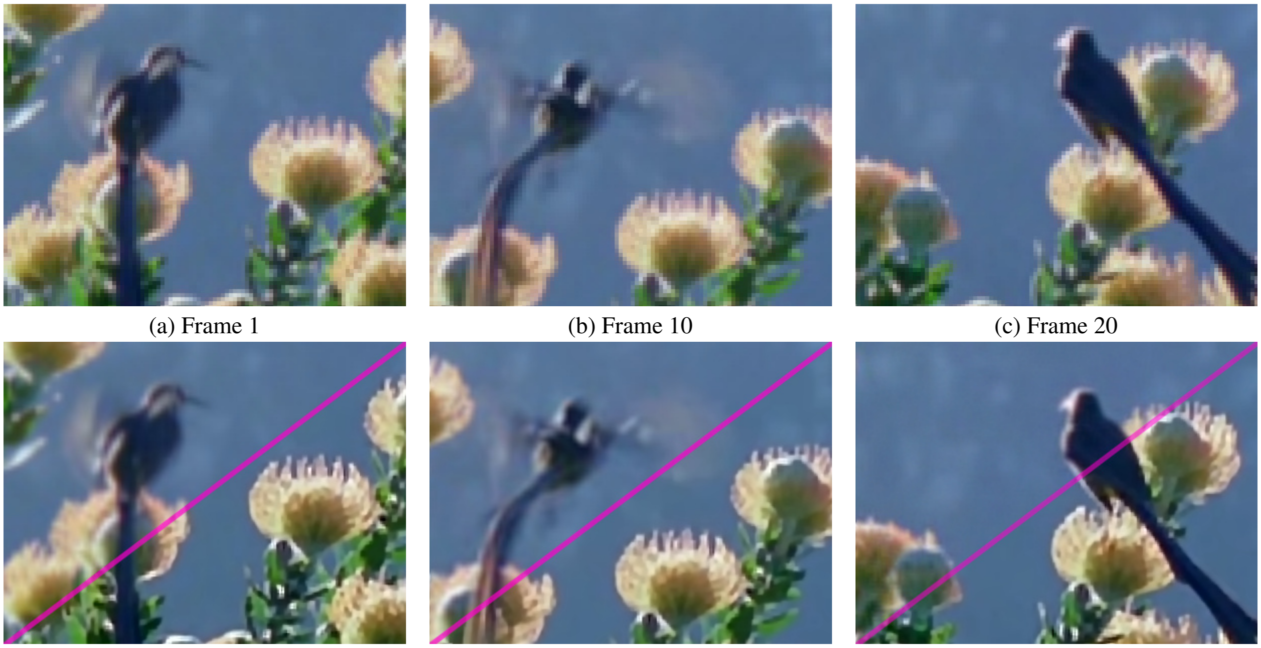
(a) Input Images

(b) Our Results

(c) Close ups of (a)

(d) Close ups of (b)

Figure 8: *More image results with the upsampling factors of 8 (the first row) and 4 (the lower two rows). (a) Input images. (b) Our results. (c) Close-ups of (a). (d) Close-ups of (b).*



(a) Frame 1

(b) Frame 10

(c) Frame 20

(d) Bilinear vs our algorithm

(e) Bicubic vs our algorithm

(f) “Video enhancer” vs our algorithm

Figure 9: *Video upsampling results. We show three frames from an input video in (a)–(c) respectively. For comparison, we juxtaposed our results with the results produced by (d) bilinear interpolation, (e) bicubic interpolation, and (f) “video enhancer” [Mon 2006].*

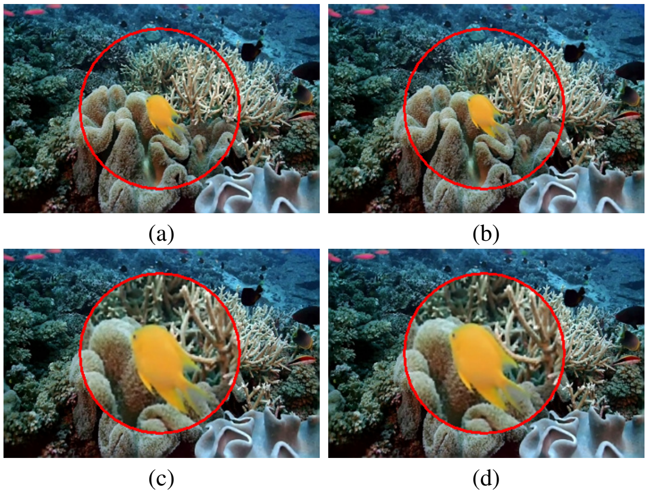


Figure 10: *On-the-fly local upsampling. Two input frames from a "sea world" sequence are shown in (a) and (b). (c) and (d) respectively show the locally upsampled frames.*

in the input low-resolution image, missing edges or textures during the decimation process cannot be recovered. We believe that if example-based methods are appropriately combined and work in tandem with our method, more relevant structural information can be injected into the upsampled results. This is one potential research direction for further pursuit.

Acknowledgements

We thank all reviewers for their constructive comments. This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. 412708 and 620006).

References

AVIDAN, S., AND SHAMIR, A. 2007. Seam carving for content-aware image resizing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 10.

BAKER, S., AND KANADE, T. 2000. Limits on super-resolution and how to break them. In *CVPR*, IEEE Computer Society, 2372–2379.

BANDO, Y., AND NISHITA, T. 2007. Towards digital refocusing from a single photograph. In *Pacific Graphics*.

BHAT, P., ZITNICK, C. L., SNAVELY, N., AGARWALA, A., AGRAWALA, M., CURLESS, B., COHEN, M., AND KANG, S. B. 2007. Using photographs to enhance videos of a static scene. In *Eurographics Symposium on Rendering*, Eurographics, 327–338.

BISHOP, C. M., BLAKE, A., AND MARTHI, B. 2003. Super-resolutioin enhancement of video. In *In 9th Conf. on Artificial Intelligence and Statistics*.

CHAN, T. F., OSHER, S., AND SHEN, J. 2001. The digital tv filter and nonlinear denoising. *IEEE Trans. on Image Processing 10*, 2, 231–241.

FATTAL, R. 2007. Image upsampling via imposed edges statistics. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 95.

FERGUS, R., SINGH, B., HERTZMANN, A., ROWEIS, S. T., AND FREEMAN, W. T. 2006. Removing camera shake from a single photograph. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006) 25*, 787–794.

FREEMAN, W. T., JONES, T. R., AND PASZTOR, E. C. 2002. Example-based super-resolution. *IEEE Computer Graphics and Applications 22*, 56–65.

FRIGO, M., AND JOHNSON, S. G., 2006. FFTW Home Page. WWW page. `http://www.fftw.org/`.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. 2001. Image analogies. In *SIGGRAPH*, 327–340.

HUANG, J. G., AND MUMFORD, D. 1999. Statistics of natural images and models. In *CVPR*, 1541–1547.

IRANI, M., AND PELEG, S. 1993. Motion analysis for image enhancement: resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation 4*, 324–335.

KEYS, R. G. 1981. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics, Speech, and Signal Processing 29*, 1153–1160.

KONG, D., HAN, M., XU, W., TAO, H., AND GONG, Y. H. 2006. Video super-resolution with scene-specific priors. In *BMVC*.

KOPF, J., COHEN, M. F., LISCHINSKI, D., AND UYTTENDAELE, M. 2007. Joint bilateral upsampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 96.

MON, D., 2006. Video enhancer. WWW page. `http://www.thedeemon.com/VideoEnhancer/`.

OSHER, S., SOLE, A., AND VESE, L. 2003. Image decomposition and restoration using total variation minimization and the H^{-1}. *Multiscale Modeling and Simulation 1*, 3, 349–370.

PATTI, A., SEZAN, M., AND TEKALP, A. 1997. Super resolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Trans. on Image Processing 6*, 1064–1076.

QELABS, 2005. Qe super resolution. WWW page. `http://www.qelabs.com/index`.

SCHULTZ, R. R., AND STEVENSON, R. L. 1996. Extraction of high-resolution frames from video sequences. *IEEE Transactions on Image Processing 5*, 996–1011.

SHAN, Q., JIA, J. Y., AND AGARWALA, A. 2008. High-quality motion deblurring from a single image. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*.

SUN, J., SUN, J., XU, Z. B., AND SHUM, H. Y. 2008. Image super-rosolution using gradient profile prior. In *CVPR*.

TAPPEN, M. F., RUSSELL, B. C., AND FREEMAN, W. T. 2003. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *Intl. Workshop on Statistical and Computational Theories of Vision*.

TAPPEN, M. F., RUSSELL, B. C., AND FREEMAN, W. T. 2004. Efficient graphical models for processing images. In *CVPR*, IEEE Computer Society, 673–680.

TIPPING, M. E., AND BISHOP, C. M. 2002. Bayesian image super resolution. In *NIPS*, MIT Press, 1279–1286.

YUAN, L., SUN, J., QUAN, L., AND SHUM, H. Y. 2008. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*.

ZHAO, W. Y., AND SAWHNEY, H. S. 2002. Is super-resolution with optical flow feasible. In *ECCV*, Springer, 599–613.