DIGITAL IMAGE SUPER RESOLUTION

LIU SHUAICHENG

(B.Sc., SICHUAN UNIVERSITY, 2008)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2010

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Assoc. Prof. Michael S. Brown, for his instructive advice and useful suggestions on my thesis. I am deeply grateful of his help in the completion of this thesis. I am also deeply indebted to all my colleagues in Computer Vision Laboratory, National University of Singapore. I really enjoyed the pleasant stay with these brilliant people for the past 2 years. Special thanks should go to my friends who have put considerable time and effort into their comments on my thesis draft. Finally, I am indebted to my parents for their continuous support and encouragement.

Contents

Chapter 1

Introduction

1.1 Overview of Super Resolution

Image super resolution (SR) is a process that estimates a fine-resolution image from a coarse-resolution image. SR is a fundamentally important research topic with the main purpose to recover sharp edges and estimate missing high frequencies while suppressing other visual artifacts. Traditionally, there are both multiple-frame and single-frame variants in the SR [3]. In multiple-frame SR [9, 2, 25, 18, 36] a set of low resolution (LR) images of the same scene are available. Usually, it is assumed that there is some relative motion between the camera and the scene. Therefore, the first step is to register or align these LR images. The high resolution (HR) image is constructed from these aligned LR images by multiple-frame SR algorithms. Single image SR [5, 7, 13, 15, 39] methods attempt to magnify the image with the purpose of preserving edges or recovering missing details. These methods obtain missing information from the input image itself or other similar

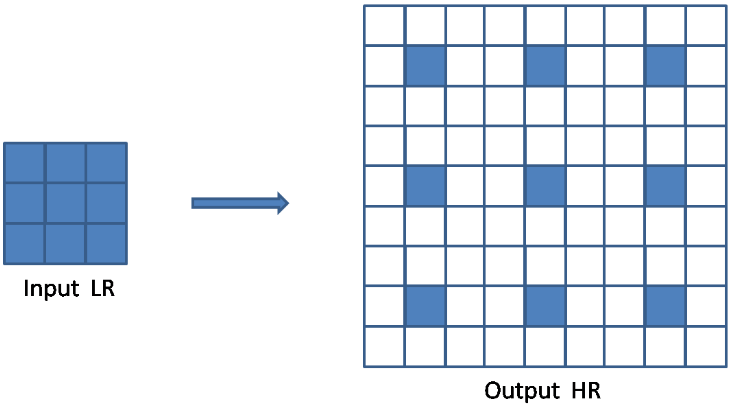


Figure 1.1: An example of upsampling 3 ×, one pixel in the input image corresponds to 9 unknow pixels.

images. This paper focuses on single image SR approaches.

Single image SR is necessary when multiple inputs of the same scene are not available. As the number of the unknown pixels to be inferred is much more than the size of the input data, the problem can be challenging. For example if we upsample an image by a factor of three, one pixel in the input image corresponds to nine unknown pixels (see figure 1.1). In the past years a wide range of very different approaches has been taken to improve single image SR. They can be broadly classified into three families: (1) Interpolation-based methods,(2) Reconstruction-based methods, (3) Learning-based methods.

Interpolation-based approaches [1, 29, 37, 24, 27, 17] have their foundations in sampling theory and try to interpolate the high resolution (HR) image from the LR input. These approaches run fast and are easy to implement. However, they usually blur high frequency details and often have noticeable aliasing artifacts along edges.

Reconstruction-based approaches [5, 7, 34, 39, 41, 38, 10] estimate an HR image by enforcing some prior knowledge on the upsampled image. These approaches usually require the appearance of the upsampled image to be consistent with the original input LR images. This is achieved by back projection. The enforced priors are typically designed to reduce edge artifacts. These types of methods are also referred to as edge-directed SR in this report. The performance of reconstruction based approaches depends on the priors and its compatibility with the given image.

Learning-based approaches [13, 5, 15, 22, 33] are sometimes termed "image hallucination". In learning-based SR, correspondences between low and high resolution image patches are learned from a database consisting of low and high resolution image pairs. The learned patches are applied to a new LR image to recover its most likely HR version. The high frequencies of the upsampled image which are learned from the training data are not guaranteed to be the true high resolution details. The performance of learning based approaches depend on the effectiveness of the supporting image training database, especially for edges.

1.2 Thesis Objective

The objective of this thesis is to design algorithms for single image SR. Two algorithms are proposed. The first algorithm named 'Super resolution using Edge Prior and Single Image Detail Synthesis' focus on the traditional single image SR problem. Sharp edges and image details are recovered under large zoom in factors. Another algorithm named 'Single image super resolution' addresses color issues in single image SR and trying to handle color bleedings which happens in many existing SR methods.

1.3 Thesis Organization

The remainder of this paper is organized as follows: in Chapter 2, we survey a variety of techniques and provided a tentative classification according to their properties; in Chapter 3, the proposed algorithm of SR named 'Super resolution using Edge Prior and Single Image Detail Synthesis' is discussed in details. A method for addressing color in SR, is given in Chapter 4. Chapter 5 concludes the thesis.

Chapter 2

Literature Survey

2.1 Interpolation Based Methods

Interpolation is the process of determining the values of a function at positions lying between samples. Common used interpolation methods include nearest neighbor, bilinear, bicubic. Super resolution through these simple interpolation method is computational efficient and is widely used in image processing software.

Nearest neighbor

The simplest interpolation method is nearest neighbor (pixel replication), where each interpolated output pixel is assigned the value of the nearest sample point in the input image. The kernel of nearest neighbor interpolation is defined as:

$$h(x) = \begin{cases} 1 & 0 \leq |x| < 0.5, \\ 0 & 0.5 \leq |x|, \end{cases}$$

The kernel $h(x)$ helps to decide which neighbor values to choose at the interpolated position based on the $|x|$. The term $|x|$ refers to the distance between the given position and a specific neighbor. Due to the fact that nearest neighbor interpolation simply copy the nearest pixel, a jaggy artifacts is obvious.

Linear Interpolation

Linear interpolation is a method of curve fitting using linear equations. Unlike the nearest neighbor method, the interpolated pixel value is computed by its neighbors. The kernel of linear interpolation is defined as:

$$h(x) = \begin{cases} 1 - |x| & 0 \leq |x| < 0.5, \\ 0 & 1 \leq |x| \end{cases}$$

For the 2D case , bilinear interpolation is used where four neighbors are considered for the interpolated value. Linear interpolation produces reasonably good results, but still tend to blur edge detail.

Cubic convolution

The cubic convolution interpolation kernel is composed of a piecewise cubic polynomials defined on the subintervals (-2,-1),(-1,0),(0,1),(1,2). Outside the interval (-2,2), the interpolation kernel is zero. Compared to the linear in-

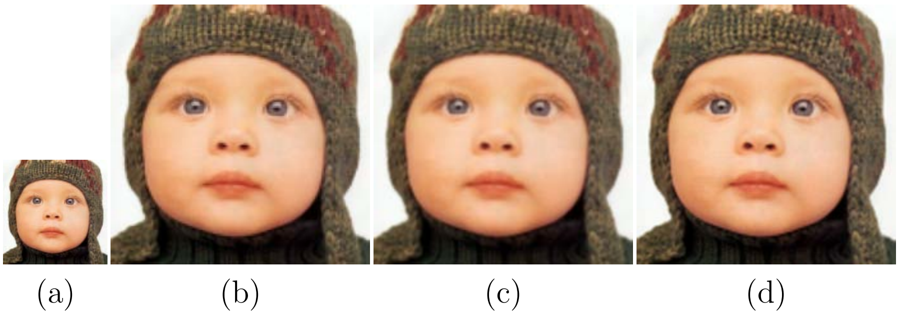


Figure 2.1: Example of interpolation based methods. (a) low resolution image. (b)Nearest neighbor 4x. (c)Linear interpolation 4x. (d)Cubic interpolation 4x.

terpolation, more samples are used to compute the newly interpolated value. The kernel is defined as:

$$h(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 \leq |x| < 1, \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \leq |x| < 2, \\ 0 & 2 \leq |x| \end{cases}$$

The performance of the interpolation kernel depends on a. For different images, different values of a gives the best performance. Cuibc interpolation is more computational expensive compared to linear and nearest neighbor interpolation. However, the results are smoother and have fewer interpolation artifacts.

2.2 Reconstruction Based Methods

2.2.1 Back Projection

Back projection (BP) [19, 6] is an efficient algorithm which minimize the reconstruction error with an iterative procedure. It is widely used in SR algorithms. Back projection makes the reconstructed HR image consist with the input LR image. The main contribution of back projection is that the reconstructed HR have the same look and feel as the LR image after applying BP. Usually, a BP algorithm is used together with other super resolution algorithm to enhance the SR result during the reconstruction phase or at the final step.

Back Projection algorithm

The generation process of producing a LR image can be modeled by a combination of the blur effect and the down-sampling operation as shown in [3]. By simplifying the blur effect with a single filter g for the entire image, the generation process can be formulated as follows:

$$I^l = (I^h \otimes g) \downarrow_s, \tag{2.1}$$

where I^l and I^h are the LR and HR images respectively, $\otimes$ represents convolution with filter g, and $\downarrow_s$ is the down-sampling operator with scaling factor s.

The Back Projection algorithm can be summarized as iteratively updating HR image to minimize the reconstruction error. The algorithm is described as follows:

- Compute the LR error: $Error(I_t^h) = I_t^l - (I_t^h \otimes g) \downarrow_s$
- Update the HR image by back-projecting the error as follows:

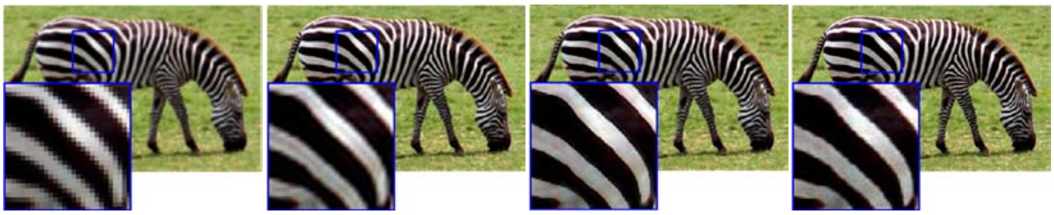
$$I_{t+1}^h = I_t^h + Error(I_t^h) \uparrow_s \otimes p$$

where I_t^h is the HR image at the t-th iteration, ↑ is the upsampling operator, p is a constant back-projection kernel. These two steps are computed iteratively until the reconstruction error $Error(I_t^h)$ drops under a given threshold. During each iteration, the current reconstruction error is back-projected to adjust the image intensity. By updating the HR image with back-projection iteration, I_t^h will converge to a desired image which satisfies Eqn. 2.1

Bilateral Back Projection

The algorithm described above can produce visually appealing result, however, this method suffers from the chessboard effect and ringing effect, especially along strong edges. The underlining reason is that there is no edge guidance in the error correction process. During each iteration , the LR error $Error(I_t^h)$ is back-projected to HR image by a isotopic kernel p. The error correction step propagates the error without considering the local edge direction and strength. The cross-edge error propagation may produce ringing effect, and the isotropic kernel results in chessboard effect.

Bilateral back projection [6] using a bilateral filter during the back projection process. Bilateral filter is a non-linear filtering technique which can combine image information form both of the space domain and the feature domain in the filter process. Rather than simply replacing a pixel's value



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

Figure 2.2: Example of back projection algorithms [6]. (a) low resolution image. (b)Back projection 4x. (c)Bilateral back projection 4x. (d)Ground truth.

with a weighted average of its neighbors, as for instance the Gaussian filter does, the bilateral filter replaces a pixel's value by a weighted average of its neighbors in both space and range,thus the edge sharpness is preserved by avoiding the cross edge smoothing.

The main difference between simple BP and bilateral BP is that the bilateral filter is applied on the HR error image $Error(I_t^h) \uparrow_s$ during each iteration. For homogeneous regions , the bilateral BP algorithm is the same as the simple BP, for regions near step edges, the error will be only propagated in the part on the sides of the edges. With bilateral BP, clear and sharp edges are obtained compared to simple BP.

2.2.2 Gradient Profile Prior

The Gradient Profile Prior [39] is a parametric prior describing the shape and the sharpness of the image gradients. Unlike previous smoothness prior, the gradient profile prior is not a smoothness constraint. Both small scale and large scale magnification can be well recovered. The common artifacts in super resolution, such as ringing artifacts can be avoided by working in the gradient domain using the gradient profile prior. The reconstructed gradient

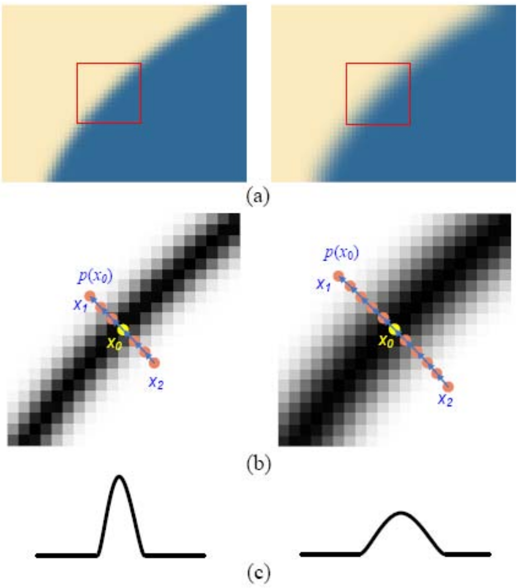


Figure 2.3: (a)Two edges with different sharpness. (b)Gradient map, $p(x_0)$ is a gradient profile.(c)1D curves of two gradient profiles. Image from [39].

field is much closer to the ground truth gradient field. Generally, SR through the gradient profile constraints produces results with sharper edges than other techniques. Fig.2.3 from [39] shows an example of gradient profile of $p(x_0)$ with different sharpness.

The gradient profile $p(x_0)$ is a 1-D profile along the gradient direction of the zero-crossing pixel in the image. The gradient profile prior is a parametric distribution describing the shape and the sharpness of the gradient profiles in natural image. One observations is that the shape statistics of the gradient profiles in natural image is quit stable and invariant to the image resolution. With this stable statistics, statistical relationship of the sharpness of the gradient profile between the HR image and the LR image can be learned. Using the learned gradient profile prior and relationship, we are able to provide a constraint on the gradient field of the HR image. Combining

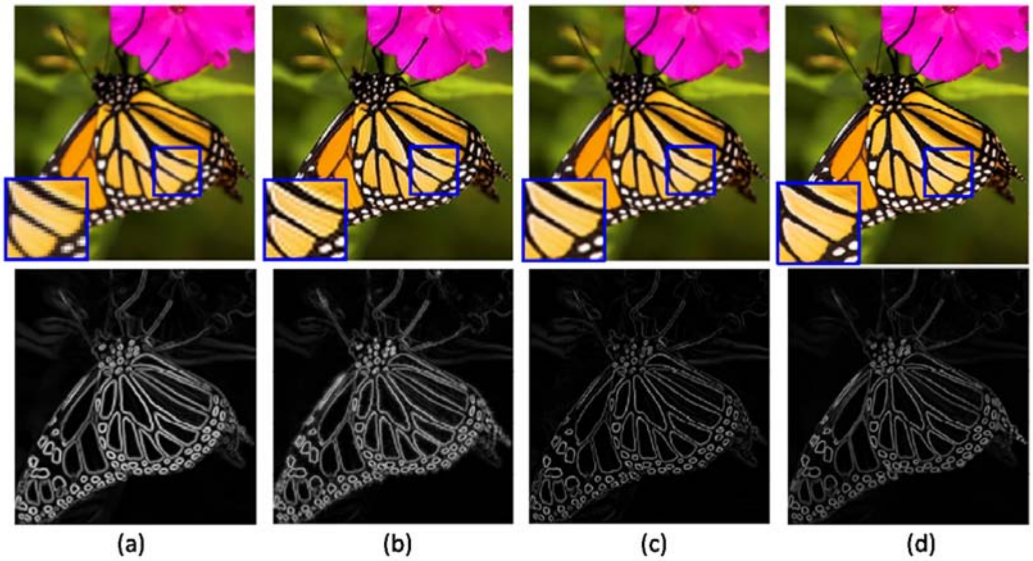


Figure 2.4: (a) LR image and its gradient field. (b) result of back-projection and its gradient field. (c)GPP result and its gradient field. (d) ground truth image and its gradient field. Image form [39]

with the reconstruction constraint, hi-quality HR image can be recovered.

Figure 2.4 gives an example of GPP method. Figure 2.4(a) are input LR image and the gradient field of bicubic upsampled image. Figure 2.4(d) are ground truth HR image and its gradient field. Figure 2.4(b) are back-projection result using the reconstruction constraint only. The bottom image in Figure 2.4(c) is GPP transformed gradient field. The transferred gradient field is used as the gradient domain constraint for the HR image reconstruction. As we can see, the transformed gradient field Figure 2.4(c) is much closer to the ground truth gradient Figure 2.4(d).

2.3 Learning Based Methods

2.3.1 Example-based

The interpolation-based image SR (bilinear, bicubic) usually result in the blurring of images. While edge directed interpolation can preserve the edges to some extent, it still suffers from lost of image detail in homogenous regions. Example-based SR [13] tries to recover the lost high frequency details. The recovered plausible high frequency comes from a database which consists of a set of training images. Example-based SR is the most important learning-based approach which has inspired many other learning-based algorithms.

Training Set

The training set contains a set of HR and LR image pairs. The LR image is generated by down sampling the corresponding HR image. It is believed that the highest frequency components of the low resolution image are most important in predicting the extra details. The low frequency are filtered out and only the high frequency component are stored. The low resolution patch has the size of 7×7 and the corresponding high resolution patch size is 5×5. The reason why the LR patch size is bigger than its HR counterpart is that big patch can capture more spatial information than small ones. Fig.2.5 from [13] shows the pre-processing steps for the training set generation. LR image Fig.2.5(a) is a down sampled version of original image(c). Fig.2.5(b) is the interpolation version of (a). Images (b) and (c) becomes a pair of image pairs in the pixel domain. Band-pass filtering and contrast normalizing (b) get (d). Fig.2.5(e) is high frequency of (c). Training set stores corresponding pairs of patches from (d) and (e).

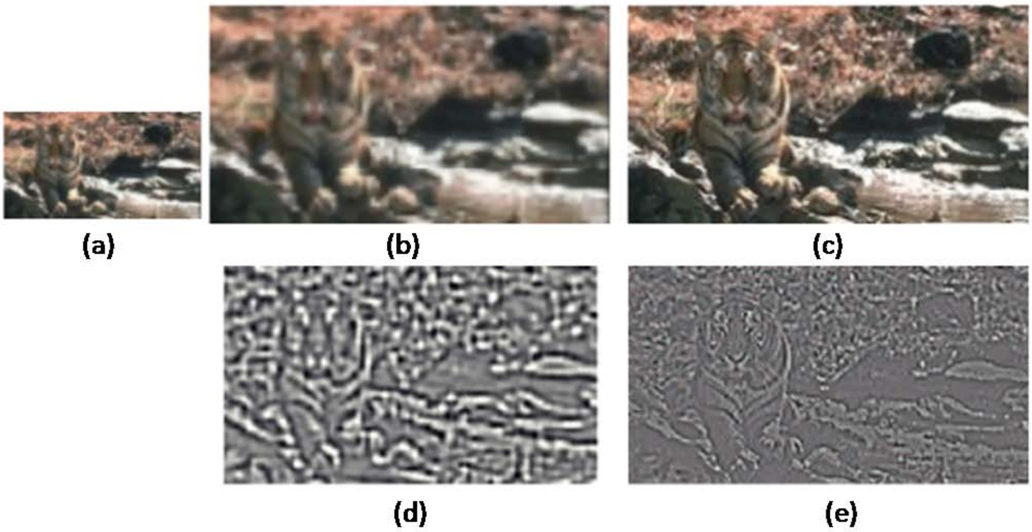


Figure 2.5: Training set images generation. (a)Low resolution input image. (b)initial cubic interpolation image. (c)orginal full frequency image. (d)Band-pass filtered and contrast normalized of (b). (e)True high frequencies of (c). Image from [13]

Markov network model

The local image information alone is not sufficient to predict the missing high resolution details. If we take a look at a input patch and its K nearest patches searched in the database, it is easy to find that although the K nearest patches are similar to the input patch and also have a similar look between each other, the corresponding HR patches are quite different from each other. This indicates that a nearest neighbor algorithm is not sufficient , spatial context must also be considered. The spatial relationships between patches are modeled as a Markov network shown in Fig.2.6 [13]. The term y is the observed node corresponding to the interpolated version of input image and x is the underlying scenes. The term y_i and x_i refer to LR patches and HR patches respectively. Each observed node y_i has many underlying

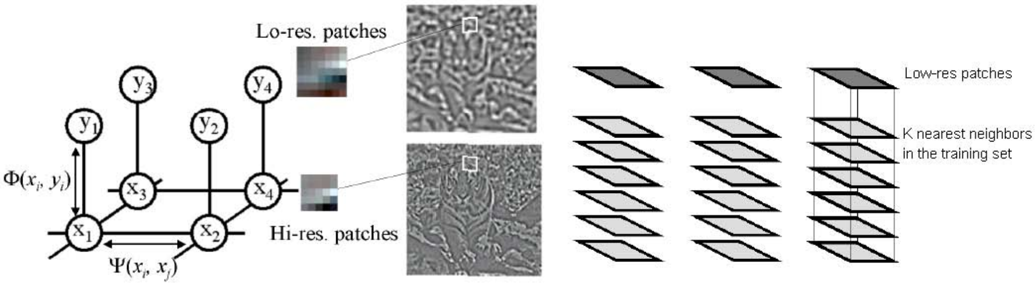


Figure 2.6: Markov network model for example-based super resolution. Image from [13]

candidate scenes by K nearest neighbor search in the training set. For the MRF, the joint probability over the scenes x and observed images y can be written as:

$$P(x_1, x_2..., x_N, y_1, y_2..., y_N) = \prod_{i,j} \Psi(x_i, x_j) \prod_{k} \Phi(x_k, y_k)\,, \tag{2.2}$$

where (i, j) indicates neighboring nodes i,j and N is the number of image and scene nodes. The term Ψ and Φ are pairwise compatibility functions where Φ is data cost and Ψ is smoothness cost in the MRF model. Data cost Φ is defined as the Euclidean distance between the input image patches and patches extracted from LR images in the training set. A K nearest neighbor search algorithm is used for each node. To specify smoothness constraint Ψ, the nodes are sampled from the input image so that the HR patches overlap with each other by one or more pixels. Let d^l_{jk} be a vector of the pixels of the l-th possible candidate for scene patch x_k which lie in the overlap region with patch j. Likewise, let d^m_{kj} denote m-th candidate vector. We say that scene candidates x^l_k (candidate l at node k) and x^m_j are compatible with each other if the pixels in their overlap regions agree. The term Ψ defines compatibility

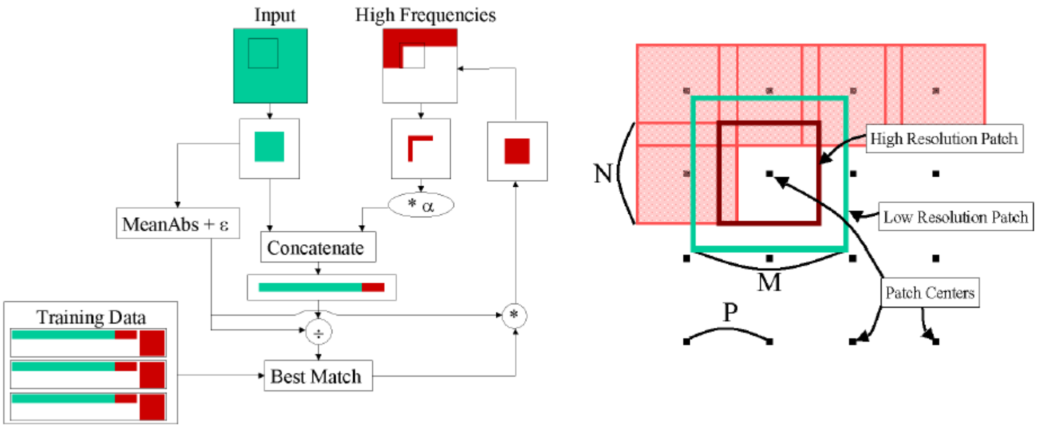


Figure 2.7: A single pass algorithm without MRF. Image form [13]

of node k and j defined as:

$$\Psi(x_k^l, x_j^m) = exp^{-|d_{jk}^l - d_{kj}^m|^2/2\sigma_s^2}\,, \tag{2.3}$$

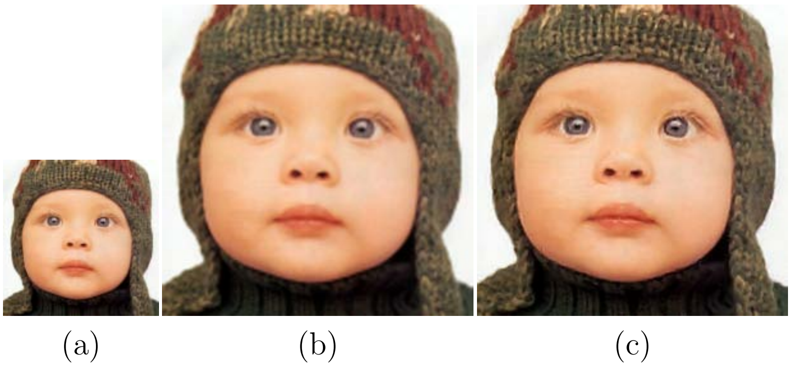
We say that a scene candidate x_k^l is compatible with an observed image patch y_0 if the image patch y_k^l in the training database matches y_0.

$$\Phi(x_k^l, y_k) = exp^{-|y_k^l - y_0|^2/2\sigma_i^2}\,, \tag{2.4}$$

The MRF model can be solved by Belief Propagation [21]. For each node x_i a compatible patch is found from the training database by solving the Markov network. The result is reconstructed by these patches.

An algorithm without MRF

Fig.2.7 from [13] illustrates an algorithm of example-based SR without introducing the Markov network while still preserving the smooth constraint. The algorithm is more efficient than solving the Markov network. The algorithm



(a) (b) (c)

Figure 2.8: Example of learning based methods. (a) Low resolution image. (b)Cubic interpolation 4x. (c)Learning 4x.

works in raster-order from left to right and top to bottom. At each step the search vector is formed by the LR input and the overlap region of previous selected HR patches. The training data is also generated by concatenated vectors. Therefore, the nearest search in the training set is not only trying to find the underling sense patch for each x_i but also trying to find the most compatible patch with previous generated patches.

Chapter 3

Edge Prior and Detail Synthesis

3.1 Introduction

As previously mentioned, approaches addressing the SR problem can be categorized as *interpolation based*, *reconstruction based(edge-directed)*, and statistical or *learning based* (for a good survey see [44]).

The major drawback of edge-directed SR approaches is their focus on preserving edges while leaving relatively "smooth" regions untouched. As discussed in [3, 31], if a SR algorithm targets only edge preservation, there exists a fundamental limit (about 5.5× magnification) beyond which high frequency details can no longer be reconstructed. Loss of these details leads to unnatural images with large homogeneous regions. This effect is demonstrated in Figure 3.1 that plots the gradient statistics of SR images with different magnification factors. Shown are bicubic upsamling (b) and edge-directed SR [39] (c). The respective gradient statistics plots shown in Figure 3.1(d-e) increasingly deviate from the heavy-tailed distribution of natural image statistics [11] as the magnification factor increases.

To produce photo-realistic results for large magnification factors, not only must edge artifacts be suppressed, but image details lost due to limited resolution need to also be recovered. Learning based techniques can achieve the latter goal; however, as mentioned in many previous works, the performance of learning based SR depends heavily on the similarity between training data and the test images. In particular, the quality of edges in the SR image can be significantly degraded when corresponding edges in the training data do not match or align well. Accurate reconstruction of edges is critical to SR, as edges are arguably the most perceptually salient features in an image.

We propose an approach that reconstructs edges while also recovering image details. This is accomplished by adding learning-based detail synthesis to edge-directed SR in a mutually consistent framework. Our method first reconstructs significant edges in the input image using an edge-directed super-resolution technique, namely the gradient profile prior [39]. We then supplement these edges with missing detail taken from a user-supplied example image or texture. The user-supplied texture represents the look-and-feel that the user expects the final super-resolution result to exhibit. To incorporate this detail in a manner consistent with the input image, we also identify significant edges in the example image using the gradient profile prior, and perform a constrained detail transfer that is guided by the edges in the input and example images.

While similar ideas have been used for single image detail- and style-

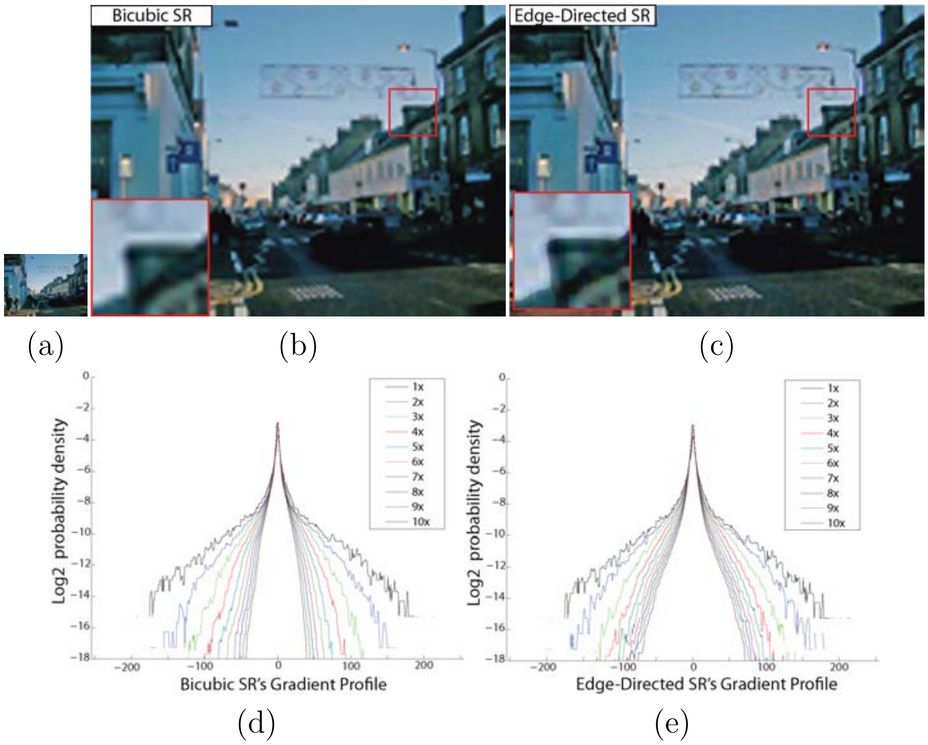


Figure 3.1: Gradient statistics of HR images using increasing magnification. (a) Input LR image; (b) 10× upsampling using bicubic interpolation; (c) 10× upsampling by edge-directed SR [39]; (d,e) gradient statistics for bicubic interpolation and edge-directed SR with 1× to 10× upsampling. For greater levels of magnification, the gradient statistics increasingly deviate from natural image statistics [11].

transfer (e.g. [16, 8, 35]), our approach is unique in that it is framed together with edge-directed SR. This gives the user flexibility in specifying the exemplar image – we can still obtain quality edges in the upsampled image even if they are not present in the example image. Experimentally, our procedure produces compelling SR results that are more natural in appearance than edge-directed SR and are on par or better than learning based approaches that require a large database of images to produce quality edges. This is exemplified by the images in Figure 3.2.

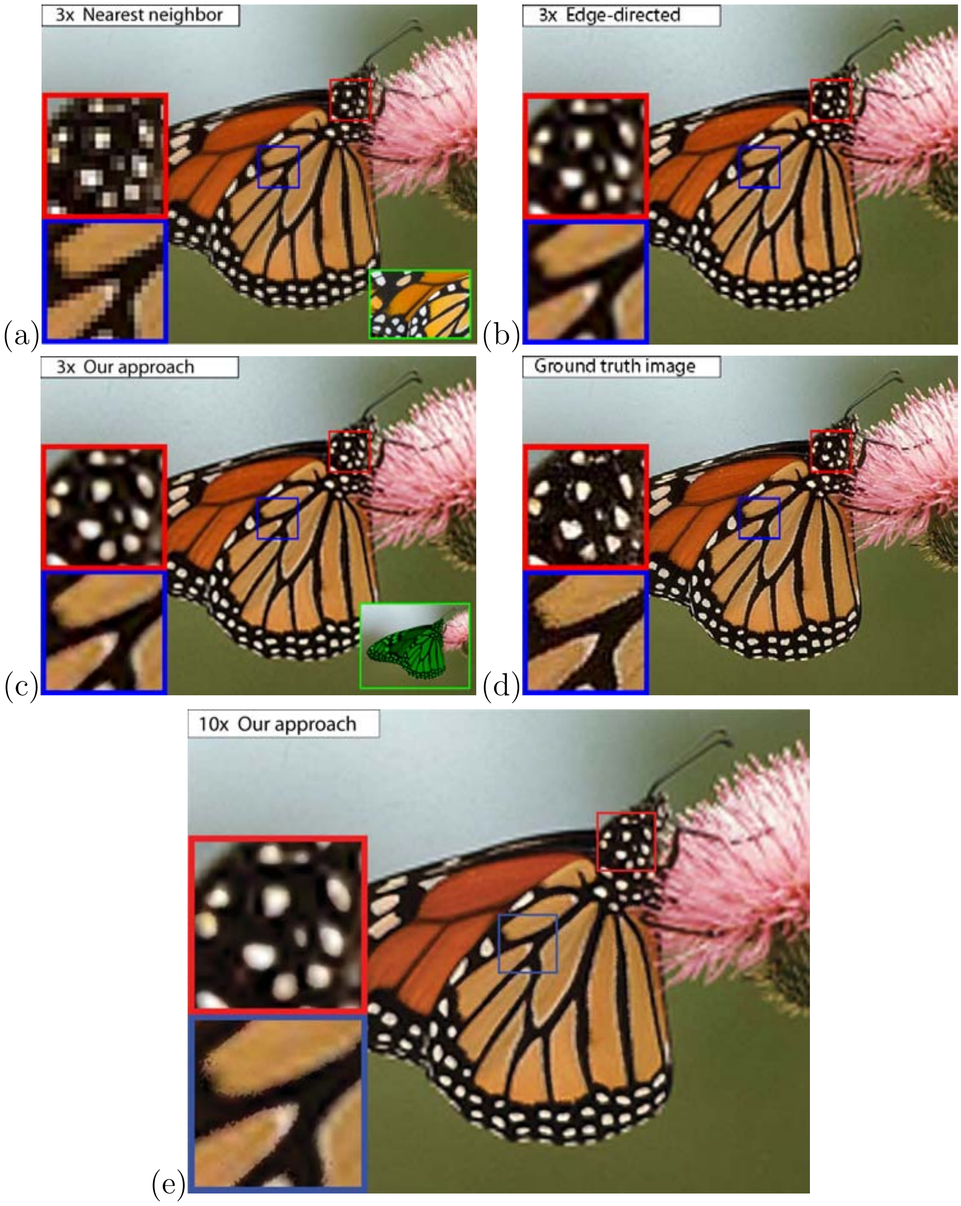


Figure 3.2: Example-based detail synthesis. (a) 3× magnification by nearest neighbor upsampling of an input low resolution (LR) image with a user supplied example image; (b) result using edge-directed SR [39]; (c) result from our approach that synthesizes details from the input example. The region where detail is transferred is shown in the lower right inset; (d) ground truth image; (e) 10× magnification using our approach. The example texture was found using Google image search with the keyword "monarch wing".

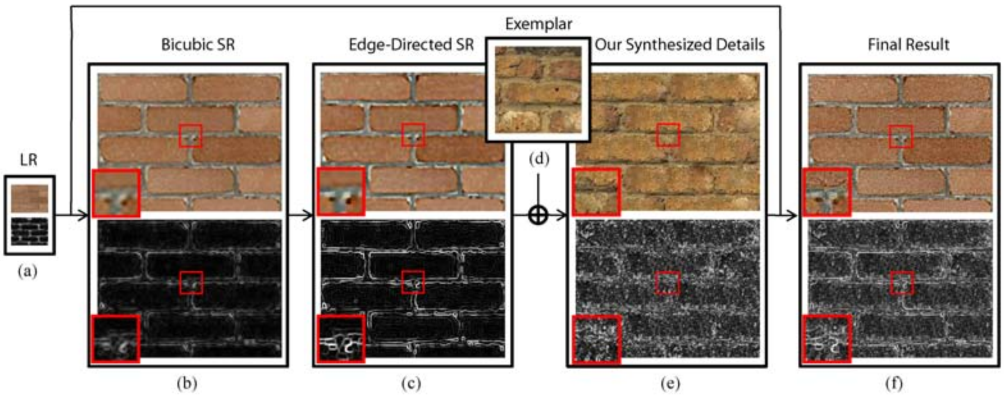


Figure 3.3: The processing pipeline of our algorithm. (a) Input LR image with its corresponding gradient profile. (b) Upsampled image and gradient profile using bi-cubic interpolation. (c) Transformed gradient field of (b) using the gradient profile prior [39] to produce sharp SR gradients. (d) Example texture. (e) High resolution gradient field constructed from the high frequency details in (d) with the image structure in (c). (f) Combined gradient field of (c) and (e) used in a reconstruction-based SR to produce the final result.

3.2 Reconstruction Framework

The processing pipeline of our approach is shown in (Figure 3.3). Given an LR image (Figure 3.3(a)) and a user supplied image/texture (Figure 3.3(d)), our goal is to produce a high resolution image (Figure 3.3(f)) such that its high frequency details resemble those in the example image/texture while preserving the edge structure from the original low resolution image. To be specific, given the LR input image, the GPP algorithm is applied to get the transformed gradient (Figure 3.3(c)). Similar procedure is applied to the user provided example image/texture (Figure 3.3(d)).(Result not shown in the Figure3.3) Then high resolution gradient field (Figure 3.3(e)) is constructed from the high frequency details in (Figure 3.3(d)) with the image structure in (Figure 3.3(c)). Finally, combine the gradient of (Figure 3.3(c)) and (Figure 3.3(e)) to obtain the guidance gradient field (Figure 3.3(f)) used in a reconstruction-based SR to produce the final result.

Our approach is framed in the standard back-projection formulation typical of reconstruction algorithms [12, 43, 31, 3, 40, 39]. The difference among these various approaches is the prior imposed on the HR image. Our approach is fashioned similar to the gradient profile prior in [39] in which a guidance gradient field, $\nabla_p I_H$, is imposed on the estimated HR image. Unique to our approach is how this $\nabla_p I_H$ is computed. This will be discussed in Section 3.3.2. First, we describe the main reconstruction algorithm which is necessary for implementation.

Within the reconstruction framework, the goal is to estimate a new HR image, I_H, given the low resolution input image I_L and a target gradient field $\nabla_p I_H$. This can be formulated as a Maximum Likelihood (ML) problem as follows:

$$\begin{aligned} I_H^* &= \arg\max_{I_H} P(I_H|I_L, \nabla_p I_H) \\ &= \arg\min_{I_H} L(I_L|I_H) + L(\nabla_p I_H|\nabla I_H) \\ &= \arg\min_{I_H} ||I_L - d(I_H \otimes h)||^2 + \beta||\nabla_p I_H - \nabla I_H||^2 \qquad (3.1) \end{aligned}$$

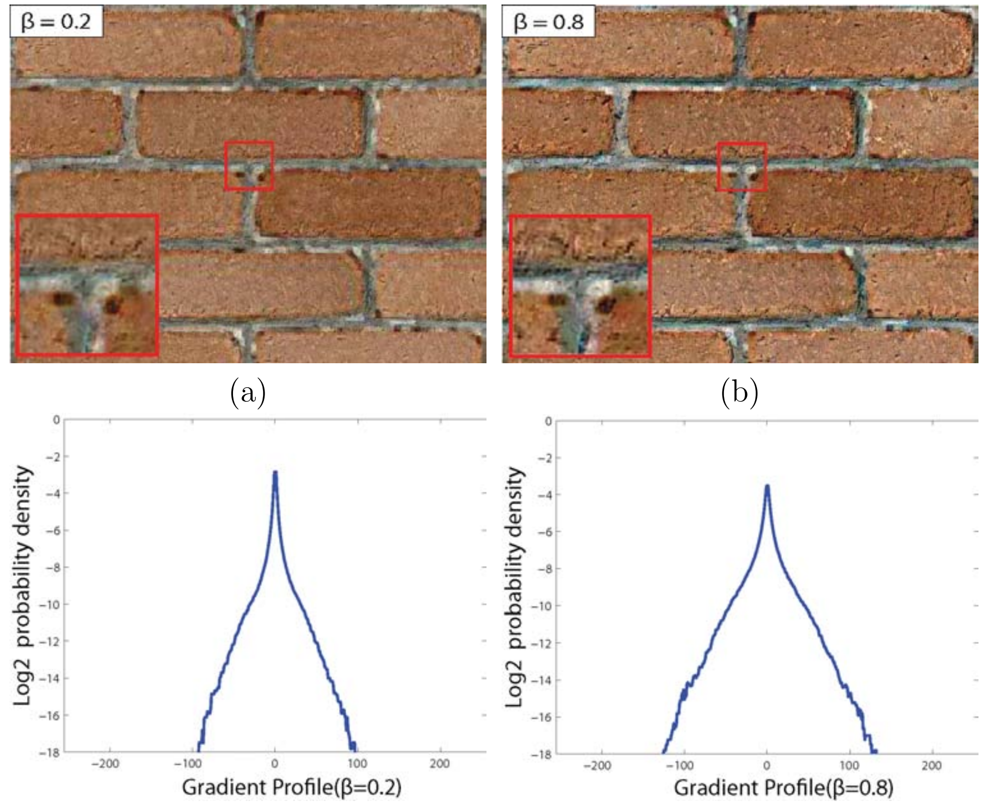
where, $L = -logP(\cdot)$, $||I_L - d(I_H \otimes h)||^2$ is the data-cost from the LR image and provides the back-projection constraint, $d(\cdot)$ is the downsampling operator, and $\otimes$ represents convolution with filter h. The term $||\nabla_p I_H - \nabla I_H||^2$ is the data-cost from the guidance gradient field $\nabla_p I_H$, and β is a weight for balancing the two data-costs. Assuming that these data-costs follow a Gaussian distribution, this objective can be cast as a least squares minimization problem with an optimal solution I_H^* obtained by gradient descent with the following iterative update rule [20, 39]:

$$I_H^{t+1} = I_H^t + \tau(I_L - u(d(I_H^t \otimes h)) \otimes p + \beta(\nabla_p^2 I_H - \nabla^2 I_H)) \tag{3.2}$$

where t is an iteration index, $\otimes$, h, $d(\cdot)$ are defined as in Equation 3.1, p is the back-projection filter, $u(\cdot)$ is the upsampling process, ∇^2 is the second derivative Laplacian operator and τ is the step size for gradient descent. In the absence of a prior, h and p are chosen to be Gaussian filters with a size proportionate to the super-resolution factor. Satisfactory results are obtained within 30 iterations with $\tau = 0.2$. The parameter β balances the amount of detail in the HR image and the back-projection constraint. The effect of β is demonstrated in Figure 3.4.

3.3 Gradient Field Estimation ($\nabla_p I_H$)

The core of our approach involves the transfer of details from the example texture to $\nabla_p I_H$ with respect to structure edges present in I_L. Our approach first upsamples edges from I_L using a reconstruction-based image SR [39]. This is described briefly in Section 3.3.1 as necessary for implementation; further details can be found in [39]. This edge-directed SR generates sharp edges in the high-resolution target gradient field, and serves as the starting point for our detail synthesis. We will also use this edge-directed SR to identify *structure edges* in the texture example. Details on the constrained texture transfer are provided in Section 3.3.2



(a) (b)

(c) (d)

Figure 3.4: The effect of β on detail synthesis. (a) Results with $\beta = 0.2$; (b) Results with $\beta = 0.8$. To evaluate the amount of detail that has been transferred, we plot the gradient statistics of (a) and (b) in (c) and (d) respectively. The value of β has a direct relationship with the amount of transferred detail.

3.3.1 Edge-Directed SR via Gradient Profile Prior

As discussed in section 2.2.2, work in [39] has shown that the 1D profile of edge gradients in natural images follows a distribution that is independent of resolution. This so-called gradient profile prior (GPP) provides an effective constraint for upsampling LR images.

The gradient profile distribution is modeled by a generalized Gaussian distribution (GGD) as follows:

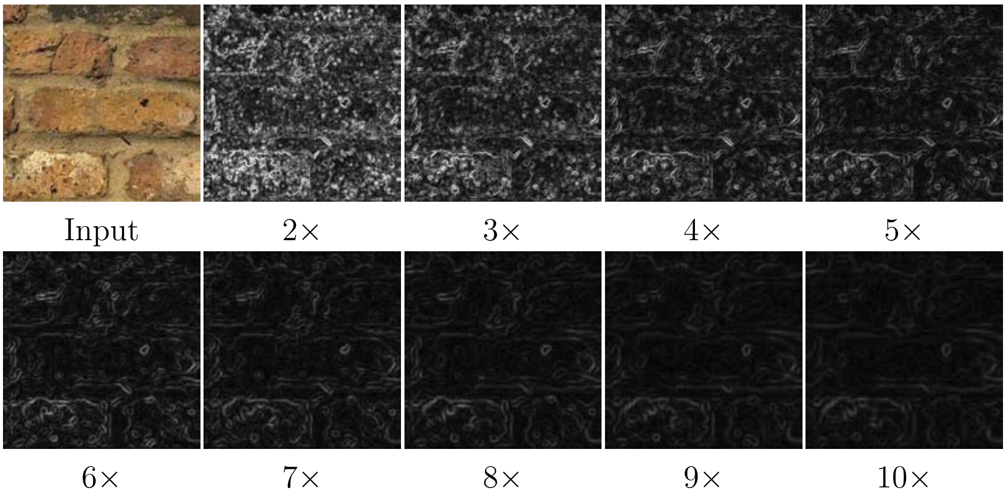
$$g(x;\sigma,\lambda) = \frac{\lambda\alpha(\lambda)}{2\sigma\Gamma(\frac{1}{\lambda})}\exp(-(\alpha(\lambda)|\frac{x}{\sigma}|)^{\lambda}) \tag{3.3}$$

where $\Gamma(\cdot)$ is the gamma function and $\alpha(\lambda) = \sqrt{\Gamma(\frac{3}{\lambda})/\Gamma(\frac{1}{\lambda})}$ is a scaling factor that makes the second moment of the GGD equal to σ^2 and thus allows estimation of σ from the second moment. The parameter λ controls the shape of the generalized Gaussian distribution. Based on a database of over 1000 images, [39] found that the gradient profile distribution of natural images has a shape approximated by a GGD with $\lambda = 1.6$.

To estimate a sharp SR gradient field based on the GPP, we can transform the gradient field of the bicubic upsampled LR image by multiplying the ratio between the gradient profiles of natural images and the gradient profiles of bicubic upsampled LR images as follows:

$$\nabla_g I_H = \frac{g(d;\sigma_h,\lambda_h)}{g(d;\sigma_l,\lambda_l)}\nabla I_L^{\uparrow} \tag{3.4}$$

where $\nabla_g I_H$ is the transformed gradient field, $\nabla I_L^{\uparrow}$ is the gradient field of the bicubic upsampled LR image, d denotes distance of a pixel to an edge maxi-



Input 2× 3× 4× 5×

6× 7× 8× 9× 10×

Figure 3.5: The amount of structure edges $\nabla_g I_E$ versus magnification factor. As the magnification factor increases, the constraints for detail synthesis decrease quadratically, which allows more (larger) details to be transferred to the super-resolution result.

mum, and $g(d; \sigma_h, \lambda_h)$ and $g(d; \sigma_l, \lambda_l)$ represent the learned gradient profiles of natural images and bicubic upsampled images, respectively. After gradient transformation, a sharper and thinner gradient field is obtained as shown in Figure 3.3(c). This procedure serves as the starting point of our detail synthesis described in the following section.

3.3.2 Synthesis of Details via Example

Given the edge-directed SR gradient field $\nabla_g I_H$ obtained using GPP, and an example image I_E, we now compute the full gradient field prior $\nabla_p I_H$ that includes synthesis of details. By synthesizing details in the gradient domain, issues with illumination and color differences between the LR image and example image are avoided. The input example image I_E represents the look-and-feel for the desired HR image and is assumed to be at the resolution of the HR image. From I_E, example patches are extracted for detail synthesis.

Extracting Structural and Detail Patches In order to better represent edge structure, we extract structure patches from the example image I_E in the following manner. We first downsample I_E to match the scale of the LR image, and then upsample its gradient field using GPP to obtain $\nabla_g I_E$, which represents the salient edge structure in I_E. Note that the amount of extracted structure edges decreases as the magnification factor increases as shown in Figure 3.5. We now form a set of exemplar patch pairs $\{\nabla E_i, \nabla_g E_i\}$, where *texture patches*, ∇E_i, come directly from ∇I_E(Figure 3.3(e) lower row shows an example of ∇I_E) and the corresponding *structural patches*, $\nabla_g E_i$, come from the $\nabla_g I_E$(Figure 3.3(c) lower row shows an example of $\nabla_g I_E$). Structural patches $\nabla_g E_i$ are different from ∇E_i, especially as magnification increases.

Detail Synthesis Our detail synthesis is formulated as a constrained texture synthesis using a Markov Random Field (MRF):

$$\begin{aligned}\nabla E^* &= \arg\min_E \sum_i P(\nabla_g I_H|\nabla_g E_i) + \sum_{(i,j)} P(\nabla E_i, \nabla E_j) \\ &= \arg\min_E \sum_i \sum_x ||\nabla_g I_H(x) - \nabla_g E_i(x)||^2 \\ &\quad + \sum_{(i,j)} \sum_{x' \in \Theta} ||\nabla E_i(x') - \nabla E_j(x')||^2 \end{aligned} \tag{3.5}$$

where $P(\nabla_g I_H|\nabla_g E_i) = \sum_x ||\nabla_g I_H(x) - \nabla_g E_i(x)||^2$ is the data-cost for aligning structural edges in $\nabla_g E_i$ with the GPP $\nabla_g I_H$, $P(\nabla E_i, \nabla E_j) = \sum_{x' \in \Theta} ||\nabla E_i(x') - \nabla E_j(x')||^2$ is the pairwise energy term to ensure neighborhood patches have similar contents among overlapping regions Θ, $\{x, x'\}$ are local patch coordinates and $\{i, j\}$ are index of nodes in the MRF network.

Since a huge number of exemplar patches can be generated from example image I_E, it is impossible to assign a discrete label to each patch in the MRF process. Therefore, for each image patch location i, we first find the best $K = 15$ candidate exemplar patch pairs that minimize the data term (using the structural patch) and the smoothness term (using the corresponding texture patch). We use patches of size 11×11 that are placed at 7-pixel intervals, providing a 4 pixel overlap. The MRF energy can be optimized using Belief Propagation (BP) [13, 40]. The final result is constructed from the exemplar texture patches, ∇E_i. Structural patches $\nabla_g E_i$ serve to help facilitate better edge alignment in the synthesis process. Feathering is used to blend patches in Θ in the final output of ∇E^*. This optimization procedure for computing ∇E^* is iterated three times, and at each iteration the best $K = 15$ candidate exemplars at each image patch location will be re-evaluated.

Final $\nabla_p I_H$ The final gradient field $\nabla_p I_H$ is then obtained by combining $\nabla_g I_H$ (edge-directed gradient) and ∇E^* (synthesized gradient) as follows:

$$\nabla_p I_H = \begin{cases} \nabla E^*, & \text{if } \nabla E^* \geq \alpha \nabla_g I_H \\ \nabla_g I_H, & \text{otherwise} \end{cases} \tag{3.6}$$

where α is set to the reciprocal of the magnification factor to maximize detail synthesis. The attenuation factor α is used to counter balance the gradient strengthening effect that edge-directed SR has on $\nabla_g I_H$.

If the user supplies stochastic texture examples with no salient edge structure, the data-cost term will have little effect and the smoothness term will dominate the MRF, resulting in standard texture synthesis. The user may choose to limit the detail synthesis only to selected regions in an image. To facilitate region selection, we currently use a fast interactive image segmentation algorithm [30].

With the estimated $\nabla_p I_H$, we can apply the reconstruction formulation with back-projection as discussed in Section 3.2 to produce the final HR image.

3.4 Results

We show results of our algorithm on a variety of examples. In addition, comparisons against other SR approaches are also presented. For all examples, the balance factor in Equation 3.2 is set as $\beta = 0.5$.

In Figure 4.7, we compare our approach with GPP [39] for 3× magnification of a monarch butterfly image. An example image was found using Google image search with the query term "Monarch Wing". This example also shows the ground truth image in Figure 3.2(d). In addition, we show a large 10× magnification in Figure 3.2(e). Such large magnification appears especially unnatural with edge-directed SR.

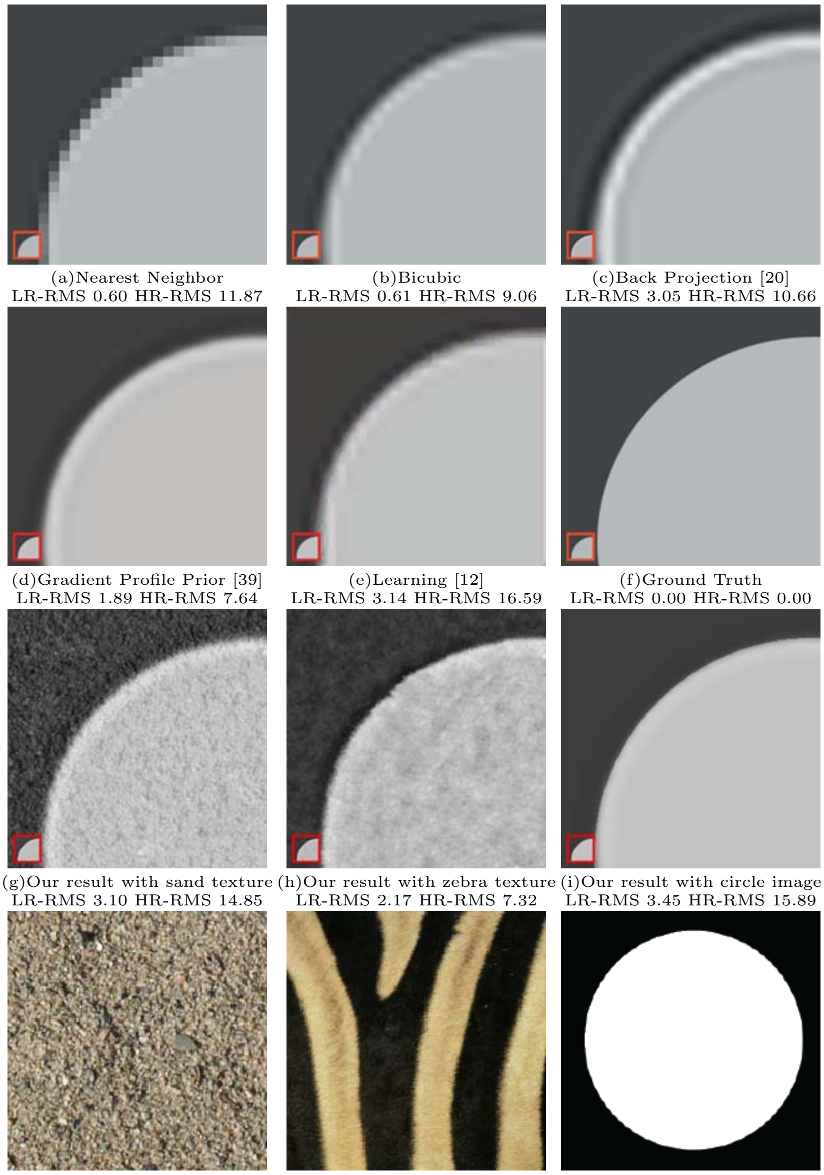
Figure 3.6 shows results with a synthetically generated circle. In this example, the root mean squared (RMS) errors are reported with respect to both the HR image and LR image. For comparison, results with bicubic interpolation, back-projection [20], GPP [39] and Learning [12] are shown. Figure 3.6(f,g,h) show three examples where different example textures/images have been used. The results exhibit the desired output with details that match the supplied examples. Our method's use of edge-directed SR and constrained detail synthesis produces detail while still preserving edge structure as evident in Figure 3.6(g,h). Although the results in Figure 3.6(f) are highly textured, the LR-RMS errors remain small under back-projection. Note that for the Learning approach [12], a generic database is used for super-resolution and hence details in regions are not synthesized. Also, since [12] does not reconstruct high resolution edges before patch matching, some aliasing artifacts remain especially under large scale magnification. This is because using low resolution edges for patch matching contains greater ambiguity. In contrast, our approach uses high resolution edges from reconstruction-based techniques to guide the patch matching, which provides a better and stronger constraint to remove aliasing artifacts. When an example similar to the ground-truth image is used as an example, our method produces a sharper and clearer result (both subjectively and in terms of RMS errors) as shown in Figure 3.6(h).

Figure 3.7 demonstrates SR results for an LR image of a boy's face with noticeable freckles (an example first used in [12]). This image is upsampled with 4× magnification in this experiment. We compare our method against generic learning based SR [12] and two edge-directed techniques ([7] and [39]). Here, we used an image of a different face with significantly different freckle pigmentation to serve as the image example (Google image search "freckle boy" for extra-large images). Our result is shown in Figure 3.7(e), and a 10× magnification is shown in Figure 3.7(g). We also compare our result in terms of HR-RMS errors against previous methods. Although our result has larger HR-RMS errors compared with edge-directed techniques ([7] and [39]), our result has much smaller HR-RMS errors compared with generic learning based SR [12]. To better evaluate our result, we show the Mean of Structural

Similarity (MSSIM) scores for our results. The MSSIM score is an image quality assessment method that closely matches the human visual system by using local means and variances for measurement [45]. Our result produces the best MSSIM score, because the synthesized details in our result match the “missing” details of the original image in terms of local variances. Previous methods over-smooth the results resulting in lower MSSIM scores.

Figure 3.8 shows a comparison of our result to the single image super resolution approach presented in [15]. From the zoom insets, we can see that the single image approach can produce very nice edges similar to edge-directed approaches (without explicit edge priors). Our result, however, can help synthesize the missing detail to make the result appear more realistic.

Several results under 8× magnification are shown in Figure 3.9. The LR input image (upsampled using nearest neighbor) and the user-supplied example image are shown in Figure 3.9(a). Comparisons with GPP [39] (Figure 3.9(b)) and a standard learning-based approach [12] (Figure 3.9(c)) are given. Figure 3.9(d) displays our result with the detail-transfer region shown in the inset and highlighted in green. Each result shows the same zoomed in region for comparison. The images used for example textures were found with Google image search as follows: (first row) “marble texture”, (second row) “bark”, (third row) “tree sparrow”. Our results have sharp edges as well as detail not obtainable with edge-directed SR or standard learning-based SR. Note for the results using [12] we include our example image into the image training database. Even with our example image included with [12], our method still produces better results.



(a)Nearest Neighbor
LR-RMS 0.60 HR-RMS 11.87

(b)Bicubic
LR-RMS 0.61 HR-RMS 9.06

(c)Back Projection [20]
LR-RMS 3.05 HR-RMS 10.66

(d)Gradient Profile Prior [39]
LR-RMS 1.89 HR-RMS 7.64

(e)Learning [12]
LR-RMS 3.14 HR-RMS 16.59

(f)Ground Truth
LR-RMS 0.00 HR-RMS 0.00

(g)Our result with sand texture
LR-RMS 3.10 HR-RMS 14.85

(h)Our result with zebra texture
LR-RMS 2.17 HR-RMS 7.32

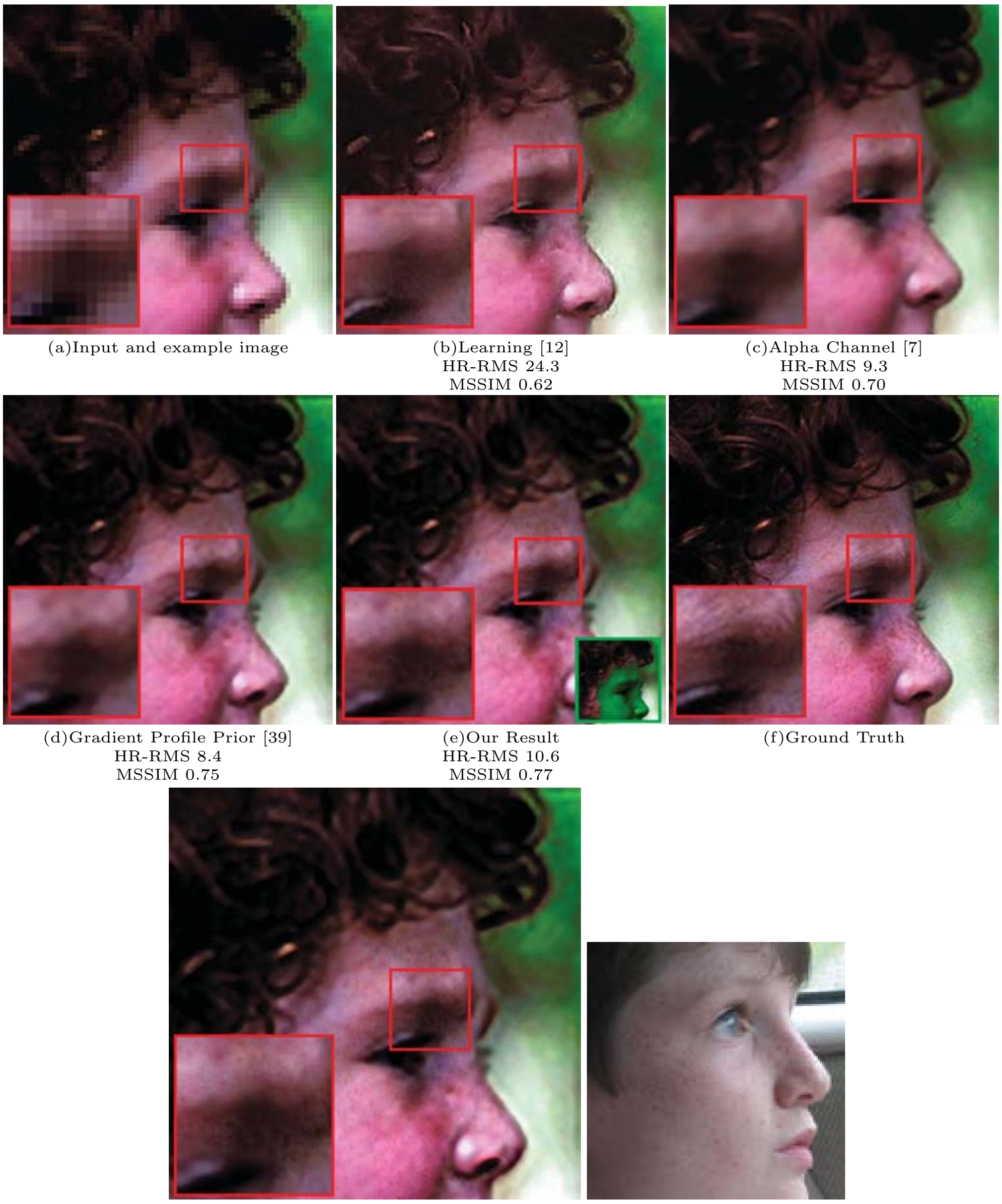
(i)Our result with circle image
LR-RMS 3.45 HR-RMS 15.89

(j)sand texture

(k)zebra texture

(l)circle image

Figure 3.6: 10× super-resolution on a synthetic example. Our approach generates different results depending on the supplied texture. The lower left corner shows the result image after 10× downsampling. Note that for all results, the down-sampled images are approximately identical. Listed below each result are the LR-RMS errors (RMS errors with respect to the low resolution input), and the HR-RMS errors (RMS errors with respect to the high resolution ground truth image).



(a)Input and example image

(b)Learning [12]
HR-RMS 24.3
MSSIM 0.62

(c)Alpha Channel [7]
HR-RMS 9.3
MSSIM 0.70

(d)Gradient Profile Prior [39]
HR-RMS 8.4
MSSIM 0.75

(e)Our Result
HR-RMS 10.6
MSSIM 0.77

(f)Ground Truth

(g)Our 10× magnification

(h)Example image

Figure 3.7: Face with freckles. (a-e) 4× magnification result of various approaches. (f) Ground truth. (g) Our result with a 10× magnification. (h)Example image. The HR-RMS errors and the MSSIM score with respect to the 4× ground truth image are listed below each result.

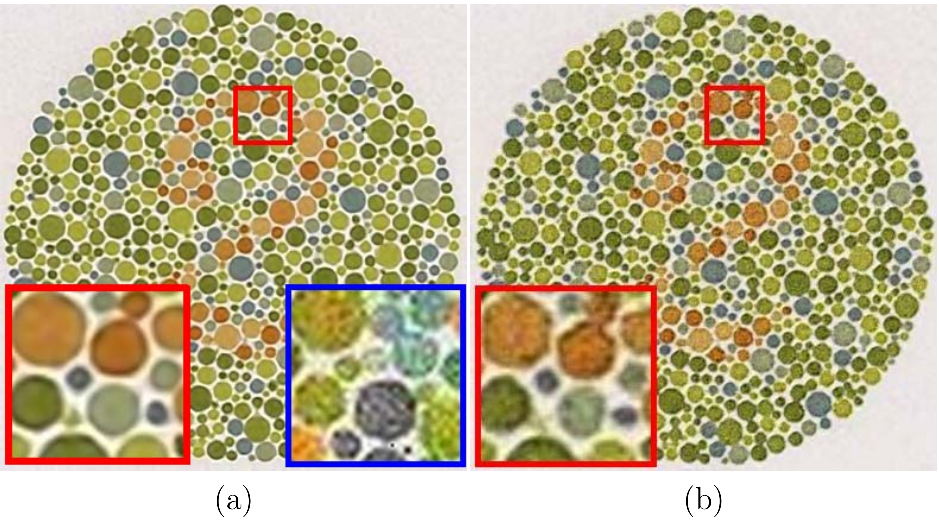


Figure 3.8: (a) Single image super resolution result from [15] with 3x magnification. The image patch in the blue border is exemplar texture, and the region in the red border is a zoom-in region. (b) Our result which synthesizes details from exemplar texture.

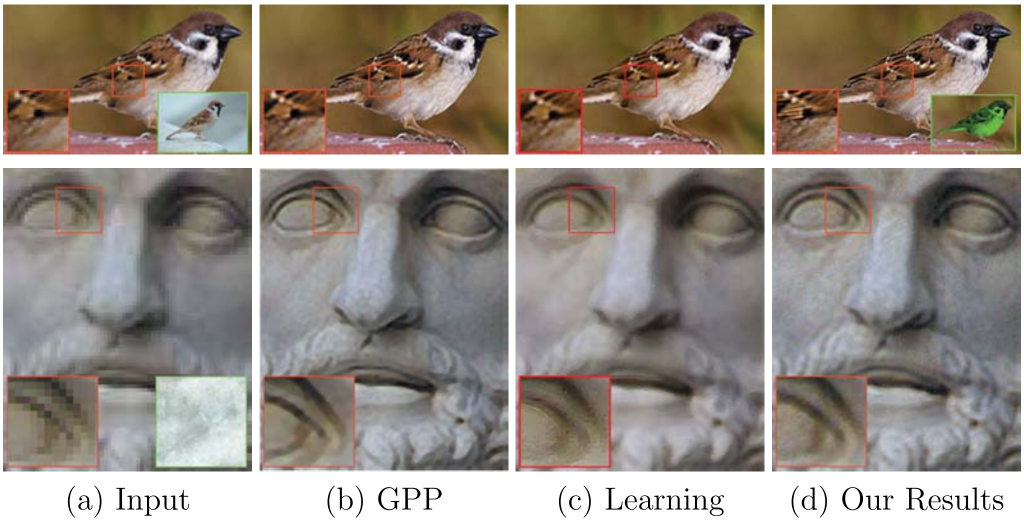


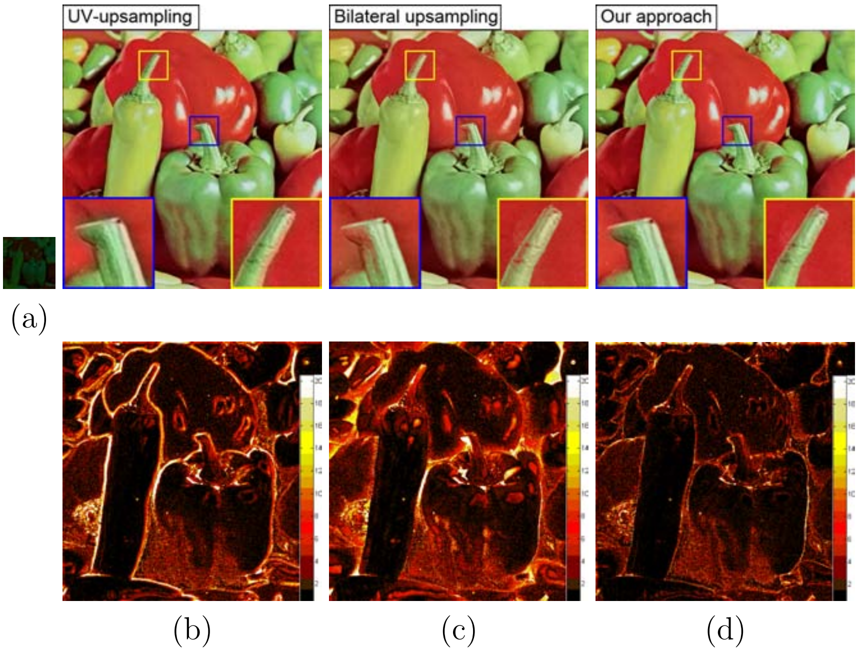
Figure 3.9: Examples with 8× magnification. (a) Input LR image (shown with nearest neighbor upsampling) and an example image/texture provided by the user; (b) results from GPP [39]; (c) results from Learning [12] with a generic database; (d) our results which synthesize details from the example image in the inset of (a). The lower left inset image in (d) highlights regions where details are transferred.

Chapter 4

Addressing Color for SR

4.1 Introduction

The existing SR techniques have successfully demonstrated ways to enhance image quality through priors or detail hallucination – how to handle color in the SR process has received far less attention. Instead, two simple approaches are commonly used to assign color. The first approach is to perform color assignment using simple upsampling of the chrominance values. This approach, used extensively in both reconstruction-based and learning-based SR (e.g. [39, 38, 5, 22]), first transforms the input image from RGB space to another color space (notably YIQ, YUV). Super resolution is applied only to the luminance channel. The chrominance channels are then upsampled using interpolation methods (e.g. bilinear, bicubic) and the final RGB is computed by recombining the new SR luminance image with the interpolated chrominance to RGB. The second approach, used primarily in learning-based techniques (e.g. [12, 32, 13]), is to use the full RGB channels in patch matching for detail synthesis, thus directly computing an RGB output.



(a)

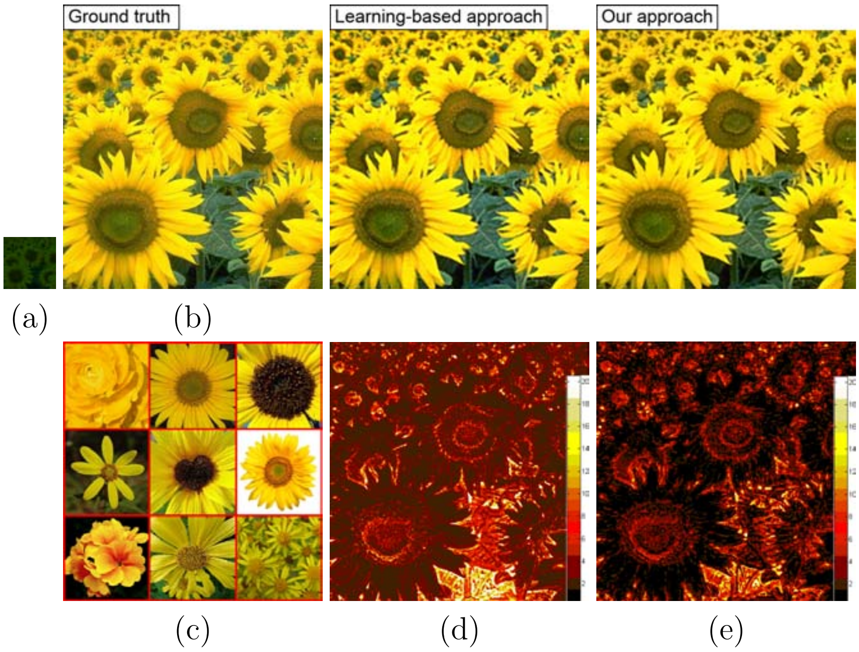
(b) (c) (d)

Figure 4.1: (a) LR chrominance input. (b) Results from bicubic interpolation of the UV channels. (c) Results from joint-bilateral upsampling [26] (d) Our result. Color difference maps are computed based on the CIEDE2000 color difference formula (e.g. see [23, 14])

These two existing approaches for SR color assignment have drawbacks. The basis for the UV-upsampling approach is that the human visual system is more sensitive to intensities than colors and can therefore tolerate the color inaccuracies in this type of approximation. However, color artifacts along the edges, are still observable, especially under large magnification factors as shown in Fig 4.1. Performing better upsampling of the chrominance, e.g. by weighted average [10] or joint-bilateral filtering [26], can reduce these artifacts as shown in Fig 4.1(c), but not to the same extent as our algorithm (Fig 4.1(d)). In addition, techniques such as joint-bilateral upsampling requires parameter-tuning to adjust the Gaussian window size and weighting parameters between the spatial and range data to obtained optimal results.

For learning-based techniques, the quality of the final color assignment depends heavily on the similarity between training data and the input image. The techniques that perform full RGB learning can exhibit various color artifacts when suitable patches cannot be found in the the training data. Approaches that apply learning-based on the luminance channel in tandem with UV-upsampling can still exhibit errors when the estimated SR luminance images contains contrast shifts due to training set mismatches. Since back-projection is often not used in learning-based techniques, this error in the SR luminance image can lead to color shifts in the final RGB assignment. Fig. 4.2 shows examples of the color problems often found in learning-based approaches.

Here, we propose a new approach to reconstruct colors when performing image super resolution. As with chrominance upsampling, our approach applies super resolution only to the luminance channel (Y). Unique to our approach, however, is the use of image colorization [28, 46] to assign the chrominance values. To do this, we first compute a chrominance map that adjusts the spatial locations of the chrominance samples supplied by the LR input image. The chrominance map is then used to colorize the final result based on the SR luminance channel. When applying our approach to learning-based SR techniques, we also introduce a back-projection step to first normalize the luminance channel before image colorization. We show that this back-projection procedure has little impact on the synthesized detail. Our approach not only shows improvements both visually and quan-



(a) (b)

(c) (d) (e)

Figure 4.2: (a) LR chrominance input. (b) ground truth image. (c)training images (d) result using learning based SR [13]. (e) our result. Color differences computed using CIEDE2000 metric.

titatively, but is straight-forward to implement and requires no parameter tuning. Moreover, our approach is generic and can be used with any existing SR technique.

4.2 Colorization Framework for SR

The pipeline of our approach is summarized in Fig. 4.3. Given a LR color image (Fig.4.3(a)), our goal is to produce a SR color image (Fig.4.3(h)). To achieve this goal, the input LR image is first decomposed into the luminance channel Y_L and chrominance channels U_L and V_L . For simplicity, we use only the U channel to represent chrominance since the operations on the U and V channels are identical. Next, the HR luminance channel Y_H is

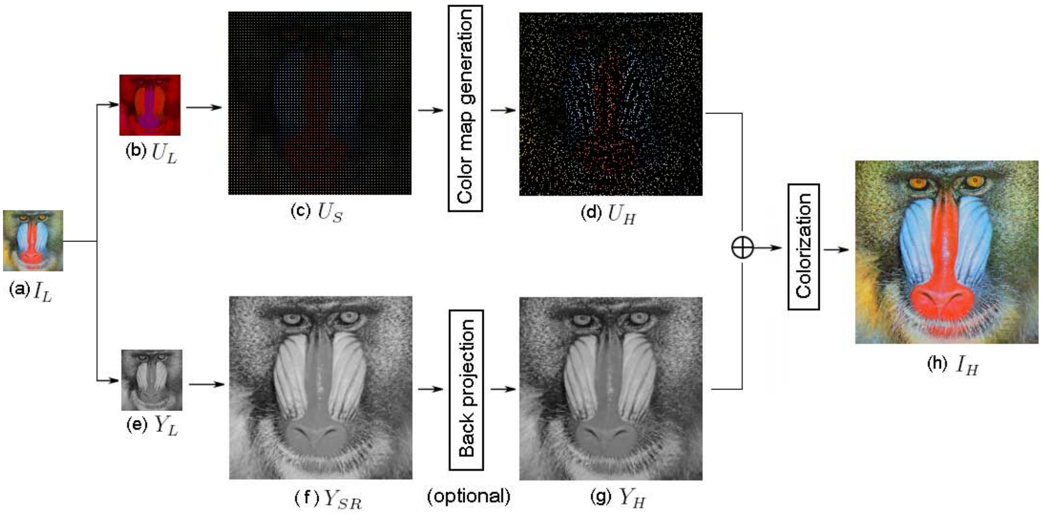


Figure 4.3: The processing pipeline of our algorithm. (a) LR input image. (b) The chrominance component of input image. (c) Initial chrominance map produced by expanding (b) with desired scale without any interpolation. (d) Adjusted chrominance map (e) The luminance component of input image. (f) Upsampled image using any single channel SR algorithm. (g) Upsampled image produced by adding back projection constraint (if necessary). (h) Combined color map (d) and SR image (g) using image colorization to produce the final result.

constructed from Y_L. This can be done by using any preferred SR algorithm. To add colors to the final SR image I_H, we use the colorization framework introduced in [28]. For the colorization, we introduce a method to generate the chrominance samples which act as the seeds for propagating color to the neighboring pixels. The chrominance samples are obtained from the low resolution input, U_L, however the spatial arrangement of these chrominance values are generated automatically from the relationships between intensities in Y_L and Y_H.

Before we explain the colorization scheme, we note that we apply back-projection for computing Y_H from Y_L when the selected SR algorithm does not already include a back-projection procedure. We explain the reason for this first before describing the colorization procedure.

4.2.1 Luminance Back-projection

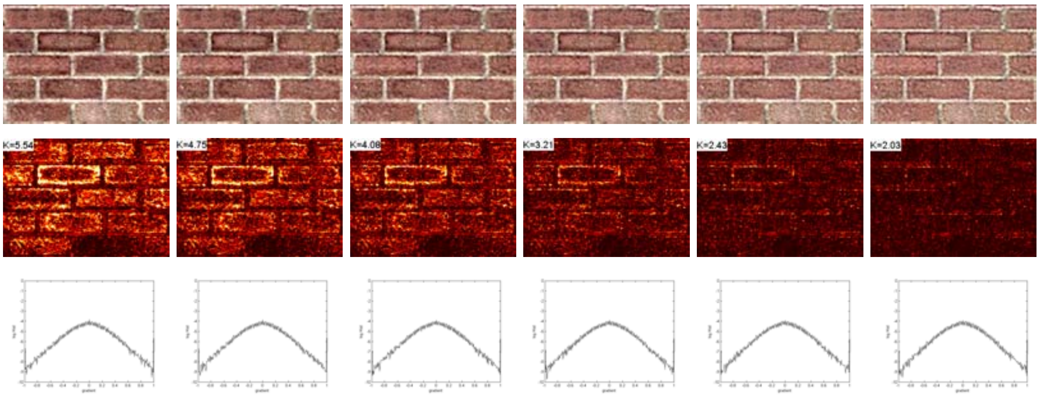
Enforcing the reconstruction constraint is a standard method which is used in many reconstruction based algorithms [41, 7, 4, 39, 38]. The difference among these various approaches is the prior imposed on the SR image. In our framework, the reconstruction constraint is enforced by minimizing the back projection error of the reconstructed HR image Y_H on the LR image Y_L without introducing extra priors. This can be expressed as as:

$$Y_H = \arg\min_{Y_H} \|Y_L - (Y_H \otimes h) \downarrow \|^2, \tag{4.1}$$

where $\downarrow$ is the downsampling operator and $\otimes$ represents convolution with filter h with proportional to the magnification factor.

Assuming the data-cost term $Y_L - (Y_H \otimes h) \downarrow$ follows a Gaussian distribution, this objective equation can be cast as a lest squares minimization problem with an optimal solution Y_H obtained by iterative gradient descent [19].

The reason to incorporate the reconstruction constraint is that the desired output should have the similar intensity values as the input image. As discussed in Section 4.1, learning-based techniques can often suffer from luminance shifts due to training example mismatches. Conventional wisdom is that the back-projection may remove hallucinated details. However, we found that adding this procedure had little effect on the synthesized details. Fig. 4.4 shows an example of the gradient histogram of the original Y_{SR} as more iterations of back-projection are applied. We can see that the gradients exhibit virtually no change, while the color errors are significantly reduced. This is not too surprising given that the estimated luminance image is downsampled by the kernel h in the back-projection process described in Eqn. (1). Thus, back-projection is correcting luminance mismatches on the low-pass image, allowing the fine details to remain. For SR techniques that already included back-projection this step can be omitted.



0 iteration 2 iterations 4 iterations 8 iterations 16 iterations 32 iterations

Figure 4.4: Illustration of the back projection procedure. Images and their color difference maps are shown at different iterations based on Eqn. (1) Colorization alone is not sufficient to correct the color shift if the luminance channel is not already normalized.

4.3 Colorization Scheme

The core of our approach lies in using image colorization to propagate the chrominance values from the LR input to the upsampled SR luminance image. In [28], chrominance values are assigned via scribbles drawn on the image by the user. In our approach, the chrominance assignment comes from the LR image and needs to be adjusted to better fit the SR luminance channel. The procedure to build the chrominance map is detailed in Section 4.3.2, we first review image colorization for the sake of completeness.

4.3.1 Image Colorization

Image colorization [28] computes a color image from a luminance image and a set of sparse chrominance constraints. The unassigned chrominance values are interpolated based on the assumption that neighboring pixels **r** and **s** should have similar chrominance values if their intensities are similar. Thus, the goal is to minimize the difference between the chrominance $U_H(\mathbf{r})$ at pixel **r** and the weighted average of the chrominance at neighboring pixels:

$$E = \sum_{\mathbf{r}} (U_H(\mathbf{r}) - \sum_{s \in N(\mathbf{r})} w_{\mathbf{rs}} U_H(\mathbf{s})) \tag{4.2}$$

where $w_{\mathbf{rs}}$ is a weighting function that sums to unity. The weight $w_{\mathbf{rs}}$ should be large when $Y_H(\mathbf{r})$ is similar to $Y_H(\mathbf{s})$, and small when the two luminance values are different. This can be achieved with the affinity function[28]:

$$w_{\mathbf{rs}} \propto e^{-(Y_H(\mathbf{r}) - Y_H(\mathbf{s}))^2 / 2\sigma_{\mathbf{r}}^2} \tag{4.3}$$

where σ_r is the variance of the intensities in a 3×3 window around **r**. The final chrominance image is obtained by minimizing the Eqn. 4.2 based on the input luminance image and chrominance constraints. The final RGB image is computed by recombining the luminance and estimated chrominance. As shown in Fig. 4.5(a), the resulting chrominance values are sensitive to the

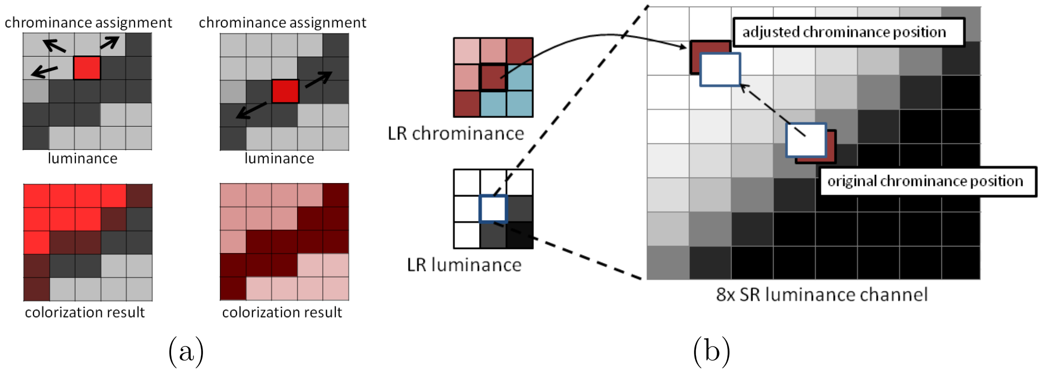


Figure 4.5: (a) An example illustrates the adjustment of chrominance values on the final colorization result. (b) Adjusting a chrominance sample after 8× upsampling the color point is shifted based on its luminance value.

position of the seed points (i.e. hard constraints), especially about the edges.

4.3.2 Chrominance map generation

Since the nature of image SR is to introduce image detail by either enforcing image priors or via hallucination, the corresponding upsampled pixels contain image content not captured by the LR pixels. Fig. 4.5(b) shows an example, where a pixel has been upsampled by a factor of 8. Blindly assigning the chrominance value to the middle of the patch may not produce the best result and can likely result in undesired color bleeding.

Our strategy is to place the chrominance value in a region of the SR luminance image that most resembles the original pixel's intensity value in input LR image as shown in Fig. 4.5(b). This approach, however, is sensitive to noise and we therefore introduce a simple Markov Random Field (MRF) formulation to regularize the search direction. Fig. 4.6 outlines the approach using an example with 8× upsampling[1]. The search directions are discretized

[1] 8× upsampling is to help illustrate our approach, our experiments are performed on

into four regions (Fig. 4.6 (a)) which serve as the four labels of the MRF ($l_x \in \{0, 1, 2, 3\}$). Let $\mathbf{x}$ be a point in the LR image and $\mathbf{X}$ be the upsampled coordinate of the point $\mathbf{x}$ ($\mathbf{X} = k\mathbf{x}$, where k is the magnification factor). Let $N_i(\mathbf{X})$ be the neighborhood of X in the direction i ($i \in \{0, 1, 2, 3\}$). Then a standard MRF formulation is derived:

$$E = E_d + \lambda E_s, \tag{4.4}$$

where E_d is the data cost of assigning a label to each point $\mathbf{x}$ and E_s is the smoothness term representing the cost of assigning different labels to adjacent pixels. The balancing term is λ is set to 1. Each cost is computed as follows :

$$E_d(l_{\mathbf{x}} = i) = \min_{\mathbf{Z} \in N_i(\mathbf{X})} |Y_L(\mathbf{x}) - Y_H(\mathbf{Z})|, \tag{4.5}$$

and

$$E_s(l_p, l_q) = f(l_p, l_q) \cdot g(Y_{pq}), \tag{4.6}$$

where $f(l_p, l_q) = 0$ if $l_p = l_q$ and $f(l_p, l_q) = 1$ otherwise. The term $g(\xi) = \frac{1}{\xi+1}$ with $Y_{pq} = \|Y_L(p) - Y_L(q)\|$, where p and q are neighboring pixels. This weighting term encourages pixels with similar LR luminance intensity values to share the same directional label. The MRF labels are assigned using belief propagation (BP) [42].

After MRF regularization, the chrominance values are adjusted to the pixel with the most similar luminance value in the regularized search direc-

4× upsampling which offers more spatial coherence for regularization

tion. Fig. 4.7 shows an example of the results obtained before and after applying the chrominance map adjustment. Bleeding is present without adjustment, however, with adjustment the results is much closer to the ground truth.

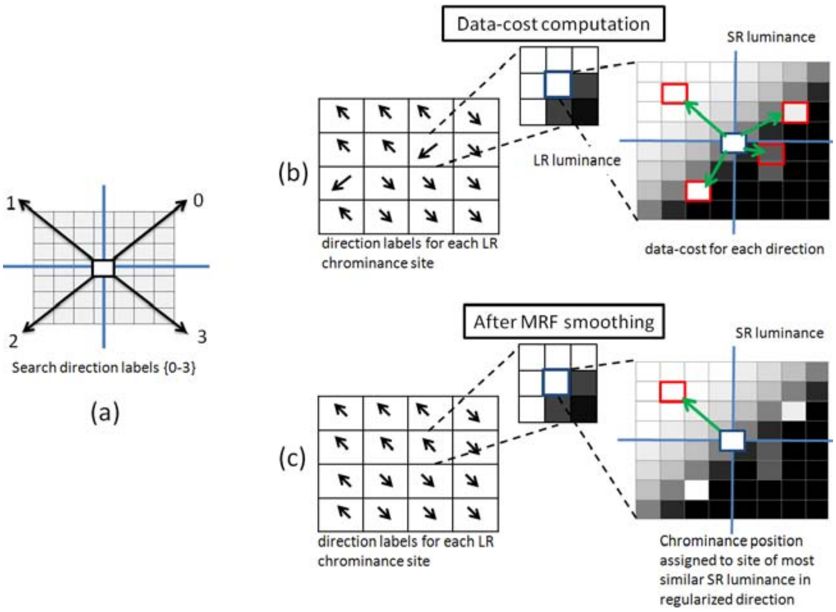

Data-cost computation
SR luminance
LR luminance
(b)
direction labels for each LR chrominance site
data-cost for each direction
1
0
2
3
Search direction labels {0-3}
(a)
After MRF smoothing
SR luminance
(c)
direction labels for each LR chrominance site
Chrominance position assigned to site of most similar SR luminance in regularized direction

Figure 4.6: The MRF example: (a) Discretized search directions. (b) Data cost computation in each search direction. (c) Smoothness constraint to regularize results.

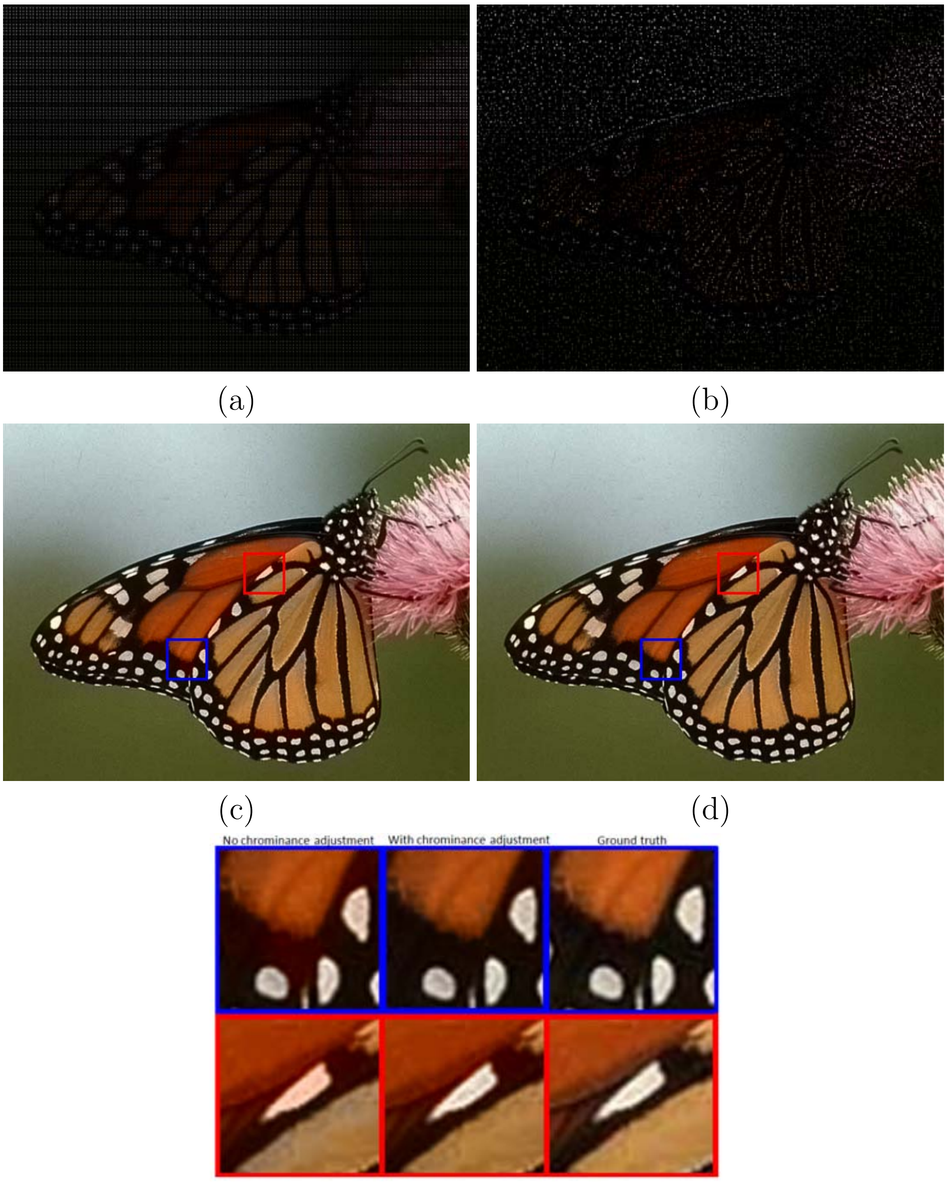
4.4 Results

Here we show results on four representative images shown in Fig. 4.4(top). For brevity, we only show the error maps and selected zoomed regions. Full resolution images of our results, together with additional examples, are available in the supplemental material. For the color difference measure, we use the CIEDE2000 metric [23, 14] together with a “hot” map. The mean color errors for all pixels as defined by CIEDE2000 metric are provided.

The first two results are shown in Fig. 4.9 and Fig. 4.10. The images have been upsampled using 4× magnification using the recent reconstruction based SR algorithm in [38]. The result was produce with executable code available on the author's project page. Our results are compared with the defacto UV-upsampling technique (also used in [38]). The overall error map for our results are better. For the zoomed regions, we can see that artifacts about edges are less noticeable using our technique.

The second two results are shown in Fig. 4.11 and Fig. 4.12. Fig. 4.4(bottom) shows the training images used for the learning examples – which are the the same images used in the [13]. We use our own implementation of the full RGB learning method using the one-pass algorithm described in [13]. For our results, we first apply back-projection on the SR luminance channel before performing the colorization step. Learning-based techniques exhibit more random types of color artifacts, however, our approach is still able to improve the results as shown in the errors maps and zoomed regions.

Our final example demonstrates the benefits of the optional back-projection procedure when the SR luminance image exhibits significant intensity shifting. In this example, only two of the training images are used to produce the SR image. Fig. 13(a) shows the result and associated error. Fig. 13(b) shows our results obtained by only applying the colorization step. Fig. 13(c) shows the results when back-projection is used followed by our colorization approach. We can see the error is significantly reduced when back-projection is incorporated.



(a) (b)

(c) (d)

Figure 4.7: (a)Initial color map U_S. (b)Color map U_H. (c)Colorization result using (a). (d)Colorization result using (b). Color map (b) produce better results without leakage at boundaries since the color points are well located.

Figure 4.8: (Top) Images used for comparison. (Bottom) Images used for learning-based SR.

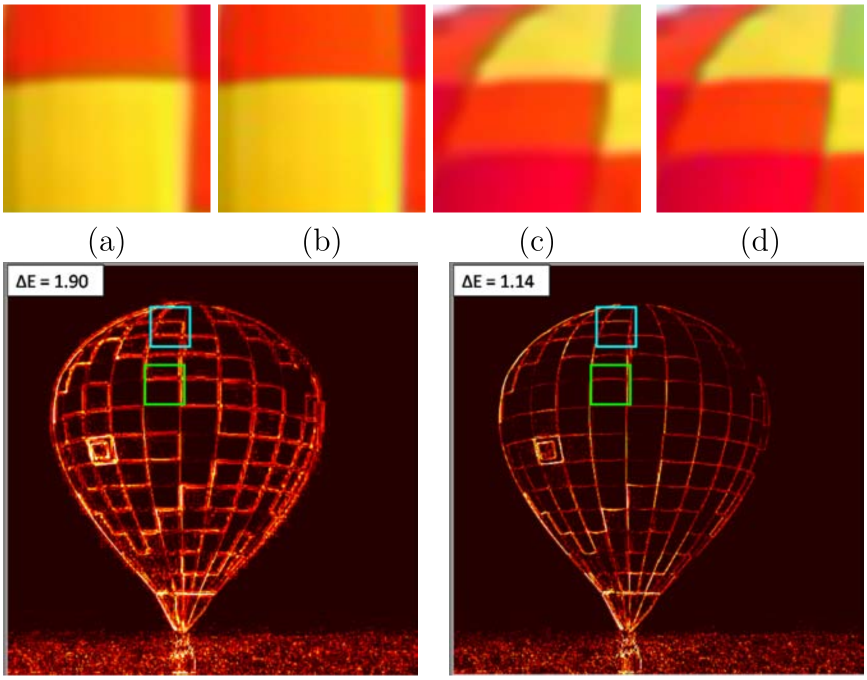


Figure 4.9: Example 1 (Ballon): 4× reconstruction-based upsampling has been applied to the "ballon" image. UV-upsampling (a,c) is compared with our result (b,d).

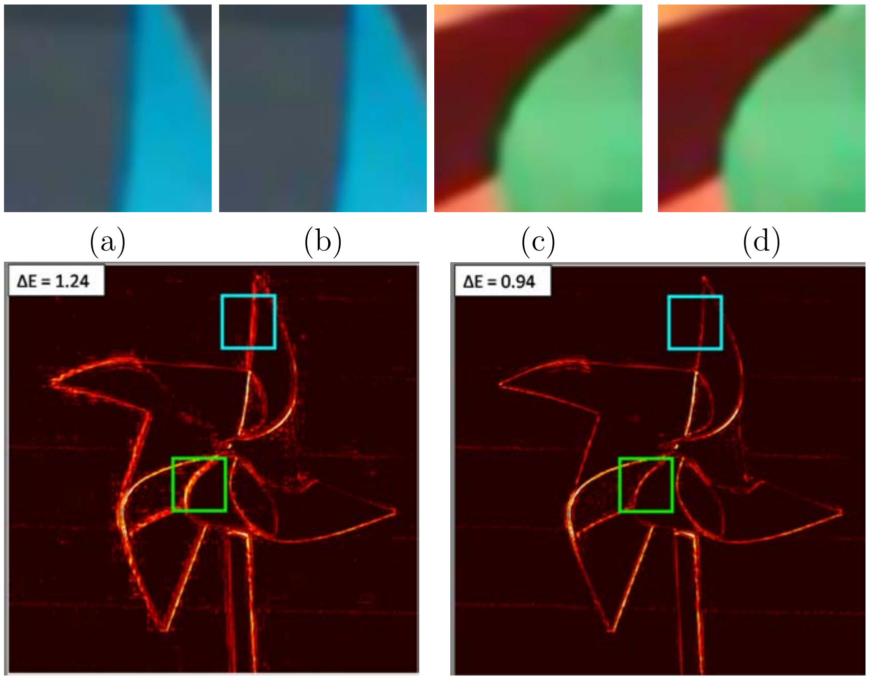


Figure 4.10: Example 2 (Pinwheel): 4× reconstruction-based upsampling has been applied to the "pinwheel" image. UV-upsampling (a,c) is compared with our result (b,d).

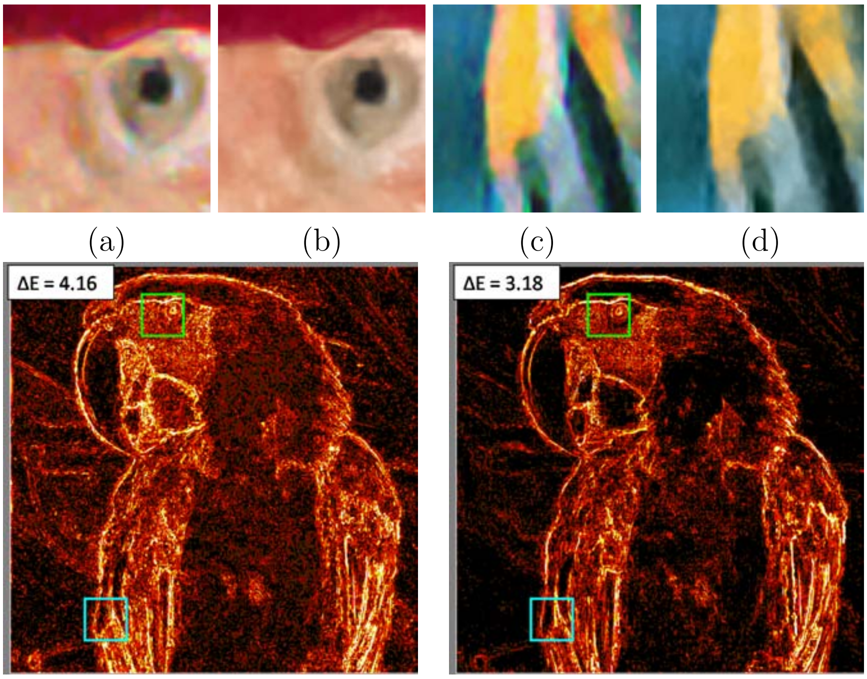


Figure 4.11: Example 3 (Parrot): 4× learning-based upsampling (a,c) has been applied to the the "parrot" image. Full RGB SR is compared with our result (b,d).

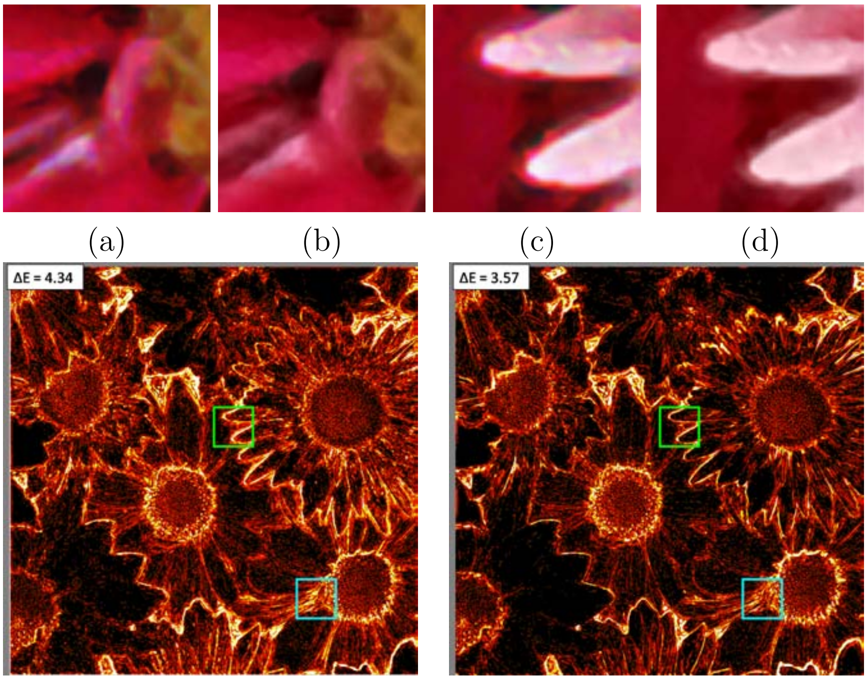


Figure 4.12: Example 4 (Flowers): Example 2 (Parrot): 4× learning-based upsampling (a,c) has been applied to the the “parrot” image. Full RGB SR is compared with our result (b,d).

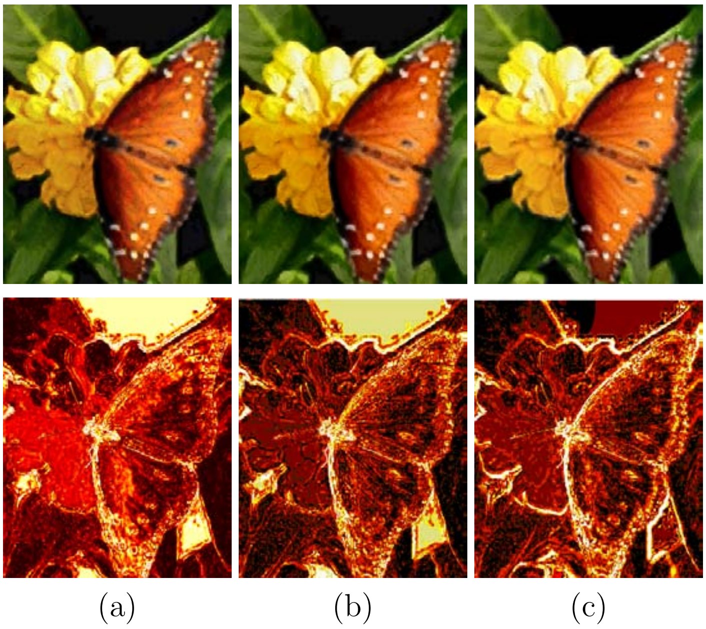


Figure 4.13: Example showing the benefits of back-projection. (a) learning-based result; (b) our approach without back-projection; (c) our approach with back-projection.

Chapter 5

Conclusion

Super resolution is a fundamentally important research topic and is widely used in many applications. In this thesis, existing super resolution algorithms are reviewed and two super resolution related algorithms are proposed.

In chapter 3, we have presented a new framework for image SR that combines edge-directed SR with detail synthesis from a user supplied example image. Our approach uses edge-directed SR to obtain sharp edges by upsampling the LR image, as well as to extract texture structure from the user supplied example. From the example, detail synthesis in the gradient domain is then applied using the edge-directed HR image. Consistency of the synthesis detail to the input image is then enforced in a reconstruction framework to produce compelling HR images that appear more natural than using learning based or edge-directed SR alone. In addition, our approach is particularly well-suited to leverage the vast example images made available by Internet image search engines and other online image repositories.

In chapter 4, we have introduced a new approach for assigning colors to

SR images based on image colorization. Our approach advocates using back-projection with learning-based techniques and describes a method to adjust the chrominance values before performing image colorization. Our approach is generic and can be used with existing SR algorithms.

Bibliography

[1] J.P. Allebach and P.W. Wong. Edge-directed interpolation. In *ICIP*, 1996.

[2] S. Baker and T. Kanade. Super-resolution reconstruction of image sequences. *IEEE TPAMI*, 21:817 – 834, 1999.

[3] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE TPAMI*, 24(9):1167–1183, 2002.

[4] M. Ben-Ezra, Z.C. Lin, and B. Wilburn. Penrose pixels: Super-resolution in the detector layout domain. In *ICCV*, 2007.

[5] H. Chang, D.Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *CVPR*, 2004.

[6] W. Ying G. Yihong D. Shengyang, H. Mei. Bilateral back-projection for single image super resolution. *Multimedia*, 2007.

[7] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong. Soft edge smoothness prior for alpha channel super resolution. In *CVPR*, 2007.

[8] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proc. ACM SIGGRAPH*, 2001.

[9] M. Elad and A. Feuer. Restoration of single super-resolution image from several blurred, noisy, and down-sampled measured images. *IEEE TIP*, 6:1646 – 1658, 1997.

[10] R. Fattal. Image upsampling via imposed edge statistics. *Proc. ACM SIGGRAPH*, 2007.

[11] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graphics*, 25(3), 2006.

[12] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *IJCV*, 40:25 – 47, 2000.

[13] W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[14] W. Wu G. Sharma and E. Dalal. The ciede2000 color difference formula: Implementations notes, supplementary test data and mathematical observations. *Color Res.Appl.*, 2004.

[15] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009.

[16] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proc. ACM SIGGRAPH*, 2001.

[17] H. Hou and H. Andrews. Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoust Speech, Signal Processing*, 26:508 – 517, 1978.

[18] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP*, 3, 1991.

[19] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion and transparency. *JVCIR*, 1993.

[20] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *JVCIR*, 4:324–335, 1993.

[21] Y. Weiss J. S. Yedidia, W. T. Freeman. *Understanding belief propagation and its generalizations.* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2003.

[22] Y. Jianchao, W. John, H. Thomas, and M. Yi. Image super-resolution as sparse representation of raw image patches. In *CVPR*, 2008.

[23] G. Johnson and M. Fairchild. A top down description of s-cielab and ciede2000. *Color Res.Appl.*, 2002.

[24] R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust Speech, Signal Processing*, 29:1153 – 1160, 1981.

[25] S. Kim and W.-Y. Su. Recursive high-resolution reconstruction of blurred multiframe images. *IEEE TIP*, 2:534 – 539, 1993.

[26] J. Kopf, M. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. In *Proc. ACM SIGGRAPH*, 2007.

[27] S. Lee and J. Paik. Image interpolation using adaptive fast b-spline filtering. *IEEE Trans. Acoust Speech, Signal Processing*, 5:177 – 180, 1981.

[28] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *Proc. ACM SIGGRAPH*, 2004.

[29] X. Li and M.T. Orchard. New edge-directed interpolation. In *ICIP*, 2000.

[30] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graphics*, 23(3):303–308, 2004.

[31] Z. Lin and H.Y. Shum. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE TPAMI*, 26(1):83–97, January 2004.

[32] C. Liu, H. Y. Shum, and C. S. Zhang. Two-step approach to hallucinating faces: global parametric model and local nonparametric model. In *CVPR*, 2001.

[33] C. Liu, H.Y. Shum, and W.T. Freeman. Face hallucination: Theory and practice. *IJCV*, 75:115–134, 2007.

[34] B.S. Morse and D. Schwartzwald. Image magnification using level-set reconstruction. In *CVPR*, 2001.

[35] G. Ramanarayanan and M. K. Bala. Constrained texture synthesis via energy minimization. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):167–178, 2007.

[36] M. Elad and P. Milanfar. S. Farsiu, M. Robinson. Fast and robust multiframe super resolution. *IEEE TIP*, 10:1327–1344, 2004.

[37] R. W. Schafer and L. R. Rabiner. A digital signal processing approach to interpolation. *Proc. IEEE*, 61:692 – 702, 1973.

[38] Q. Shan, Z. Li, J. Jia, and C.-K. Tang. Fast image/video upsampling. *ACM Trans. Graphics*, 27(5), 2008.

[39] J. Sun, J. Sun, Z. Xu, and H.Y. Shum. Image super-resolution using gradient profile prior. In *CVPR*, 2008.

[40] J. Sun, N. N. Zheng, H. Tao, and H. Y. Shum. Generic image hallucination with primal sketch prior. In *CVPR*, 2003.

[41] Y. W. Tai, W. S. Tong, and C. K. Tang. Perceptually-inspired and edge-directed color image super-resolution. In *CVPR*, 2006.

[42] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *ICCV*, 2003.

[43] M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *Third International Workshop on Statistical and Computational Theories of Vision*, 2003.

[44] J.D. van Ouwerkerk. Image super-resolution survey. *Image and Vision Comput.*, 24(10):1039–1052, 2006.

[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13:600 – 612, 2004.

[46] L. Xiaopei, W. Liang, Q. Yingge, W. Tien-Tsin, L. Stephen, L. Chi-Sing, and H. Pheng-Ann. Intrinsic colorization. In *Proc. ACM SIGGRAPH ASIA*, 2008.