Fall 2016

Region-Based Approach for Single Image Super-Resolution

Min Zhang
Old Dominion University

Recommended Citation

Zhang, Min. "Region-Based Approach for Single Image Super-Resolution" (2016). Doctor of Philosophy (PhD), dissertation, Electrical/Computer Engineering, Old Dominion University, DOI: 10.25777/xfkx-2v34
https://digitalcommons.odu.edu/ece_etds/12

REGION-BASED APPROACH FOR SINGLE IMAGE SUPER-RESOLUTION

by

Min Zhang
B.S. July 2005, Guangxi University, China
M.S. December 2010, North Carolina University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
December 2016

Approved by:

Chung Hao Chen (Director)

Chunsheng Xin (Member)

Jiang Li (Member)

Gene J. Hou (Member)

ABSTRACT

REGION-BASED APPROACH FOR SINGLE IMAGE SUPER-RESOLUTION

Min Zhang
Old Dominion University, 2016
Director: Dr. Chung Hao Chen

Single image super-resolution (SR) is a technique that generates a high-resolution image from a single low-resolution image [1,2,10,11]. Single image super-resolution can be generally classified into two groups: example-based and self-similarity based SR algorithms. The performance of the example-based SR algorithm depends on the similarity between testing data and the database. Usually, a large database is needed for better performance in general. This would result in heavy computational cost. The self-similarity based SR algorithm can generate a high-resolution (HR) image with sharper edges and fewer ringing artifacts if there is sufficient recurrence within or across scales of the same image [10, 11], but it is hard to generate HR details for an image region with fine texture.

Based on the limitation of each type of SR algorithm, we propose to combine these two types of algorithms. We segment each image into regions based on image content, and choose the appropriate SR algorithm to recover the HR image for each region based on the texture feature. Our experimental results show that our proposed method takes advantage of each SR algorithm and can produce natural looking results with sharp edges, while suppressing ringing artifacts. We compute PSNR to qualitatively evaluate the SR results, and our proposed method outperforms the self-similarity based or example-based SR algorithm with higher PSNR (+0.1dB).

ACKNOWLEDGEMENTS

First I would like to express my sincere thanks to my advisor Dr. Chen for his guidance, support and suggestions throughout my Ph.D. study. My deep appreciation goes out to my thesis committee members Dr. Hou, Dr. Li, and Dr. Xin. Furthermore, I would like to thank Old Dominion University for supporting me in fulfilling my education.

I would like to thank my family for their support and encouragement. I would also like to thank my friends and colleagues at Old Dominion University.

NOMENCLATURE

dB	Decibel
EM	Expectation-Maximization
HR	High Resolution
MAP	Maximum A Posteriori Probability
ML	Maximum-Likelihood
LR	Low Resolution
PSNR	Peak Signal-To-Noise Ratio
PSF	Point Spread Function
SR	Super Resolution

TABLE OF CONTENTS

Page

NOMENCLATURE v

LIST OF FIGURES vii

LIST OF TABLES ix

Chapter

1. INTRODUCTION 1
   1.1 WHAT IS SUPER-RESOLUTION 1
   1.2 CHALLENGE FOR SUPER-RESOLUTION 5
   1.3 MOTIVATION 6
   1.4 PROBLEM STATEMENT 9
   1.5 CONTRIBUTION AND ORGANIZATION 11

2. STATE-OF-THE-ART SINGLE IMAGE SR APPROACHES REVIEW 13
   2.1 EXAMPLE-BASED SR ALGORITHMS REVIEW 13
   2.2 SELF-SIMILARITY BASED SR ALGORITHMS REVIEW 16

3. PROPOSED SINGLE IMAGE SUPER-RESOLUTION METHODOLOGY 25
   3.1 OVERVIEW OF PROPOSED SR ALGORITHM 25
   3.2 IMAGE PRE-PROCESSING 30
   3.3 SELF-SIMILARITY BASED IMAGE SUPER-RESOLUTION 34
   3.4 EXAMPLE BASED IMAGE SUPER-RESOLUTION 46
      3.4.1 SPARSE CODING BASED SR METHOD 49
      3.4.2 OVER-COMPLETE DICTIONARY TRAINING 54
      3.4.3 JOINT DICTIONARY TRAINING 58

4. EXPERIMENTAL RESULTS 63
   4.1 COMPARISON OF TWO EXISTING SR ALGORITHMS 63
   4.2 COMPARISON OF OUR PROPOSED METHOD VERSUS EXISTING METHODS 83

5. CONCLUSION AND FURTURE WORK 96
   5.1 CONCLUSION 96
   5.2 FUTURE RESEARCH WORK 98

REFERENCES 99

VITA 114

LIST OF FIGURES

Figure Page

1. FRAMEWORK OF INTERPOLATION BASED SUPER RESOLUTION [18] 2

2. IMAGES SHOWING ARTIFACTS. 3

3. FRAMEWORK OF EXAMPLE-BASED SUPER-RESOLUTION APPROACH 14

4. ARCHITECTURE OF SELF-SIMILARITY BASED SR ALGORITHM [11] 19

5. FRAMEWORK OF SUB-BAND SELF-SIMILARITY SR ALGORITHM [36] 20

6. FRAMEWORK OF PROPOSED SINGLE IMAGE SUPER-RESOLUTION 26

7. DIVIDE IMAGE INTO SMALL REGIONS. 29

8. EXAMPLE RESULTS OF IMAGE SEGMENTATION ON THE 21-CLASS MSRC DATASET [113]. 33

9. IMAGE PATCH RECURRENCE ACROSS DIFFERENT SCALES OF THE SAME IMAGE.. 36

10. SELF-SIMILARITY BASED SINGLE IMAGE SUPER-RESOLUTION 37

11. EXAMPLE DEMONSTRATING TRANSFORMATION OPERATION 42

12. DIFFERENT TYPES OF GEOMETRIC TRANSFORMATIONS APPLIED TO TARGET IMAGE PATCH...45

13. MARKOV NETWORK MODEL FOR THE SUPER-RESOLUTION PROBLEM [12] ..48

14. RESULTS OF THE GIRL IMAGE MAGNIFIED BY A FACTOR OF 2 64

15. RESULTS OF IMAGE LENA MAGNIFIED BY A FACTOR OF 2 67

16. RESULTS OF FLOWER IMAGE MAGNIFIED BY A FACTOR OF 2 69

17. RESULTS OF LITTLE GIRL MAGNIFIED BY A FACTOR OF 4 71

18. RESULTS OF IMAGE LENA MAGNIFIED BY A FACTOR OF 4 73

Figure Page

19. RESULTS OF FLOWER IMAGE MAGNIFIED BY A FACTOR OF 4 75

20. RESULTS OF URBAN ARCHITECTURE MAGNIFIED BY A FACTOR OF 4 77

21. RESULTS OF URBAN ARCHITECTURE MAGNIFIED BY A FACTOR OF 4 78

22. RESULTS OF SQUIRREL MAGNIFIED BY A FACTOR OF 4..................................80

23. RESULTS OF PENGUIN MAGNIFIED BY A FACTOR OF 4....................................81

24. RESULTS OF HORSE MAGNIFIED BY A FACTOR OF 4 85

25. RESULTS OF TIGER MAGNIFIED BY A FACTOR OF 4 86

26. RESULTS OF NATURAL SCENE MAGNIFIED BY A FACTOR OF 4. 88

27. RESULTS OF PENGUIN MAGNIFIED BY A FACTOR OF 4....................................90

28. RESULTS OF BUTTERFLY MAGNIFIED BY A FACTOR OF 4 92

29. RESULTS OF HUMAN MAGNIFIED BY A FACTOR OF 4.......................................93

LIST OF TABLES

Table Page

1. SUPER-RESOLUTION QUALITY MEASUREMENT ON EXAMPLES IN PSNR 76

2. QUANTITATIVE COMPARISON ON 100 URBAN IMAGES IN PSNR 82

3. QUANTITATIVE COMPARISON ON 100 IMAGES OF NATURAL SCENE IN PSNR.82

4. SUPER-RESOLUTION COMPARISON ON EXAMPLES IN PSNR 94

CHAPTER 1

INTRODUCTION

A digital image is made of pixels; spatial image resolution refers to the pixel density of an image [1, 18]. An image with higher resolution contains more high frequency details. Image spatial resolution is usually limited by image acquisition devices, optical distortions, and motion blurs [18]. Super-resolution refers to methods that are used to generate or reconstruct high-resolution (HR) images from a single or a set of low-resolution (LR) images from the same scene.

1.1 What Is Super-Resolution

One simple and computationally efficient method is the interpolation based SR method. As shown in Figure 1, the high-resolution image is generated by combing multiple LR images. Those LR images are blurred and are a noisy version of the true HR image. Let $\boldsymbol{X}$ denote the HR image; $\boldsymbol{Y_i}$ is the i^{th} LR image from the same scene. Given $\boldsymbol{k}$ LR images of $\boldsymbol{X}$**,** the interpolation based method recovers HR image $\boldsymbol{X}$ through three steps: 1) low resolution images registration; 2) interpolation; 3) de-blur and remove noise [18]. In the registration step, motion model such as shifting needs to be estimated. This estimation error will affect the quality of the final HR image. In the interpolation step, non-uniform interpolation is applied. The goal of the last step is to remove noise and blur. However, since no additional HR information is provided, the resulting image lacks HR details and tends to have ringing or aliasing artifacts. The registration errors from the first step will affect the quality of the HR image as well. All those drawbacks prevent the interpolation based SR algorithm from being used in real-world applications.

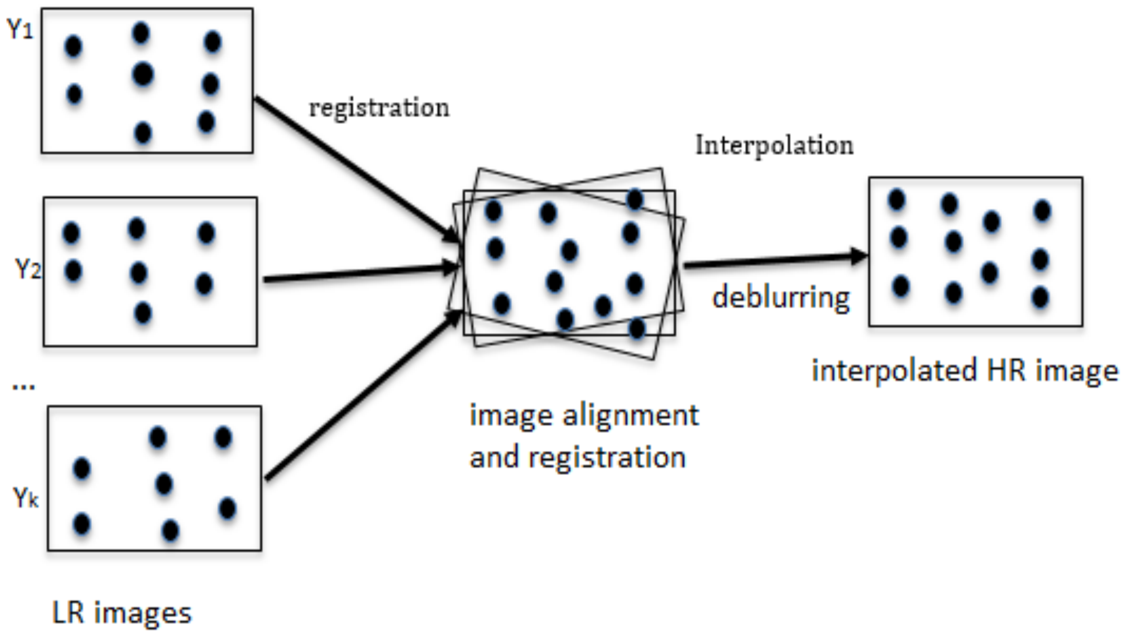


Figure 1: Framework of Interpolation Based Super Resolution [18]

Images from figure 2 demonstrate the ringing artifacts and aliasing artifacts by comparing them with the same images without artifacts. The ringing artifact usually appears near edges as we see in figure 2(a). As explained by Mitchell [114], the ringing artifacts are caused by the Gibbs phenomenon. In the frequency domain, when a single image that does not contain high frequency information passes through a low-pass filter, it will cause ringing artifacts. Thus improper shifting estimation or alignment between sub-images results in ringing artifacts. The Nyquist-Shannon sampling theorem states that aliasing happens when a signal is sampled at less than the double of the highest frequency contained in the signal [115]. Figure 2(c) shows an example of aliasing artifacts.

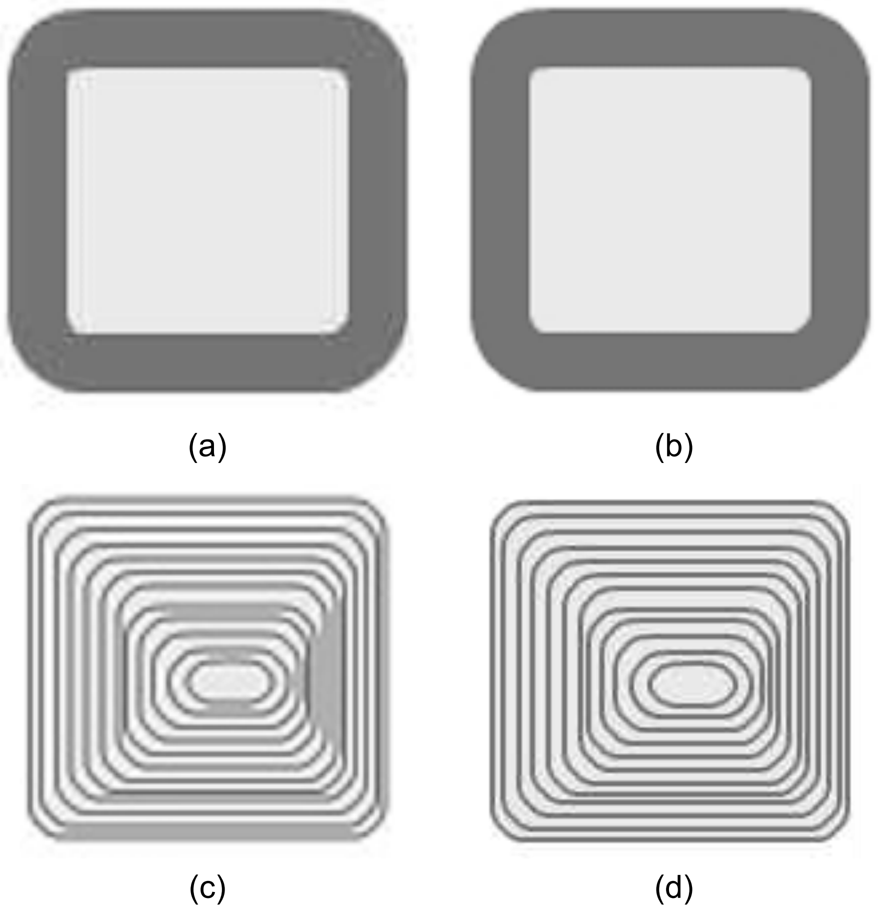

(a)
(b)
(c)
(d)

Figure 2: Images showing Artifacts. (a) image with ringing artifacts (b) same image without ringing artifacts. (c) image with aliasing artifacts (d) same image without aliasing artifacts

Example-based SR approaches [10, 11] train a set of LR-HR image patch pairs; $\{x_i\}_{i=1}^{n}$ are HR image patches, and $\{y_i\}_{i=1}^{n}$ are their corresponding LR image patches. For each image patch x_i, y_i is the blurred and down-sampled version of x_i. Each image patch pair $\{x_i,\ y_i\}$ is constrained by $y_i = DHx_i + v$, where D is the down-sampling operation, $\boldsymbol{H}$ is the blurring operation, and *v* is random noise. The relationship between the high-resolution image patch and low-resolution image patch is learned from the training dataset. This relationship will be used as a priori information. This learned information could be used to reconstruct the high-resolution details for the LR input image [18].

Statistical SR algorithms treat the blur effect and motions or shifting among low-resolution images as stochastic variables, and the desired high-resolution image is considered as stochastic variable as well [18]. Let ***X*** denote the desired HR image, and Y denote the LR version image of ***X***. ***Y*** is affected by the blurring operation and noise. Let ***M*** denote the blurring and down-sampling operation, then the SR problem can be described as:

$$X = \arg\max_X \Pr\left(X \middle| \underline{Y}\right) \qquad (1.1).$$

There are four popular statistical approaches to the estimate ***X*** [18]. One popular method is the Maximum a Posteriori (MAP) approach, where ***M*** is assumed given. Then Eqn. 1.1 becomes

$$X = \arg\max_{X} \Pr(\underline{Y}|X, M)\Pr\,(X)$$

$$= \arg\min_X \left\{\left\|\underline{Y} - MX\right\|^2 + \lambda A(X)\,\right\} \qquad (1.2).$$

Another popular statistical method is the Maximum Likelihood (ML) statistical method, where blurring estimation ***M*** is assumed as a prior. ***M*** is assumed uniform over ***X,*** and the most likely estimation for ***X*** becomes:

$$\hat{X}_{ML} = \arg\min_X \left\|\underline{Y} - MX\right\|^2 \qquad (1.3).$$

If all the parameters are known, $\hat{X}_{ML}$ can be calculated directly by:

$$\hat{X}_{ML} = (M^T M)^{-1} M^T \underline{Y} \qquad (1.4).$$

The ML method is ill-suited for real world practice; it will widely spread the noise and registration errors [18]. What's more, if $\boldsymbol{M^T M}$ is singular, there will be infinite possible solutions for equation 1.4.

Given several low-resolution images, Bayesian approaches [20-23] reconstruct HR image through three steps: 1) Use a Gaussian process prior to estimate the high-resolution image; 2) Estimate the blurring parameters and point spread function (PSF) by integrating over the estimated high-resolution images; 3) Fix the estimated parameters, and apply MAP to estimate the HR image.

Joint MAP restoration methods divide the SR problem into three steps: 1) registration, 2) restoration, and 3) interpolation [18]. The registration and restoration parameters are estimated simultaneously using Expectation-Maximization (EM). By doing this, the blurring parameters estimation and HR estimation can benefit each other.

1.2 Challenge for Super-Resolution

In recent years, researchers have focused on different ways to improve the quality of HR images recovered from the LR image, and they have made certain progress. However, there are still many challenges that prevent the application of SR algorithms in real world practice.

One well-known challenge in image processing is image registration. This problem is even worse in multi-image SR techniques. The registration errors will widely spread during the SR image reconstruction process and, as a result, will generate annoying artifacts, ringing effects or aliasing [18].

The second challenge in SR techniques is computation efficiency. For example-based SR algorithms, the quality of the recovered HR image depends on the similarity between the input image and training dataset. In order to generate high quality HR images in general, a huge database is needed. This training database needs to contain as many different types of image contents as possible. As a result, it will cause extremely heavy computation costs while searching for the best matched image patch from the database [18]. For self-similarity based SR algorithms, building the internal database through the image itself, and searching for similar image patches using a different transformation matrix also requires expensive computation costs.

1.3 Motivation

We have introduced several popular super resolution approaches in the previous section. In this section we will discuss why we are interested in reconstructing a high-resolution image from a single image instead of using multiple low-resolution images. Then we will introduce the state-of-the-art single image super-resolution methods followed with our proposed SR method.

Multiple image based super-resolution assumes multiple LR images of the same scene are available; the high-resolution value is estimated by combining the information from each low-resolution image. Each low-resolution image contributes a new constraint on estimating the high-resolution image. In practice, multiple LR image based super-resolution algorithms face different challenges such as image registration, motion estimation, and proper low resolution blur filter estimation [18]. Image registration is a well-known ill-posed image processing problem; researchers are still working on the algorithm to increase registration accuracy. Registration errors will propagate in the HR image estimation step and result in severe artifacts.

In real world applications, it is hard to get several images from the same scene. Thus, generating a high-resolution image from a single LR image becomes very important. It is also worth mentioning that single image SR algorithms do not involve image registration or shift estimation between multiple images. As a result, registration errors can be avoided, so in this paper, we will propose approaches to generate an SR image from a single low-resolution (LR) image.

The goal of single image super-resolution is to reconstruct a high-resolution image with a sharp edge from a single low-resolution image. Single image super-resolution algorithms can be broadly classified into two categories: example-based and self-similarity based algorithms [18].

The basic idea of example-based super resolution methods can be divided into two steps: 1) model the relationship between the high-resolution and lower-solution image pairs through the training data; 2) apply this model to the new input low-resolution image in order to generate its corresponding high-resolution image. Example-based super-resolution algorithms usually need an external database composed of high-resolution (HR) image patches and their corresponding low-resolution (LR) patches. For each LR patch from the input image, one of the first *k* best-matched LR patches will be found from the database. The corresponding HR patches will be used to predict the HR details of the given LR image patch.

Self-similarity based super-resolution algorithms [1, 10, 11] are based on the observation that patches of a natural image repeat within or across different scales of the same image, so an internal database containing LR-HR image patch pairs can be built through a pyramid of the given image itself [1, 10]. This internal database will be used to learn the relationship between LR and HR images.

Each SR algorithm may have its own strength and limitation. One method may perform better than other state-of-the-art methods on some types of images with particular textural structure. The performance of example-based algorithms depends on the similarity between the testing image and image database. Usually a huge set of training data is needed for better performance in general, which will result in heavy computation costs. It is unclear how big training LR-HR image patches are needed in order to generate good results in general. Database retraining is needed for different scaling factors. What’s more, the high-resolution details learned from LR-HR image patches of the database cannot be guaranteed accurate [1,18].

As for self-similarity based SR methods, the internal database built from the image itself contains more relevant LR-HR image patches. The reconstructed HR image has fewer ringing and artifact effects compared to example-based algorithms. One common limitation of self-similarity algorithms is the ability to recover textural details for super low-resolution input images.

1.4 Problem Statement

Single image super-resolution is intended to reconstruct a high-resolution image from the single low-resolution input image. The recovered high-resolution image should obey the following constraints: 1) the recovered high-resolution image contains more high frequency details with sharp edges, and 2) the recovered image needs to be consistent with the original low-resolution input image. Single image super-resolution algorithms can be classified into two groups: 1) example-based super-resolution, and 2) self-similarity based super-resolution. Each type of image super-resolution algorithm has its own advantages and limitations.

As we know, example-based super-resolution algorithms [2, 12, 103] learn the relationship between high-resolution images and their corresponding low-resolution images through an external training database. They then apply this relationship to recover the high-resolution details for the testing low-resolution input image. Example-based image super-resolution algorithms have shown the ability to produce high-resolution images with more details and sharper edges when there exists an image in the database that is similar to the testing low-resolution input image. However, the general performance of such algorithms highly depends on the similarity between the testing image and the images from the training database. What’s more, the recovered high-resolution details are not always consistent with the ground truth [12], and it may generate some details that do not exist in the ground truth image.

Research found that image content tends to repeat both within an image itself at different locations and across different scales of the original image [11]. Self-similarity based super-resolution algorithms [1, 11] are proposed to create an internal high-resolution and low-resolution image pairs database through an image itself, and high-resolution image details can be recovered by referring to this internal database. Since this internal database contains more relevant image content, the recovered high-resolution image can preserve the sharpness of the edge when there exists multiple image recurrence across different scales. However, it will fail to produce the high-resolution image details for image patches when there is not enough recurrence exists for such an image patch or when the input low-resolution image has poor resolution quality.

Given the limitation of each type of image super-resolution algorithm, one super-resolution algorithm cannot always produce the best high-resolution image for any low-resolution input image. In this paper we propose to take advantage of both super-resolution algorithms (example-based and self-similarity based), and develop a scheme to combine the two approaches to obtain high-resolution details. The main idea of our proposed method is to divide an image into multiple regions based on image content information and utilize appropriate super-resolution techniques to generate the high-resolution image for each image region. To our knowledge, we are the first to take image contextual information into account in image super-resolution techniques. In order to determine the proper super-resolution technique for different image content, we compared and evaluated the performance of both example-based and self-similarity based super-resolution techniques on a variety of image contents. Our proposed method focuses on learning the performance of super-resolution algorithms on different image contextual information and applies the appropriate super-resolution technique to recover the high-resolution details.

1.5 Contribution and Organization

Current methods for image super-resolution algorithms are dealing with the whole image. To our knowledge, no existing method considers employing contextual information to a super-resolution algorithm. We are the first to take image content into account in image super-resolution techniques. We proposed a new concept that segments the image into multiple regions based on spatial context and chooses the appropriate SR algorithm for each region based on image content.

We quantitatively compared the existing example-based and self-similarity based SR algorithms on a wide variety of real world image content, which included faces, animals, architecture, and nature scenes. The purpose of doing this is to evaluate the performance of existing example-based and self-similarity based SR algorithms on different types of image content. Our proposed SR method will refer to the evaluation results in order to choose the appropriate SR algorithm for each image region. Comparing two of the state-of-the-art SR algorithms, our proposed SR algorithm can produce better results in terms of visual quality and PSNR (+0.1dB).

Chapter 1 introduces the basic SR algorithms, current challenges and limitations. Chapter 2 reviews the state-of-the-art single image SR algorithms. Chapter 3 describes our proposed SR approach in detail. Quantitative evaluation and supplementary experiments are performed in chapter 4. Finally, the thesis carries out an overall conclusion in chapter 5.

CHAPTER 2

STATE-OF-THE-ART SINGLE IMAGE SR APPROACHES REVIEW

Compared to singe image SR methods, multiple image SR methods have the following limitations that prevent their application to real world problems: 1) it is hard to get several images from the same scene in practice, and 2) multiple image SR methods involve image registration and blur estimation. The registration errors and estimation errors will spread more widely in the HR image reconstruction phase. In real world applications, recovering a high-resolution image through single image SR methods can successfully avoid the limitations mentioned above. Thus, researchers have become more interested in reconstructing a HR image from a single LR image in recent years.

Single image super-resolution algorithms can be grouped into two categories in general: 1) example-based SR algorithms that rely on an external training dataset to recover a high-resolution image, and 2) self-similarity based SR algorithms that recover the high-resolution detail through an image itself. In this section, we will introduce several state-of-the-art single image SR algorithms.

2.1 Example-Based SR Algorithms Review

The basic idea of example-based super-resolution methods contains two steps: first model the relationship between the high-resolution and low-resolution image pairs through the training dataset. Then apply this model to the new LR image in order to generate its HR image. The performance of the learning-based methods largely depend on the similarity between the training data and the testing data.

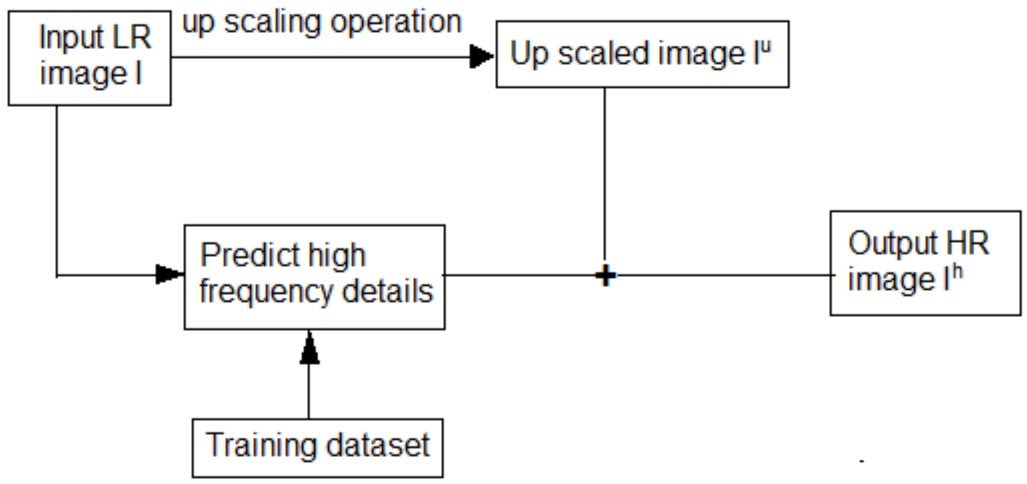


Figure 3: framework of example-based super-resolution approach

Example-based super-resolution was first proposed by Freeman [12], who suggested recovering the missing high-resolution details from a training dataset. The training dataset is composed of high-resolution images and their corresponding all possible low-resolution images. Figure 3 shows the general overview of the example-based super-resolution approaches. The up scaled image using the interpolation operation lacks high frequency details; those missing high-resolution details will be estimated through the training data. Freeman [12] proposed to build a training dataset containing a set of low-resolution and high-resolution image pairs, through two steps: 1) generate the corresponding LR images by applying blurring and subsample operations to the HR images; 2) break the images into small patches, and store the high-resolution image patches and their corresponding low-resolution image patches in the database. The

Markov network model is used to describe the relationship between LR image patches and HR image patches.

The input low-resolution image is divided into small patches, for each image patch, searching for its most similar image patch from the training dataset, the corresponding high-resolution image patch will be used to predict the missing high frequency details of the input low-resolution image patch. There are two main drawbacks for this method. 1) expensive computation costs. Since the size of training database is very large, searching for the nearest neighbor for each image patch is time consuming. 2) this method may generate annoying artifacts due to noise.

In order to reduce searching time, Sun et al. [94] improved the approach by employing a sketch prior into the high-resolution estimation. His idea is motivated by the observation that; the human eye is more sensitive to high frequency details. In order to accelerate computation, he proposed classifying the image patches into two groups: 1) high frequency image patches, which contain edges, and 2) low frequency image patches, which contain no sharp edges, and are low contrast. Instead of estimating the high-resolution details for the entire image patch, Sun suggested enhancing the sharp edge only. As a result, for image patches containing high frequency details, search the training dataset in order to estimate the missing high-resolution details. While for image patches containing only low frequency information, a bicubic interpolation method is used to estimate the missing image information. This strategy reduced the searching time by a certain degree.

Instead of using image information directly, Yang et al. [2] extract image features using a sparse coding technique. Their approach is motivated by: 1) research results from Compressed sensing [3], which conclude that the linear representation for high-resolution signals can be successfully reconstructed from their low-dimensional projections [2, 3], and 2) sparse representation has been widely used in image processing, such as image denoising, object recognition, and restoration [3, 26, 27]. Yang generated two over complete dictionaries D_h and D_l by training the LR-HR image patch pair's database. Each LR image patch can represented by a linear combination of vectors to form the over complete dictionary [2]. The sparse representation vector will be used to generate the HR image patch.

Compared to other example-based SR algorithms, the sparse-coding based SR algorithm has two advantage: 1) the sparse representation is robust to noise, while most other methods cannot perform denoising and super-resolution simultaneously, and 2) As we know, most elements in sparse coefficient are zeros; this will simplify the computation process and accelerate the processing time. The Sparse representation technique is adopted in our proposed algorithm.

2.2 Self-Similarity Based SR Algorithms Review

Research shows image patch texture tends to repeat within or across different scales of the same image at different locations [48,69]. Recurrence or redundancy from different scales of image makes it possible to build an internal image database though the input image itself. As a result, without the assistance of an external training dataset, the high-resolution image details can be recovered through these self-samplers.

The idea of recovering high-resolution image details by using self-similarity examples from the image itself was first proposed by Glasner et al. [11], it does not need any prior sample data. Given an input LR image I_0, first construct a set of image pyramid {I_{-1}, I_{-2},…, I_{-n} }, where I_{-i} is a blurred and downscaled version of input image I_0. Image I_{-i} is generated from image I_0 by:

$$I_{-i} = (I * B_i) \downarrow_{si} \qquad (2.1).$$

where,

B_i is Gaussian blurring operator.

$\downarrow_{si}$ is down-sampling operator with downscale factor setting to i.

Then divide the input image I_0 into small patches of size n*n. Since image patch recurrence exists across different scales, for each image patch, a similar image patch can be found from the down-sampled image I_{-i} (i = 1, 2 …, n). The corresponding HR version of the image patch from image I_0 will be used to recover the high-resolution details. As shown in figure 4, for the image patch in light blue from the input image I_0, a similar image patch (marked in the dark blue box) was found from the image pyramid I_{-2} by using nearest neighbor searching. The image patch in I_{-2} is generated by the image patch (marked by the dark blue box) from image I_0 by using a blurring operator and downscaling operator. The HR image patch marked in the dark blue box from image I_0

contains the missing details, which can be used to recover the HR details for the image patch marked in light blue. The same is true for the image patch in the light purple box; the nearest neighbor (marked in the dark purple box) is found from the down-sampled image I_{-n}. the corresponding HR version image patch from image I_0 will be used to recover the high frequency details of the image patch in light purple.

The self-similarity based image SR algorithm is very similar to the traditional example-based image SR algorithm. The only difference is instead of searching through external training data for a similar low-resolution/high-resolution image patch pair, the self-similarity based SR algorithm searches across different scales of the image itself at different locations. The relationship between the high-resolution image patch and low-resolution image patch are learned from the image itself. This method can preserve edge sharpness when several recurrences exist, since the internal database generated through itself contains more relevant image information. However, it does not always guarantee redundancy for each of the image patches; as a result, the quality of the high-resolution image is not guaranteed.

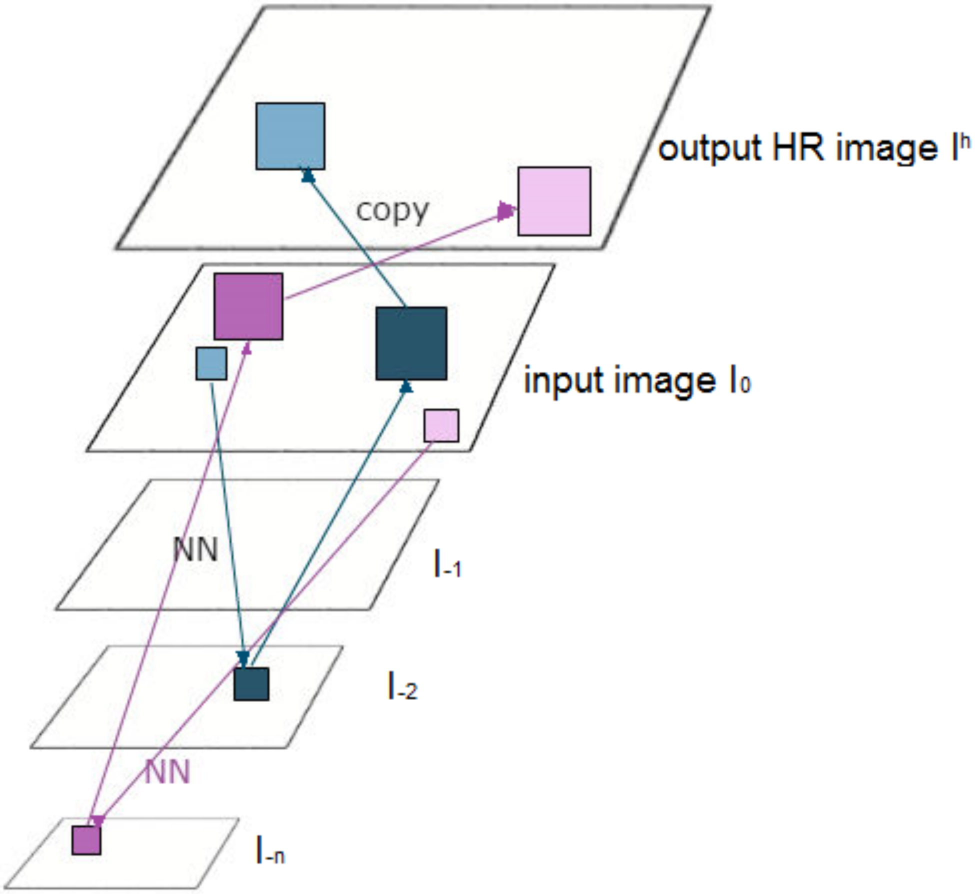


Figure 4: Architecture of self-similarity based SR algorithm [11]

Later, Freedman and Fattal [10] extended their example-based image SR algorithm with self-similarity. Their research shows that image patch recurrence tends to happen within a local neighborhood, and it can produce good high-resolution results if the up scaling factor is small. Based on those observations, they accelerated the computation cost by only searching the local spatial neighborhood across different scales for the best-matched image patch. The similarity between image patches is calculated by traditional L2 distance.

However, some limitations exist for the self-similarity based SR algorithm. One limitation of self-similarity based SR algorithm is that the accuracy of finding the best-matched image patch is relevant to or depends on the size of image patch. When the size is small, little image information is contained in the image patch, which will lead to a lower probability of finding the best matched image patch accurately. However, if the size of the image patch is too big, it will result in heavy computation cost, so we have to leverage the computation cost and accuracy and choose the proper image patch size. Another limitation for the self-similarity base SR algorithm is that, the number of self-samplers learned from different scales of image itself is small. Thus, for image patches that contain complex texture such as hair, fur, tree leaves, the algorithm is less likely to recover fine details and image patches with those types of textures tend to generate over-smoothed HR results [36].

In order to overcome the limitations mentioned above, Singh et al. [36] proposed a sub-band self-similarity SR algorithm. Figure 5 shows the framework of this method.

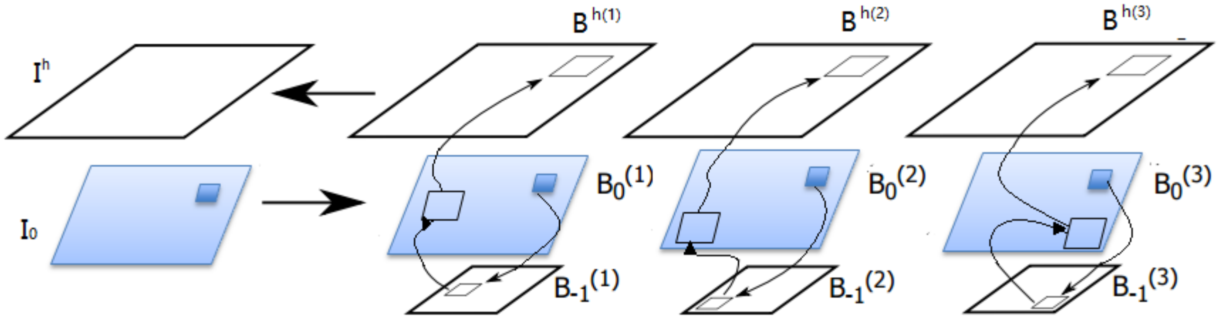


Figure 5: Framework of Sub-Band self-similarity SR algorithm [36]. I_0 is the given input image. Divide image I_0 into small sub-bands B_0^i, i=1... K. For each patch of the given sub-band, find the best-matched patch in the down-sampled image sub-band. The corresponding patch in B_0^i will be used as a HR prior. This patch is used to recover the

missing HR details. For the patch shown in blue, this algorithm allows for its various sub-bands to find the match image patches in different spatial locations independently.

Given an input LR image I_0, Singh [36] decomposed image I_0 into K small sub-bands $\{B_0^j\}_{j=1}^{K}$ and recovered the high-resolution details of each sub-band separately through the following steps:

1. For each sub-band B_0^j, generates its down-sampled version of the sub-band by

$$B_{-1}^{j} = \{B_0^j * f_{psf}\}\downarrow \quad (2.2).$$

   where

   f_{psf} is a point spread function

   $\downarrow$ is down-sampling operator.

2. Divide sub-band B_0^j into small image patches, and create an internal LR-HR image patches database for all the image patches.

3. For each image patch P in sub-band image B_0^j, find its first k nearest neighbors from the internal database generated using image patches from the same sub-band. The corresponding k HR image patches will be used to recover the HR details of image patch P. Repeat this step for all the image patches in order to generate the HR version of the sub-band image B_0^j.

4. Combine all those HR sub-bands to reconstruct the HR image I_0^h of the given input image I_0.

5. Perform a back-projection operation in order to make the HR image consistent with the original input image. The back-projection function is defined as:

$$J(I^h) \ = \ \left\|(I^h * f_{psf}) \downarrow - I_0\right\|_2^2 \qquad (2.3).$$

   The optimal solution of equation 2.3 can be calculated by using gradient decent iteration with I_0^h from step 4 as the initial value.

Given that the number of recurrence for each image patch is limited, it cannot be guaranteed that there exist sufficient numbers of similar image patches from the image itself. As a result, the performance of the self-similarity based SR algorithm is not stable. In order to overcome this problem, we expand the internal database by using image patches from different scales of image itself. Huang [1] applied geometric transformation to each image patch when searching for the best-matched image patch. Given an input LR image *I*, the main steps to recover its high-resolution image using Huang's approach are as follows:

1. Blur and down sample the input image to get its low resolution images I_{-1}, I_{-2}, …, I_{-k} at different down scales.

2. Divide the input image *I* into small patches P_1, P_2,…P_n. For each patch *P* in image *I*, find the best-matched patch Q in the down-sampled image I_{-l}, where *Q* is the nearest neighbor of *P* after applying geometric transformation matrix *T*.

3. Extract the corresponding HR version of Q from the original input image I, which is Q_H. Use the inverse of transformation matrix T to obtain the HR version of P from Q_H.

$$P_H = T^{-1} Q_H \qquad (2.4).$$

4. Repeat steps 2 and 3 for all input image patches to generate the HR image I_H.

5. In order to make the HR image consistent with the original input image, run back-projection iteratively in order to deliver optimal results.

So far, we have introduced several state-of-the-art single image super-resolution algorithms. Because of the limitations we discussed above, it is impossible to always generate good HR image results for all types of input image using a particular image super-resolution algorithm. The performance of example-based SR algorithms always relies on the similarity between the testing image texture and the training dataset. It is impossible to cover all image texture cases in the training dataset, because the huge set of the training dataset will result in super heavy computation when searching for the nearest neighbor, and the performance of the method for recovering high-resolution image details from self-exemplars is also not reliable. Such methods can deliver poor high-resolution results if there is not a sufficient number of image patches that reappear at different locations in the downscaled image. What’s more, those self-similarity based SR methods will not be able to recover the high-resolution details for image patches with complex structure.

Given the limitation of each type of single image SR algorithm, we propose to take the advantage of each type of SR algorithm, and combine those two types of SR algorithm. Our goal is to better recover the HR details by using both the internal database and external training dataset.

CHAPTER 3

PROPOSED SINGLE IMAGE SUPER-RESOLUTION METHODOLOGY

It is impossible to improve image resolution by overcoming the Nyquist limit through a single image itself. The high-resolution details must be provided from elsewhere. Example-based SR algorithms learn the relationship between high-resolution images and low-resolution images from training dataset. Freeman et al. [12] used Markov Random Field (MRF) to model the relationship between LR and HR image patch pairs, and used Belief Propagation to generate the missing high-resolution details. Tappen et al. [39, 40] suggested to learn the high-resolution image prior through natural image statistics, and improve high-resolution quality on the image edge. Self-similarity based SR algorithms are based on the research result that image texture tends to recur within or across different scales of the same image itself, so instead of learning the relationship between HR and LR image pairs through an external database, self-similarity based algorithms build an internal database from an image itself.

3.1 Overview of Proposed SR Algorithm

In this section, we will first describe the overview of our proposed scheme. Then we will describe details of the self-similarity SR algorithm and example–based SR algorithm that are adopted in our proposed method.

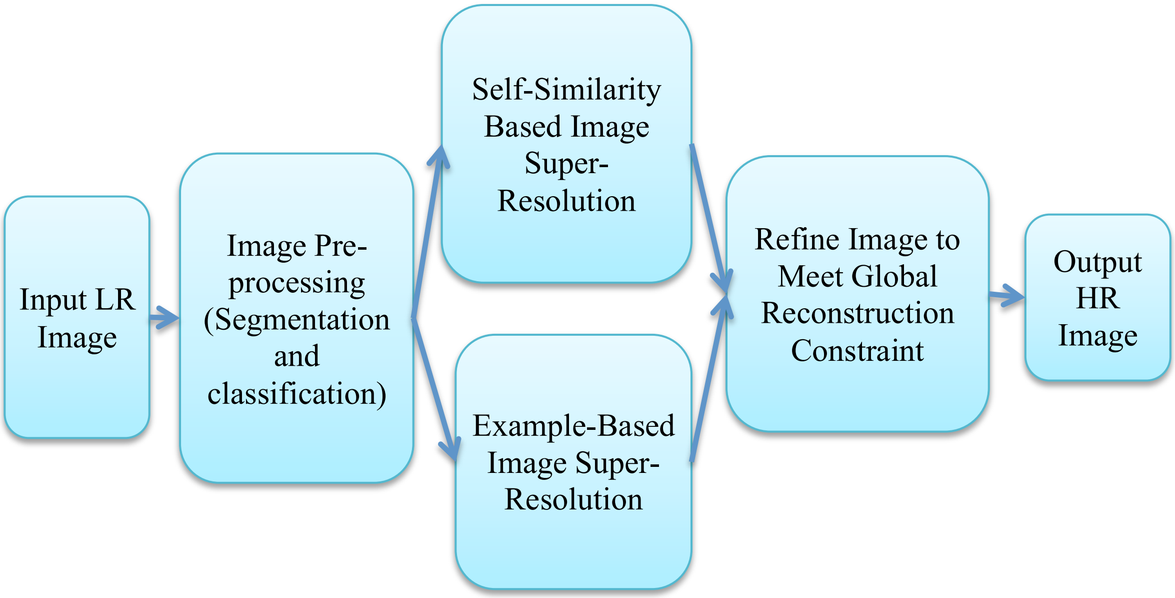

Input LR Image
Image Pre-processing (Segmentation and classification)
Self-Similarity Based Image Super-Resolution
Example-Based Image Super-Resolution
Refine Image to Meet Global Reconstruction Constraint
Output HR Image

Figure 6: Framework of Proposed Single Image Super-Resolution

The framework of our proposed method is shown in figure 6. The main steps are as follows:

1. Segment input LR image *I* into multiple regions based on image content. For example, divide image *I* into two regions R_1 and R_2.

2. Apply self-similarity algorithm [1] to generate the HR image I_{h1} for image region R_1, and apply example-based algorithm [2] to reconstruct HR image I_{h2} for region R_2.

3. Combine I_{h1} and I_{h2} to get the estimated HR image $I_{h(0)}$.

4. In order to satisfy the global reconstruction constraint, we need to refine $I_{h(0)}$ iteratively using back projection as follows [2, 36, 37]:

$$I_{h\,(t+1)} = I_{h(t)} + \lambda_1[H^T S^T\big(I - SHI_{h(t)}\big) + \lambda_2(I_{h(t)} - I_{h(0)})] \qquad (3.1).$$

where

$I_{h(t)}$ is the HR image after the t-th iteration.

S is the down-sampling operator.

H is the blurring filter, such that $I = SHI_h$.

In the image pre-processing step, we segment the image into multiple regions, and classify those regions based on the image content in that region. In order to choose the appropriate SR approach for each image region, we quantitatively compare the performance of two state-of-the-art super-resolution algorithms (example-based and self-similarity based SR algorithms) on a variety of real world image textures, which include faces, animals, architecture, and nature scenes. We tested 100 natural scene images, taken from the Berkeley segmentation dataset and 100 urban images that contain urban and architectural scenes. The purpose of this evaluation is to identify which types of image texture's high-resolution details can be successfully recovered using the self-similarity based approach and for which types of image texture, a high-resolution image can be better recovered by utilizing example-based super-resolution algorithm.

Our evaluation shows the self-similarity based SR algorithm always outperforms the example-based SR algorithm on urban architectures, animals, and insects while the example-based SR algorithm generates more natural looking faces, trees, flowers, and grasses. For image patches containing tree leaves and grasses, it is hard to accurately find the best-matched patch from an image itself within or across different scales. It is because the structures of grass and tree leaves are irregular; it is hard to estimate a proper geometric transformation. As a result, we are not guaranteed to find the right nearest neighbor that can provide high-resolution details. In the high-resolution image reconstruction step, our proposed SR method refers to the evaluation result in order to choose the appropriate SR algorithm for each image region.

The purpose of step 4 is to force the reconstructed HR image to be consistent with the input LR image [2]. The final HR results can be optimized by [2, 36]:

$$I_h^* = \arg min\, \lambda_{1_t} \left\| I_{h(t)} - I_{h(0)} \right\|_2^2 + \lambda_2 \left\| I - SHI_{h(t)} \right\|_2^2 \qquad (3.2).$$

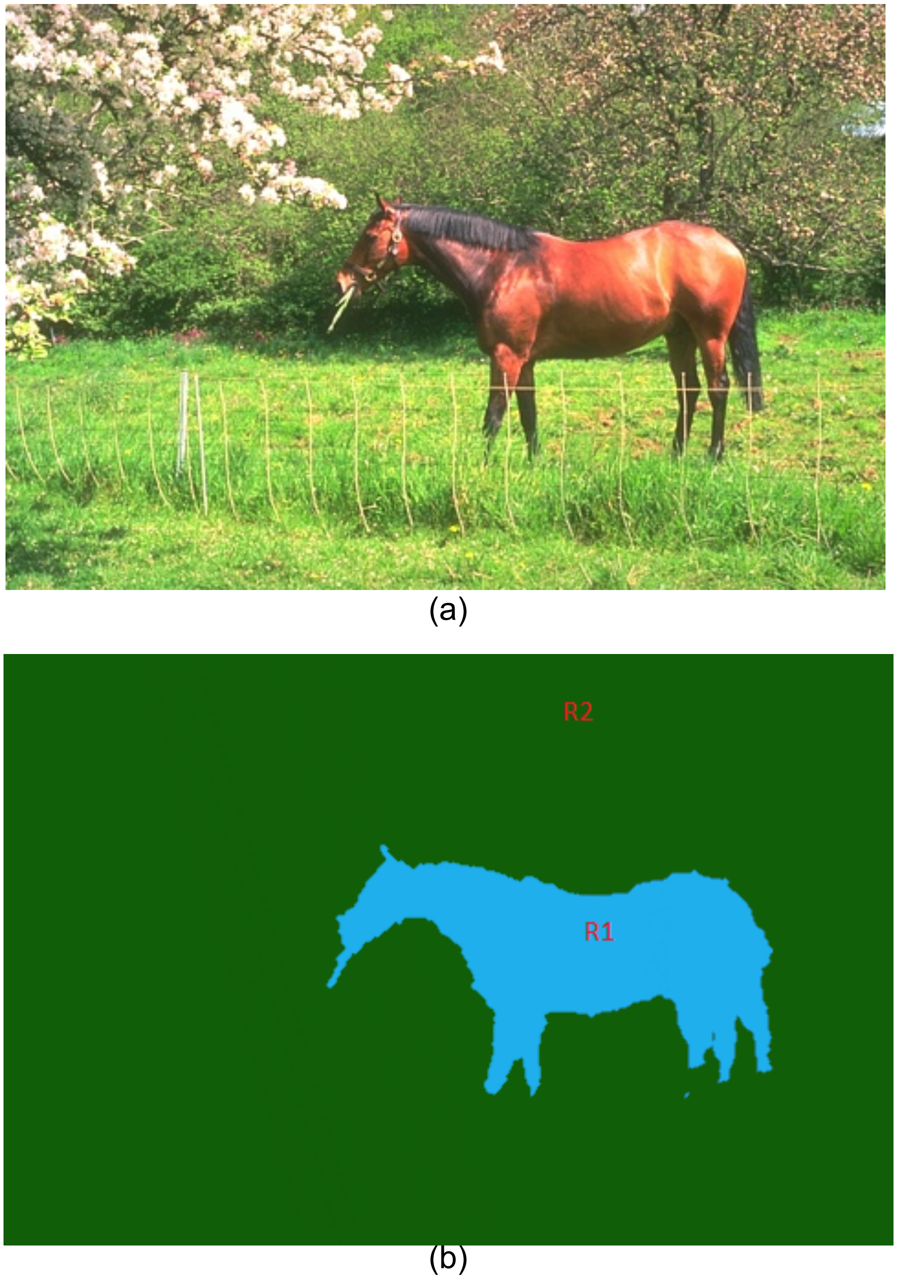
The optimal solution of equation 3.2 can be calculated by using the gradient decent algorithm [113]. In order to find the local minimum solution, we can update $I_{h(t)}$ iteratively through equation 3.3:

$$I_{h\,(t+1)} = I_{h(t)} + \lambda_1 [H^T S^T \left(I - SHI_{h(t)} \right) + \lambda_2 (I_{h(t)} - I_{h(0)})] \qquad (3.3).$$

where,

$I_{h(t)}$ is the HR results after the t-th update.

λ_1 and λ_2 are two parameters that control how fast it converges. With proper selection of these two parameters, the local optimal solution will also lead to the global optimal solution.



(a)

(b)

Figure 7: Divide Image into Small Regions (a) Original image to be segmented, (b) image segmented into two regions based on textural information.

As shown in figure 7, we divide the input image into two regions. The region containing a horse is region $R_{1,}$ and the other part, which contains trees and grasses, is region $R_{2.}$ We choose different super-resolution methods to recover the high-resolution details for region R_1 and region R_2. We apply a self-similarity based SR approach to generate the high-resolution image for image region R_1, and utilize an example-based SR algorithm to generate the high-resolution image for image region R_2. Since the two regions utilize different SR algorithms to reconstruct their high-resolution image, in order to avoid unnecessary artifacts around the boundary, and be consistent with the original input image. We need to refine the result by iteratively updating the high-resolution image using equation 3.3. The updating will stop when it converges. The final optimal solution will be the target high-resolution result.

3.2 Image Pre-processing

Our research is focused on recovering a high-resolution image; we assume the input image is segmented based on textural information and the properly labeled in the pre-processing phase. Many recent works on region-based image segmentation [107,108, 110] perform remarkably well. In implementation, we used a multi-class image segmentation tool provided by the Darwin software library [109] to segment and label the regions.

There are two main goals in image segmentation:

1. Decomposing an image into multiple regions based on image content information.

2. Classifying and labeling each region.

The Darwin software library is implemented using an algorithm from [108,110]; the main steps of this algorithm are:

1. Generate a segmentation dictionary Ω by running a mean-shift algorithm [111]; this dictionary will be referred to during region assignment or movement.
2. Initialize each pixel in the image to one region in $R_p \in \{1, 2, \ldots, K\}$ by comparing with the labeled training data.
3. Given current region assignment, propose to assign pixels to the new regions $\{R_p : p \in \omega\} \leftarrow r$.
4. Update the corresponding features and region appearance that was affected by the proposal assignment in step 3.
5. Update the a priori location information v^{hz}.
6. Compute total energy E, which will be used to decide whether to accept the region update or not. If $E < E^{min}$, then accept the move and set $E^{min} = E$; otherwise, no change is needed.
7. Repeat steps 3 to 6 until convergence.

The energy function contains five terms used to determine which region the pixel belongs to. It can be defined as:

$$E = \psi^{hz}(v^{hz}) + \sum_{r} \psi_{r}^{reg}(S_r, v^{hz}) + \sum_{r,s} \psi_{rs}^{bdry} + \sum_{o} \psi_{o}^{obj}(C_o, v^{hz}) + \sum_{o,r} \psi_{or}^{ctxt}(C_o, S_r)$$

(3.4).

where

$\psi^{hz}(v^{hz})$ captures the a priori location in the scene. It is implemented using Log-Gaussian.

The term $\sum_{r} \psi_{r}^{reg}(S_r, v^{hz})$ shows the likelihood to assign a region to different image classes.

The term $\sum_{r,s} \psi_{rs}^{bdry}$ measures similarity between regions, and the contrast between boundaries. It encourages regions with similar content to merge into one region.

The term $\sum_{o} \psi_{o}^{obj}(C_o, v^{hz})$ measures the likelihood of a group of regions being assigned to a given object label.

The term $\sum_{o,r} \psi_{or}^{ctxt}(C_o, S_r)$ measures the relationship between objects and their background.

In the image segmentation stage, the image would be segmented into different regions based on textural information: tree, road, face, flower, building, mountain, human, grass, animal. The image regions can be grouped into one of 21 classes compared with the 21-class MSRC dataset [112].

In our experiment, we will first evaluate the performance of the self-similarity based SR algorithm and example-based SR algorithm on different types of image regions. We observe which types of image region can better generate HR details. For the other image regions, an external database will be used to recover the corresponding HR details. In the following sections, we will detail the self-similarity based SR algorithm [1] and example-based SR algorithm [2].

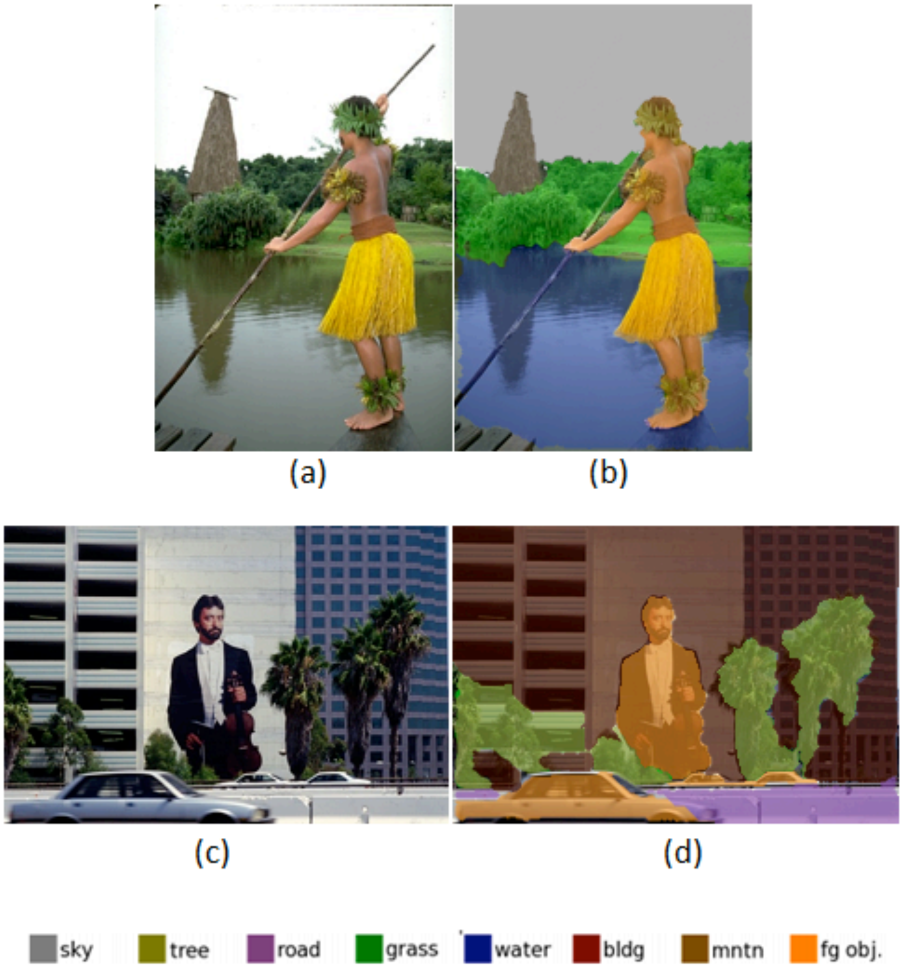


Figure 8: Example Results of Image Segmentation on The 21-Class MSRC Dataset [113]. Images in (a) and (c) are original images. (b) and (d) are results of segmented images into regions, and classified region into different group based on image content.

As shown in figure 8, the sample images are properly segmented into multiple regions based on image content; each region has been correctly labeled based on content. Input image 4(a) has been segmented into four regions: human, mountain, tree, and water. Image 4(b) has been segmented into five different regions: human, tree, road, building, and vehicle.

3.3 Self-Similarity Based Image Super-Resolution

Research shows image patch texture tends to repeat within or across different scales of the same image at different locations [11]. This self-similarity within an image makes it possible to build an internal LR-HR image patch pairs database through input image itself and reconstruct high-resolution image details without using an external database. As shown in figure 9, image recurrence from different scales makes it possible to recover high-resolution details.

In this section, we will briefly describe the self-similarity based algorithm developed by [1] that is used in our proposed method to generate the HR image. The main steps are as follows:

1. Given an input LR image *I*, blur and down-sample the input image to get several low resolution images $I_{-1}, I_{-2}, \ldots, I_{-k}$ at different scales.

2. Divide input image *I* into small patches $P_1, P_2, \ldots P_n$.

3. For each patch P in image I, use the PatchMatch algorithm [5] to find the best matched patch Q in down-sampled image I_{-l}, where Q is the nearest neighbor of P after applying geometric transformation matrix T.

4. Extract the corresponding HR version of Q from the original input image I, which is Q_H.

5. Use the inverse of transformation matrix T to obtain the HR version of P from Q_H.

$$P_H = T^{-1} Q_H \tag{3.5}$$

6. Repeat steps 3 to 5 for all input image patches; we can generate the HR image I_H.

7. At the end, run back-projection iteratively in order to satisfy the global reconstruction constraint.

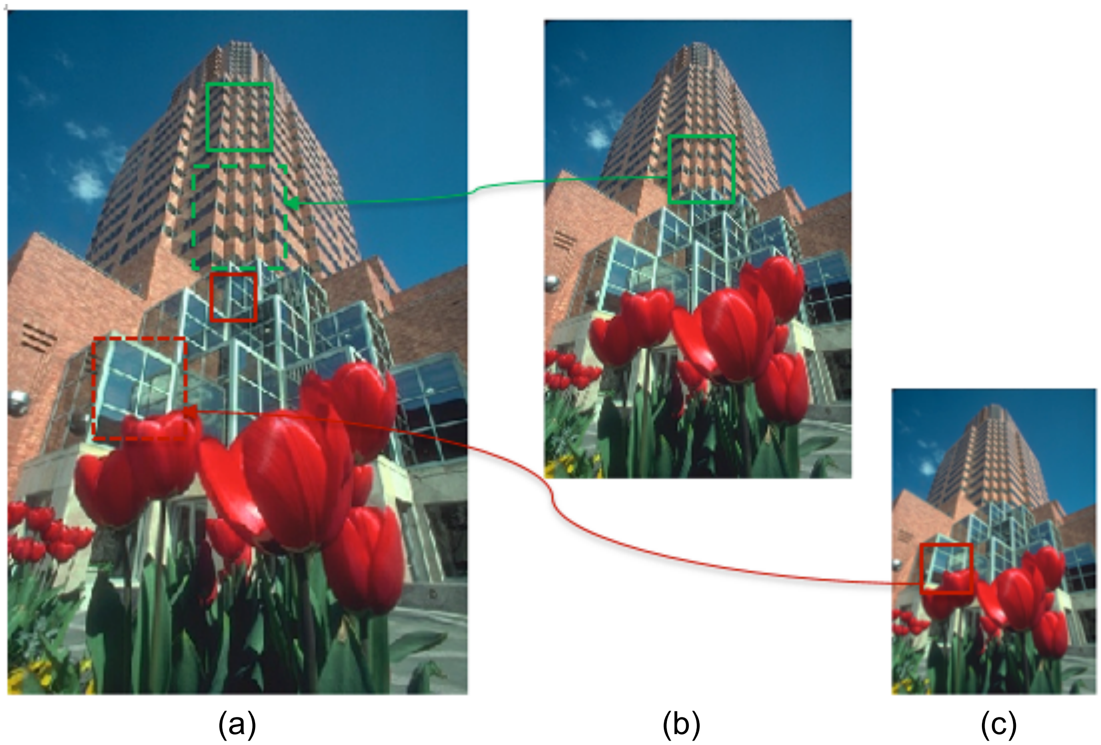


Figure 9: Image Patch Recurrence Across Different Scales of The Same Image. Images in (b) and (c) are smaller and downscaled version of the original image in (a). Image patch marked in solid red box in (a) reappears in image (c) at different location (marked in solid red box). The corresponding parent image patch (marked in dashed red box) from input image (a) could be used to generate the HR details. The same for image patch marked in green box.

When searching for the best matched patches, affine transformation, shearing or rotation operation is applied to the image patch. The use of the transformed operation results in lower matching errors, and can predict the HR details more accurately. Since the PatchMatch algorithm [5] is well known for finding the correspondence between image patches in a short time, the method is used to quickly estimate the transformation function between image patches.

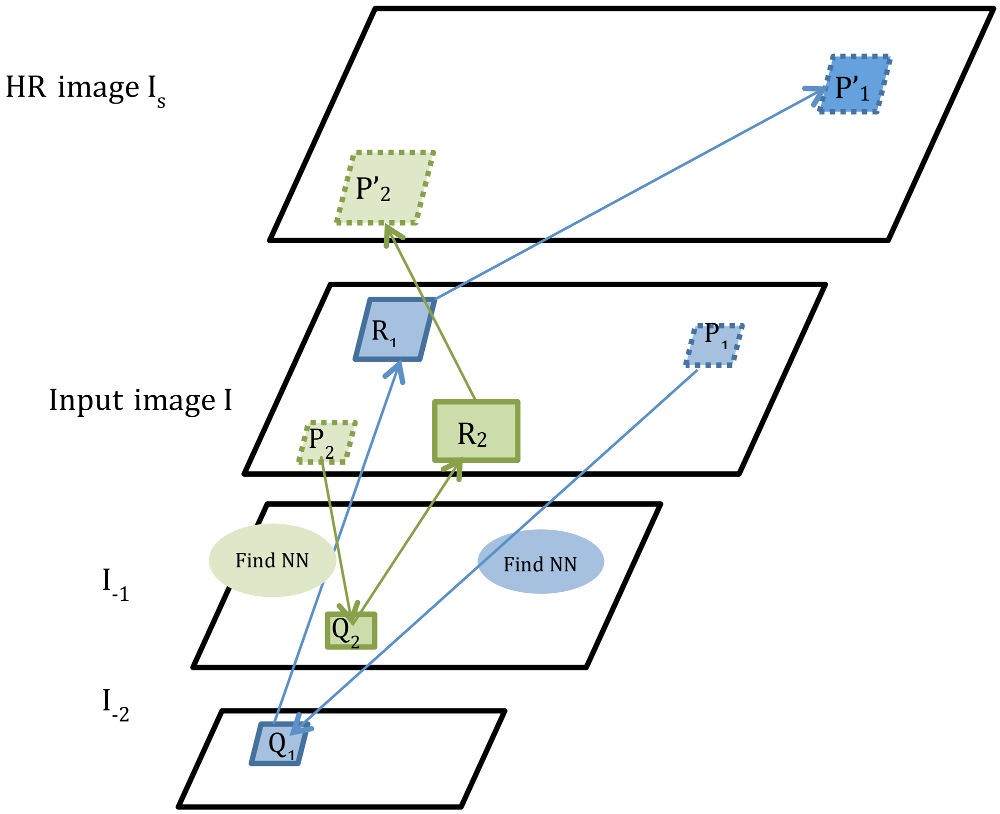


Figure 10: Self-Similarity Based Single Image Super-Resolution

As shown in figure 10, given a LR input image *I*, blur it using Gaussian low pass filter and down-sample it with different scale factors to generate the low frequency images I_{-1}, …, I_{-n}. Image *I* are divided into small patches P_i of size *a*a*. For image patch P_1 in I, the best match patch (Q_1) is found in lower-resolution images (I_{-2}) by using the algorithm described in the following section. For the found image patch (Q_1), a corresponding region (R_1) in image *I* will be used to generate the corresponding HR version of patch P_1. Similarly, for image patch P_2, R_2 will be used to recover the HR image patch of P_2. A proper size of the image patch plays an important role in recovering the HR image details. Since the image information in the small image patch is limited; as a result, it will affect the performance of finding the best-matched image patch. On the other hand, if we choose a large size for the image patch, the computation cost will be very expensive. Thus, in implementation, we need to tune the size in order to balance the computation cost and match accuracy.

The core of the self-similarity based image super-resolution algorithm is to find the best-matched patch from the down-sampled image patches accurately. Since the number of similar image patches within an image itself is limited. Simply apply transition cannot guarantee to find the nearest neighbor accurately. In order to expand the number of repeating image patches, geometric transformation is used when searching for the best-matched patch. The corresponding image patch from the input image will be used to recover the HR details. In this section, we will describe the detail of best-matched patch algorithm.

The self-similarity based super-resolution algorithm built an internal LR-HR image patch pairs within and across scales of the input LR image itself. The size of the internal database is much smaller than the external database used in the example-based super-resolution algorithm; however, this internal database contains more similar image patch pairs. The self-similarity based super-resolution algorithm is based on the observation that image patches repeat inside an image at different locations, both within the same scale and across different image scales [1]. For each image patch, in order to recover the fine details accurately, enough recurrence must exist across different scales. Transformation operation, such as rotation, shearing mapping, scaling expands the chance of recurrence.

Given an input low resolution image *I*, we divide it into small image patches P_i of size *a*a*, *i*=1, 2… *m*. For each image patch $\mathrm{P_i}$ centered at position (x_i, y_i), find the best match patch *Q* from down-sampled images. Since the internal image patch pairs database is small, it may not generate a satisfactory result by simply searching for the most similar patch from sub-image patches. It is possible the image patch *Q'* from the down-sampled image becomes the best match patch after applying the transformation operation, such as rotation, scaling operation, or shearing mapping. In such a case, instead of simply comparing P_i with every patch *Q* from down-sampled images, and computing the similarity between P_i and *Q*, we apply a translation operation *T* to *Q* and compare the similarity between *P* and *TQ*. The best matched patch can be found by solving:

$$\min_{\{\theta_i\}} \sum_{i=1}^{m} E_{app}(i, \theta_i) + E_{plane}(i, \theta_i) + E_{scale}(i, \theta_i) \qquad (3.6).$$

where,

θ_i is a set of parameters used to build the transformation matrix $T_{i.}$

E_{app} measures the similarity between P_i and the best match patch *Q* from down sampled images after applying transformation matrix T_i.

E_{plane} is plane compatibility cost, which encourages us to search for multiple planes in order to find the best match patch.

E_{scale} is scale cost, which encourages to search the best match patch from the down-sampled image with a larger scaling factor.

The similarity cost E_{app} is measured by

$$E_{app}(i, \theta_i) = \|w_i(P_i - Q(i, \theta_i))\|_2^2 \qquad (3.7).$$

where,

w_i is the Gaussian weights.

$Q(i, \theta_i)$ is a transformation operation to the image patch from down-sampled image I_{-i}. For example, Q_i is an image patch from the down-sampled image after applying $Q(i, \theta_i)$ operation; image patch Q_i becomes T_i Q_i. Instead of measuring the similarity between P_i and Q_i straightly, the similarity cost E_{app} measures the similarity between P_i and T_i Q_i.

The transformation matrix T_i is defined as [1]:

$$T_i\,(\theta_i) = H(i, s_i^x, s_i^y, m_i)\; S(s_i^s, s_i^\theta)\, A(s_i^\alpha, s_i^\beta) \qquad (3.8).$$

where

$S(.)$ is the scaling and rotation transformation

$$S(s_i^s,\, s_i^\theta) = \begin{bmatrix} s_i^s R(s_i^\theta) & 0 \\ 0 & 1 \end{bmatrix} \qquad (3.9).$$

Here, s_i^s is a scaling factor, and $R(s_i^\theta)$ is a rotation operation.

$A(.)$ is the shearing mapping transformation

$$A(s_i^{\alpha},\, s_i^{\beta}) = \begin{bmatrix} 1 & s_i^{\alpha} & 0 \\ s_i^{\beta} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.10).$$

H(.) is the perspective deformation given the position of P_i from input image and the matching patch Q_i from down-sampled images.

As shown in figure 11, for image patch Q_1 marked in the red box in (a), the best-matched patch P_1 (marked in the green box) is from the down-sampled image (b). Image patch P_1 is similar to Q_1 after applying the scaling transformation. For image patch P_2 (marked in the blue box), without any transformation operation, the difference between image patches P_2 from down-sampled image (b) to Q_1 is very big. But after applying the shearing mapping and rotation operation, the translated version of P_2 becomes the nearest neighbor of $Q_{1.}$

As figure 11 illustrates, transformation matrix *T* expands the internal LR-HR image patches database space and makes it possible to find the best matched image patch accurately.

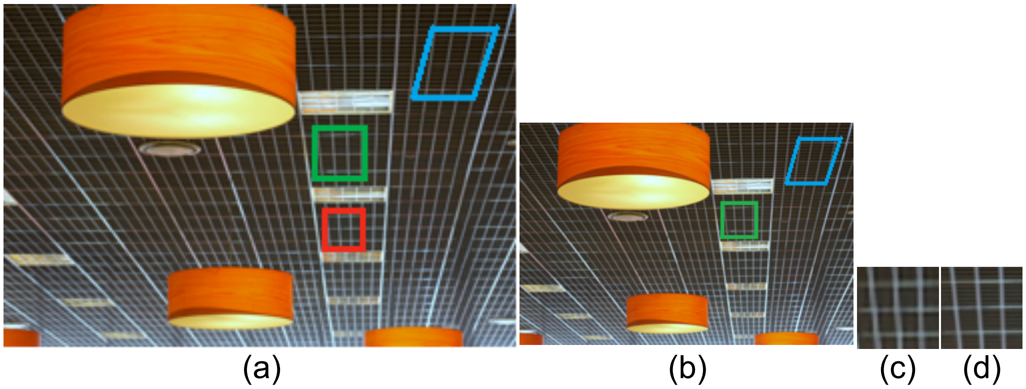


Figure 11: Example Demonstrating Transformation Operation. (a). original input image, (b) down-sampled image, (c) 2x HR image patch recovered by using green box, (d) 2x HR image patch recovered by using blue box

The example in figure 11 demonstrates the reason why we need to apply the transformation operation in best-matched patch search. For the image patch P in the red box from the LR input image, in order to construct its 2x up-scaling HR image patch. Without using the transformation operation, we compare image patch P with all image patches from the down-samples image. By comparing the similarity between those image patches, we found the image patch in the green box is the nearest neighbor of image patch *P*. In such case, the corresponding image patch of the green box from input image *I* is used to recover the HR detail of image patch *P*. The image in figure 7(c) is the 2x HR image of image patch *P* recovered by using the image patch in the green box. However, if we apply a different affine transformation operation to the image patches from down-sampled while searching for the nearest neighbor, we found the image patches in blue box are similar to image patch after applying affine transformation. In such case, the corresponding image patch of the blue box from input image will be used to generate the HR image patch of P. As we can see, the corresponding image patch of the blue box contains more detail than the image patch from the green box. As a result, the 2x HR image patch generated by using the image patch in the blue box recovers more HR details. The image in figure 7(d) is the 2x HR image of image patch P recovered by using the image patch in the blue box. This example shows why using the transformed version of the image patch from the down-sampled image when comparing similarities between image patches is useful in estimating the HR details.

The compatibility cost E_{plane} is used to encourage search over different plane label:

$$E_{plane} = -\lambda_{plane} log(Pr[m_i|(s_i^x, s_i^y)] x Pr[m_i|(t_i^x, t_i^y)]) \quad (3.11).$$

where,

$Pr[m_i|(x, y)]$ is the posterior probability that pixel (x,y) belongs to plane m_i.

The purpose of scale cost E_{scale} is to avoid matching a patch to itself in the down-sampled image.

$$E_{scale} = \lambda_{scale}\, max(0, SRF - Scale(T_i)) \quad (3.12).$$

where

SRF is the desired SR factor.

Scale(.) is the scale estimation of the source patch using T_i

$$scale(T_i) \; = \; \sqrt{det(\begin{bmatrix} T_{1,1} - T_{1,3}T_{3,1} & T_{1,2} - T_{1,3}T_{3,2} \\ T_{2,1} - T_{2,3}T_{3,1} & T_{3,1} - T_{2,3}T_{3,2} \end{bmatrix})} \qquad (3.13).$$

In implementation, seven types of geometric transformations were estimated over all target image patches in order to find the best-matched patch. The geometric transformations include translations, scaling, shearing, rotation, homograph, non-linear and perspective deformation. There are three main steps [1]:

1. Initialization

   Research results from [106, 107] conclude that a good matched image texture can always be found in the local neighborhood. Based on this statement, the nearest neighbor field form scale equal to the desired SR magnified factor will be used for initialization. This strategy will result in faster convergence.

2. Propagation

   Use shearing, homography, scaling and rotation transformation at different plane indexes to widely spread good match neighbors.

3. Randomization

   Perform randomized search to refine the current solution at the end of each iteration. In the meantime, search for the best transformation for the input image patches, and reduce the matching errors between the input patch and the best-matched image patch.

The process to estimate the geometric transformations between two image patches involves heavy computation. Seven different types of geometric transformation need to be estimated in order to map and measure the difference between the target image patch and the source image patch. The PatchMatch algorithm is employed in order to speed up the mapping process. Figure 12 demonstrates the geometric transformations involved in the matching process.

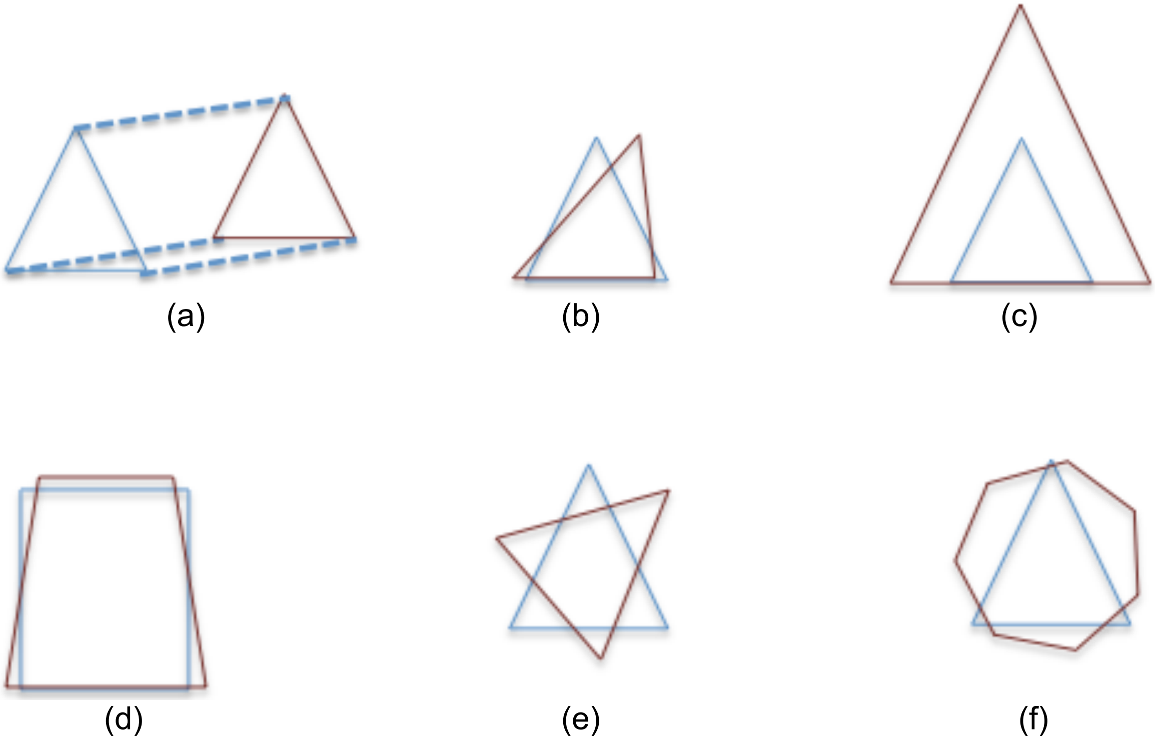


Figure 12: Different Types of Geometric Transformations Applied to Target Image Patch. (a) Translation, (b) Shearing, (c) Scaling, (d) Homography, (e) Rotation, and (f) non-linear transformation

Examples in figure 12 demonstrate the results after applying each geometric transformation. The objects in blue are the original image shape, and the objects in red are the results after applying the corresponding geometric transformation. As it shows, translation is simply a moving operation, the shape and size of the original object will not change, and Shearing, scaling, and homography are linear transformations. Those operations will distort the shape of the original objects, but the number of edges will keep the same after those transformations. Rotation operation does not change the shape and size of the original object as well. As we can see from figure 8(f), the shape, size and edge will change after non-linear transformation. The purpose of employing those geometric transformations to the image patch is to enlarge the searching database.

3.4 Example Based Image Super-Resolution

Let X and Y denote the HR and LR images respectively. The example-based super resolution algorithm reconstructs the HR image by satisfying the following constraint [2]:

$$Y = D_s H X \qquad (3.14).$$

where

D_s is the down-sampling operator.

H is the blurring filter.

Example based super resolution methods need a training database, which contains a set of LR-HR image patch pairs. The core of example based methods is to learn the relationships between LR and HR image patches {x_i, y_i} from the database, then apply the relationship to estimate the HR image details for the target image patch.

The nearest neighbor-based methods [12, 13, 17, 25] use the image patch pairs directly, find the first or k nearest neighbors, and the corresponding HR image patches will be used to recover the HR image details for the LR input image patch. Freeman et al. [12] generated the training dataset using a set of high-resolution images through two steps: 1) generate the corresponding LR images by applying blurring and subsample operation to the HR images; 2) break the images into small patches, and store the high-resolution image patches and their corresponding low-resolution image patches in the database. A Markov network is then used to model the relationship between LR image patches and HR image patches. As shown in figure 13, the circles indicate network nodes; y_i at each node represents the observed low-resolution image patch, and x_i represents the corresponding high-resolution image patch that we are going to estimate. The lines represent statistical dependencies between nodes. The Markov network probability is defined as:

$$P(x|y) = \frac{1}{z} \prod_{(i,j)} \psi_{ij}(x_i, x_j) \prod_{i} (x_i, y_i)$$

$$\text{s.t. } \psi_{ij}(x_i, x_j) = \exp\left(-\frac{d_{ij}(x_i, x_j)}{2\delta^2}\right) \qquad (3.15).$$

where,

z is a constant, which is used for normalization.

$d_{ij}(x_i, x_j)$ measures the difference between image patch x_i and x_j.

The corresponding high-resolution image patches that can maximize the Markov network probability will be used to reconstruct the high-resolution image [12].

In order to produce good results in general, this type of method requires a huge training database which includes as many image textures as possible; this will result in heavy computation cost.

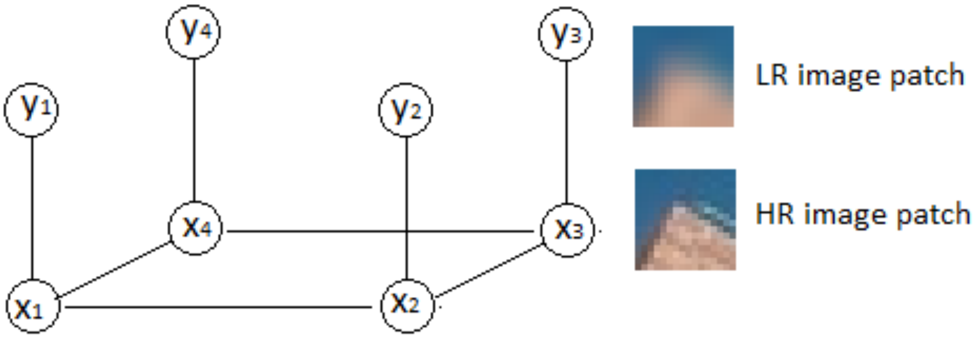


Figure 13: Markov Network Model for The Super-Resolution Problem [12]

Another more effective way is sparse-coding based methods. The LR-HR image patch pairs database is trained to generate an over complete dictionary. Each LR image patch can be represented by a linear combination of vectors from the over complete dictionary [2]. This sparse representation vector will be used to generate the HR image patch. One limitation of those methods is that the performance heavily depends on the similarity between the training image patches in the database and testing image patch.

We will briefly present the sparse coding based SR method proposed by [2] in the following section.

3.4.1 Sparse Coding Based SR Method

Recently sparse representation has drawn a lot of interest in computer vision and image processing. It has been widely used in image processing, such as image denoising, object recognition, and restoration [3, 26, 27]. Sparse coding based SR approach is motivated by the research results from Compressed sensing [3], which conclude that the linear representation for high-resolution signals can be successfully reconstructed from their low-dimensional projections [2, 3].

Let X and Y denote the HR and LR images respectively; $\{x_i\}$ are image patches of HR images, and $\{y_i\}$ are corresponding LR image patches. The basic idea of the sparse coding algorithm [2] is that each image patch y_i can be represented as

$$y_i \doteq Lx_i = LD\alpha \qquad (3.16).$$

where

α is a sparse representation vector with few nonzero entries.

$D = [d_1,\ldots.,d_k] \in \mathbb{R}^{mxk}$ is an over-complete dictionary learned from training image examples; each element d_i in the dictionary is a basis vector. We will introduce the over-complete dictionary training process in the section follows.

Research shows the sparsest solution α can be sparse and unique under certain conditions, and if D satisfies restricted isometry property at a certain level, the linear representation of a HR image patch can be accurately reconstructed from the corresponding low-resolution image patch [2, 28].

For the sparse coding based SR method proposed by [2], there are two constraints it has to satisfy:

1. Reconstruction constraint: the reconstructed high-resolution image X and the corresponding input low-resolution image Y should satisfy:

$$Y = SHX \tag{3.17}$$

where,

S is subsampling operation.

H is blurring operation.

2. Sparsity constraint: each low-resolution and high-resolution image patch pair should have the same sparse representation. Two over-complete dictionaries are trained from the LR-HR image patch database using a tool developed by [4]. D_h is learned from HR image patches $\{x_i\}$, and D_l is trained using LR image patches $\{y_i\}$ simultaneously by enforcing proper constraint. Each image patch y_i from low-resolution image Y can be described by a linear sparse combination of dictionary $D_{l:}$

$$y_i \approx D_l\alpha$$

$$\text{s.t. } \|\alpha\|_0 \ll \mathrm{K}, \alpha\epsilon\mathbb{R}^k \quad (3.18).$$

The sparse representation, which is calculated from low-resolution image patch y_i could be used to recover the corresponding high-resolution image patch x_i by using the high-resolution dictionary D_h:

$$x_i \approx D_h\alpha \quad (3.19).$$

The main steps of the sparse coding based super-resolution algorithm [2] are as follows:

1. Use LR and HR image patch pairs to train over complete dictionary D_h and D_l.
2. Divide the input LR image Y into small patches of size $n*n$ with 1-pixel overlap. For each patch y_i, solve the optimization problem

$$\min_\alpha \left\|\widetilde{D}\alpha - \widetilde{y\imath}\right\|_2^2 + \lambda\|\alpha\|_1 \quad (3.20).$$

3. Once the optimal solution α* from step 2 has been calculated, the HR patch can be obtained by x_i=D_h α*
4. Repeat step 2 and 3 for all the patches to generate the HR image X_0.
5. Optimize X_0 in order to meet the global reconstruction constraint.

$$X^{*} = \arg\min_{X} \|SHX - Y\|_2^2 + c\|X - X_0\|_2^2 \qquad (3.21).$$

X* is the final super-resolution version of input image *Y*.

In dictionary training step, instead of using absolute intensity values, each image patch subtracts its mean pixel value. By doing this, the LR and HR image patches used for dictionary training represent image texture. In step 2, the input LR image patch also needs to subtract its mean pixel value before calculating the optimal sparse representation α. After generating the HR image patch from step 3, the mean pixel value will be added back to reconstruct the HR absolute intensity [2].

D_h and D_l are jointly trained in order to have the same sparse coefficient α for each LR and HR image patch pair. The sparse representation α for LR image patch y_i using [2]

$$y_i = D_l\, \alpha \qquad (3.22).$$

can be used to recover the corresponding HR image patch x_i *by*

$$x_i = D_h\, \alpha \qquad (3.23).$$

For each input LR patch y_i, the parse representation of y_i can be calculated as:

$$\min \|a\|_0$$

$$s.t.\ \ \|FD_l a - Fy\|_2^2 \leq \epsilon \qquad (3.24).$$

where,

F is a function that extracts features from image patch. *F* controls how closely the sparse representation approximate *y*. In practice, *F* can be one of the following filters [2]:

1. High-pass filter, which is used to extract the edge and high frequency content from the LR image patch.
2. Gaussian derivative filter, which is used to extract contour information from the LR image patch.
3. First and second order gradient, which is used to extract the derivative information. *F* is composed of four 1-D filters:

$$\begin{aligned} f_1 &= [-1, 0,1], \\ f_2 &= f_1^T \\ f_3 &= [1, 0, -2, 0, 1], \\ f_4 &= f_3^T \qquad (3.25). \end{aligned}$$

Equation 3.24 is a NP-hard problem; it is very difficult to get the optimal solution. According to research in [29, 30], minimizing the L1 norm can also produce an optimal sparse solution.

In order to simplify the problem, we solve equation 3.26 instead:

$$\min_{\alpha} \|FD_l\alpha - Fy_i\|_2^2 + \lambda\|\alpha\|_1 \qquad (3.26).$$

where,

F is a feature extraction function, which is the same as equation 3.25.

Lagrange multiplier λ is used to balance the sparsity of α and reconstruction error.

In order to enforce the compatibility between adjacent image patches, equation 3.26 is modified as [2]

$$\min_{\alpha}\left\|\widetilde{D}\alpha-\widetilde{y\iota}\right\|_2^2+\lambda\|\alpha\|_1$$

$$\text{s.t. } \widetilde{D}=[\begin{smallmatrix}FD_l\\ \beta PD_h\end{smallmatrix}]$$

$$\tilde{y}=[\begin{smallmatrix}Fy\\ \beta\omega\end{smallmatrix}] \qquad (3.27).$$

β is a tradeoff control parameter, which is set to be 1 in experiment.

P is the overlap region between current image patch x_i and previously reconstructed image patch x_{i-1}.

ω holds the values of x_{i-1} on the overlap region.

3.4.2 Over-Complete Dictionary Training

Before describing how to train an over-complete dictionary, let's introduce what is over-complete first. If every element in Banach space X can be approximated arbitrarily well in norm by linear combinations of elements in $\{v_i\}$ $i\epsilon K$, then the system $\{v_i\}$ $i\epsilon K$ is complete. If the system is still complete after removing a $v_{j,}$ then the system {v_i} *iϵK* is over-complete [26, 27].

In this section, we will describe how to train the over-complete dictionary using the K-SVD algorithm [33]. The purpose of this algorithm is to generate an over-complete dictionary matrix $D\epsilon\mathbb{R}^{nxK}$, which contains K atoms, $\{d_i\}_{i=1}^{K}$. Such a dictionary matrix can lead to the best possible presentations for each image patch with strict sparsity constraints.

Given a set of sample image patches $X = \{x_1, x_2, ..., x_k\}$, the over-complete dictionary is trained by the following model [2]:

$$\min_{D,\alpha} \|X - Da\|_F^2$$

$$\text{S.t., } \forall i, \|a_i\|_0 \leq T_0 \text{ , i=1,2,…,M} \qquad (3.28).$$

where,

$X = \{x_1, x_2, ..., x_m\} \in \mathbb{R}^{N*M}$ is the training sample image patches.

$D \in \mathbb{R}^{N*K}$ is a dictionary matrix needed to be optimized.

a is the sparse representation matrix.

There are two main steps for the K –SVD algorithm [33]:

1. Fix dictionary *D*, find the best sparse representation matrix a.

In this stage, the optimization problem becomes finding the optimal sparse representations with coefficients summarized in the matrix a:

$$\|X - Da\|_F^2 = \sum_{i=1}^{M} \|x_i - Da_i\|_2^2 \qquad (3.29).$$

The problem in equation 2.28 becomes:

$$\min_{a_i} \|x_i - Da_i\|_2^2$$

$$\text{s.t., } \forall i, \|a_i\|_0 \le T_0 \text{ , i = 1,2,…,M} \qquad (3.30).$$

2. Given the optimal representation matrix a from step 1, update each column of *D*.

   In this step, both representation matrix a and dictionary *D* are fixed; only update one column *d_k* in dictionary *D* each time. The problem in this step becomes:

$$\|X - Da\|_F^2 = \left\|X - \sum_{j=1}^{K} d_j a_T^j\right\|_F^2$$

$$= \left\|(X - \sum_{i \neq k} d_j a_T^j) - d_k a_T^k\right\|_F^2$$

$$= \left\|E_k - d_k a_T^k\right\|_F^2 \qquad (3.31).$$

where

a_T^k is the kth row in matrix a.

E_k is the error matrix for all the terms except the kth term.

For the problem in equation 3.31, K-1terms in dictionary D are fixed; we only need to find the optimal solution for the k^{th} term.

In practice, equation 3.30 is a NP-hard problem; it is computationally difficult to find the exact solution. One easier way is to approximate a solution to the following formulation:

$$D = \min_{D,\alpha} \|X - Da\|_2^2 + \lambda \|a\|_1$$

$$s.t. \|D_i\|_2^2 \leq 1, i = 1,2, \dots, K \qquad (3.32).$$

As we see, the problem described in equation 3.32 is a convex problem; there are two variables D and a. It is impossible to keep both variables convex at the same. In order to find the optimal solution, we need to update one variable and keep the other variable fixed each time. The approximate K-SVD algorithm [33] to find the optimal over-complete dictionary is:

Input: a set of sample image patches $X = \{x_1, x_2, \dots, x_n\}$

1. Initialize $D = D_0$, with each column unit normalized

2. Fixed D, optimize the sparse representation matrix a by solving

$$a = \min_{\alpha} \|X - Da\|_2^2 + \lambda \|a\|_1 \qquad (3.33).$$

3. Using a from step 2, update D column by column by:

For all *k* = 1, 2, …, *K*

- Compute the overall error matrix E_k by:

$$E_k = X - \sum_{i \neq k} d_j a_T^j \qquad (3.34).$$

- Fix all the *K*-1 terms in the dictionary *D*, update d_k and a_T^k by:

$$\{ d_k, a_T^k\} = arg\ min\left\|E_k - d_k a_T^k\right\|_F^2$$

$$\text{s.t. } \|d_k\|_2 = 1 \qquad (3.35).$$

4. Repeat step 2 and step 3, alternately update sparse coding matrix and the dictionary until they converge.

Output: over-complete dictionary *D*.

3.4.3 Joint Dictionary Training

In the previous section, we mentioned that we need to add constraint to D_h and D_l, so that they can be guaranteed to have the same sparse coefficient α for each LR and HR image patch pair. Given the sample image patch pairs {*X, Y*}, where X = {$x_1, x_2, ..., x_n$} are the set of high-resolution image patches and *Y* = {$y_1, y_2, ..., y_n$} are the corresponding low-resolution image patches, in order to enforce D_h and D_l share the same sparse coefficient, instead of training D_h and D_l separately using the method discussed in over-complete dictionary training section, Yang et al. [2] suggested training

D_h and *D_l* jointly by adding a constraint, which force the two over-complete dictionary to share the same spare representation. The problem can be defined as:

$$\min_{\{D_h, D_l, a^h, \alpha^l\}} \frac{1}{2}\ \{\|X - D_h a^h\|_2^2 + \lambda\|a^h\|_1\} + \frac{1}{2}\{\|Y - D_l a^l\|_2^2 + \lambda\|a^l\|_1\}$$

$$s.t. \left\|D_{h(i)}\right\|_2^2 \leq 1, \left\|D_{l(i)}\right\|_2^2 \leq 1, a^h = \ a^l,\ i = 1,2, \dots, K \qquad (3.36).$$

which is equivalent to

$$\min_{\{D_h, D_l, a\}} \|X - D_h a\|_2^2 + \|Y - D_l a^l\|_2^2 + \lambda\|a\|_1\}$$

$$s.t. \left\|D_{h(i)}\right\|_2^2 \leq 1, \left\|D_{l(i)}\right\|_2^2 \leq 1, i = 1,2, \dots, K \qquad (3.37).$$

We can group the two reconstruction error terms together, letting

$$X_c = \begin{bmatrix} \frac{1}{\sqrt{N}} X \\ \frac{1}{\sqrt{M}} Y \end{bmatrix}, D_c = \begin{bmatrix} \frac{1}{\sqrt{N}} D_h \\ \frac{1}{\sqrt{M}} D_l \end{bmatrix} \qquad (3.38).$$

Then Eqn. 3.36 becomes:

$$\min_{\{D_h, D_l, a\}} \|X_c - D_c a\|_2^2 + \lambda(\tfrac{1}{N} + \tfrac{1}{M})\|a\|_1$$

$$\text{s.t. } \left\|D_{c(i)}\right\|_2^2 \leq 1, i = 1,2, \dots, K \qquad (3.39).$$

The algorithm of joint dictionary training is similar to the algorithm discussed in the previous section. We still use the K-SVD algorithm to find the optimal solution. This time, there are three variables in the equation: *D_h*, *D_l* and a. Again, it is a convex problem; it is impossible to keep all the three variables convex at the same time, but we can update one variable each time. In order to find the optimal solution, we need to update one variable and keep the other two variables fixed each time. The main steps for reaching the optimal solution are as follows [2, 105].

Given a set of sample image patch pairs $\{X, Y\}$, $X = \{x_1, x_2, ..., x_k\}$, and $Y = \{y_1, y_2, ..., y_k\}$. X contains high-resolution image patches, and Y contains their corresponding low-resolution image patches.

1. Initialize $D_h = D_{h0}$ and $D_l = D_{l0}$ with some random value.

2. Fixed D_h and D_l, update a by

$$a = \min_{\{a\}} \|X_c - D_c a\|_2^2 + \lambda(\frac{1}{N} + \frac{1}{M})\|a\|_1 \qquad (3.40).$$

3. Fixed D_l and a, update D_h by

$$D_h = \min_{\{D_h\}} \|X_c - D_c a\|_2^2 + \lambda(\frac{1}{N} + \frac{1}{M})\|a\|_1 \qquad (3.41).$$

4. Fixed D_h and a, update D_l by

$$D_l = \min_{\{D_l\}} \|X_c - D_c a\|_2^2 + \lambda(\frac{1}{N} + \frac{1}{M})\|a\|_1 \qquad (3.42).$$

5. Repeat step 2 to step 4, update D_h and D_l, and α iteratively until convergence.

Output the optimized solution of D_h and D_l

The over-complete dictionaries D_h and D_l generated through the methods described above can be used to present the image information since the two dictionaries are trained with the constraint that both dictionaries share the same sparse representation for low-resolution image patch and its corresponding high-resolution image. As a result, for each LR image patch, we can calculate its sparse representation with respect to the over-complete dictionary D_l. This sparse representation will be used to recover the HR image details through the over-complete dictionary D_h.

So far, we have described our proposed method for recovering the high-resolution image in detail. We use both objective and subjective methods to evaluate the performance of our proposed method. The subjective measurement is performed using our eyes. We demonstrated the HR results using our proposed method in section 4. The objective measurement is calculated using Peak Signal to Noise Ratio (PSNR), which is defined as:

$$MSE = \frac{\sum_i \sum_j [X(i,j) - Y(i,j)]^2}{M * N} \quad (3.43).$$

$$PSNR = 10\, log_{10}(255 * 255/MSE) \quad (3.44).$$

where

M*N is the dimension of the image, and

MSE measures the mean square error between image X and Y

CHAPTER 4

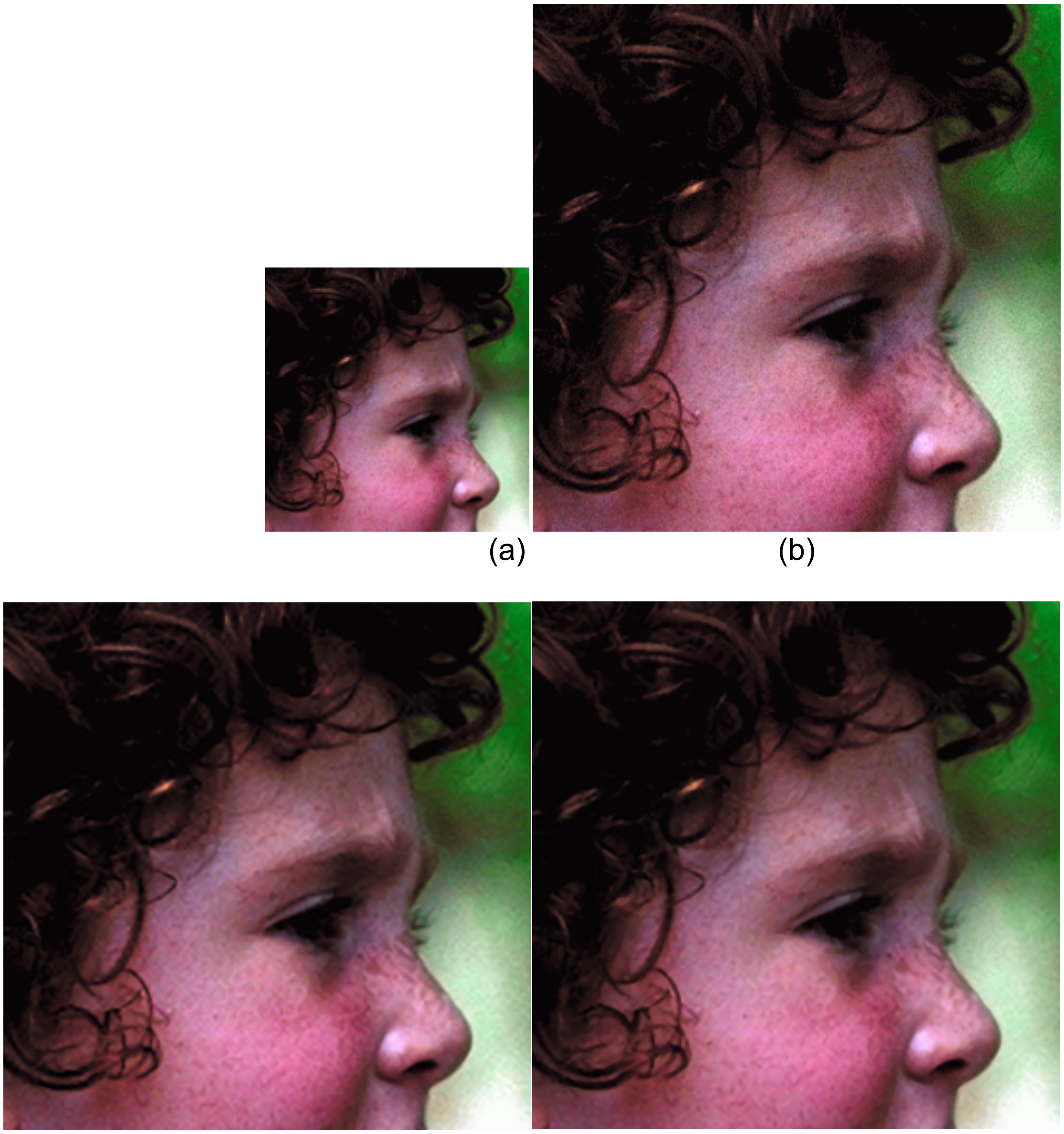
EXPERIMENTAL RESULTS

Quantitative evaluation has been conducted to compare the performance of the self-similarity based SR algorithm with the example-based SR algorithm on a variety of image contents. The purpose of this evaluation is to identify which types of an image texture's high-resolution details can be successfully recovered by using a self-similarity based approach and for which types of image texture, its high-resolution image can be better recovered by utilizing an example-based super-resolution algorithm.

4.1 Comparison of Two Existing SR Algorithms

In this section, we quantitatively compare the performance of two state-of-the-art super-resolution algorithms (example-based and self-similarity based SR algorithms) on a variety of real world image textures, which include faces, animals, architecture, and nature scenes. The purpose of doing this is to evaluate the performance of example-based and self-similarity based SR algorithms on different types of image content. Our proposed SR method will refer to the evaluation result in order to choose the appropriate SR algorithm for each image region.

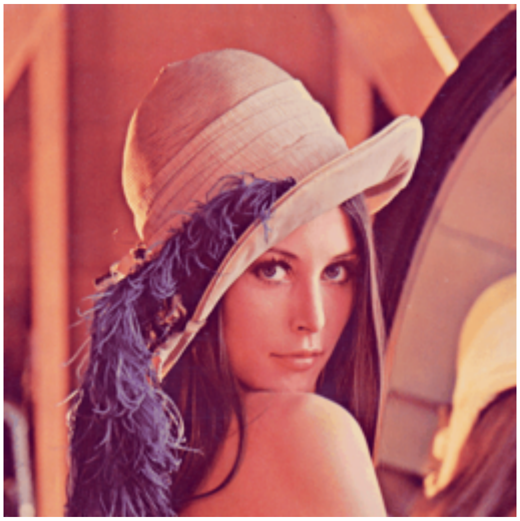
We tested 100 natural scene images taken from the Berkeley segmentation dataset, and 100 urban images that contain urban and architectural scenes. Figure 14 is an example of a human face with magnified factor 2. From the result, we can see that the self-similarity method can successfully recover the HR details through the internal database generated from the LR input image itself.



(a) (b)

(c) (d)

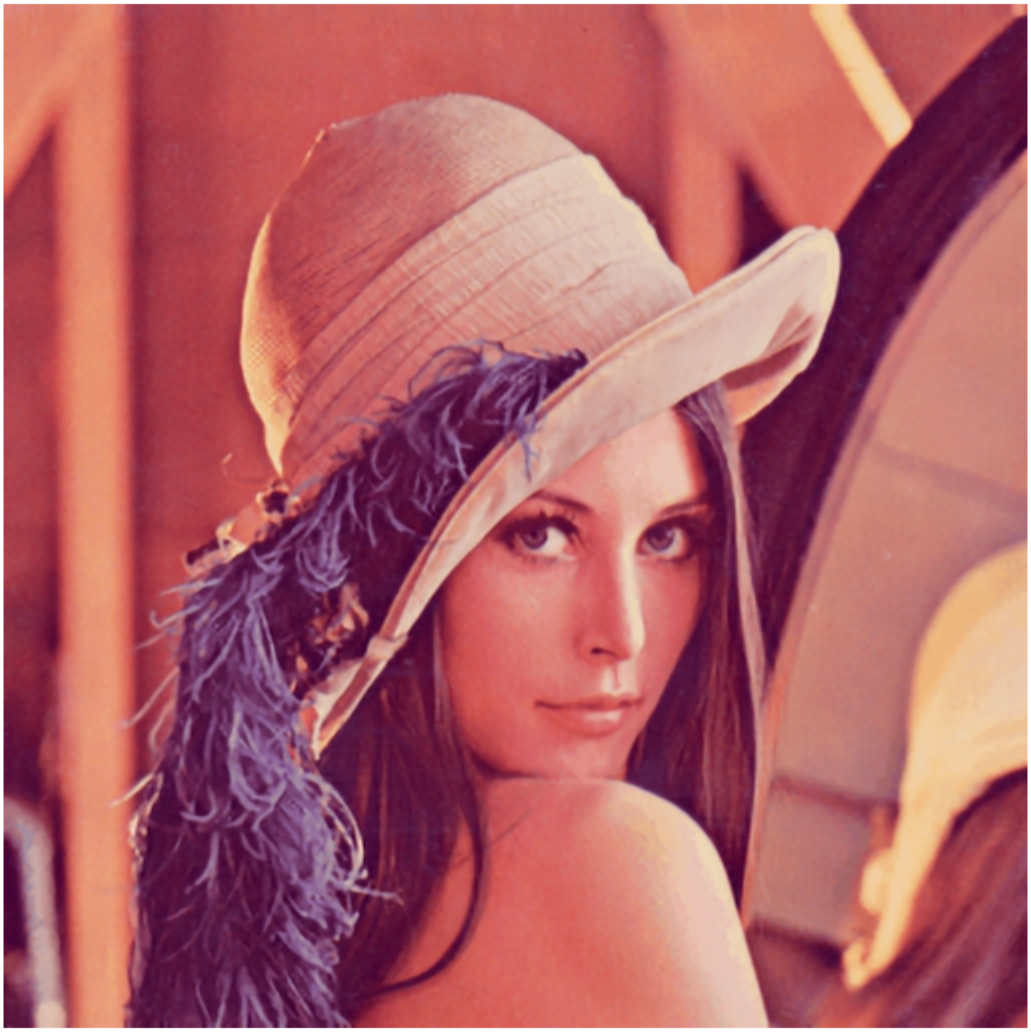
Figure 14: Results of The Girl Image Magnified by a Factor of 2. (a) LR input image, (b) ground truth HR image, (c) 2x HR image generated using self- similarity SR algorithm, (d) 2x HR image generated using example-based sparse coding SR algorithm



(a)

(b)

(c)



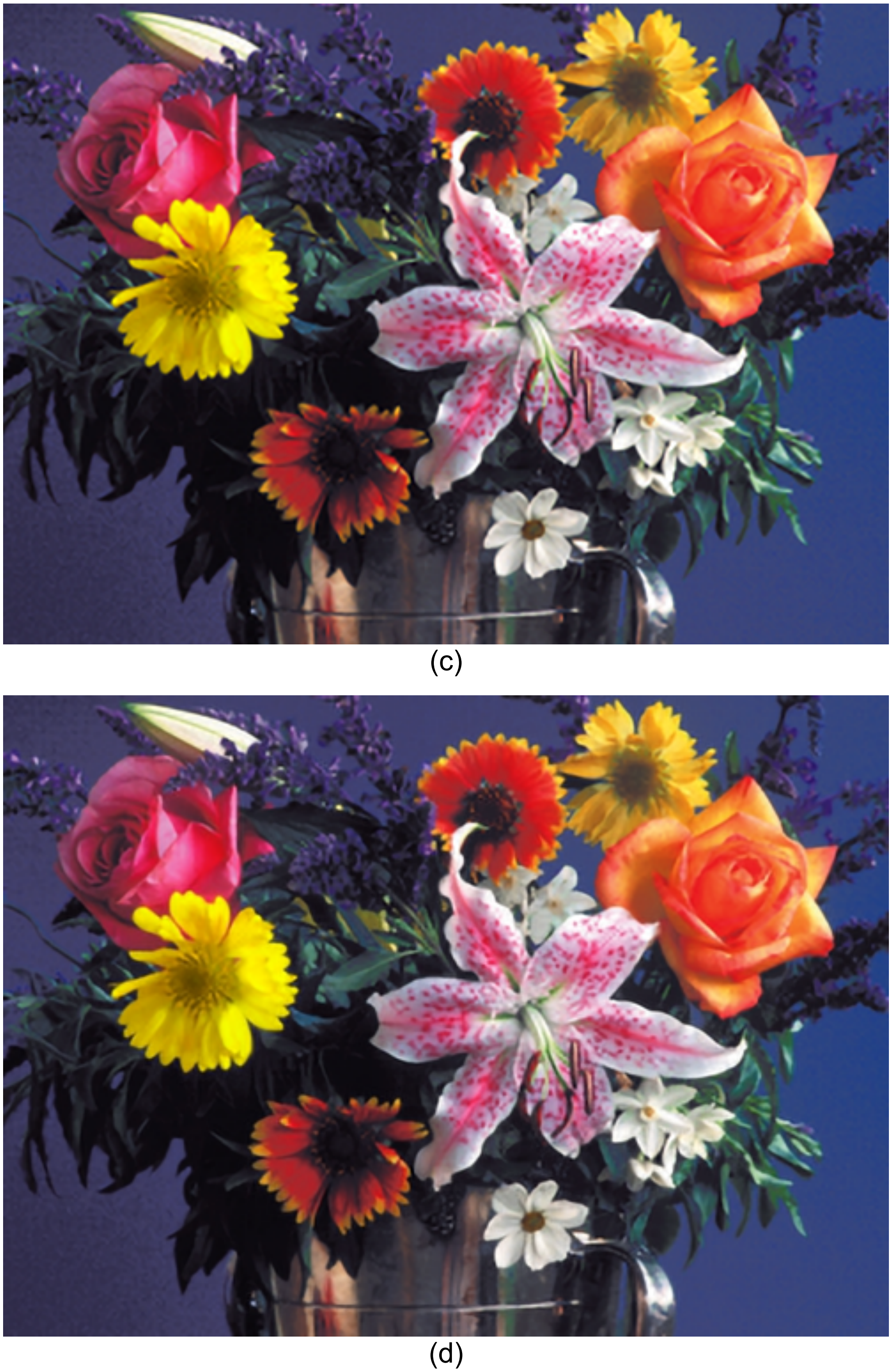
(d)

Figure 15: Results of Image Lena Magnified by a Factor of 2 (a) LR input image, (b) ground truth HR image, (c) 2x HR image generated using self- similarity SR algorithm, (d) 2x HR image generated using example-based sparse coding SR algorithm

Figure 15 is another example of a human face with magnified factor 2. From the result, we can see that the self-similarity method can successfully recover the HR details of the hat through the internal database generated from the LR input image itself.

(a)

(b)



(c)

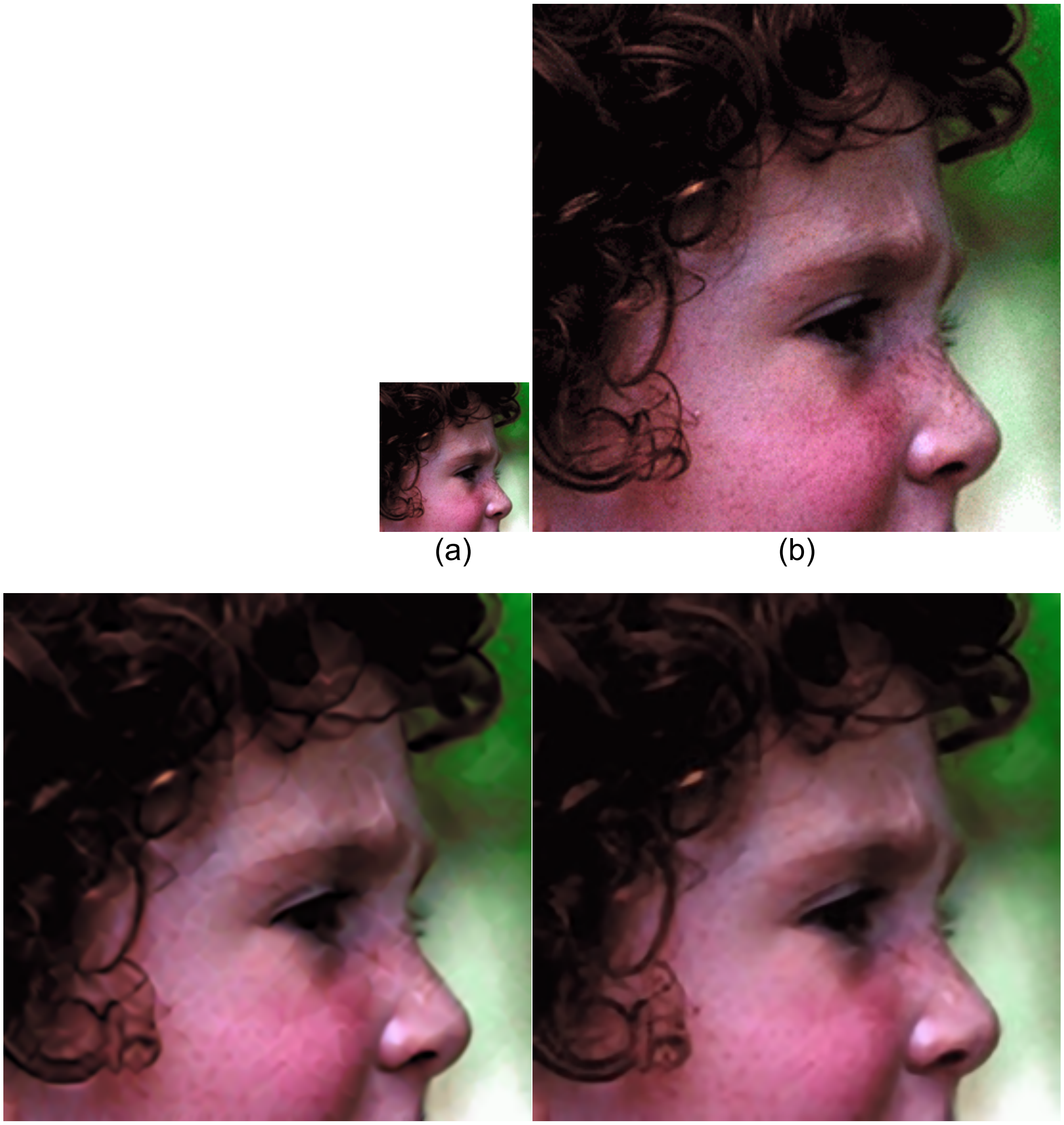
(d)

Figure 16: Results of Flower Image Magnified by a Factor of 2 (a) LR input image, (b) ground truth HR image, (c) 2x HR image generated using self- similarity SR algorithm, (d) 2x HR image generated using example-based sparse coding SR algorithm

Figure 16 is an example of a natural scene with magnified factor 2. From the result, we can see that the self-similarity method can successfully recover HR details for the flowers and leaves through the internal database generated from the LR input image itself.

Figures 14, 15, and 16 show super resolution results (2X) on the face and natural images. From the result, we can see that when the magnified factor is small (equal to or less than 2), and the resolution of the testing image is good, the self-similarity method can successfully recover HR details without using the external database. From visual inspection, we cannot tell the difference between the high-resolution images recovered using the two SR approaches.

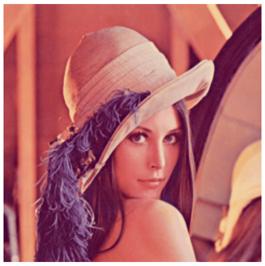
Figure 17 shows the super-resolution result with a magnified factor of 4. The testing image is a down-sampled image of the testing image in figure 14. From the result, we can see a blocky effect from the face recovered using the self-similarity based method [1]; we also can see annoying artifacts around the hair area. The face recovered using the example-based algorithm [2] looks more natural and is less noisy.



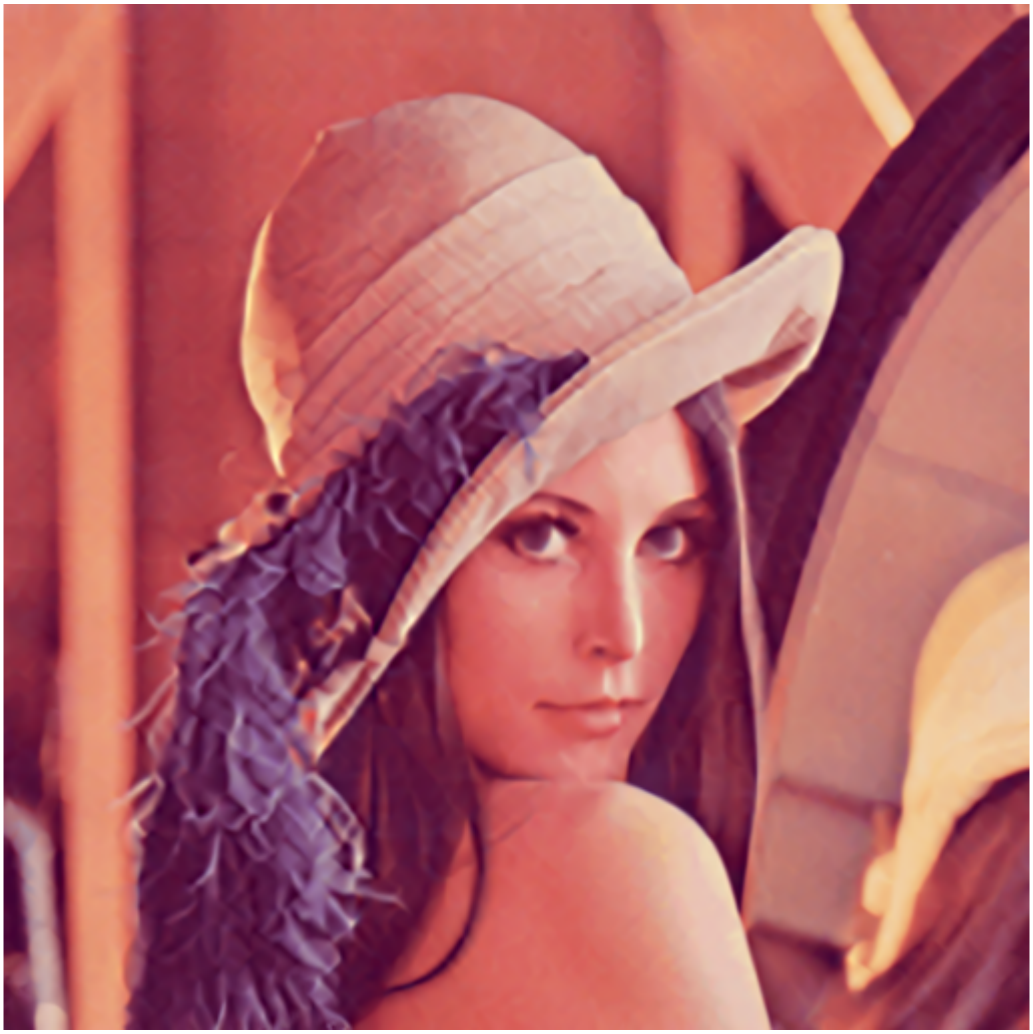
(a) (b)

(c) (d)

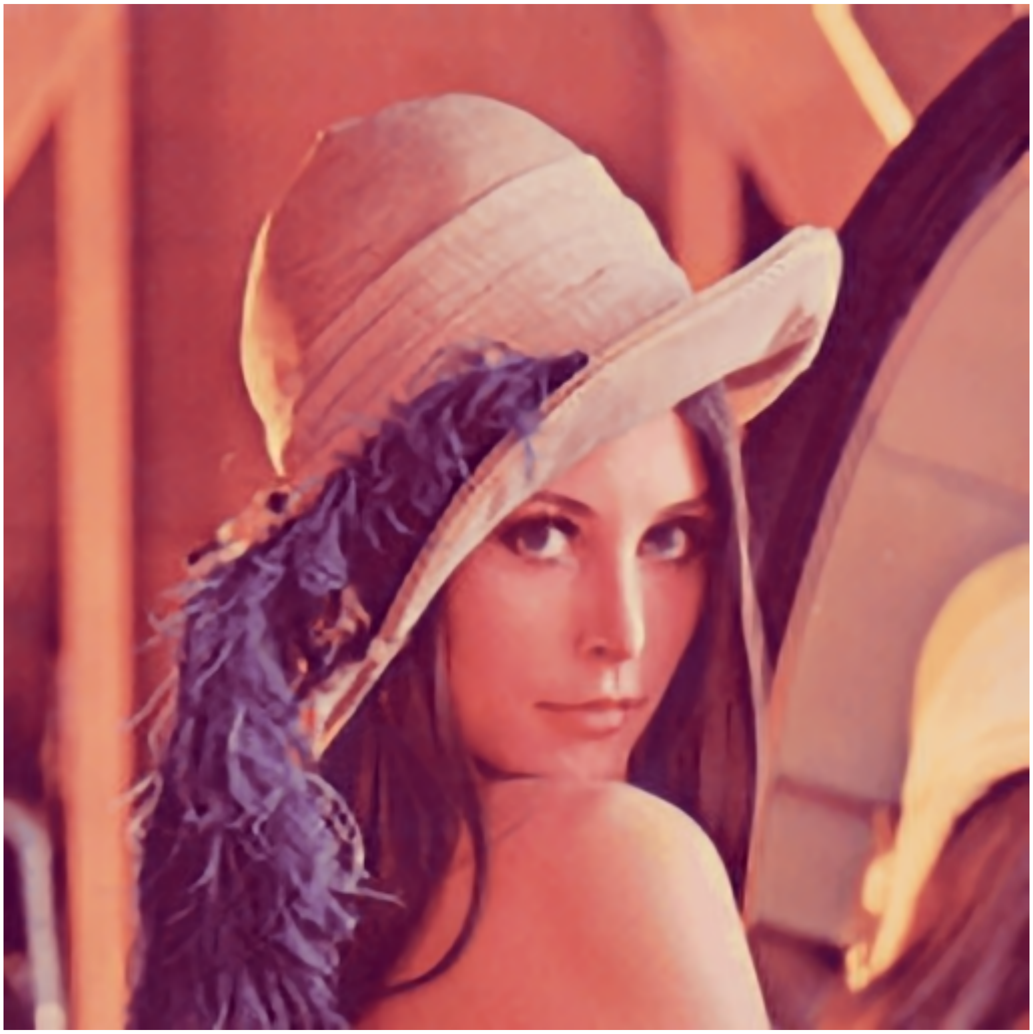
Figure 17: Results of Little Girl Magnified by a Factor of 4 (a) LR input image, (b) ground truth HR image, (c) 4x HR image generated using self- similarity SR algorithm, (d) 4x HR image generated using example-based sparse coding SR algorithm



(a)



(b)

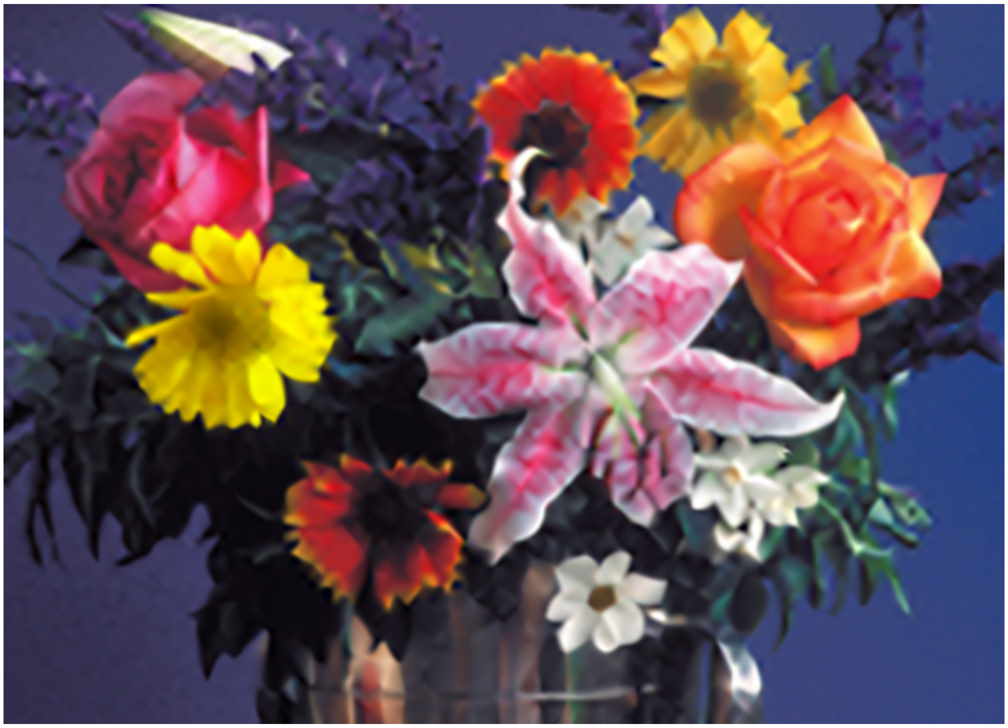


(c)

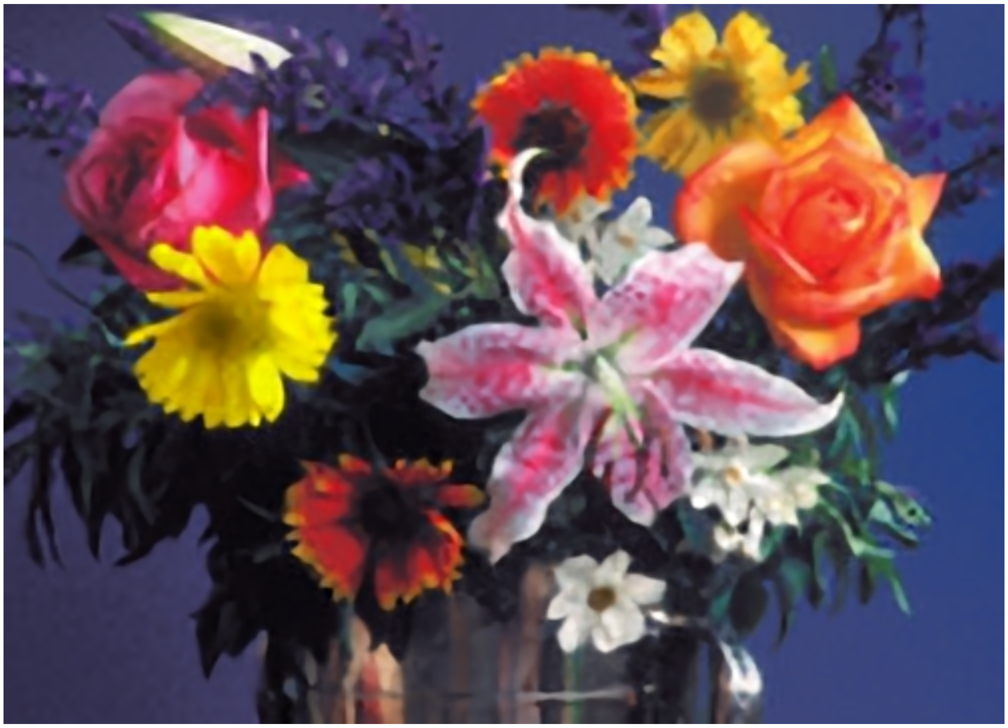
Figure 18: Results of Image Lena Magnified by a Factor of 4 (a) LR input image, (b) 4x HR image generated using self- similarity SR algorithm, (c) 4x HR image generated using example-based sparse coding SR algorithm

For the example shown in figure 18, self-similarity based algorithm generates comparable results with the example-based algorithm in most image areas, but for the decoration of the hat area, the example-based algorithm generates more natural looking details.

(a)



(b)



(c)

Figure 19: Results of Flower Image Magnified by a Factor of 4 (a) LR input image, (b) 4x HR image generated using self- similarity SR algorithm, (c) 4x HR image generated using example-based sparse coding SR algorithm.

For the flower image shown in figure 19, the self-similarity based algorithm failed to recover the fine details of the flowers and leaves. We can see the petals are over smoothed and look unnatural in the recovered 4x HR image.

In figures 17, 18, and 19, we show the 4x super resolution results for the same images. Comparing with examples in figures 14, 15, and 16, we use the same image with lower resolution as the input image. The LR input image is built by blurring and down sampling the original image at a scale of 4. For results in figure 17, we can see over smoothing on the girl’s face in the HR image recovered by the self-similarity algorithm while the example-based algorithm generates more natural looking details.

When the input LR image has good resolution quality, as the examples shown in figures 14, 15, and 16, the self-similarity algorithm is able to generate fine, natural looking HR details, and preserve image sharpness. However, when the resolution of the input LR image is low, the self-similarity based algorithm performs slightly worse than the example-based algorithm.

Table 1: Super-Resolution Quality Measurement on Examples In PSNR

	Scale	Little girl	Lena	Flower
Self-similarity based SR [1]	2x	35.6412	36.5361	33.0719
	4x	32.4178	31.4381	27.2028
Example-based SR [2]	2x	35.6125	36.5054	33.2846
	4x	32.4851	31.5048	27.2319

The figures compare the visual quality of high-resolution results using two state-of-the-art algorithms, but since visual quality is subjective, each person may have a different standard. We compared the quality of the high-resolution image in terms of PSNR as well. From the comparison results of table 1, we can see that when the resolution of the input image is good, the self-similarity based SR algorithm [1] can produce a better or comparable high-resolution results[2], but when the resolution of the input image is poor, the self-similarity based SR algorithm performs slightly worse than the example-based SR algorithm because there is limited high-resolution information that can be found through the input image itself.

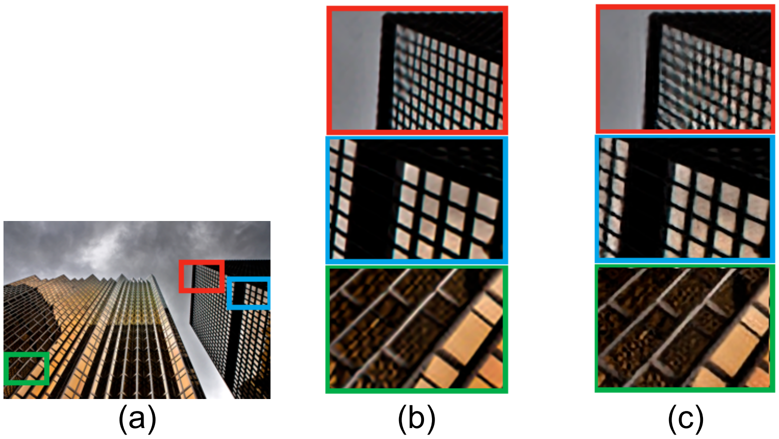


Figure 20: Results of Urban Architecture Magnified by a Factor of 4 (a) LR input image, (b) 4x HR image generated using self- similarity SR algorithm [1], (c) 4x HR image generated using example-based sparse coding SR algorithm [2].

Figure 20 shows an example of urban architecture. We cropped the HR images due to size and only show the area in the rectangular box. By comparing the two results, we can see the building recovered using the self-similarity based SR algorithm has a sharper edge and fewer artifacts. For the image region marked in the red rectangle, we can clearly see that the self-similarity based SR algorithm [1] has less noise and fewer artifacts, and preserves sharper edges than the example-based SR algorithm.

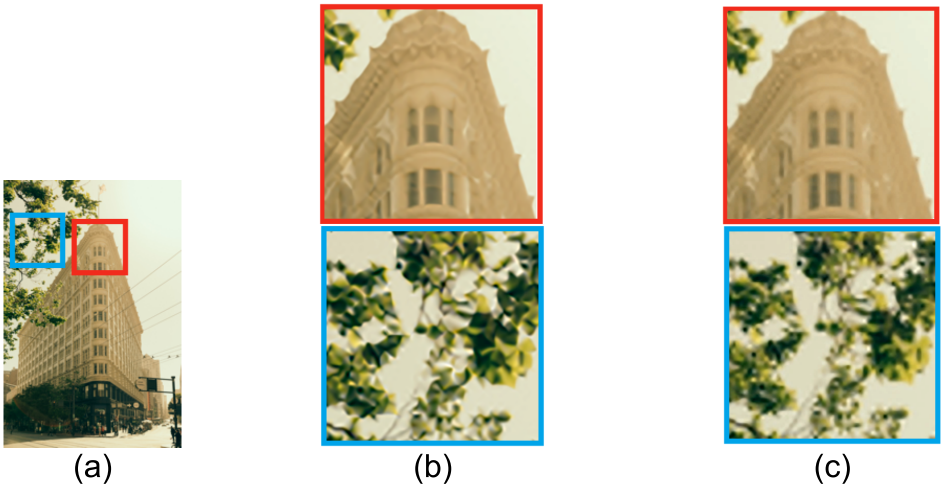
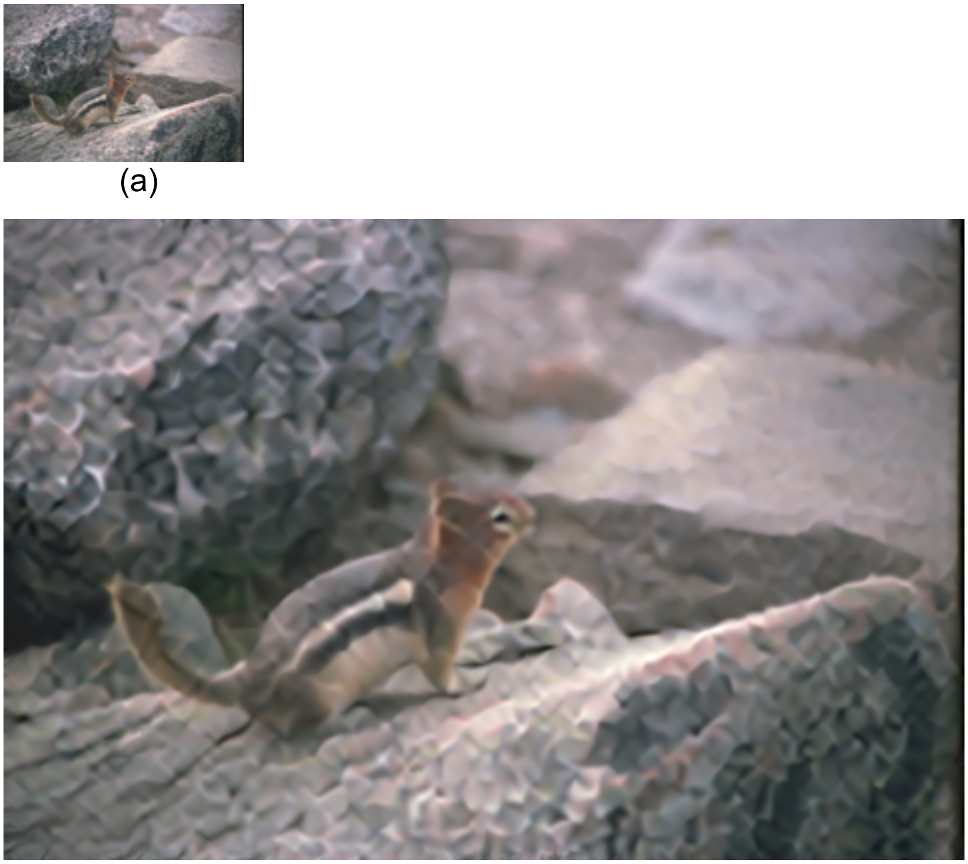


Figure 21: Results of Urban Architecture Magnified by a Factor of 4 (a) LR input image, (b) 4x HR image generated using self- similarity SR algorithm, (c) 4x HR image generated using example-based sparse coding SR algorithm.

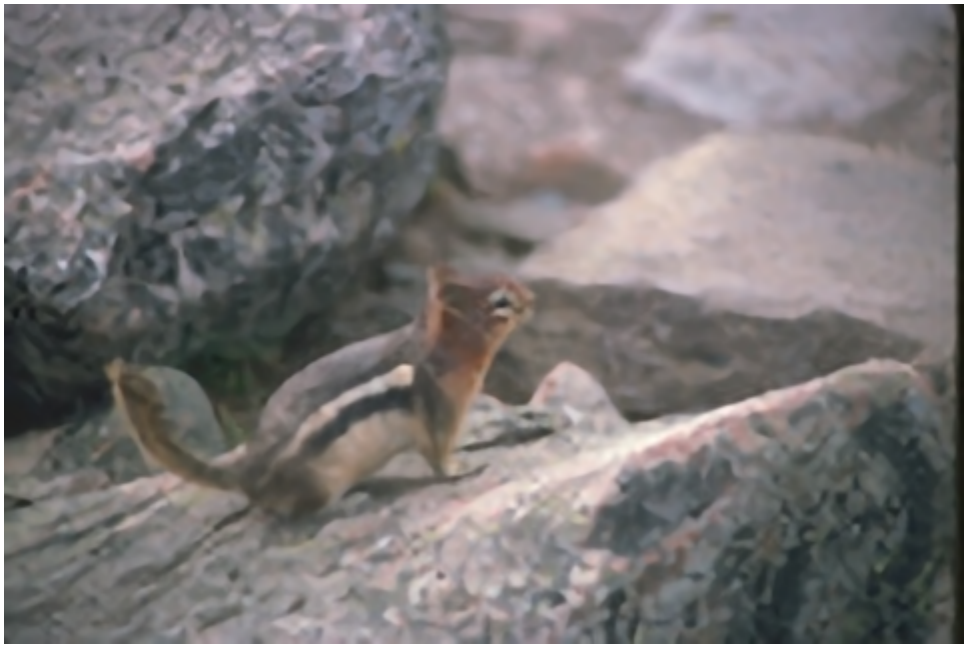
Figure 21 shows an example of urban architecture. We cropped the HR images of two regions: 1) architecture, and 2) tree leaves. By comparing the results, we can see the building recovered using the self-similarity based SR algorithm preserves sharper edges and contains fewer artifacts but for the tree leaves, the example-based SR algorithm generates better, more natural looking results.

We evaluated the self-similarity algorithm and the example-based algorithm on 100 urban architectural images using a dataset named Urban100. Our test results show that the self-similarity algorithm always outperforms the example-based SR algorithm on urban architecture. Figures 20, and 21 show two examples of urban architectural images, we can see that the HR image of architecture recovered using the self-similarity algorithm is sharper than the one generated using the example-based SR algorithm, but the tree leaves and stone generated using the example-based algorithm looks more natural than the ones reconstructed through the self-similarity approach.



(a)

(b)



(c)

Figure 22: Results of Squirrel Magnified by a Factor of 4 (a) LR input image, (b) 4x HR image generated using self- similarity SR algorithm, (c) 4x HR image generated using example-based sparse coding SR algorithm.

Figure 22 shows an example of super resolution results on a natural scene. We can see the stone obtained through the example-based SR method 22(c) has more natural looking details than the one shown in figure22 (b), which is reconstructed through the self-similarity based SR method. We can clearly see annoying blocky and over-smooth effects in figure 22(b).

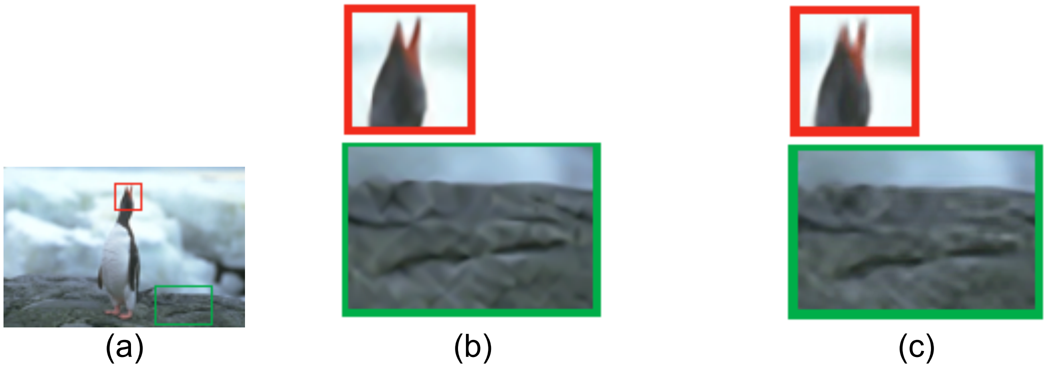


Figure 23: Results of Penguin Magnified by a Factor of 4 (a) LR input image, (b) 4x HR image generated using self- similarity SR algorithm, (c) 4x HR image generated using example-based sparse coding SR algorithm.

Figure 23 is another example of super resolution results on a natural scene containing wild animals and natural scenes. We can see the penguin reconstructed though the self-similarity approach has a slightly sharper beak. While the ground obtained through the example-based method has more natural looking details.

So far, we have shown the high-resolution results on several images with a variety of different contents. The self-similarity based SR algorithm always outperforms the example-based SR algorithm on urban architecture, animals, and insects. The example-based SR algorithm generates more natural looking faces, trees, flowers, and grasses.

We also compared the performance of the existing example-based [2] and self-similarity based [1] SR algorithms on a variety of image textures, which include faces, animals, architecture, and nature scenes. The comparison was conducted in terms of visual quality and PSNR. The purpose of this comparison is to evaluate the performance the example-based and self-similarity based SR algorithms on different types of image content. Our proposed SR method will refer to the comparison results in order to select the appropriate SR algorithm for each image region.

Table 2 Quantitative Comparisons on 100 Urban Images in PSNR

	Self-similarity algorithm	Example-based algorithm
2x	31.2546	31.1215
4x	26.8719	26.8066

Table 3 Quantitative Comparisons on 100 Images of Natural Scene in PSNR

	Self-similarity algorithm	Example-based algorithm
2x	28.8515	29.0486
4x	24.3624	24.6692

Table 2 and table 3 show the comparison results for different magnification factors (2x and 4x). Our quantitative evaluation shows that the self-similarity based algorithm can produce compelling or better results on urban images (about 0.1dB better in terms of PSNR). The self-similarity based algorithm can successfully recover the HR details by properly selecting the similar image patch across different image scales.

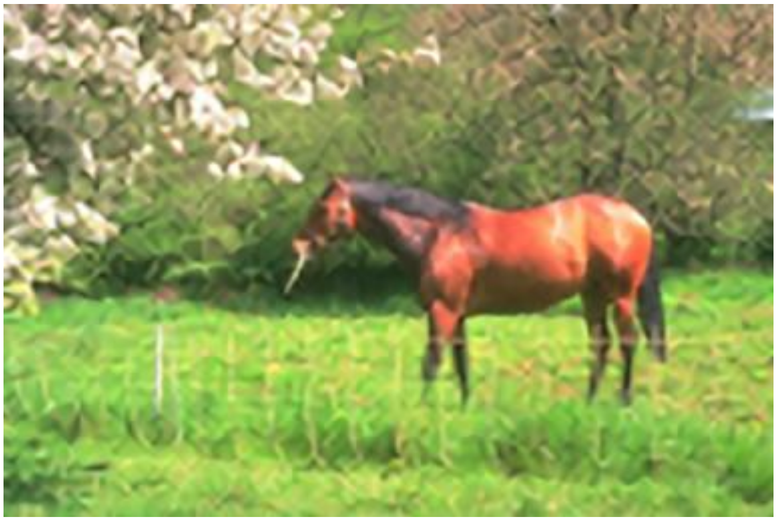
4.2 Comparison of Our Proposed Method Versus Existing Methods

Our quantitative evaluation shows that for urban images with regular geometric texture structure, the self-similarity based SR algorithm can generate a HR image with sharper edges and fewer artifacts, but for some nature images containing an irregular texture structure, it is hard to estimate the proper transformation matrix and find the best matched image patch accurately. As a result, the self-similarity based algorithm performs worse than the example based algorithm.

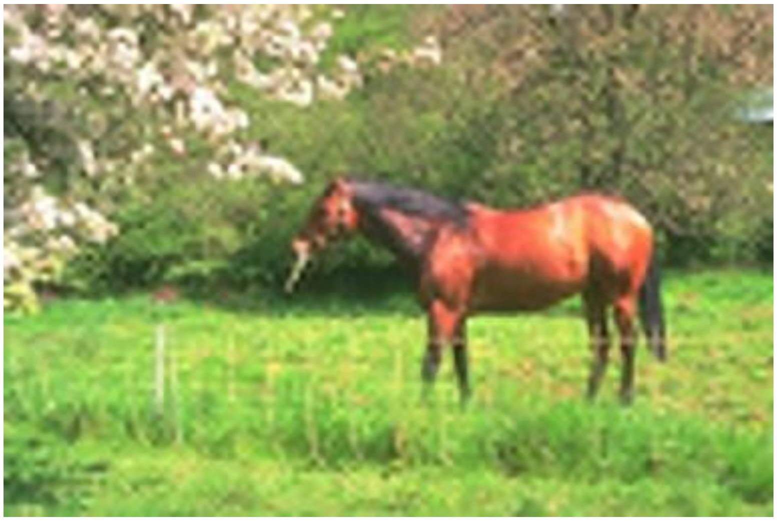
Based on our evaluation results, we divide the image into several regions based on textural structure. Instead of using the same algorithm to process the whole image, we apply a different SR method based on image texture structure. Figure 19 shows the result of using the proposed combined SR algorithm. In this example, we divide the image into two regions; the self-similarity based algorithm is used to reconstruct HR details for the region containing the horse while the example-based algorithm is applied to recover the HR details of the rest of the image region.

For testing the image in figure 24, our proposed method first segments the image into two regions: 1) one region contains the horse and 2) the other region contains trees and grass. By referring to our evaluation result, we learn that the self-similarity based SR algorithm performs slightly worse on tree and grass image textures, so the self-similarity based SR algorithm is used to recover the HR image region containing the horse, while the example-based SR algorithm recovers the other region. As a result, the recovered HR image produces better HR image than using the self-similarity based SR algorithm on the grass region.

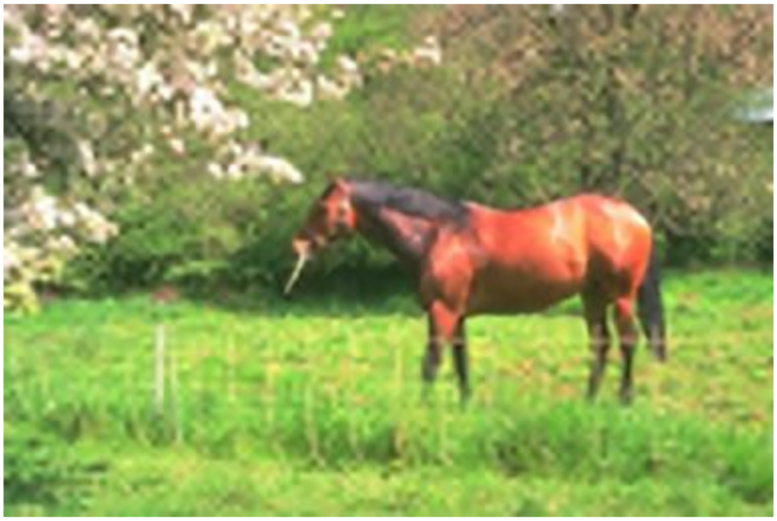
(a)



(b)



(c)

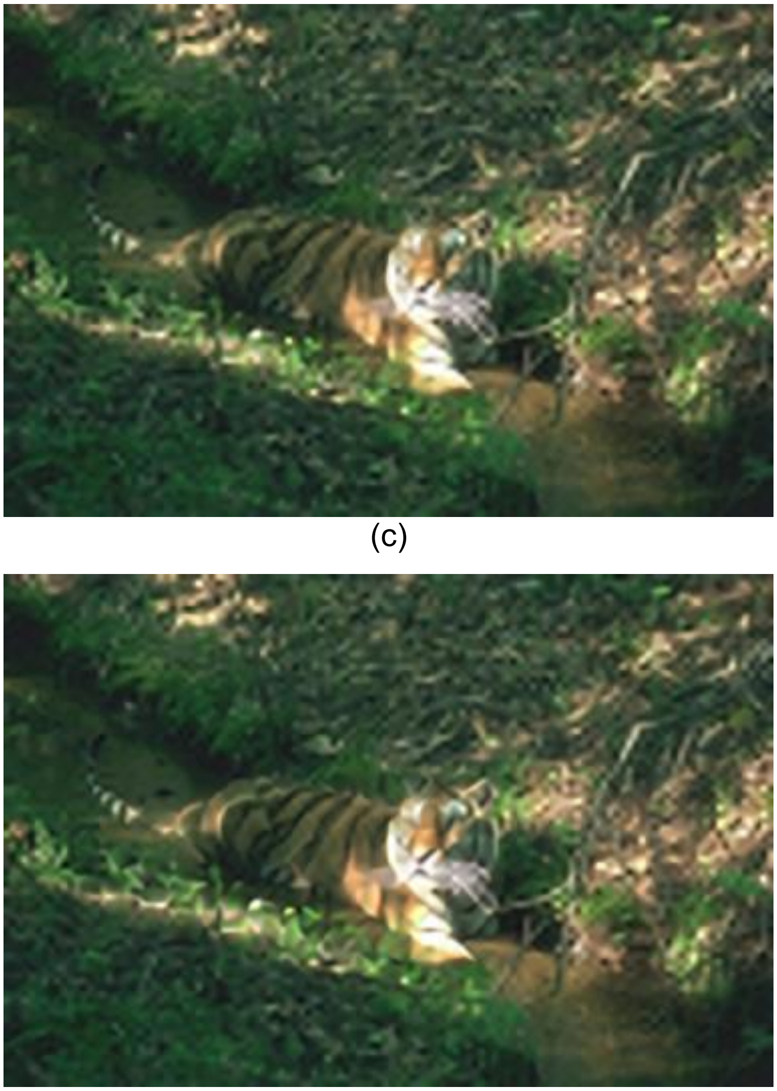


(d)

Figure 24: Results of Horse Magnified by a Factor of 4(a) Original LR image, (b) 4X up scaling SR using self-similarity based algorithm, (c) 4X up scaling SR using example-based SR algorithm, (d) 4X up scaling SR using proposed SR algorithm.

(a)

(b)

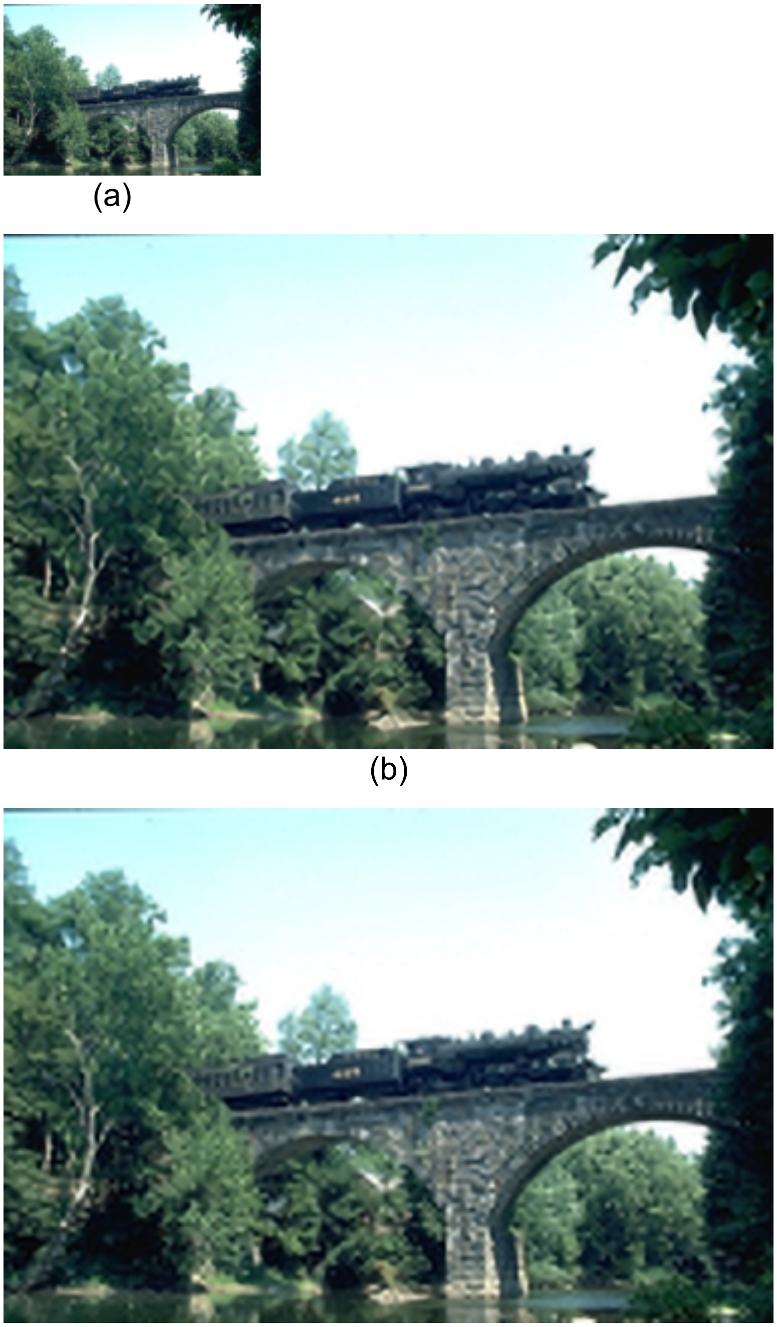


(c)

(d)

Figure 25: Results of Tiger Magnified by a Factor of 4 (a) Original LR image, (b) 4X up scaling SR using self-similarity based algorithm, (c) 4X up scaling SR using example-based SR algorithm, (d) 4X up scaling SR using proposed SR algorithm

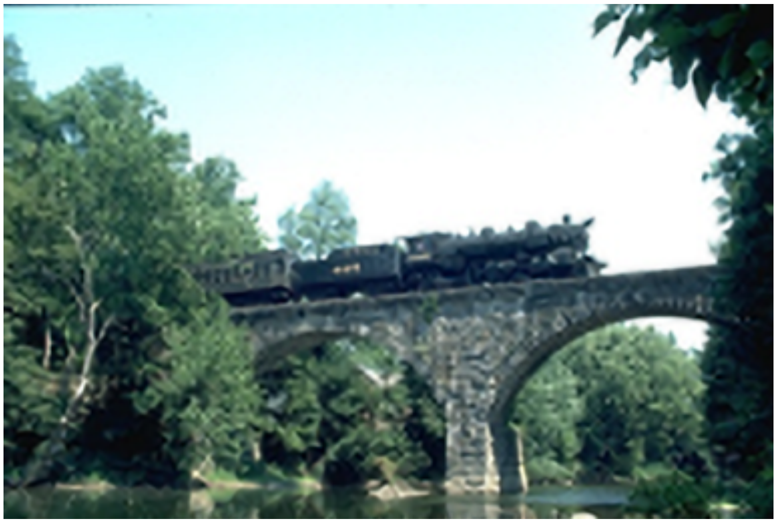
For testing the image in figure 25, our proposed method segments the image into two regions. By referring to our evaluation results, we learn that the self-similarity based SR algorithm performs slightly worse on tree or grass textures, so the self-similarity based SR algorithm is used to recover the HR image region containing the tiger, while the example-based SR algorithm recovers the other region. As a result, the recovered HR image using our proposed approach produces a better HR image than using the self-similarity based SR algorithm on the grass region.



(a)

(b)

(c)

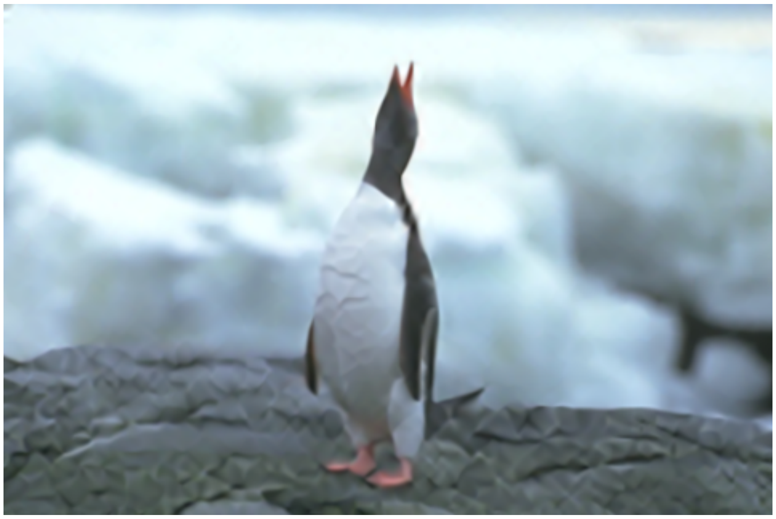


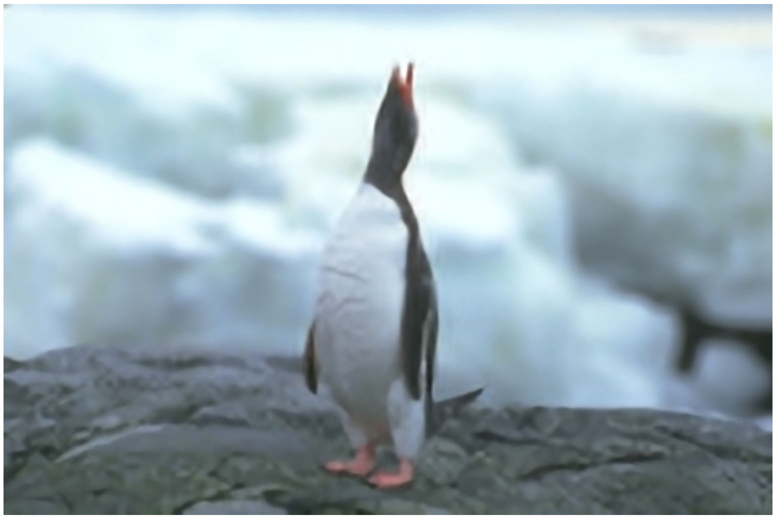
(d)

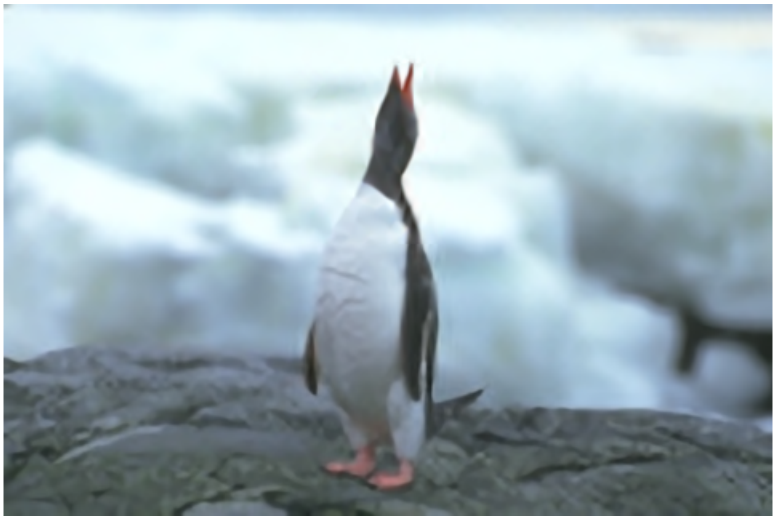
Figure 26: Results of Natural Scene Magnified by a Factor of 4 (a) Original LR image, (b) 4X up scaling SR using self-similarity based algorithm, (c) 4X up scaling SR using example-based SR algorithm, (d) 4X up scaling SR using proposed SR algorithm.

For images containing a natural scene and architecture, as shown in figure 26, our proposed algorithm divided the image into two regions: 1) one region contains manmade architecture, and 2) the other region contains trees. Our proposed method selects the self-similarity based SR algorithm to reconstruct the HR details for the region containing manmade architecture and uses the external database to recover the HR details of the tree region. As a result, the recovered HR image looks more natural in the tree region.

(a)


(b)


(c)



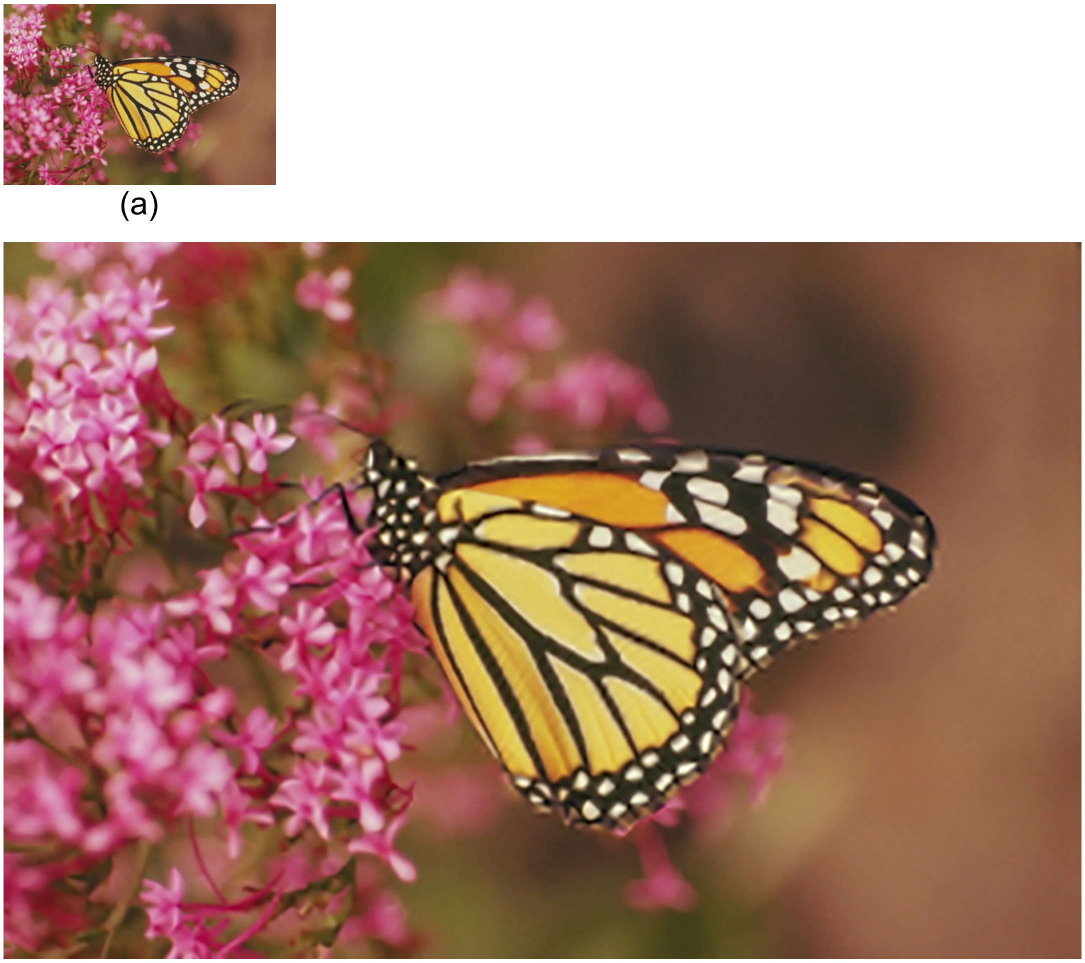
(d)

Figure 27: Results of Penguin Magnified by a Factor of 4 (a) Original LR image, (b) 4X up scaling SR using self-similarity based algorithm, (c) 4X up scaling SR using example-based SR algorithm, (d) 4X up scaling SR using proposed SR algorithm

For the example shown in figure 27, the penguin is recovered through the self-similarity method, and the other region is recovered using example-based SR approach. The penguin recovered from our proposed method is slightly sharper and has less of the ringing effect; the background generated from our proposed method is more natural. The HR image recovered by our proposed method makes it possible to keep the sharpness with natural visualization.

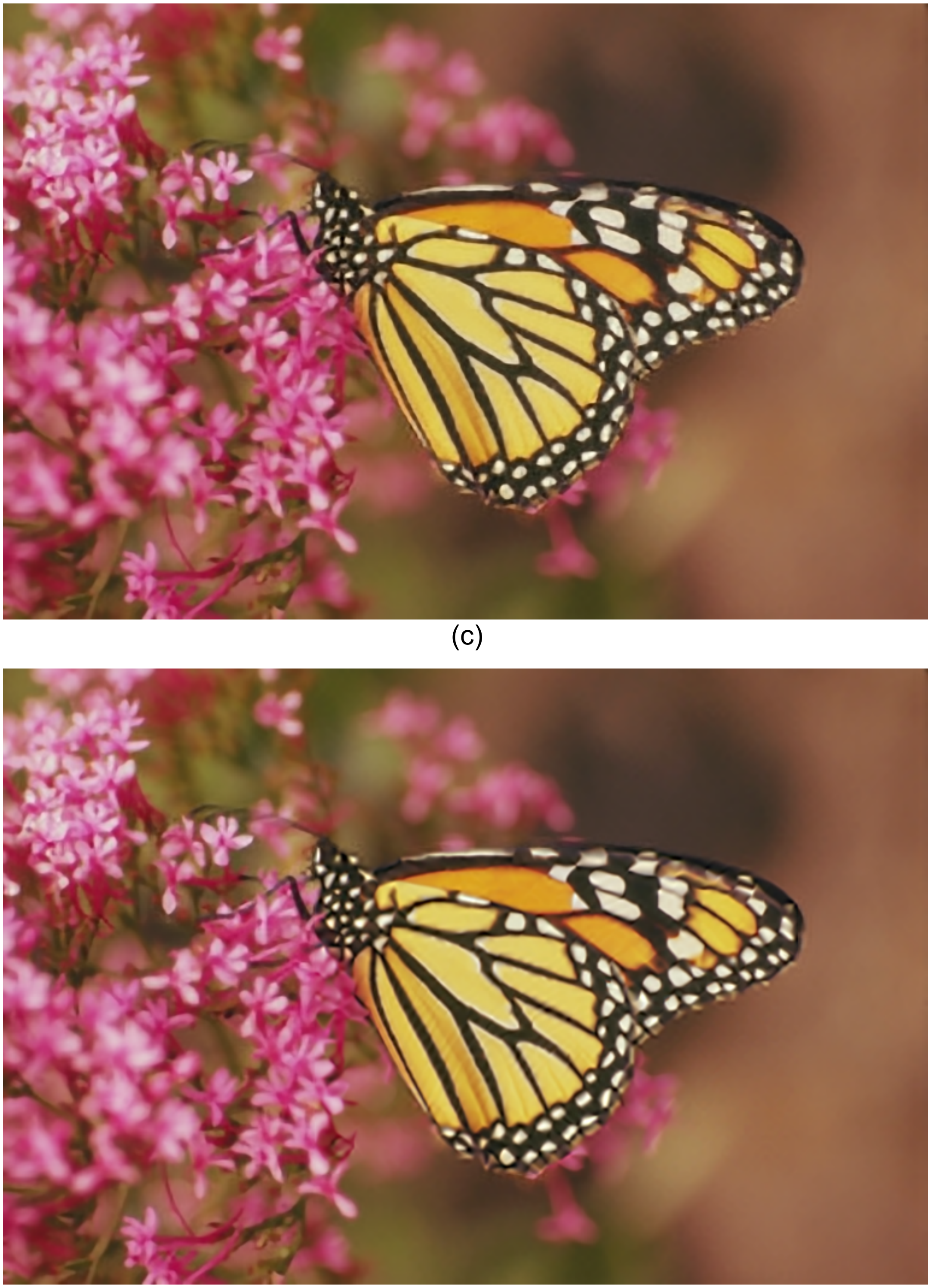
Figures 24, 25, 26, and 27 present the visual comparison of the proposed SR algorithm with the self-similarity based SR algorithm and the example-based SR algorithm. As shown in the examples, our proposed method takes advantage of those two state-of-the-art SR approaches; the recovered HR image using our proposed SR method is slightly sharper and looks more natural.

For the example in figure 28, the structure of the flower is not regular. We can see the high resolution of this area using the self-similarity based algorithm has a blocky and over smooth effect. That's because the self-similarity based algorithm is not able to accurately detect the best matches for the HR image patch. The database used by the example-based algorithm contains natural texture without regularity. We can see the HR image recovered by the example-based algorithm looks more natural. Our proposed method takes advantage of those two SR approaches. The recovered HR image using our proposed SR method is slightly sharper and looks more natural.



(a)

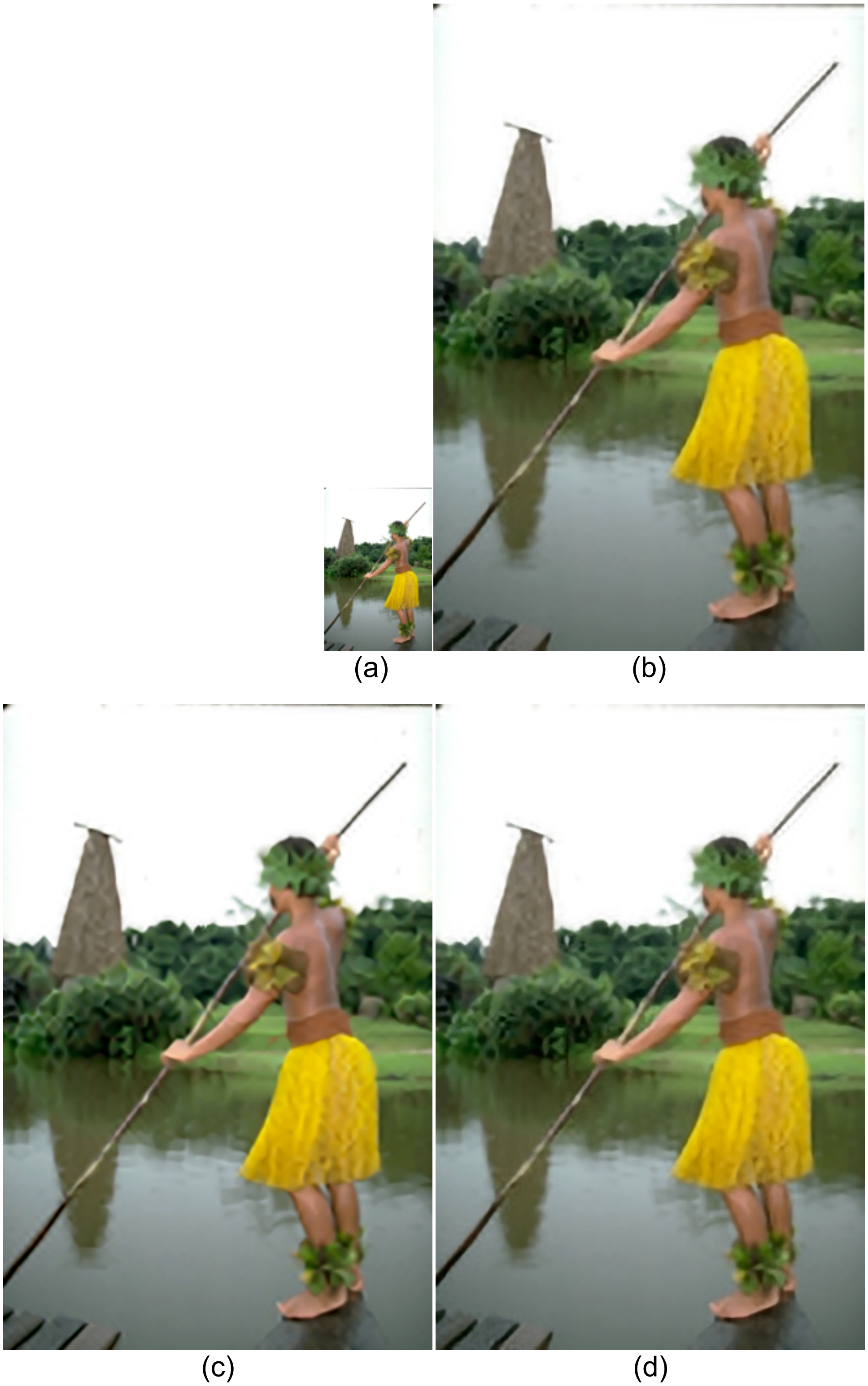
(b)



(c)

(d)

Figure 28: Results of Butterfly Magnified by a Factor of 4 (a) Original LR image, (b) 4X up scaling SR using self-similarity based algorithm, (c) 4X up scaling SR using example-based SR algorithm, (d) 4X up scaling SR using proposed SR algorithm



(a) (b)

(c) (d)

Figure 29: Results of Human Magnified by a Factor of 4 (a) Original LR image, (b) 4X up scaling SR using example-based SR algorithm, (c) 4X up scaling SR using self-similarity based algorithm, (d) 4X up scaling SR using proposed SR algorithm.

For the example shown in figure 29, we can see the high resolution of these areas using the self-similarity based algorithm have blocky and over smooth effects. That’s because the structure of tree leaves is not regular; as a result, the self-similarity based algorithm is not able to accurately detect the best matches for HR image patch. The database used by the example-based algorithm contains image patches with natural texture without regularity. We can see the HR image recovered by the example-based algorithm looks more natural. Our proposed method takes advantage of those two SR approaches; the recovered HR image using our proposed SR method looks more natural.

Table 4 Super-Resolution Comparisons on Examples in PSNR

		Self-similarity based SR	Example-based SR	Proposed method
horse	4x	21.1502	21.1737	21.2543
tiger	4x	22.8315	22.8418	22.8605
natural scene	4x	23.4748	23.2946	23.5096
penguin	4x	32.7132	32.7908	32.8461
human	4x	27.9602	27.8935	28.0563
Squirrel	4X	29.3127	29.6012	29.6408
butterfly	4x	29.5343	29.6684	30.0852

Table 4 compares image quality in terms of PSNR. as shown in the table, our proposed method takes advantage of the two existing SR algorithms, and performs slightly better than using each algorithm alone on the whole image.

CHAPTER 5

CONCLUSION AND FURTURE WORK

5.1 Conclusion

Interest in different approaches to improve the quality of high-resolution images recovered from a single LR image has increased in recent years. Researchers have made progress, but there are still many challenges and limitations than prevent the application of SR algorithms in real world practice.

This thesis has focused on the problem of single image super-resolution. The reasons why we are more interested in recovering HR image details from a single LR image are: 1) in real world applications, it is hard to get several images from the same scene; 2) registration errors between multiple images will spread in the HR image reconstruction process, which will result in annoying artifacts, and aliasing effects. Given the fact that single image super-resolution techniques can successfully avoid the problems mentioned above, we are more interested in improving the performance of generating a HR image from a single LR image.

The current state-of-the-art single image super-resolution algorithms are dealing with the whole image with the same method. To our knowledge, no existing method considers employing contextual information to super-resolution algorithm. We proposed a new concept in image SR research, which suggests segmenting the image into multiple regions based on spatial context and selecting the appropriate SR algorithm for each region based on image content.

We quantitatively evaluated the performance of example-based and self-similarity based SR algorithms on a wide variety of image contents using 200 images, which contain nature scenes, humans, animals, urban architecture. Compared to the example-based SR algorithm, we learned that the self-similarity based algorithm has difficulty reconstructing HR details for images containing contextual information such as small tree leaves, faces and grass, while it can recover the edges better with less noticeable artifacts for images with content such as architecture and animals. The example-based algorithm can generate more natural looking HR details for an image containing irregular geometrical structures, such as trees, flowers, and the ground. Our proposed method refers to the evaluation results in order to choose the appropriate SR method for each image region.

We tested our proposed method on different natural images, which contain animals, architecture, and nature scenes. From our experimental results, we can see that our proposed SR method takes advantage of the existing self-similarity based and example-based SR algorithm; it can produce better results with more high-resolution details in terms of visual quality and in terms of PSNR (+0.1dB) than using each algorithm alone on the whole image.

However, our proposed method consumes more computation time than the other two existing SR methods. For our proposed method, image pre-processing takes a considerable amount of time in order to reach proper segmentation and classification.

What’s more, since we apply different SR algorithms on different image regions based on image content, we need one more back-projection step at the end to ensure that the generated HR image satisfies the global reconstruction constraints. As a result, the computation complexity of our proposed method is higher than using each algorithm alone on the whole image.

5.2 Future Research Work

Since edge sharpness plays an important role in image quality perception, our future work may take the gradient profile prior into account. We will make efforts to improve our proposed SR method by employing rational constraints on the gradient field to preserve image sharpness. The gradient profile may vary with image contents. In our future work, we will study the relationship between gradient features and image textural content and employ the relationship to establish an appropriate gradient profile for different types of image content.

We are also interested in improving computational performance. We will make efforts to speed up our proposed SR algorithm by reducing computational complexity and try to extend our proposed SR algorithm to real-time video application.

REFERENCES

[1] J.-B. Huang, A. Singh. Single Image Super-resolution from Transformed Self-Exemplars. IEEE CVPR, 5197 – 5206, 2015

[2] J. Yang, J. Wright, T. Huang, Y. Ma. Image Super-Resolution via Sparse Representation. IEEE TIP, 19(11) 2861 – 2873, 2010

[3] H. Rauhut, K. Schnass, and P. Vandergheynst. Compressed sensing and redundant dictionaries. *IEEE Transactions on Information Theory*, 54(5), May, 2008.

[4] H. Lee, A. Battle, R. Raina, and A. Y. Ng, Efficient sparse coding algorithms, in Advances in Neural Information Processing Systems (NIPS), 2007

[5] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereostereo matching with slanted support windows. In BMVC, 2011.

[6] C.-Y. Yang, J.-B. Huang, and M.-H. Yang. Exploiting self-similarities for single frame super-resolution. In ACCV. 2010. 2

[7] C.-Y. Yang, C. Ma, and M.-H. Yang. Single-image super resolution: A benchmark. In ECCV, 2014. 5

[8] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In ACCV, 2014.

[9] C.-Y. Yang and M.-H. Yang. Fast direct super-resolution by simple functions. In ICCV, 2013

[10] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. ACM Trans. on Graphics, 30(2):12, 2011. 1

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

[12] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super resolution. IEEE Comput. Graph. Appl., vol. 22, no. 2, pp. 56–65, Apr. 2002.

[13] H. Chang, D. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. in Proc. IEEE CVPR, Jul. 2004, pp. 275–282.

[14] T.-M. Chan, J. Zhang, J. Pu, and H. Huang. Neighbor embedding based super-resolution algorithm through edge detection and feature selection. Pattern Recognit. Lett., vol. 30, no. 5, pp. 494–502, Apr. 2009.

[15] Y. Tang, P. Yan, Y. Yuan, and X. Li. Single-image super-resolution via local learning. Int. J. Mach. Learn. Cybern., vol. 2, no. 1, pp. 15–23, Mar. 2011.

[16] X. Gao, K. Zhang, D. Tao, and X. Li. Image super-resolution with sparse neighbor embedding. IEEE Trans. Image Process., vol. 21, no. 7, pp. 3194–3205, Jul. 2012

[17] K. I. Kim and Y. Kwon. Example-based learning for single-image super-resolution. in Proc. DAGM, 2008, pp. 456–465.

[18] J. Yang, T. Huang. Image super-resolution: historical overview and future challenges. In "Super-resolution imaging", CRC Press, 2010.

[19] D. Capel and A. Zisserman. Computer vision applied to super-resolution. IEEE Signal Processing Magazine, 20(3):75-86, 2003.

[20] Michael E. Tipping and Christopher M. Bishop. Bayesian image super-resolution. In Proceedings of Advances in Neural Information Proceeding Systems, pages 1279-1286, 2003.

[21] L. C. Pickup, D. P. Capel, S. J. Roberts, and A. Zisserman. Bayesian image super-resolution, continued. In Proceedings of Advances in Neural Information and Proceedings Systems, pages 1089-1096, 2006.

[22] L. C. Pickup, D. P. Capel, S. J. Roberts, and A. Zisserman. Bayesian methods for image super-resolution. The Computer Journal, 52(1):101-113, 2009.

[23] L. C. Pickup, S. J. Robert, and A. Zisserman. A sampled texture prior for image super-resolution. In Proceedings of Advances in Neural Information and Processing System, pages 1587-1594, 2003.

[24] M. Elad and D. Datsenko. Example-based regularization deployed to super-resolution reconstruction of a single image. The Computer Journal, 52(1):15-30, 2007.

[25] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing, vol. 15, pp. 3736–3745, 2006.

[26] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for

image and video restoration. Multiscale Modeling and Simulation, vol. 7, pp. 214–241, 2008.

[27] D. L. Donoho. Compressed sensing. IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, 2006.

[28] D. L. Donoho. For most large underdetermined systems of linear equations, the minimal L1-norm solution is also the sparsest solution. Communications on Pure and Applied Mathematics, vol. 59, no. 6, pp.797–829, 2006.

[29] D. L. Donoho. For most large underdetermined systems of linear equations, the minimal L1-norm near-solution approximates the sparsest near-solution. Communications on Pure and Applied Mathematics, vol. 59, no. 7, pp.907–934, 2006.

[30] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1. Vision Research, vol. 37, no. 23, pp.3311–3325, 1997.

[31] J. F. Murray and K. Kreutz-Delgado. Learning sparse overcomplete codes for images. The Journal of VLSI Signal Processing, vol. 45, pp.97–110, 2007.

[32] M. Aharon, M Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. IEEE Transactions on Signal Processing, Vol. 54, No. 11, November 2006

[33] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super-resolution. IEEE Transactions on Image Processing, vol. 13, pp. 1327–1344,

2004.

[34] M. E. Tipping and C. M. Bishop. Bayesian image super-resolution. in Advances in Neural Information and Processing Systems 16 (NIPS),2003.

[35] J. Sun, Z. Xu, and H. Shum. Image super-resolution using gradient profile prior. in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.

[36] A. Singh and N. Ahuja. Super-resolution using sub-band self-similarity. In ACCV, 2014.

[37] A. Singh, F. Porikli, and N. Ahuja. Super-resolving noisy images. In CVPR, 2014.

[38] J. Sun, Z. Xu, and H.-Y. Shum. Gradient profile prior and its applications in image super-resolution and enhancement. IEEE TIP, 20(6):1529–1542, 2011.

[39] J. Sun, J. Zhu, and M. Tappen. Context-constrained hallucination for image super-resolution. In CVPR, 2010.

[40] M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In 3rd International Workshop on Statistical and Computational Theories of Vision, 2003.

[41] L. Sun and J. Hays. Super-resolution from internet-scale scene matching. In ICCP, 2012.

[42] R. Timofte, V. De, and L. V. Gool. Anchored neighborhood regression for fast example-based super-resolution. In ICCV, 2013.

[43] Y. Zhu, Y. Zhang, and A. L. Yuille. Single image superresolution using deformable patches. In CVPR, 2014.

[44] M. Zontak and M. Irani. Internal statistics of a single natural image. In CVPR, 2011. 1

[45] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. on Graphics, 28(3):24, 2009.

[46] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In ECCV, 2010.

[47] M. Barnsley. Fractals Everywhere. Academic Press Professional, Inc., 1988.

[48] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereostereo matching with slanted support windows. In BMVC, 2011.

[49] M. Ebrahimi and E. Vrscay. Solving the inverse problem of image zooming using self-examples. In International Conference on Image Analysis and Recognition, pages 117–130, 2007.

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

[51] O. Chum and J. Matas. Planar affine rectification from change of scale. In ACCV, 2010.

[52] K. Huang and S. Aviyente. Sparse representation for signal classification. NIPS,

vol. 19, pp. 609–616, 2007.

[53] M. Ranzato, F. Haung, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1–8, 2007.

[54] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. Advances in Neural Information Processing Systems, Vancouver, B.C., Canada 2006.

[55] J. Mairal, F. Bach, J. Pnce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. Proc. of the Conference on Computer Vision and Pattern Recognition, Anchorage, AL, June 2008.

[56] J. Mairal, M. Leordeanu, F. Bach, M. Herbert, and J. Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. Proc. of the European Conference on Computer Vision, 2008.

[57] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. International Conference on Machine Learning, Montreal, Canada, June 2009.

[58] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. Advances in Neural Information Processing Systems, Vancouver, B.C., Canada, Dec. 2008.

[59] Q. Qiu, Z. Jiang, and R. Chellappa. Sparse dictionary-based representation and

recognition of action attributes. International Conference on Computer Vision, 2011.

[60] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, june 2010, pp. 2691 –2698.

[61] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, june 2011, pp. 1697 –1704.

[62] S. Baker and T. Kanade. Limits on super-resolution and how to break them. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(9):1167-1183, 2002.

[63] S. Borman and R. L. Stevenson. Simultaneous multi-frame MAP super-resolution video enhancement using spatio-temporal priors. In Proceedings of IEEE International Conference on Image Processing, volume 3, pages 469-473, 1999.

[64] Sean Borman and Robert L. Stevenson. Super-resolution from image sequences - A review. In Proceedings of the 1998 Midwest Symposium on Circuits and Systems, pages 374-378, 1998.

[65] N. K. Bose, H. C. Kim, and H. M. Valenzuela. Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframes. In Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing, volume 5, pages 269-272, 1993.

[66] O. Bowen and C. S. Bouganis. Real-time image super resolution using an FPGA.

In International Conference on Field Programmable Logic and Applications, pages 89-94, 2008.

[67] L. Brown. A survey of image registration techniques. ACM Computing Surveys, 24(4):325-376, 1992.

[68] A. Buades, B. Coll, and J. M. More. A non-local algorithm for image denoising. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 60-65, 2005.

[69] H. Chang, D. Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 275-282,2004.

[70] M. C. Chiang and T. E. Boulte. Efficient super-resolution via image warping. Image and Vision Computing, 18(10):761-771, 2000.

[71] J. Chung, E. Haber, and J. Nagy. Numerical methods for coupled super-resolution. Inverse Problems, 22(4):1261-1272, 2006.

[72] Marco Crisani, Dong Seon Cheng, Vittorio Murino, and Donato Pannullo. Distilling information with super-resolution for video surveillance. In Proceedings of the ACM 2nd International Workshop on Video Surveillance and Sensor Networks, pages 2-11, 2004.

[73] D. Datsenko and M. Elad. Example-based single document image super-resolution: a global MAP approach with outlier rejection. Multidimensional System

and Signal Processing, 18(2-3):103-121, 2007.

[74] K. M. Hanson and G. W.Wecksung. Bayesian approach to limited-angle reconstruction in computed tomography. Journal of Optical Society of America, 73(11):1501-1509, 1983.

[75] R. Hardie. A fast image super-resolution algorithm using an adaptive Wiener filter. IEEE Transactions on Image Processing, 16(12):2953-2964, 2007.

[76] R. C. Hardie, K. J. Barnard, and E. E. Armstrong. Join MAP registration and high resolution image estimation using a sequence of undersampled images. IEEE Transactions on Image Processing, 6(12):1621-1633,1997.

[77] P. H. Hennings-Yeomans, S. Baker, and B. V. K. V. Kumar. Simultaneous super-resolution and feature extraction for recognition of low-resolution faces. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 1-8, 2008.

[78] G. T. Herman, H. Hurwitz, A. Lent, and H-P. Lung. On the Bayesian approach to image reconstruction. Information and Control, 42(1):60-71, 1979.

[79] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In Proceedings of the 28th annual conference Computer Graphics and interactive techniques, pages 327-340, 2001.

[80] L. C. Pickup, S. J. Roberts, and A. Zisserman. Optimizing and learning for super-resolution. In British Machine Vision Conference, volume 2, pages 439–448, 2006.

[81] M. Protter and M. Elad. Super resolution with probabilistic motion estimation. IEEE Transactions on Image Processing, 18(8):1899–1904, 2009.

[82] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the nonlocal-means to super-resolution reconstruction. IEEE Transactions on Image Processing, 18(1):36–51, 2009.

[83] D. Robinson and P. Milanfar. Fundamental performance limits in image registration. IEEE Transactions on Image Processing, 13(9):1185–1199, 2004.

[84] D. Robinson and P. Milanfar. Statistical performance analysis of superresolution. IEEE Transactions on Image Processing, 15(6):1413–1428, 2006.

[85] L. Rudin, S. Osher, and E. Fatemi. Nonlinaer total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60(1-4):259–268, 1992.

[86] M. Elad S. Farsiu, D. Robinson and P. Milanfar. Advances and challenges in super-resolution. International Journal of Imaing Systems and Technology, 14(2):47–57, 2004.

[87] R. R. Schultz and R. L. Stevenson. A Bayesian approach to image expansion for improved definition. IEEE Transactions on Image Processing, 3(3):233–242, 1994.

[88] R. R. Schultz and R. L. Stevenson. Extraction of high-resolution frames from video sequences. IEEE Transactions on Image Processing, 5(6):996–1011, 1996.

[89] C. A. Segall, A. K. Katsaggelos, R. Molina, and J. Mateos. Bayesian resolution enhancement of compressed video. IEEE Transactions on Image Processing,

13(7):898–910, 2004.

[90] C. A. Segall, R. Molina, and A. K. Katsaggelos. High resolution images from low-resolution compressed video. IEEE Signal Processing Magazine, 20(3):37–38, 2003.

[91] H. Shen, L. Zhang, B. Huang, and P. Li. A MAP approach for joint motion estimation, segmentation and super-resolution. IEEE Transactions on Image Processing, 16(2):479–490, 2007

[92] H. Stark and P. Oskoui. High-resolution image recovery from image plane arrays, using convex projections. Journal of Optical Society of America A, 6(11):1715–1726, 1989.

[93] W. Su and S. P. Kim. High-resolution restoration of dynamic image sequences. International Journal of Imaging Systems and Technology, 5(4):330–339, 1994.

[94] J. Sun, N. N. Zheng, H. Tao, and H. Shum. Image hallucination with primal sketch priors. In Proceedings of IEEE Computer Society Conferece on Computer Vision and Pattern Recognition, volume 2, pages 729–736, 2003.

[95] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. IEEE Transactions on Image Processing, 16(2):349–366, 2007.

[96] H. Takeda, P. Milanfar, M. Protter, and M. Elad. Super-resolution without explicit subpixel motion estimation. IEEE Transaction on Image Processing, 18(9):1958–1975, 2009.

[97] J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for

nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 2000.

[98] R. Tibshirani. Regression shrinkage and selection via the Lasso. Journal of Royal Statistical Society: Series B (Statistical Methodology), 59(1):267–288, 1996.

[99] A. N. Tikhonov and V. A. Arsenin. Solution of ill-posed problems. Winston & Sons, Washington, 1997.

[100] Q. Wang, X. Tang, and H. Shum. Patch based blind image superresolution. In Proceedings of IEEE International Conference on Computer Vision, volume 1, pages 709–716, 2005.

[101] X. Wang and X. Tang. Hallucinating face by Eigen transformation. IEEE Transactions on Systems, Man, and Cybernetics, 35(3):425–434, 2003.

[102] N. A. Woods, N. P. Galatsanos, and A. K. Katsaggelos. Stochastic methods for joint registration, restoration and interpolation of multiple undersampled images. IEEE Transactions on Image Processing, 15(1):210–213, 2006.

[103] Jianchao Yang, Hao Tang, Yi Ma, and Thomas Huang. Face hallucination via sparse coding. In Proceedings of IEEE International Conference on Image Processing, pages 1264–1267, 2008.

[104] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image superresolution as sparse representation of raw image patches. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 1–8, 2008.

[105] G. Freedman and R. Fattal. Image and video upscaling from local self-examples.

ACM Trans. on Graphics, 30(2):12, 2011

[106] M. Hornacek, C. Rhemann, M. Gelautz, and C. Rother. Depth super resolution by rigid body self-similarity in 3d. In CVPR, 2013.

[107] S. Gould. Multiclass Pixel Labeling with Non-Local Matching Constraints. CVPR, 2012 IEEE Conference on, pages 2783 – 2790

[108] S. Gould, J. Rodgers, D. Cohen, G. Elidan, D. Koller. Multi-Class Segmentation with Relative Location Prior. International Journal of Computer Vision, Volume 80 Issue 3, December 2008, Pages 300 - 316

[109] http://drwn.anu.edu.au/drwnProjMultiSeg.html

[110] S. Gould, R. Fulton, and D. Koller. Decomposing a Scene into Geometric and Semantically Consistent Regions." In ICCV, 2009.

[111] D. Comaniciu, P. Meer, and S. Member. Mean shift: A robust approach toward feature space analysis. PAMI, 2002.

[112] A. Criminisi. Microsoft research cambridge object recognition image database. http://research.microsoft.com/vision/cambridge/recognition, 2004.

[113] Qian, N. (1999). On the momentum term in gradient descent learning algorithms. Neural Networks: The Official Journal of the International Neural Network Society, 12(1), 145–151.

[114] Mitchell, Don P. Netravali, Arun N. (August 1988). Reconstruction filters in computer-graphics. ACM SIGGRAPH International Conference on Computer

Graphics and Interactive Techniques. pp. 221–228

[115] Mertz, Pierre, and Frank Grey, "A Theory of Scanning and its Relation to the Characteristics of the Transmitted Signal in Telephotography and Television," Bell System Tech. J., Vol. 13, pp. 464-515, July 1934.

VITA

Min Zhang
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529

Min Zhang received a Bachelor of Science with a major in computer science from Guangxi University, China in July 2005. She enrolled in North Carolina Central University for higher education in 2008, and she received a Master of Science with a major in computer science in December 2010. In August 2012, Min enrolled in Old Dominion University, and she completed her Doctor of Philosophy in electrical and computer engineering in December 2016.