LEARNING TO SUPER-RESOLVE IMAGES USING SELF-SIMILARITIES

BY

ABHISHEK SINGH

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor Emeritus Narendra Ahuja, Chair
Professor David A. Forsyth
Professor Minh N. Do
Professor Mark A. Hasegawa-Johnson
ABSTRACT

The single image super-resolution problem entails estimating a high-resolution version of a low-resolution image. Recent studies have shown that high resolution versions of the patches of a given low-resolution image are likely to be found within the given image itself. This recurrence of patches across scales in an image forms the basis of self-similarity driven algorithms for image super-resolution. Self-similarity driven approaches have the appeal that they do not require any external training set; the mapping from low-resolution to high-resolution is obtained using the cross scale patch recurrence. In this dissertation, we address three important problems in super-resolution, and present novel self-similarity based solutions to them: First, we push the state-of-the-art in terms of super-resolution of fine textural details in the scene. We propose two algorithms that use self-similarity in conjunction with the fact that textures are better characterized by their responses to a set of spatially localized bandpass filters, as compared to intensity values directly. Our proposed algorithms seek self-similarities in the sub-bands of the image, for better synthesizing fine textural details. Second, we address the problem of super-resolving an image in the presence of noise. To this end, we propose the first super-resolution algorithm based on self-similarity that effectively exploits the high-frequency content present in noise (which is ordinarily discarded by denoising algorithms) for synthesizing useful textures in high-resolution. Third, we present an algorithm that is able to better super-resolve images containing geometric regularities such as in urban scenes, cityscapes etc. We do so by extracting planar surfaces and their parameters (mid-level cues) from the scene and exploiting the detected scene geometry for better guiding the self-similarity search process. Apart from the above self-similarity algorithms, this dissertation also presents a novel edge-based super-resolution algorithm that super-resolves an image by learning from training data how edge profiles transform across resolutions. We obtain edge profiles via a detailed and explicit examination of local image structure, which we show to be more robust and accurate as compared to conventional gradient profiles.
To my family, for their eternal love, encouragement and support.

May it always see them in health and happiness.
ACKNOWLEDGMENTS

I would first like to thank my advisor, Prof. Narendra Ahuja for his guidance throughout the course of my doctoral studies. It was a privilege to be under his esteemed tutelage at this renowned institution. I am grateful for the freedom and encouragement he gave me to pursue the ideas that I proposed. I would also like to thank Profs. David Forsyth, Minh Do and Mark Hasegawa-Johnson for serving on my doctoral committee.

I am thankful for having a wonderful family. This dissertation would not have been possible without their invaluable support. Their tremendous faith in me is only bettered by my love and gratitude toward them. I pride myself in being their son, son-in-law and brother. A special note of gratitude to my wife, Padmini, for being a great pillar of strength through thick and thin. I thank her for her patience in enduring years of separation while I carried out my research. I write this with a promise to make it all worthwhile.

I made some wonderful friends and came across many brilliant people during my stay on campus. I am grateful to each one of them for being an important part of my experience here. Apart from learning many things from them, I take back fond memories of fun-filled times together. I am also thankful to the wonderful staff of the Electrical and Computer Engineering Department and the Beckman Institute for making my time at the University of Illinois comfortable and hassle-free.

I gratefully acknowledge the Joan and Lalit Bahl Fellowship and the Computational Science and Engineering Fellowship for supporting me. I also acknowledge the US National Science Foundation and the Office of Naval Research for supporting parts of this work.

Finally, I thank the Lord Almighty for all His blessings.
# TABLE OF CONTENTS

## CHAPTER 1 INTRODUCTION ....................................... 1
  1.1 The Single Image Super-Resolution Problem .................. 1
  1.2 Learning-Based Approaches .................................. 2
  1.3 Self-Similarity-Based Approaches ............................ 5
  1.4 Internal vs. External SR ...................................... 7
  1.5 Contributions and Organization of This Dissertation .......... 9

## CHAPTER 2 SUB-BAND SELF-SIMILARITY .......................... 13
  2.1 Introduction .............................................. 13
  2.2 Overview of Proposed Method ............................... 14
  2.3 Sub-Band Self-Similarity .................................... 16
  2.4 Implications .............................................. 17
  2.5 Experiments and Results ................................. 21
  2.6 Conclusion .............................................. 31

## CHAPTER 3 SUB-BAND ENERGY CONSTRAINTS FOR SUPER-RESOLUTION .................................................. 32
  3.1 Introduction .............................................. 32
  3.2 Algorithm Overview ....................................... 33
  3.3 Self-Similarity-Based SR .................................. 35
  3.4 Analysis of Sub-Band Energies ............................... 36
  3.5 Self-Learning of Sub-Band Constraints ...................... 39
  3.6 Backprojection Constraint ................................ 40
  3.7 Results .................................................. 40
  3.8 Conclusion .............................................. 45

## CHAPTER 4 SUPER-RESOLUTION OF NOISY IMAGES .................. 47
  4.1 Introduction .............................................. 47
  4.2 Related Work ............................................ 50
  4.3 Proposed Model ........................................... 52
  4.4 Spatial Constraint ....................................... 55
  4.5 Frequency Domain Constraint ............................... 60
  4.6 Global Kurtosis Constraint ................................ 62
  4.7 Texture Adaptive Patch Averaging Perspective .............. 62
  4.8 Results .................................................. 65
CHAPTER 1
INTRODUCTION

1.1 The Single Image Super-Resolution Problem

The traditional super-resolution (SR) problem involved fusing together multiple low-resolution (LR) images of a scene, displaced by sub-pixel amounts with respect to each other, to obtain a high-resolution (HR) image of the scene [1]. This problem has been well studied by several authors\(^1\) since the pioneering work of Tsai and Huang [2]. A more practical and challenging variant of the traditional SR problem is the single image SR problem, which has been the subject of recent research in this domain and is the focus of this dissertation. The single image SR problem entails the estimation of an HR image, from a single given LR version of the scene. Since a large number of unknown pixel values need to be estimated from a fewer number of observed pixels, this SR problem is highly ill-posed even for moderate upscaling factors. Choosing appropriate priors or regularizers are therefore an important component in addressing this problem.

Perhaps the simplest priors are those which assume simple models for image smoothness (such as linear or cubic). Super-resolution then simply amounts to interpolation of the patch pixels according to the chosen model to obtain the sub-pixel values [3, 4, 5]. However, such methods tend to produce overly smooth results, and tend to produces artifacts such as chessboard effect along edges. A popular class of priors that are aimed at preserving sharpness are those which impose constraints on the marginal distributions of filterbank responses of the image [6, 7]. Studies on statistical properties of natural images have found that these distributions are well modeled as Laplacians [6] or generalized Gaussians [7]. The constraints therefore occur in terms of fits of these distribution types to the data at hand. These priors, however, are used as a global constraint over the

---

\(^1\)A complete review of the multi-image or multi-frame SR problem is beyond the scope of this dissertation, but we refer the reader to [1] for further reading.
entire image. Spatial localization is incorporated only weakly at best [8].

A more recent and highly successful approach for addressing the single image SR problem is data driven or learning-based approach. In general, these methods aim at predicting the HR image corresponding to a given LR image by learning the LR-HR mapping using a training database of LR-HR image pairs. A related class of approaches aims at learning this LR-HR relationship using self-similarities. That is, the LR-HR training database is constructed using scaled-down versions of the given LR image itself. In the rest of this introductory chapter, we elaborate more on learning-based and self-similarity-based approaches and discuss their relative advantages and drawbacks. We then describe the contributions of this dissertation and how the they are organized in the subsequent chapters.

1.2 Learning-Based Approaches

Learning-based SR algorithms typically involve first collecting a set of HR training images, each of which are downsampling to yield a corresponding set of LR images. Pairs of LR-HR patches are then extracted from this image set to create a database or dictionary of LR-HR training image patches. Different learning algorithms are then employed for learning the LR to HR mapping using this patch database. Figure 1.1 illustrates the general framework for learning-based SR.

Freeman and Pasztor [9, 10] pioneered the idea of using training examples for the SR problem, by using a simple nearest neighbor search strategy. Given an LR patch to be super-resolved, its nearest neighbor is searched for in the LR training patch database. The corresponding HR patch of this nearest neighbor is deemed to be the HR version of the given LR patch. Several search/prediction algorithms have been employed since then for making the HR prediction. Manifold learning has been used in [11], wherein the manifold of training patches is assumed to be locally linear, and a given LR patch is expressed as a linear combination of its neighboring patches in the LR dictionary. LR patches are expressed as a sparse combination of training examples in [12]. The sparse approximation approach is further extended to the case where instead of using raw image patches in the training dictionary, the dictionary is itself learned using sparsity constraints [13, 14]. More recently, approaches have been proposed that first pre-cluster the training dataset, and learn relatively simpler LR-HR mapping functions for each cluster. An LR test patch is first matched to a cluster, and then the associated prediction
function is employed to obtain its HR version [15, 16, 17]. Convolutional neural networks have also been proposed to learn the LR-HR mapping in [18].

Learning-based single image SR algorithms have become popular in the image processing and vision communities over the past few years. They have yielded significant improvement in results when compared to older techniques. Their success can also be partly attributed to the recent advances in large-scale learning algorithms that have enabled the use of large training databases that allow for training more complex models for better prediction on test samples.

1.2.1 Limitations

While learning-based approaches have become quite popular, they do suffer from some important drawbacks:

1. Perhaps the most important limitation of learning from external databases is that the training dataset may not contain enough relevant patches. We illustrate this with an example in Fig. 1.2. An outdoor natural scene as in Fig. 1.2(a) will not be super-resolved well with a training database of urban, man-made scenes such as in Fig. 1.2(b). To partially overcome this
Figure 1.2: (a) An outdoor natural scene to be super-resolved. (b) Training database consisting of mostly urban, man-made scenes. Such a database would not contain enough relevant patches for effectively super-resolving the image in (a).

problem of not having enough relevant training samples for super-resolving any given test image, recent methods have resorted to collecting very large training databases, covering a variety of different scenes [18, 17]. While this may improve the quality of results, the following problems also arise:

2. The use of large external training databases often entails employing elaborate and computationally expensive learning algorithms.

3. The need for having large training databases also limits the applicability of the SR algorithms, particularly in terms of portability on mobile or remote applications that have memory and computational constraints.

To overcome these problems associated with external training databases, a number of related SR algorithms have been proposed that are based on self-similarity. These algorithms do not require any external training database, but are rather
based on redundancy of image patches across scales. We elaborate on this class of algorithms in Section 1.3.

1.3 Self-Similarity-Based Approaches

Self-similarity-based approaches find their roots in fractal image coding from the 1990s [19, 20], and are driven by the fact that images tend to have redundant patches. More specifically, self-similarity methods are based on the idea that patches of an image tend to recur within the same image, not only in the same scale, but also across scales. This is illustrated with an example in Fig. 1.3, which shows the red patch recurring in the image at the same scale, and the green patch recurring across scales of the image [21]. For SR, the type of patches that are of interest to us are the green patches that recur across scales. This cross-scale recurrence suggests that the HR version of a patch from an image also exists within the same image. This gives rise to a powerful statistical prior that can be exploited for super-resolution without the need of an external training database [21]. The general principle involved in such a self-similarity-based SR algorithm consists of the following three steps, also illustrated in Fig. 1.4.

1. Given a patch from an input image, its best match is first found in a coarser version of the input image.

2. The HR version of this best match is then extracted from the original input image.

3. This HR patch is deemed to be the HR estimate of the input patch, and is placed at the original input patch location, in the HR grid.

Comparing Figs. 1.4 and 1.1, we see that while learning-based approaches use an external training dictionary of LR-HR patches, self-similarity-based approaches can be thought of as using an internal dictionary of LR-HR patches, that are extracted from scaled-down version(s) of the given image itself. In light of this distinction, henceforth, we refer to self-similarity-based approaches as internal dictionary-based approaches also, and the traditional learning-based approaches as external dictionary-based approaches.

Ebrahimi and Vrscay [22] proposed the first self-similarity-based SR algorithm by combining ideas from fractal coding [19, 20] and example-based algorithms
Figure 1.3: Patches in an image tend to recur within the same scale, as well as across scales of the image. Figure taken from [21].

(such as non-local means filtering [23]). Glasner et al. [24] fuse together multiple matched patches from the internal dictionary of the image to generate HR patches, in a way similar to traditional multiframe SR. Freedman and Fattal [25] show that patches tend to recur across scales within local spatial neighborhoods, which they exploit for computational speed-up. Michaeli and Irani [26] used self-similarity to not only super-resolve the image, but also recover the optimal blur kernel or point spread function of the downsampling process.

Clearly, the biggest advantage of self-similarity-based approaches is not having to use an external training dictionary. It has been shown that the internal dictionaries used by self-similarity methods tend to contain more relevant training patches, and, in general, yield nearest neighbor matches with lower error as compared to external dictionaries [21]. In terms of results, therefore, self-similarity methods often outperform, or are very comparable to external dictionary-based approaches, for many types of images.

1.3.1 Limitations

While existing internal dictionary-based methods overcome the dependency of learning-based methods on external training databases, they do have some drawbacks of their own:

1. The size of the internal dictionary used in the self-similarity methods depends on the size of the input image. Small input images consist of fewer
patches, and therefore the size of the internal dictionary yielded by them might be extremely small. Small dictionaries are, in general, not expressive enough. Self-similarity approaches therefore tend to suffer while super-resolving very small images since the small internal dictionaries yielded by the images may not contain sufficiently good patch matches.

2. While the patch self-similarity property of natural images has been well justified by several statistical experiments, it is also known (and stands to reason) that as the complexity of the patches increases, the quality of self-similar patch matches falls. This suggests that fine textural details like hair, fur, etc., often do not find sufficiently good matches in the internal dictionary. Indeed, we demonstrate this with an example in Fig. 1.5, that shows the patch-matching error for each location in the input image. We see that textures like the fur tend to find matches with high error. This suggests that such details tend to be smoothed out by self-similarity methods.

1.4 Internal vs. External SR

We now summarize the relative advantages and disadvantages of internal vs. external dictionary-based SR methods. The important distinctions between these
Figure 1.5: (a) An input image containing fine textural details. (b) Error map showing the nearest neighbor matching error while searching for patch matches in the internal search space. Textural details often fail to find low-error matches in the internal dictionary.

classes of methods can be stated in the following five points:

1. Perhaps the biggest and most obvious advantage of internal dictionary-based methods over external SR methods is in not requiring any external training database. Not requiring external training images is advantageous in terms of ease of implementation and portability on practical systems that are constrained on memory resources.

2. Another equally important advantage of internal SR methods is that the internal dictionary generated by them generally consists of more relevant patches, when compared to an arbitrary external database. This generally leads to better results when compared to external SR methods.

3. Internal SR methods also have the advantage of not requiring expensive and cumbersome training procedures on large training sets.

4. Internal SR methods tend to suffer while super-resolving small images, since small images yield small internal dictionaries which may not be expressive enough. On the other hand, external SR methods use a database that can be as large as desired.

5. Patches containing textural details tend to find poor self-similar matches in internal SR methods, and therefore tend to get smoothed out. On the other hand, since external SR methods can have an arbitrarily large training set, textural patches are likely to find better matches, given a large enough dictionary.
1.5 Contributions and Organization of This Dissertation

The primary contribution of this dissertation is to present self-similarity-based SR algorithms that push the state-of-the-art in SR along three unique directions. More specifically, we propose multiple self-similarity-based SR algorithms that are respectively able to (1) better synthesize textural details in the scene (thereby overcoming the drawbacks in items 4 and 5 listed in Section 1.4), (2) jointly super-resolve as well as denoise noisy LR inputs, and, (3) better exploit geometric regularities present in scenes containing urban environments, man-made structures, etc. In the next three subsections, we elaborate on each of these three contributions and introduce the chapters that discuss them in detail.

Lastly, moving away from self-similarity approaches, this dissertation also presents a novel edge or segmentation driven SR algorithm that improves over existing edge-based approaches [27]. We present a summary of this algorithm in Subsection 1.5.4, and present the details in Chapter 6.

1.5.1 Improving Synthesis of Stochastic Textures

As described earlier, self-similarity-based approaches tend to smooth out fine textural details in the super-resolved results. We present two algorithms that address this problem in different ways as summarized in the following.
In Chapter 2, we propose a self-similarity-based SR algorithm, that, instead of seeking self-similar patches directly in the image domain, uses the self-similarity principle independently on each of a set of different sub-band images. These sub-band images are obtained using a bank of orientation selective band-pass filters. Therefore, we allow the different directional frequency components of a patch to find matches independently, which may be in different image locations. Essentially, we decompose local image structure into component patches defined by different sub-bands, with the following advantages: (1) The sub-band image patches are simpler and therefore easier to find matches, than for the more complex textural patches from the original image. (2) The size of the dictionary defined by patches from the sub-band images is exponential in the number of sub-bands used, thus increasing the effective size of the internal dictionary. (3) As a result, our algorithm exhibits a greater degree of invariance to parameters like patch size and the dimensions of the LR image. We demonstrate these advantages and show that our results are richer in textural content and appear more natural than several state-of-the-art methods.

In Chapter 3, we discuss the metric or the matching criterion used in searching for patch matches in the internal dictionary. We argue that metrics like pixel-wise sum of squared differences ($L_2$ distance) make it difficult to find matches for high-frequency textured patches in the internal dictionary. The matching criterion is therefore another reason why textural details are often smoothed out in the final image. In this chapter, we propose a method to compensate for this loss of textural detail. Our algorithm uses the responses of a bank of orientation-selective band-pass filters to represent texture, instead of using the spatial variation of intensity values directly. Specifically, we use the energies contained in different sub-bands of an image patch to separate different types of details of a texture, which we then impose as additional priors on the patches of the super-resolved image. Our experiments show that for each patch, the low-energy sub-bands (which correspond to fine textural details) get severely attenuated during conventional $L_2$ distance-based SR. We propose a method to learn this attenuation of sub-band energies in the patches, using scaled-down version(s) of the given image itself (without requiring external training databases), and thus propose a way of compensating for the energy loss in these sub-bands. We demonstrate that as a consequence, our SR results appear richer in texture and closer to the ground truth as compared to several other state-of-the-art methods.
1.5.2 Handling Noise in Input Images

In Chapter 4, our goal is to obtain a noise-free, high-resolution (HR) image, from an observed, noisy, low-resolution (LR) image. Conventional approaches for handling noise typically simply preprocess the image with a denoising algorithm, and then apply a super-resolution (SR) algorithm. However, such a processing framework has an important limitation: Along with noise, some high-frequency content of the image (particularly textural detail) is invariably lost during the denoising step. This “denoising loss” restricts the performance of the subsequent SR step, wherein the challenge is to synthesize such textural details. In this chapter, we show that high-frequency content in the noisy image (which is ordinarily removed by denoising algorithms) can be effectively used to obtain the missing textural details in the HR domain. To do so, we first obtain HR versions of both the noisy and the denoised images, using a patch-similarity-based SR algorithm. We then show that by taking a convex combination of orientation and frequency selective bands of the noisy and the denoised HR images, we can obtain a desired HR image where (i) some of the textural signal lost in the denoising step is effectively recovered in the HR domain, and (ii) additional textures can be easily synthesized by appropriately constraining the parameters of the convex combination. We show that this part-recovery and part-synthesis of textures through our algorithm yields HR images that are visually more pleasing than those obtained using the conventional processing pipeline. Furthermore, our results show a consistent improvement in numerical metrics, further corroborating the ability of our algorithm to recover lost signal.

1.5.3 Exploiting Geometric Regularity in Urban Scenes

In Chapter 5, we propose a self-similarity-based SR algorithm that expands the internal patch search space (internal dictionary) by allowing geometric transformations of the patches. We do so by explicitly localizing planes and planar surfaces that exist in several urban and man-made scenes such as cityscapes. After identifying planar surfaces, the detected perspective geometry is used to geometrically transform patches. Effectively we use the planar structures present in the scene to guide the patch search process. We also incorporate additional affine transformations to accommodate local shape variations. We propose a compositional model to simultaneously handle both these types of transformations.
extensively evaluate the performance of this method in urban scenes. In spite of not using any external training database, we achieve significantly superior results on urban scenes as compared to other state-of-the-art SR algorithms.

1.5.4 A New Edge-Based SR Method

In Chapter 6, we propose a new edge-based SR algorithm, that makes use of explicitly identified image structure. We treat the image as a layout of homogeneous regions, surrounded by ramp edges of a larger contrast. The SR problem is then viewed primarily as one of super-resolving these ramps, since the relatively homogeneous interiors can be handled using simpler methods. Our approach involves learning how these ramps transform across resolutions. Ramps are characterized by 1D intensity profiles across them, derived from sequences of monotonically increasing (or decreasing) intensity values along multiple directions through the ramp pixels. Conventional edge-based SR methods are based on gradients, which use different filters with heuristically chosen parameters and these choices result in different gradient values. This sensitivity gets amplified when learning gradient domain correspondences across different resolutions. We show that ramp profiles are more adaptive, stable and therefore reliable representations for learning edge transformations across resolutions. Additionally, existing gradient-based SR methods are often unable to sufficiently constraint the brightness levels in the intensity domain. Our approach on the other hand, operates directly in the image intensity domain, jointly enforcing sharpness and color consistency. Unlike previous gradient-based methods, we also explicitly incorporate dependency between closely spaced edges while learning ramp correspondences. This allows for better recovery of contrast across thin structures such as in high spatial frequency areas. We obtain results that are sharper and more faithful to the true image color, and show almost no ringing artifacts.
CHAPTER 2

SUB-BAND SELF-SIMILARITY

2.1 Introduction

Self-similarity-based methods have limitations while super-resolving textural regions. Indeed, [21] shows that the likelihood of finding a good internal match for a patch decreases as the gradient content of the patch increases. This suggests that textural details like hair, animal fur etc. often find suboptimal matches, using a self-similarity approach, and are thus averaged or smoothed out in the final SR result. One reason behind such a limitation is that the internal dictionary obtained from the given image generally has fewer number of LR-HR patch pairs than external dictionaries, which can potentially be as large as desired. Due to the limited size of the internal dictionary, textural patches (which contain complex structures) fail to find suitable representations. The size of the self-learned dictionary furthermore depends on the dimensions of the given image; smaller images consist of fewer patches and thereby yield fewer LR-HR patch pairs. Additionally, the quality of matches depends on the patch size chosen. For example, the complexity of structures in the patches increases with increase in patch size, making it difficult to find accurate matches.

In this chapter, we propose an SR algorithm that alleviates the abovementioned problems of self-similarity-based approaches, without resorting to any external training database. We propose a self-similarity driven algorithm wherein, instead of seeking self-similar patches directly in the image domain, we use self-similarity-based SR independently on images corresponding to different sub-bands. These sub-bands are the responses of the image to a bank of spatially localized, orientation selective band-pass filters. Effectively, we unravel the complexity of the structure by representing it in terms of simpler components, which, being simpler, are easier to find matches for. Unlike in the case of patch matching in the image domain, we allow the different directional frequency components of the
patch to independently find their best matches in different locations in the image. Therefore, we synthesize HR patches by combining different frequency components from the best matches found at different locations. Such a combinatorial expansion of the internal dictionary allows for finding better (lower error) patch matches for a test image produces a better quality HR image. Our SR results appear richer in texture and more natural than those produced by state-of-the-art methods. We also show that our algorithm leads to improvements in two other important aspects of the SR problem that have not received much attention in the past. We show that our approach has a greater degree of invariance to the choice of patch size, which can be a sensitive parameter, particularly for self-similarity methods. We also show that due to the ability of our algorithm to generate richer internal dictionaries, we are able to super-resolve extremely small images much better, thereby achieving greater invariance to the size of the input image, as compared to the existing self-similarity approach.

In Sections 2.2 and 2.3, we describe the steps involved in our algorithm, which is conceptually quite straightforward and easy to implement. In Section 2.4, we discuss a number of important implications and corollaries resulting out of the proposed algorithm, and discuss the key advantages it brings over existing schemes. We demonstrate our performance vis-a-vis several other state-of-the-art methods, and corroborate our claims through a number of systematic experiments in Section 2.5.

### 2.2 Overview of Proposed Method

**Notation:** We denote the given image to be super-resolved as $I_0$. By $I_1$ we denote the HR version of $I_0$, whose linear dimension, or scale, is larger by a factor of $s$. Similarly, we denote by $I_{-1}$, the smaller version of $I_0$, by the scaling factor of $1/s$. We denote the super-resolved image(s) obtained using our algorithm using a hat ($\hat{}$) symbol. Therefore, our objective is to super-resolve $I_0$ to obtain an HR image $\hat{I}_1$, that best approximates the true HR image $I_1$. We use scripted letters to denote sets, we use lowercase boldface letters to denote image patches, and lowercase italicized letters to denote scalars and indices.
Figure 2.1: (a) Conventional self-similarity-based SR framework. Each patch of the given image $I_0$ is matched to a patch in $I_{-1}$ in step 1. The corresponding patch (in the same location) in $I_0$ serves as the HR predictor (step 2). This patch is then pasted in the HR image $\tilde{I}_1$ (step 3). (b) The proposed sub-band self-similarity framework. Our method follows a series of similar steps as (a), but on each sub-band independently. Note that for super-resolving the patch shown in red, our algorithm allows for its various sub-bands to find matches in different spatial locations. See Sections 2.2, 2.3, 2.4 for details.
2.2.1 Algorithm Summary

To super-resolve the image $I_0$, our algorithm consists of the following steps, also summarized in Fig. 2.1(b):

1. We decompose the image $I_0$ into $N$ sub-bands $\{B^{(j)}_0\}_{j=1}^N$, which are obtained as the responses of the image $I_0$ to a bank of spatially localized, orientation selective, bandpass filters. We use the steerable pyramid decomposition [28, 29] for our work, although other schemes such as contourlet transform [30] may also be used.

2. We then apply a self-similarity-based SR algorithm to each of the sub-bands $\{B^{(j)}_0\}_{j=1}^N$, independently, to yield the set of HR sub-bands $\{\tilde{B}^{(j)}_1\}_{j=1}^N$. We describe this step in detail in Section 2.3 and discuss the key advantages it brings in Section 2.4.

3. We then recombine the HR sub-bands $\{\tilde{B}^{(j)}_1\}_{j=1}^N$, by inverting the sub-band decomposition, to yield an HR image $\tilde{I}_1$.

4. Finally, in order to ensure that the downsampled version of our estimated HR image is close to the given LR image, we enforce the backprojection constraint [31] by minimizing,

$$J(\hat{I}_1) = |(\hat{I}_1 * f_{psf}) \downarrow - I_0|^2$$

(2.1)

Starting with $\tilde{I}_1$ as initialization, we minimize the above cost function using a few iterations of gradient decent, to yield our final HR image $\hat{I}_1$.

2.3 Sub-Band Self-Similarity

We independently super-resolve each sub-band $B^{(j)}_0$ of $I_0$, using a self-similarity approach adopted from previous work [24, 21], summarized in the following:

For the sub-band $B^{(j)}_0$, we first obtain its downsampled version,

$$B^{(j)}_{-1} = \left( B^{(j)}_0 * f_{psf} \right) \downarrow$$

(2.2)

where $f_{psf}$ is an assumed point spread function. We then create internal dictionaries $\mathcal{L}^{(j)}$ and $\mathcal{H}^{(j)}$ that contain patches from $B^{(j)}_{-1}$ and their corresponding (higher
resolution) patches from $B_0^{(j)}$, respectively. The sets $\mathcal{L}^{(j)}$ and $\mathcal{H}^{(j)}$ serve as our internal training database of LR-HR training patches, for super-resolving the sub-band $B_0^{(j)}$. To super-resolve $B_0^{(j)}$ to $\tilde{B}_1^{(j)}$, we do the following: For every patch $l$ of $B_0^{(j)}$, we look for its $k = 5$ most similar patches $\{l_i\}_{i=1}^k$ in the LR set $\mathcal{L}^{(j)}$, based on $L_2$ distances. Their corresponding HR patches $\{h_i\}_{i=1}^k$ from the set $\mathcal{H}^{(j)}$ serve as individual predictors for the patch $l$. We compute a weighted average of $\{h_i\}_{i=1}^k$ to estimate the HR patch $\tilde{h}$ of $l$ as follows,

$$\tilde{h} = \frac{\sum w_i \cdot h_i}{\sum w_i}, \text{ where, } w_i = \exp \left( -\frac{||l - l_i||_2^2}{2\sigma^2} \right)$$

(2.3)

Using a larger number of patch matches ($k$) tends to cause oversmoothing, whereas very small values such as $k = 1$ or 2 produces sharper images but with some artifacts. We repeat the above procedure for every patch $l$ of $B_0^{(j)}$, to get the corresponding HR patches. These together constitute the super-resolved sub-band $\tilde{B}_1^{(j)}$.

### 2.4 Implications

Matching image patches based on intensity differences is often difficult if the patches contain complex structures such as textural detail [21]. Using sub-band decomposition, our algorithm essentially aims at decomposing complex textural structures into relatively simpler ones, that are easier to find matches for. For each image patch, our algorithm allows each of its sub-band components to find its optimal matches at different spatial locations in the image. This is illustrated in Fig. 2.1(b). The sub-bands of the red patch are allowed to find their optimal matches in different spatial locations in the LR sub-bands $B_1^{(1)}, B_1^{(2)}, B_1^{(3)}$. This is in contrast to the conventional way of matching raw patches as shown in Fig. 2.1(a), where all frequency components of the matched patch are restricted to be from the same spatial location, since no sub-band decomposition is performed.

These properties of our algorithm have useful implications discussed in the following:

**1) Lower matching error:** Our approach is guaranteed to find nearest neighbor (NN) matches with lower error, as compared to the traditional image domain patch matching. The NN error can be thought of as the error produced by a given LR image $I_0$ while reconstructing itself using its internal dictionary $\mathcal{L}$. In the SR
algorithm, the NN error therefore denotes the “training error” (in pattern recognition parlance).

We prove our claim of lower NN error as follows: Let $p_0$ denote a patch from the given LR image $I_0$. Let $p_{-1}(r)$ denote a patch at spatial location $r$ in the downsampled image $I_{-1}$. In conventional self-similarity schemes [24, 25], the objective is to first search for the patch $p_{-1}(r)$ that best matches $p_0$. Using the $L_2$ distance as the matching criterion, we can define this nearest neighbor matching error for the patch $p_0$ to be,

$$E_{\text{conv}}(p_0) = \inf_r \| p_0 - p_{-1}(r) \|_2^2$$  \hspace{1cm} (2.4)

Using the generalized Parseval’s theorem and the linearity property of sub-band transforms [28], we can write the above expression in terms of sub-bands of the patches as,

$$E_{\text{conv}}(p_0) = \inf_r \sum_j \| p_{0}^{(j)} - p_{-1}^{(j)}(r) \|_2^2$$  \hspace{1cm} (2.5)

where $p_{0}^{(j)}$ and $p_{-1}^{(j)}(r)$ denote the $j^{\text{th}}$ sub-band patches of $p_0$ and $p_{-1}(r)$ respectively.

Let us now compute the nearest neighbor matching error for our approach. Our approach performs the nearest neighbor search in the sub-band domain. That is, for a sub-band patch $p_0^{(j)}$ from the $j^{\text{th}}$ sub-band of the given LR image $I_0$, we look for its best matching patch $p_{-1}^{(j)}(r)$ in the $j^{\text{th}}$ sub-band of the downsampled image $I_{-1}$. The nearest neighbor error in this matching procedure can be written as,

$$E_{\text{out}}(p_0^{(j)}) = \inf_r \| p_0^{(j)} - p_{-1}^{(j)}(r) \|_2^2$$  \hspace{1cm} (2.6)
Using generalized Parseval’s theorem, the effective nearest neighbor matching error for the patch $p_0$ can be simply written as a sum of the errors in each sub-band, as follows:

$$E_{\text{our}}(p_0) = \sum_j E_{\text{our}}(p_0^{(j)})$$

(2.7)

$$= \sum_j \inf_r \| p_0^{(j)} - p_{-1}^{(j)}(r) \|^2_2$$

(2.8)

Let us now compare our nearest neighbor error (2.8) and the error in the conventional self-similarity patch matching (2.5). Using the property that the sum of infima is always less than or equal to the infimum of the sum, we can conclude that,

$$E_{\text{our}}(p_0) \leq E_{\text{conv}}(p_0)$$

(2.9)

We now verify this empirically. Figure 2.2, Center and Right show the error maps obtained by the conventional approach and by our approach, respectively. Clearly, the errors are much lower for our algorithm, particularly in textured regions such as the fur surrounding the faces. We show in our results in Section 2.5 that this lower NN error translates to better reconstruction of textural details.

(2) **Invariance to patch size:** The choice of patch size has an important effect on the quality of the SR results, particularly for self-similarity-based methods. Using larger patch sizes for conventional patch matching leads to greater difficulty in matching textural regions since the complexity of image structures is larger. On the other hand, using extremely small patch sizes is also not expected to improve results since very small patches may not contain enough structural information to learn their transformations across resolutions. For a given image, the optimal patch size to use is difficult to determine a priori. Using the proposed approach, complex patches are broken down into relatively simpler sub-bands. The simpler structure of the sub-bands decreases the variety of the sub-band patches and thus reduces the error of the best matching patch for a given dictionary size. Therefore, we expect our algorithm to suffer less if the patch size chosen is sub-optimal. Indeed, as compared to traditional self-similarity-based SR, we find our results to be less sensitive to the choice of patch size. We show this in our experiments later in Section 2.5.

(3) **Exponentially larger internal dictionary:** Allowing different sub-bands of the HR patch to come from different spatial locations of the LR image has
an important corollary. Combining sub-bands from different locations effectively allows us to synthesize new patches, originally not present in the dictionary of raw image patches. This, in a sense, leads to a combinatorial expansion of the internal dictionaries $L$ and $H$, resulting in a dictionary whose size increases exponentially with the number of sub-bands. Further, this is achieved without the use of external databases. We illustrate this with a simple example in Fig. 2.3. We assume here that our raw patch dictionary $L$ consists of only two patches as shown in the blue box. In this example we decompose these patches into $N = 3$ sub-bands as depicted in the black dotted box. Now, if using traditional image domain patch-matching, one is restricted to choosing among only two possible candidate matches while searching for a nearest neighbor match. However, if patch-matching is done independently for each sub-band, the number of unique combinations possible is $2^N = 8$. In Fig. 2.3 on the right, we show the patches resulting from each of the unique sub-band combinations. Clearly, in addition to the original two patches, several more new textural patches have been synthesized in this expanded dictionary. Note that one never has to explicitly obtain such an expanded dictionary. Such an expansion is an implicit consequence of independently finding best matches for the different sub-band patches.

(4) Invariance to image size: We have shown that super-resolving sub-bands independently has the overall effect of performing conventional patch-similarity-based SR, but using a much larger internal dictionary, whose elements are generated by combining different sub-band patches from different locations in the scene. While the use of a larger dictionary is expected to be always beneficial in general, it becomes particularly useful in cases where the original internal dictio-
nary is small, such as while super-resolving extremely small images. Indeed, as we show in Section 2.5, in such cases we observe a much greater improvement in our results over the conventional self-similarity approach. Our algorithm therefore yields relatively more consistent levels of performance across different image sizes. We corroborate this claim in Section 2.5.

### 2.5 Experiments and Results

![Images comparing different super-resolution methods](image)

(a) Ground truth  
(b) Glasner et al.[24]  
(c) Freedman and Fattal [25]  
(d) Ours

Figure 2.4: Dog (2X): The dog fur, and the details on the wooden pole are better reconstructed using our method, and bear closer resemblance to the ground truth.

**Implementation Details:** For the steerable pyramid, we use eight different orientation bands, and a single scale decomposition. Using more orientation bands improved results in general, but the improvements became marginal beyond eight bands. We use only a single (highest) scale decomposition since the lower-scale bands contain lower-frequency information which does not pose much challenge for SR. We perform SR in two steps. Therefore, for 3X SR, we perform $\sqrt{3}$X SR twice. Our algorithm is used only on the luminance channel of color images. The chroma components are separately upscaled using bicubic interpolation and combined with our output to obtain the final color image.
We compare our results to eight popular single image SR methods [24, 25, 14, 15, 32, 33, 34, 31], as described in the paragraphs following.

**Comparison with self-similarity methods:** Our most important comparison is with other self-similarity methods. We compare our results to [24] and [25], which are two very popular self-similarity-based SR methods in the literature. Figure 2.4 shows results on the *Dog* image. Our result shows more detail and richer texture in the dog fur and the wooden pole. The self-similarity methods [24, 25] in general are quite good at preserving sharpness of high contrast edges, but tend to smooth out finer details. The method in [25] tends to smooth details more than [24] since it performs only a very localized search for nearest neighbors, for computational reasons. Our result bears closer resemblance to the ground truth.

Figure 2.5 shows results on the *Kangaroo* image. Notice here that both [24] and [25] almost completely lose the textural details of the tail. Our algorithm is able to better preserve this texture.

Figure 2.6 shows results on the *Koala* image. Here as well, our algorithm is able to synthesize richer texture in the fur and the tree trunk, than both [24] and [25]. Note that the *Koala* and *Kangaroo* images do not have ground truth available.

**Comparison with external dictionary-based methods:** We now compare our results with methods that use external dictionaries for SR. Specifically, we consider [14] which is a popular method based on dictionary learning and sparse representations, the method in [32] that uses ridge regression for predicting HR patches, and the more recent method [15], which is based on using simple regression functions on a pre-clustered training dictionary. We also compare to the classic iterative backprojection algorithm [31] for reference.

Figure 2.7 shows the results on the *Tiger* image. While Kim [32] reconstructs high-contrast edges almost as sharp as ours, textural details appear highly washed out. The result of [14] also appears a little soft, both along high-contrast edges as well as in textural regions such as the grass (red box). The result of [15] appears slightly more detailed than [14], but it shows excessive ringing artifacts such as along the stripes of the tiger (yellow box), much like the backprojection algorithm [31]. Overall, our result has richer textural details without excessive ringing artifacts.

Figure 2.8 shows results on the *Sunlight* image. Notice that the woman’s hair appears most natural in our result. The results of [15] and [31] clearly show more ringing artifacts in the hair, whereas [14] and [32] are not able to reconstruct sufficient detail.
Figure 2.5: Kangaroo (3X): Both [24] and [25] almost completely lose the textural details of the kangaroo’s tail. Our algorithm is able to better synthesize this. Ground truth for this image is not available so absolute error cannot be obtained.
Figure 2.6: Koala (3X): Our result shows richer texture in koala’s fur and the tree trunk. Ground truth for this image is not available so absolute error cannot be obtained.
Figure 2.7: Tiger (4X): Notice the grass above and below the tiger. Our result shows greater textural detail in the grass regions (red box), as compared to most methods. While [15] also seems to produce rich texture, it also produces ringing artifacts such as on the stripes of the tiger (yellow box).
Figure 2.8: Sunlight (4X): Notice the woman’s hair. [32] and [13] do not produce sufficient detail in the hair, whereas [15] and [31] show excessive ringing artifacts. Our result appears more natural.

**Comparison with other methods:** We also compare our approach with two other methods popularly used in literature – the gradient profile prior (GPP) method [33], that is based on learning gradient profile transformations across resolutions, and the method in [34] that uses iterative feedback-based upsampling, without any external databases. Figure 2.9 shows our results on the Red hair image. The fine strands of hair in the blue box are clearly visible in our result, but is lost in the results of [33] and [34]. Our result appears almost indistinguishable from the ground truth in this example.

**Performance vs. patch size:** The chosen patch size can have a significant effect on the quality of the SR results particularly for internal dictionary-based methods. We have shown earlier that using the proposed approach, complex patches are broken down into simpler sub-bands, that can find closer (lower error) matches. Therefore, our algorithm should suffer less if patch size is increased. To
Figure 2.9: *Red Hair (2X)*: Notice the details of the hair as shown in the blue box. Fine strands of hair are discernible in our result, whereas they are smoothed out in the result of [33] and [34]. Our result seems almost indistinguishable from the ground truth in this example.
verify this empirically, we do the following: We super-resolve 100 natural images (with known ground truth) using our method and also using the conventional self-similarity method of [24], with several different patch sizes, ranging from $2 \times 2$ to $11 \times 11$. We then plot the average output image quality (in terms of PSNR and SSIM [35]) as a function of the patch size used. The plots in Fig. 2.10 show our results. As expected, the performance of our algorithm not only remains higher throughout the tested range, but the loss of PSNR and SSIM is also much slower than the conventional self-similarity approach.

Figure 2.10: Left: Plots of PSNR and SSIM as a function of the patch size used, for our algorithm as well as the conventional self-similarity method of [24]. Right: An example showing the effect of patch size on the results of both algorithms. Our result remains more consistent with patch size variation as compared to [24]. Numbers in parentheses denote PSNR in dB and SSIM [35].

**Performance vs. image size:** Earlier, we showed that our algorithm has the effect of synthesizing a much larger internal dictionary, by combining sub-bands from different spatial locations in the image. We therefore expected our algorithm to perform significantly better than conventional self-similarity, if the input image size was very small. To verify this claim, we perform the following experiment: Consider super-resolving the set of images as shown in Fig. 2.11 on the left. Each image here is a cropped version of the image on its right. The leftmost (smallest) image, therefore, is a sub-image of all the other images, and appears in all of them, as marked by the red box. We now wish to see how well this sub-image gets super-resolved in each of these images. Clearly, in the rightmost (largest) image, the sub-image has access to all the patches from its surrounding regions as well, which should therefore result in better SR. We compute SR quality (in
terms of PSNR and SSIM [35]) of this sub-image, as a function of the size of the image containing it, and plot the result in Fig. 2.11 on the right. As expected, the conventional self-similarity approach [24] shows a more drastic reduction in performance for smaller image sizes, as compared to our method.

Figure 2.11: Left: Data used for studying the performance of our algorithm as a function of the size of the input image. We use a series of cropped images as shown. We study how the common sub-image (red box) gets super-resolved in each of these images. Right: Plots showing the PSNR (in dB) and SSIM of the super-resolved sub-image as a function of the size of the image containing it. Our algorithm shows a much gradual decline in performance for smaller images, as compared to the conventional self-similarity method [24].

To visualize this effect of image size in a more practical SR problem, we perform the following experiment: We consider super-resolving two input images, as shown in the black dotted box in Fig. 2.12. The first image shows a group photograph, whereas the second is a cropped version containing just one of the faces, measuring only $20 \times 25$ pixels. We super-resolve both these images using the method of [24] as well as our proposed algorithm and show the results in the blue and red dotted boxes respectively. We compare the quality of the super-resolved faces obtained using each method, in both the images. We make the following two observations: (1) In both images, the face is super-resolved better (visually) by our algorithm than the conventional internal dictionary-based approach [24]. (2) There is a significant difference in the quality of the super-resolved faces from the bigger and the cropped images, using either method. Using our algorithm, however, this difference is smaller. Our algorithm is able to super-resolve the extremely small cropped image better than the conventional self-similarity approach.

In practice, small images are more commonly encountered as candidates for super-resolution than large ones. Our algorithm is therefore useful for practical applications like super-resolution of thumbnail images, detection/recognition of distant (small) faces in images/videos captured using surveillance cameras, etc. Like any self-similarity-based algorithm, our algorithm does not require manu-
Figure 2.12: An example showing the performance of our algorithm for very small input images. We super-resolve the two images shown in the black dotted box, the right one being a one face sub-image cropped from the left image. Our algorithm is able to super-resolve this small face image much better than the conventional self-similarity approach of [24].
ally chosen training images, which makes it all the more attractive in terms of portability and ease of implementation.

**Computational Cost:** Our algorithm applies a self-similarity SR algorithm (such as [24]) on $R$ different sub-bands. A naive implementation would be $R$ times slower than the corresponding self-similarity SR algorithm. But since each sub-band is super-resolved independently, they can be easily parallelized. Using such a parallelization, our algorithm is just around 1.5 times slower than the baseline self-similarity SR algorithm of [24].

### 2.6 Conclusion

While external dictionary-based methods can produce good results in general, they are hindered by the problems associated with the choice and construction of the external training database. Internal dictionary-based methods provide an attractive way to circumvent these issues, but also sacrifice some ability to reconstruct textural details well, particularly while super-resolving small-sized images and/or when the optimal patch size not used. In this chapter we have proposed a self-similarity-based algorithm that overcomes these limitations. Our algorithm produces better SR results that remain fairly consistent across several scenarios commonly encountered in practice.
CHAPTER 3

SUB-BAND ENERGY CONSTRAINTS FOR SUPER-RESOLUTION

3.1 Introduction

Recent studies have shown that the likelihood of finding a good match for a patch in the internal dictionary, falls, as the gradient content of the image increases [21]. This suggests that textural details like hair, animal fur etc. often find suboptimal matches, using a self-similarity approach. This problem can be partly attributed to the limitation in using distance metrics such as pixel-wise sum of squared difference ($L_2$ distance) for matching textural patches. The $L_2$ distance between two patches is largely determined by the high contrast and prominent structures (macrostructures) in the patch, and is less sensitive to the fine details (microstructures) of the patch. Indeed, this problem manifests itself in the final results of patch-based SR reconstruction methods – poor patch matches lead to inconsistent explanations of pixels in textural regions, and fine textural details or microstructures are thus averaged out.

In this chapter, we propose a solution to the above problem. We argue that the $L_2$ distance by itself is not a sufficient criterion to find suitable matches for textural patches. Indeed, metrics based on pixelwise differences have been rather unsuccessful in applications such as texture classification or texture retrieval. On the other hand, texture descriptors based on responses to a multi-orientation bank of bandpass filters have been effective for such tasks [36, 37, 38]. In the SR application at hand, we therefore combine the conventional $L_2$ distance-based patch-matching procedure with additional prior constraints on the energies of the different orientation selective sub-bands of the patch. We observe through experiments in this chapter that for each patch, the low-energy sub-bands (which correspond to fine textural details) get severely attenuated during conventional $L_2$ distance-based SR. Based on this observation, we propose a method to learn this attenuation of sub-band energies in the patches, using scaled-down version(s) of
Figure 3.1: Conventional self-similarity-based SR. Given LR image $I_0$ is shown in red. Each patch of $I_0$ is matched to $k = 5$ most similar patches in $I_{-1}$ in step 1. For simplicity we show only $k = 1$ most similar patch in this figure. The corresponding patch (in the same location) in $I_0$ serves as the HR predictor (step 2). This patch is then pasted in the HR image $\tilde{I}_1$ (step 3). See Section 3.3 for details.

the given image itself (without requiring external training databases), and thus propose a way of compensating for the energy loss in these sub-bands of each patch. More specifically, we propose the use of scaling coefficients to boost the sub-bands of the patch that constitute the fine textural details (microstructures). As a consequence, our SR results appear richer in texture and more natural as compared to state-of-the-art methods, as shown by our experiments.

In Section 3.2, we present a stepwise summary of the proposed algorithm. The subsequent sections present details of the steps involved.

### 3.2 Algorithm Overview

**Notation.** We denote the given image to be super-resolved as $I_0$. By $I_1$ we denote the HR version of $I_0$, whose linear dimension, or scale, is larger by a factor of $s$. Similarly, we denote by $I_{-1}, I_{-2}$ etc., the smaller versions of $I_0$, by scaling factors of $1/s, 1/2s$ etc., respectively. We denote the super-resolved image(s) obtained using our algorithm using a hat ($\hat{\cdot}$) symbol. Therefore, our objective is to super-resolve $I_0$ to obtain an HR image $\hat{I}_1$, that best approximates the true HR image $I_1$.

We use scripted letters to denote sets, we use boldface lowercase letters to denote image patches, and lowercase italicized letters to denote scalars and indices.

**Algorithm Summary.** To obtain $\hat{I}_1$, from $I_0$, our proposed algorithm involves the
Figure 3.2: Schematic summary of proposed algorithm. The left part of the figure depicts the conventional self-similarity-based SR procedure. Each HR patch obtained using the conventional SR algorithm is further enhanced by amplifying its sub-bands using scaling coefficients. These scaling coefficients are obtained by learning the attenuation caused in the sub-bands of the patches of $\tilde{I}_0$, compared to those of $I_0$. The image pair $I_0$ and $\tilde{I}_0$ (green dotted box) therefore serves as a source of training patches. $\tilde{I}_0$ is obtained by using the conventional self-similarity-based SR (blue dotted box) with $I_{-1}$ as input.
following steps:

1. Using $I_0$, we first compute an intermediate HR image $\tilde{I}_1$ that is obtained by the conventional self-similarity-based SR approach, along the lines proposed earlier [21, 24, 25]. We present the general framework of such an algorithm in Fig. 3.1, and its details in Section 3.3.

2. For each patch $\tilde{p}$ of $\tilde{I}_1$, we compute the response of bank of $R$ orientation selective bandpass filters, yielding the sub-bands $\{\tilde{p}^{(1)}, \tilde{p}^{(2)}, \ldots, \tilde{p}^{(R)}\}$. To selectively amplify the patch macrostructure vs. microstructure, we differentially scale the patch’s energy contents in different sub-bands by using the coefficients $\{\alpha^{(j)}\}_{j=1}^R$, to yield a transformed set of bandpass patches,

$$\hat{p}^{(j)} = \alpha^{(j)}\tilde{p}^{(j)} \quad j = 1, 2, ..., R \quad (3.1)$$

We discuss our algorithm that learns these coefficients in Sections 3.4 and 3.5. The scaling coefficients $\alpha^{(j)}$ allow us to impose sub-band energy constraints on each patch of the super-resolved image $\tilde{I}_1$, to minimize the loss of textural detail in the patch.

3. The rescaled sub-bands $\{\hat{p}^{(j)}\}_{j=1}^R$ of each patch are recombined to yield the texture-enhanced patch $\hat{p}$, and all such enhanced patches constitute the super-resolved image $\hat{I}_1$. Finally, we also then run a few iterations of the classical backprojection constraint [31], to ensure that the blurred and downsampled version of $\hat{I}_1$ matches the given LR image $I_0$. We elaborate on this step in Section 3.6.

A schematic summary of our algorithm is presented in Fig. 3.2. The details of each step follow.

### 3.3 Self-Similarity-Based SR

The conventional self-similarity-based SR approach that we adopt to obtain $\tilde{I}_1$ from $I_0$ follows similar steps as done in existing work [21, 24, 25], and is summarized in Fig. 3.1. Given the LR image $I_0$, we first obtain its downsampled version,

$$I_{-1} = (I_0 * f_{psf}) \downarrow \quad (3.2)$$
where \( f_{psf} \) is an assumed point spread function. We then create two sets of image patches \( \mathcal{L} \) and \( \mathcal{H} \), that contain patches from \( I_{-1} \) and their corresponding (bigger) patches extracted from \( I_0 \), respectively. The sets \( \mathcal{L} \) and \( \mathcal{H} \) serve as our database of LR-HR training patches. To super-resolve the given image \( I_0 \) to \( \tilde{I}_0 \), for every patch \( l \) of \( I_0 \), we look for its \( k = 5 \) most similar patches \( \{l_i\}_{i=1}^{k} \) in the LR set \( \mathcal{L} \), based on \( L_2 \) distances. Their corresponding HR patches \( \{h_i\}_{i=1}^{k} \) from the set \( \mathcal{H} \) serve as individual predictors for the patch \( l \). We average \( \{h_i\}_{i=1}^{k} \) to estimate the HR patch \( \tilde{h} \) of \( l \) as follows,

\[
\tilde{h} = \frac{\sum w_i \cdot h_i}{\sum w_i}, \text{ where, } w_i = \exp \left( -\frac{||l - l_i||^2}{2\sigma^2} \right)
\] (3.3)

We repeat the above procedure for every patch \( l \) of \( I_0 \), and get their corresponding HR patches, which constitute the HR image \( \tilde{I}_1 \).

### 3.4 Analysis of Sub-Band Energies

We argue that the self-similarity-based SR algorithm described in Section 3.3 tends to smooth out fine textural details, due to the limitation of the \( L_2 \) distance in capturing textural similarity between patches. To quantify this loss of textural detail, we now perform a simple experiment. We use the \textit{baby} image of Fig. 3.3(a) as our example. We denote \( I_1 \) to be the ground truth HR version of this image, as shown in Fig. 3.3(a). We compute its LR version \( I_0 \) by blurring and downsampling. We then use the SR algorithm described in Section 3.3 to super-resolve this LR image \( I_0 \) to obtain the image \( \tilde{I}_1 \) as shown in Fig. 3.3(b). We now examine the textural loss in \( \tilde{I}_1 \) when compared to the ground truth image \( I_1 \).

Let \( \tilde{p} \) and \( p \) represent corresponding patches from the super-resolved image \( \tilde{I}_1 \) and the ground truth image \( I_1 \), as illustrated by the blue box in Fig. 3.3. Let \( \{\tilde{p}^{(j)}\}_{j=1}^{R} \) and \( \{p^{(j)}\}_{j=1}^{R} \) denote the decomposition of these patches into \( R \) orientation sub-bands, as illustrated in Figs. 3.3(d) and 3.3(c). We use the steerable pyramid decomposition [28, 29] to obtain the orientation selective sub-bands. The steerable pyramid provides jointly localized (space/frequency) representation of images using an invertible multi-scale, multi-orientation image decomposition [28, 29], as shown in Fig. 3.4. We use \( R = 16 \) orientations (and just a single scale) in our algorithm.

Let \( \tilde{e}^{(j)} \) and \( e^{(j)} \) be the energies of the \( j^{th} \) sub-bands \( \tilde{p}^{(j)} \) and \( p^{(j)} \) respectively.
Figure 3.3: (a) Ground truth HR image $I_1$. (b) Image $\tilde{I}_1$ obtained after super-resolution using conventional approach of Section 3.3. (c) An example patch from the ground truth image $I_1$, along with its decomposition into orientation selective sub-bands. (d) Similar decomposition for the corresponding patch from $\tilde{I}_1$. We analyze the loss of energy in the sub-bands of patches from $\tilde{I}_1$, compared to those from $I_1$. The patches shown here are chosen large for illustration purpose.

\[ e^{(j)} = ||\tilde{p}^{(j)}||^2_2, \text{ and } e^{(j)} = ||p||^2_2 \]  

(3.4)

We now sort the sub-band energies $\{\tilde{e}^{(j)}_i\}_{j=1}^R$ and $\{e^{(j)}_i\}_{j=1}^R$ according to decreasing values of $\tilde{e}^{(j)}_i$. The sorted set of energy values helps us observe the relative energy distribution between the macrostructure (high-energy sub-bands) and the microstructures (low-energy sub-bands) in the patch, irrespective of their orientations. The sorting helps us achieve this rotation invariance. Therefore, if an image patch recurs in the image in a rotated form, both the patches would yield the same sorted set of sub-band energy values.

Figure 3.4: An example showing the multi-orientation image decomposition yielded by the steerable pyramid [28, 29], on a synthetic image.

We repeat the above procedure for all patch pairs $\tilde{p}$ and $p$ from the images $\tilde{I}_1$.
Figure 3.5: (a) Average sub-band energy value of all patches in the Baby image, sorted in decreasing order. Clearly, high energy sub-bands (macrostructures) are reasonably well recovered, whereas low energy sub-bands (microstructures) get severely attenuated using conventional patch-similarity-based SR. (b) Blue plot shows the sub-band scaling coefficients obtained using the ground truth image \( I_1 \) (i.e. by comparing patches from \( I_1 \) and \( \tilde{I}_1 \)). Red plot shows the coefficients obtained using the proposed self-learning scheme (i.e. by comparing patches from \( I_0 \) and \( \tilde{I}_0 \)).

and \( I_1 \), and obtain a sorted array of sub-band energy values for each patch. We then compute an average of these sorted arrays or sets, across all the patches. Figure 3.5(a) shows this average set of sorted energy values, for patches from the super-resolved image \( \tilde{I}_1 \) (blue bars) and from the ground truth \( I_1 \) (red bars).

We make the following two interesting observations: (1) The energy in the high energy bands of the super-resolved image \( \tilde{I}_1 \) is much closer to those of the ground truth image \( I_1 \). This shows that the patch-similarity-based SR algorithm using \( L_2 \) distances is able to preserve the macrostructures quite well. (2) Relatively, the low-energy sub-bands suffer from severe attenuation, confirming our hypothesis stated earlier that fine textures (microstructures) are much less preserved by such an SR algorithm.

Can we recover or compensate for this loss? Based on examining the bar plot of Fig. 3.5(a), a possible way to “optimally” compensate for the sub-band attenuation is to amplify each sub-band \( \tilde{p}^{(j)} \) of the patch \( \tilde{p} \) by multiplying with scaling factors \( \alpha^{(j)} \), where,

\[
\alpha^{(j)} = \frac{e^{(j)}}{\tilde{e}^{(j)}}, \quad j = 1, 2, ..., R \tag{3.5}
\]

Using the coefficients \( \alpha^{(j)} \), the sub-bands can be amplified such that their en-
energies match those of the ground truth. The blue curve in Fig. 3.5(b) shows the values of these coefficients computed using (3.5) for the baby image. As expected, the lower-energy sub-bands have higher scaling coefficients as they are more severely attenuated.

An obvious problem in using (3.5) is that the ground truth image \( I_1 \) is never available in any practical SR problem. Therefore, the sub-band energies \( \{e^{(j)}\}_{j=1}^{R} \) of the ground truth image patches are never available, and the coefficients \( \alpha^{(j)} \) of (3.5) cannot be determined. In Section 3.5, we propose a method to learn these coefficients.

### 3.5 Self-Learning of Sub-Band Constraints

Given an input image \( I_0 \), our analysis showed that patches of the super-resolved image \( \tilde{I}_1 \), obtained using the conventional self-similarity approach of Section 3.3, suffer attenuation of the low-energy sub-bands. We saw that the scaling coefficients \( \alpha_j \) of (3.5) could compensate for this attenuation by appropriately boosting the sub-bands of each patch. However, computing these coefficients required knowledge of the ground truth HR image \( I_1 \), which is not available in practical scenarios.

A solution to the above problem is to estimate these coefficients from training patches extracted from natural images and treat these learned coefficients as a statistical prior. Such a prior would indicate the relative amplifications required for different sub-bands of the super-resolved image patch.

Instead of resorting to an external database of image patches for learning such a prior, we propose a self-learning scheme, that operates as follows: We utilize scaled-down versions of the given image \( I_0 \), to generate training data for learning the scaling coefficients. More specifically, we first obtain \( I_{-1} \) from \( I_0 \) by a blurring and downsampling operation. We then compute a super-resolved image \( \tilde{I}_0 \), by using the patch-similarity-based SR algorithm of Section 3.3 with \( I_{-1} \) as the input image. The computation of \( \tilde{I}_0 \) is schematically illustrated in the blue dotted box of Fig. 3.2.

Our training image pair consists of the super-resolved image \( \tilde{I}_0 \), and its corresponding “ground truth” \( I_0 \), which is available to us. Our objective is now to learn the attenuation in the sub-bands of the patches of \( \tilde{I}_0 \), when compared to those from \( I_0 \). We extract around 1000 randomly sampled patches from \( \tilde{I}_0 \) along with their
corresponding ground-truth patches from $I_0$. Using these two sets of patches, we repeat the analysis presented in Section 3.4 to obtain the scaling coefficients $\alpha^{(j)}$ using (3.5).

The red plot in Fig. 3.5(b) shows the coefficients thus obtained using the proposed self-learning scheme (using $\tilde{I}_0$ and $I_0$) for the Baby image. We can see that these coefficients closely approximate the “optimal” coefficients learned with knowledge of the ground truth image $I_1$ (blue plot).

### 3.6 Backprojection Constraint

Once the coefficients $\{\alpha^{(j)}\}_{j=1}^R$ have been determined, we use it to amplify or boost the respective sub-bands of each patch from the image $\tilde{I}_1$, using (3.1). The enhanced patches thus obtained form the super-resolved image $\hat{I}_1$ that we set out to achieve. However, we must also ensure that the image $\hat{I}_1$ on blurring and down-sampling, yields the LR image $I_0$. We therefore need to minimize the cost function,

$$J(\hat{I}_1) = ||(\hat{I}_1 * f_{psf}) \downarrow - I_0||_2^2$$  \hspace{1cm} (3.6)

To satisfy this constraint, we run around 10 iterations of the following gradient-based update rule,

$$\hat{I}_1^+ = \hat{I}_1 - \mu \nabla J(\hat{I}_1)$$ \hspace{1cm} (3.7)

where we choose the stepsize $\mu = 1$. The above procedure is called the iterative backprojection algorithm [31].

### 3.7 Results

**Implementation Details.** We use the proposed algorithm for upscaling images with a relatively small scaling factor, not exceeding $s = 2$. Therefore, for super-resolving images to $4X$ resolution, we apply the proposed algorithm twice, each time with scaling factor $s = 2$. Similarly, for an overall super-resolution of $3X$, we apply our algorithm twice with scaling factor $s = \sqrt{3}$ each time. For super-resolving color images, we use our algorithm only on the luminance channel. The chroma channels are upscaled using simpler methods such as bicubic interpolation, and then recomputed to obtain the color image.

We first run our algorithm on images that have known ground truth HR versions.
Figure 3.6: *Sunlight* (2X): Best viewed when zoomed in. Our result shows much richer texture in the hair, facial features and the blue shoulder strap etc., as compared to the conventional patch-similarity-based SR. Our result appears almost indistinguishable from the ground truth. Numbers in brackets denote SSIM [35] values.
Figure 3.7: *Fur* (4X): The fur is reconstructed better in our result, and it appears sharper and richer in texture. Numbers in brackets denote SSIM [35] values.
We compare our approach with the conventional self-similarity-based method as described in Section 3.3 and see the improvement in results our algorithm brings. Figure 3.6 shows our result on the *Sunlight* image. Clearly, our result shows much richer texture in the hair, facial features, the blue shoulder strap etc. Visually, our result appears almost indistinguishable from the ground truth in this example. We report the structural similarity measure (SSIM) [35] below each result, although the correlation of numerical metrics with human perception of image quality is debatable.

Figure 3.7 shows our result on the *Fur* image. In this case as well, our result looks visually more appealing and bears closer visual resemblance to the ground truth.

We now compare our results to those obtained in the past work. Specifically, we compare our results to those of two state-of-the-art methods, the self-similarity-based methods of Glasner et al. [24] and Freedman and Fattal [25], taken from the respective authors’ websites. Figure 3.8 shows the results on the *Koala* image. We can see that our result better shows the fine details in the animal fur and the tree trunk than the other two methods. Figure 3.9 shows another set of results on the *Girl* image, where fine details of the hair are more clearly visible in our result.
Figure 3.9: *Girl* (3X): Textural details of the hair are more enhanced in our result. The freckles on the face also are clearer if seen while zoomed in.
Finally, in Fig. 3.10, we also compare against two more methods, that are based on learning from external databases - the dictionary learning-based method of Yang et al. [14] and the edge statistics-based method of Fattal [39]. Yang et al. [14] is not able to produce sufficiently sharp edges (e.g. the lips). The textures produced by the edge-based method of Fattal [39] tend to appear un-natural, e.g., as in the green box. Our result appears richer in texture, with sharper edges.

3.8 Conclusion

We have presented an SR algorithm that delivers better super-resolved texture. Our algorithm is based on an observation we have made, that the conventional $L_2$ distance-based patch-matching does not sufficiently characterize fine textures. Additional criteria are needed to ensure that the subtle textural elements are super-resolved better. To take advantage of oriented bandpass filters in characterizing textures, we have presented an algorithm that additionally constrains the energies of the sub-bands of the super-resolved patches. We have proposed a self-learning scheme that determines an optimal set of scaling coefficients, to balance the energies in the sub-bands to mimic their distributions in the natural images. Our algorithm does not use any external training database.
Figure 3.10: *Baby* (4X): Textures in the woolen cap appear richer in our result as compared to other methods. Our edges are also sharp. Details in the eyes are slightly better.
CHAPTER 4

SUPER-RESOLUTION OF NOISY IMAGES

4.1 Introduction

Noise corruption is a ubiquitous phenomenon that affects many image processing tasks. For addressing any practical image processing problem, the designed algorithm needs to be either robust to the presence of noise in the input images, or denoising needs to be performed as an explicit preprocessing step before using the algorithm. In this chapter, we bring to light the problems caused by noise and the use of denoising algorithms for the super-resolution (SR) problem, and present a framework for performing SR effectively, in the presence of noise.

Image denoising algorithms have evolved from local averaging-based techniques to non-local, patch similarity driven state-of-the-art approaches [40, 23, 41, 42]. In methods such as BM3D [40] and non-local means (NLM) [23], each noisy patch is denoised by seeking several similar patches within the noisy image and computing their mean, with the intention of averaging out the noise, while retaining the underlying image structure. Such approaches are justified by studies on statistics of natural images which suggest that image patches tend to recur within the image [21].

Like denoising, the single image SR problem is also commonly addressed using patch-similarity. As discussed earlier, many state-of-the-art SR algorithms are based on seeking high-resolution (HR) versions of each low-resolution (LR) image patch, using a training database of LR-HR pairs [14, 10]. In [24, 25, 43, 44], refined versions of this approach are proposed wherein the LR-HR training database is created using scaled-down version(s) of the given LR image itself. Such self-similarity-based approaches are again driven by natural image statistics which suggest that patches recur in an image not just at one scale but at multiple scales [24, 21].

While both denoising and SR use patch-similarity-based priors, they are used
(a) A typical patch-based denoising scheme searches for a large number of patch matches of the same scale in a patch database, in order to average out the noise in the output (denoised) image.

(b) A typical patch-based SR algorithm searches for only a few (just one in this figure) most similar patches, which are then mapped to a finer scale (using an LR-HR patch database). The use of only a few patches helps limit the loss of high-frequency content in the image.

Figure 4.1: (a) A typical patch-based denoising algorithm (e.g. NLM [23]), and (b) a typical patch similarity-based SR algorithm (e.g. [24, 21, 25]).

toward different objectives. The goal in denoising is to seek a large number of similar patches of the same scale so as to average out noise. On the other hand, SR usually seeks fewer patches, which are mapped to a finer scale and averaged to obtain an HR patch. Since only a few patch-matches are averaged, the resulting patch tends to retain high-frequency content. Since denoising seeks a larger number of matches, greater error is tolerated in the patch-matches. On the other hand, SR requires a greater level of similarity, and therefore uses only a few, lowest error patch-matches. In a noisy image, the SR algorithm would therefore tend to match even the noise part, and would thus “overfit” while searching for similar patches in an effort to preserve textural details. Due to these conflicting objectives, it is difficult to perform effective denoising and SR of a noisy LR image simultaneously using a unified patch-recurrence driven algorithm. Figure 4.1 shows schematically the differences and similarities between typical patch-based denoising and SR algorithms, as described above.

To super-resolve an image containing noise, the conventional approach is therefore to first preprocess with a denoising algorithm, followed by using an SR algorithm of choice. Note that the reverse approach of super-resolving first, followed by denoising, yields unacceptable results as shown in Fig. 4.2. This happens because SR introduces spatial correlation in the noise, and most denoising algo-
Figure 4.2: (a) Noisy low-resolution image as input. (b) Result obtained using the conventional processing approach of denoising followed by super-resolving, using state-of-the-art methods [40, 24]. (c) Result obtained by super-resolving first, followed by denoising. (d) Our result.

rithms fail at removing correlated noise. Using denoising as a preprocessing step before SR, however, leads to another problem. Being an ill-posed problem, denoising is subject to inherent performance bounds [45, 46, 47]. Some components of the underlying signal are bound to be attenuated or lost by any denoising algorithm. In general, this denoising loss is more severe in areas containing complex structures such as fine textures. This loss of textural detail is particularly detrimental for super-resolution, since the synthesis of such high-frequency details is the challenge in SR algorithms.

In this chapter, we present a framework for obtaining a clean, HR image from a noisy LR image, that addresses the above problems of conventional scheme(s). Our key contribution and the motivation behind it can be summarized as follows: Since super-resolved images typically lack high-frequency content, and since denoising algorithms typically discard excess high-frequency content, we propose an algorithm that attempts to utilize the high-frequency content discarded by denoising algorithms for the benefit of super-resolution.

Our algorithm begins by obtaining two HR images from the given noisy LR image. The first image is obtained by denoising the given LR image followed by super-resolving it (as is conventionally done). We call this the denoised HR image. The second image is obtained by directly super-resolving the noisy LR
We call this the noisy HR image. While also containing noise, the noisy HR image contains some of the textural components which are not present in the denoised HR image due to the denoising loss. In order to obtain a noise-free image that also contains these textural details, we propose a linear framework that obtains the desired HR image as a convex combination of the denoised HR image and the noisy HR image. This linear combination is performed on orientation and frequency selective bands of the two images, such as those obtained using the steerable pyramid decomposition \cite{28, 29}. As we show in Section 4.3, on doing so we can obtain a desired HR image where: (1) a part of the denoising loss is recovered in the HR domain, and, (2) the resultant image can be simultaneously enhanced by synthesizing more textures from the noise components by appropriately constraining the parameters of the linear combination. These parameters are determined based on our experiments which reveal where (in spatial and oriented frequency domains) signal loss is most prevalent. We describe these constraints and procedures to obtain the parameters in Sections 4.4, 4.5 and 4.6. In Section 4.7 we discuss this convex combination model from a different perspective. We show that our algorithm can be viewed as a texture-adaptive patch-averaging-based SR algorithm, where each patch is super-resolved by seeking multiple patch matches at a finer scale, and where the number of patches to be averaged is determined based on local texture analysis at each spatial location.

We show in our results that the part-recovery and part-synthesis of textures using our algorithm yields HR images that are visually more pleasing and richer in textural content than those obtained using the conventional strategies. Since our algorithm allows for treating denoising and SR steps as abstractions or black boxes, we implement our algorithm using different state-of-the-art denoising and SR algorithms and observe consistent improvement over the respective baselines. To corroborate our hypothesis that our algorithm does indeed recover the denoising loss, we also compute quantitative metrics (PSNR and SSIM \cite{35}) over several test images and observe a consistent improvement in these metrics.

4.2 Related Work

Although both denoising and SR have been extensively studied independently in the past, addressing both these problems in a joint setting has received rather little attention. While a number of single image SR algorithms have been proposed of
late, most algorithms assume a noise-free low-resolution image as input. In this section, we briefly comment on the few SR methods that consider the effect of noise in their algorithms.

A Bayesian approach for video SR is presented in [48], where a joint model for estimating the clean HR video frames, along with motion, blur kernel and the noise level is proposed. Such a model, however, requires multiple corrupted LR frames as observations, and noise is modeled using the deviations of pixel values among these multiple registered frames. This approach, therefore, is valid for the more traditional multiframe SR problem, and is not applicable to the more difficult single image SR, which is considered in this chapter.

An edge-based SR algorithm is presented in [49], wherein the image is super-resolved by learning the transformations of gradient profiles across resolutions. The authors acknowledge the sensitivity of their algorithm to noise, while computing gradients. For noisy images, they suggest denoising the image first, then using their algorithm to super-resolve the denoised image, then using bilinear interpolation to super-resolve the noise part, and then adding the noise back to the super-resolved image. However, doing so still retains the noise (in a bilinearly interpolated form) in the final result.

Among patch-based SR methods, [14] comments on the noise robustness of their method, which is based on sparse representation of image patches in a pre-trained dictionary of training patches. The authors argue that “overfitting” of noise in the SR procedure can be alleviated to some extent by using higher values of the $L_1$ regularization parameter. However, as seen in their results, such a strategy only yields moderate amounts of noise smoothing, and that too under relatively low noise conditions ($\sigma < 10$). Satisfactory levels of denoising under moderate to heavy noise variance levels cannot be achieved without severely smoothing out the image structures.

With respect to the abovementioned ideas, our algorithm has the following novelties: (1) Our algorithm is designed to explicitly handle high noise levels, and minimize the loss in image signal that would occur if denoising were to be performed as a preprocessing step. (2) Our algorithm is also designed to exploit noise as a substrate for synthesizing high-frequency textures in the final super-resolved image, which none of the abovementioned methods do.
4.3 Proposed Model

**Notation.** We use capital letters to denote images/matrices, as well as scalar constants, as appropriately defined. We use scripted letters ($S, U, B$ etc.) to denote operators, and/or sets, as appropriate. We use the tilde symbol to denote HR versions of LR images. Therefore, if $S$ is a super-resolution operator, $\tilde{I} = S(I)$. We denote indices using superscripts.

Consider a noisy observation $I_n = I + N(\sigma)$ of an LR image $I$ under additive white Gaussian noise $N(\sigma)$ of variance $\sigma$. Our goal is to obtain the best estimate of the HR version $\tilde{I}$ of the noise-free image $I$.

Let $D$ be a denoising operator such that $I_{dn} = D(I_n)$. If $D(\sigma)$ is the signal loss caused by $D$, we can write, $I_{dn} = I - D(\sigma)$. Now, on super-resolving this denoised image $I_{dn}$ (and assuming the SR operation to be linear\(^2\)), we get,

$$\tilde{I}_{dn} = \tilde{I} - \tilde{D}(\sigma)$$  (4.1)

Here $\tilde{I}_{dn}$ denotes the denoised HR image. $\tilde{I}_{dn}$ is the result obtained using the conventional approach of denoising as a preprocessing step before super-resolving. Such an approach results in loss of signal, given by $\tilde{D}(\sigma)$.

Can we obtain a better estimate for $\tilde{I}$ than (4.1)? To answer this, let us now super-resolve the noisy LR image $I_n$,

$$\tilde{I}_n = \tilde{I} + \tilde{N}(\sigma)$$  (4.2)

Now, consider a new estimate $\tilde{I}_{new}$ of $\tilde{I}$ that is obtained by taking a convex combination of $\tilde{I}_{dn}$ and $\tilde{I}_n$,

$$\tilde{I}_{new} = (1 - A) \cdot \tilde{I}_{dn} + A \cdot \tilde{I}_n$$  (4.3)

where “$\cdot$” denotes the Hadamard or entry-wise product, and the weighting matrix $A$ contains values in $[0, 1]$. Substituting $\tilde{I}_{dn}$ and $\tilde{I}_n$ from (4.1) and (4.2),

$$\tilde{I}_{new} = (1 - A) \cdot \left[ \tilde{I} - \tilde{D}(\sigma) + A \cdot \tilde{N}(\sigma) \right]$$  (4.4)

$$= \tilde{I} - (1 - A) \cdot \tilde{D}(\sigma) + A \cdot \tilde{N}(\sigma)$$  (4.5)

$$= \tilde{I}_{dn} + A \cdot \tilde{D}(\sigma) + A \cdot \tilde{N}(\sigma)$$  (4.6)

\(^2\)We make this assumption to simplify our analysis and clarify the motivation of our algorithm. We discuss more about this assumption later in this section.
Figure 4.3: Summary of proposed approach for obtaining a noise-free HR image from a noisy LR image. Using a convex combination framework, our algorithm facilitates part-recovery and part-synthesis of lost textures. Our result (blue box) appears richer in texture as compared to the current state-of-the-art (red box).

We now compare this new estimate $\tilde{I}_{\text{new}}$ of (4.6), with the conventionally obtained image $\tilde{I}_{\text{dn}}$ in (4.1). We observe that in addition to $\tilde{I}_{\text{dn}}$ that is obtained by conventional processing, (4.6) contains two more terms: The first additive term, $A \cdot \tilde{D}(\sigma)$, recovers a fraction ($A$) of the underlying textural signal that is lost during the denoising step. The second term, $A \cdot \tilde{N}(\sigma)$, introduces high frequency (noisy) components into $\tilde{I}_{\text{new}}$. As we describe later, appropriately filtering the noisy components to align with underlying local image structure serves as a way to synthesize additional texture. In order to facilitate such texture synthesis, we reformulate the convex combination model of (4.3) in terms of orientation and frequency selective bands of the images [28]. Given an image $I$, let $\{B^{(r,s)}\}$, $r = 1, ..., R$, $s = 1, ..., S$
denote its responses to a filter bank consisting of \( S \) scales and \( R \) orientation bands per scale. We rewrite the model of (4.3) in terms of frequency bands as,

\[
\tilde{B}_{\text{new}}^{(r,s)} = (1 - A^{(r,s)}) \cdot \tilde{B}_{dn}^{(r,s)} + A^{(r,s)} \cdot \tilde{B}_{n}^{(r,s)}
\] (4.7)

Note that we have now replaced the weighting matrix \( A \), with a set of weighting matrices \( A^{(r,s)} \), one for each band \((r, s)\). We propose a further re-parameterization of \( A^{(r,s)} \) to the form,

\[
A^{(r,s)} = \alpha V \cdot W^{(r,s)}
\] (4.8)

As we discuss below, such a re-parameterization allows for incorporation of several prior constraints, without which determining the optimal coefficients for the convex linear combination of \( \tilde{I}_{dn} \) and \( \tilde{I}_{n} \) is difficult.

The matrix \( V \) with values in \([0, 1]\) is called the variance map, and for every pixel location in the scene, it measures the “texturerness” of the local neighborhood. We explain our procedure for its estimation in detail in Section 4.4. The variance map allows us to perform the linear mixing of (4.7) in a spatially selective manner. In smooth, textureless regions, \( V \) favors greater influence from the denoised HR image, since there is little textural loss expected in such regions.

Our convex combination model presents a trade-off: We see in (4.6) and (4.7) that choosing high values of the mixing weights would help recover more of the signal lost during denoising, but would also introduce more noise through \( \tilde{N}(\sigma) \) (present in \( \tilde{I}_{n} \)). We show through experiments in Section 4.5 that at any location in the image, denoising loss is prevalent only in the most dominant orientation bands. Therefore, instead of uniformly combining all orientation bands of \( \tilde{I}_{n} \) and \( \tilde{I}_{dn} \), it would suffice to combine only those bands corresponding to the dominant local texture orientation. The advantage of doing so is that only a filtered version of the noisy components from \( \tilde{I}_{n} \) would be introduced in the resulting image \( \tilde{I}_{\text{new}} \).

Such orientation selective addition of noisy components in fact serves to perceptually enhance the local texture. Indeed, this has been the key idea behind several “texture-from-noise” synthesis algorithms in the literature [50, 36, 51]. The matrices \( W^{(r,s)} \) allow us to perform the linear mixing in such a band selective manner. We elaborate more on this in Section 4.5.

The scalar parameter \( \alpha \in [0, 1] \) globally controls the relative weights of the overly smooth \( \tilde{I}_{dn} \) and the noisy \( \tilde{I}_{n} \) in the resultant linear combination \( \tilde{I}_{\text{new}} \). While \( V \) and \( W^{(r,s)} \) determine where to blend and which frequencies to blend, the scalar
parameter $\alpha \in [0, 1]$ determines how much to blend $\tilde{I}_{dn}$ and $\tilde{I}_n$. We choose an optimal $\alpha$ such that the resultant image $\tilde{I}_{\text{new}}$ obeys the kurtosis invariance properties of noise-free natural images [52]. We elaborate this procedure in Section 4.6.

Once we have determined the weights of the linear combination, we use (4.7) to combine the bands of $\tilde{I}_{dn}$ and $\tilde{I}_n$. The resulting bands are used to invert the bandpass decomposition to obtain our final result. Figure 4.3 summarizes our algorithm.

Note that our analysis in this section is based on the assumption that the SR algorithm used is linear, in the sense that (4.1) and (4.2) are true. However, most state-of-the-art SR algorithms are non-linear. The key difference between linear and non-linear SR algorithms is that the latter can synthesize new high-frequency components while linear SR methods can only reshape the existing spectrum. Our algorithm holds for non-linear SR algorithms as well, since our algorithm does not take away any advantage brought by a non-linear SR method, as these synthesized high-frequency components are retained in our result. For example, (4.2) can be generalized for a non-linear SR algorithm if we hypothesize a different decomposition of the form,

$$\tilde{I}_n = \tilde{I} + X$$  \hspace{1cm} (4.9)

Now, $\tilde{I}$ would contain the extra synthesized frequency components brought in by the non-linear SR algorithm. However, $X$ may not exactly be $\tilde{N}(\sigma)$ as in the case of using a linear SR algorithm. But $X$ would still be a noise-like signal (containing high frequencies) that can be exploited for texture synthesis as we propose. Note that we never have to actually compute the decomposition of (4.9), as our algorithm operates using only $\tilde{I}_n$ and $\tilde{I}_{dn}$. We have assumed linearity of SR in this section for ease of analysis and to simplify and make more clear the motivation of our algorithm.

### 4.4 Spatial Constraint

In this section we discuss the estimation of the variance map $V$, which can be easily computed as a by-product of any patch-based SR algorithm, without significant overhead. We first briefly outline two SR algorithms that we use in our work, and then explain how we obtain $V$ from each of them.
Figure 4.4: Given an LR image (a), we use a patch-similarity-based SR algorithm to obtain the HR image (c). In the process, we obtain the variance map (d), by computing the variance across multiple predictions obtained through overlapping patches, for every HR pixel (shaded square in (b)).

Figure 4.5: This plot shows that the calculated variance map values (as described in the text and in Fig. 4.4) of patches bear significance correlation with the signal loss in the patches (patch RMSE).
4.4.1 Super-Resolution Algorithms

The first SR algorithm we use follows the self-similarity principles described in [24, 25]. Given an LR image $I$, we first create a database of LR-HR image patches, from the image $I$ as follows: We first create an LR version $I_L$ of the input image using a filter-and-downsample operation as,

$$I_L = (f_{psf} \ast I) \downarrow$$

where $f_{psf}$ is an assumed point spread function. We then create two sets of image patches $\mathcal{P}_H$ and $\mathcal{P}_L$, that contain patches extracted from $I$ and the corresponding (smaller) patches from $I_L$ respectively. The sets $\mathcal{P}_L$ and $\mathcal{P}_H$ serve as our database of LR-HR training patches.

To super-resolve the given image $I$ to $\tilde{I}$, for every patch $p$ in $I$, we find its most similar patch $p_L$ in the LR set $\mathcal{P}_L$. Let $p_H \in \mathcal{P}_H$ be the HR patch corresponding to $p_L$. We place the patch $p_H$ in the same location as $p$, but in the HR domain, and repeat for all patches to obtain the HR image $\tilde{I}$.

Figure 4.1(b) schematically summarizes the above SR procedure. For details and variations of such a self-similarity-based SR framework, we refer the reader to [24, 25, 22, 43]. In this chapter, we call the above SR procedure “SsSR” (self-similarity-based super-resolution).

The second SR algorithm that we use in this work is the algorithm proposed in [14], that uses external training images to construct the LR-HR patch databases $\mathcal{P}_L$ and $\mathcal{P}_H$. For better generalization, $\mathcal{P}_L$ and $\mathcal{P}_H$ need to contain a very large number of patches from a large number of training images. Since nearest neighbor searches can become expensive for large databases, the authors of [14] propose to learn compact dictionaries from $\mathcal{P}_L$ and $\mathcal{P}_H$ that support sparse representations for all the training patches. Patches of a test image are super-resolved using linear combinations of atoms from these learned dictionaries. We refer the reader to [14, 12] for details. We call this SR algorithm “ScSR” (sparse-coding-based super-resolution).

4.4.2 Variance Map Estimation

Both the above SR procedures effectively replace each patch of the LR image with an HR patch computed using the database $\mathcal{P}_H$. To avoid blocking artifacts,
overlap is allowed between the extracted patches. Therefore, if the patch size is 7-by-7, each pixel in the HR image would belong to 49 overlapping patches, and would receive 49 predictions during the SR process. In textural regions, these multiple explanations for the pixel are likely to be inconsistent since finding high quality patch-matches in textured regions is difficult [21]. Therefore, the variance of the multiple predictions of a pixel obtained during the SR procedure serves as a measure of the textural content of its local neighborhood. We compute this variance across all the pixels in the HR image and normalize the values to lie between 0 and 1 to thus obtain the variance map $V$. Figure 4.4 illustrates this procedure with an example.

We now verify through an experiment that $V$ does indeed indicate pixels where signal loss occurs in the denoised SR image. We obtain 50 images from the Berkeley segmentation database [53], downsample them by a factor of two, and add Gaussian noise. This creates set of noisy LR observations. We then denoise the images using the BM3D algorithm [40], and super-resolve the denoised images using the algorithm presented above, to yield the denoised HR images. In the process, we also obtain the variance maps for each image. We then extract around 1000 $7 \times 7$ patches from all the variance maps. For each patch, we plot its average variance map value against its intensity domain RMSE value (difference between the denoised HR image and the ground truth image). In Fig. 4.5, we show the resulting scatter plot. Clearly, there is a strong correlation between the values in the variance map and the amount of signal lost in the denoised HR image. Regions with high values in the variance map lose more signal and are therefore expected to benefit more using our proposed convex combination model of (4.3), justifying our use of $V$ in (4.8).
Figure 4.7: Average distribution of patch energy, across orientation and scale, for the denoised HR image ($\tilde{I}_{dn}$), noisy HR image ($\tilde{I}_n$) and the ground truth HR image. The signal lost in $\tilde{I}_{dn}$ as compared to the ground truth is primarily in the first few largest orientation bands.
4.5 Frequency Domain Constraint

In this section we discuss the estimation of the parameters $W^{(r,s)}$ that facilitates frequency and orientation band selective blending.

We first examine the behavior of signal power in small, textured patches of $\tilde{I}_{dn}$ and $\tilde{I}_n$, across oriented frequency bands. We again use 50 images from the Berkeley database and create sets of noisy HR and denoised HR images, along with their variance maps. We then compute a steerable pyramid decomposition for each image in the two sets. The steerable pyramid provides jointly localized (space/frequency) representation of images using an invertible multi-scale, multi-orientation image decomposition [28, 29, 54], as shown in Fig. 4.6. We use $S = 4$ scales and $R = 16$ orientations for the decomposition. We then extract around 1000 patches of size $7 \times 7$ across all bands from the 50 images, from areas containing significant textures ($V > 0.5$). We compute the average energies in these patches, in the different orientation and scale bands. To achieve rotation invariance, for each patch we sort the orientation bands in decreasing order of energy before averaging.

Figure 4.7 shows the average distribution of energy across two scales and all orientation bands, for patches from the denoised HR images (red bars), the noisy HR images (green bars) and the corresponding ground truths (blue bars).

We make a simple yet important observation: Signal loss is most prevalent in the orientation bands with higher energies. In the high-energy bands, we observe that the ground truth bands lie within the convex hulls of the corresponding denoised HR and noisy HR bands. This, in a way, further justifies our convex combination model of (4.3).

Based on this, we propose the following technique for choosing the weight matrices $W^{(j)}$, given a noisy HR image $\tilde{I}_n$ and the denoised HR image $\tilde{I}_{dn}$: For any spatial location $x$, we first consider a patch centered at $x$ in the image $\tilde{I}_{dn}$. Let $B^{(s)}_\lambda(x)$ be the set of the most dominant orientation bands in the scale $s$ in this image patch, as shown in Fig. 4.8. This set is determined by a scalar parameter $\lambda \in (0,1)$ that controls the fraction (in terms of energy) of the total number of orientation bands, that are present in the set $B^{(s)}_\lambda(x)$. For the location $x$, we assign $W^{(r,s)}(x)$ the following binary valued weight:

$$W^{(r,s)}(x) = \begin{cases} 1 & \text{if } r \in B^{(s)}_\lambda(x) \\ 0 & \text{else} \end{cases} \quad (4.11)$$
Figure 4.8: Given a patch (red box) at any location $x$, and its oriented bandpass decomposition, the set $B^{(s)}_\lambda(x)$ contains the most dominant orientation bands in the patch. The proposed convex combination is selectively done only on these bands.

The above weights effectively allow for blending the images $\tilde{I}_{dn}$ and $\tilde{I}_n$ only along the most dominant orientations of $\tilde{I}_{dn}$. These are the bands where maximum signal loss occurs. As far as the noise in $\tilde{I}_n$ is concerned, it is also added into $\tilde{I}_{new}$, only along the direction of the underlying texture. Adding noise which is filtered along the texture orientation has the effect of perceptually enhancing the texture. We illustrate this in Fig. 4.9. In this simplified example, since the third orientation band has the highest energy among all bands in $\tilde{I}_{dn}$, it is combined with the corresponding band of $\tilde{I}_n$ to obtain the band for $\tilde{I}_{new}$. The other bands of $\tilde{I}_{new}$ are simply copied from $\tilde{I}_{dn}$. The resulting patch $\tilde{I}_{new}$ appears richer in texture than $\tilde{I}_{dn}$.

Figure 4.9: A simplified example showing how the orientation bands of $\tilde{I}_{dn}$ and $\tilde{I}_n$ are combined to get $\tilde{I}_{new}$. Since the third orientation band has the most energy in $\tilde{I}_{dn}$, the convex combination is performed on this band. Although, in this process, noise is also introduced from $\tilde{I}_n$, it is done so only along the texture orientation. This enhances the texture in $\tilde{I}_{new}$.
4.6 Global Kurtosis Constraint

We now discuss the estimation of the scalar parameter $\alpha$ of (4.8). A low value results in an overly smooth image (close to $\tilde{I}_{dn}$), whereas high values may result in excessive high-frequency content.

To optimally choose $\alpha$, we again resort to the statistical behavior of natural images across bandpass decompositions. It is well known that the marginal responses of natural images to bandpass filters is highly non-Gaussian [55, 56]. This deviation from the Gaussian model can be measured by the kurtosis of the responses. In fact, studies have shown that the kurtosis of natural images remains constant across different frequency bands [57, 52].

Kurtosis of a distribution is defined as, $\kappa = \frac{\mu_4}{\sigma^4} - 3$, where $\mu_4$ is the fourth moment about the mean, and $\sigma$ is the standard deviation of the distribution. By this definition, the kurtosis of a Gaussian is zero. It has been shown in [52] that in noisy images, the kurtosis values in higher frequency bands are smaller than those in lower frequencies. This is indeed expected since noise (which predominantly affects higher frequency bands) has Gaussian statistics, and therefore has the overall effect of reducing kurtosis. We observe that on the other hand, excessive smoothing dramatically increases the kurtosis values of the high-frequency bands.

We propose to choose the $\alpha$ that results in minimum variation of the kurtosis values across bands. Let $\kappa_{new}^{(r,s)}(\alpha)$ be the kurtosis value of the band $\tilde{B}_{new}^{(r,s)}$ of our image $\tilde{I}_{new}$. We obtain the optimum $\alpha$ as,

$$
\alpha^* = \arg\min_{0 \leq \alpha \leq 1} \sum_{r,s} \left[ \kappa_{new}^{(r,s)}(\alpha) - \bar{\kappa}_{new}(\alpha) \right]^2
$$

(4.12)

$\bar{\kappa}_{new}(\alpha)$ is the mean kurtosis value across all bands. We numerically solve the above optimization problem. Alternatively, one may use Matlab’s \textit{fminsearch} function.

4.7 Texture Adaptive Patch Averaging Perspective

So far in this chapter, we have presented our algorithm from a systems perspective, in the sense that denoising and SR are treated as independent black boxes. Such a perspective allowed us to think of denoising and SR blocks as abstractions, and
we motivated our algorithm in terms of minimizing the signal loss caused by the
denoising block, and utilizing the excess noise for texture enhancement. Naturally,
such a framework allows us to use different SR and denoising algorithms in our
technique. Indeed, we run our method using different combinations of state-of-
the-art denoising and SR algorithms, and show results for each of them in Section
4.8.

In this section, we comment on another viewpoint on our algorithm, that is
based on studying the relationship between denoising and SR from first principles.
As we had alluded to in Section 4.1, a typical patch-based denoising algorithm
(such as NLM [23]) removes noise from a patch by seeking several similar patches
of the same scale and averaging them out. On the other hand, a patch-based SR
algorithm looks for just one (or a few) similar patch(es), and maps them to a
finer scale, to obtain the super-resolved version of the patch. From these first
principles, one could hypothesize a “joint denoising + SR” algorithm, wherein a
patch is simultaneously denoised and super-resolved by seeking a large number
of similar patches, mapping them to a finer scale, and averaging them out. Such
a procedure is illustrated in Fig. 4.10. The key difference between an “SR only”
algorithm such as in Fig. 4.1(b), and the joint denoising + SR algorithm of Fig.
4.10, is the number of patches sought and averaged for each input patch.

Figure 4.11 shows the effect of using successively larger number of patch matches,
or nearest neighbors (NN) while performing super-resolution of a noisy LR image.
We see clearly that as the number of NNs are increased, there is greater averaging
or smoothing out of the noise. However, averaging out a large number of patches
also tends to smooth out the useful high-frequency content of the image such as
the boy’s hair, etc. This essentially leads to similar problems of “denoising loss”,
if denoising were performed as an independent preprocessing step, before using
an SR algorithm.

Figure 4.10: Schematic figure summarizing a “joint denoising + SR” scheme,
where each patch is simultaneously denoised and super-resolved by seeking a
large number of patch-matches of a bigger scale and averaging.
Figure 4.11: \textit{Boy}. Result(s) of performing SR on a noisy LR image, using different numbers of nearest neighbor patch-matches. As can be seen, if the number of nearest neighbors to be averaged is increased, noise is smoothed out, but high-frequency components of the underlying image are also lost. If the number of nearest neighbors is chosen adaptively, based on local textural analysis (using (4.14)), the high-frequency components of the image are better preserved, as can be seen.
How is our proposed algorithm related to the joint SR + denoising framework as described above? Our final result is obtained by borrowing signal components from the noisy super-resolved image, and the denoised super-resolved image, adaptively, by examining local texture properties. Therefore, in terms of the joint denoising + SR framework, our algorithm amounts to choosing the number of patches to be averaged in an adaptive manner, based on the same local texture analysis.

Our proposed algorithm can therefore be viewed as an improved version of a joint denoising + SR algorithm, where the number of patches to be averaged varies with image location, as well as frequency sub-bands as follows:

\[
N^{(r,s)} \propto V \cdot W^{(r,s)}
\]

where the matrix \( N^{(r,s)} \) determines, for each location and sub-band \((r, s)\) of the image, the number of sub-band patch-matches to be averaged. \( V \) and \( W^{(r,s)} \) are the spatial and frequency domain constraint matrices of our earlier model. The proportionality constant \( \beta \) can again be determined by enforcing the global kurtosis constraints of noise-free natural images as described in Section 4.6. Figure 4.11 shows the result of choosing the number of patch matches (nearest neighbors) adaptively using (4.14), as compared to using a fixed number.

Although most of this chapter describes our algorithm using the convex combination model that tries to minimize denoising loss while simultaneously exploiting noise to enhance texture, the interpretation presented in this section may help to highlight its links to the conventional patch-based denoising and SR algorithms, based on first principles.

4.8 Results

4.8.1 Implementation Details

We implement our method with both non-local means (NLM) [23] and BM3D [40] denoising algorithms, and the SsSR and ScSR super-resolution algorithms that were described in Section 4.4. We denote the baseline algorithms as NLM-SsSR, BM3D-SsSR, etc., which indicates first denoising with the specified de-
Figure 4.12: Lenna. The plot shows the kurtosis values across bandpass decompositions, for the denoised HR image ($\tilde{I}_{dn}$), noisy HR image ($\tilde{I}_n$) and our result ($\tilde{I}_{new}$). Higher component numbers correspond to higher frequency bands. The images above the plot shows visual comparison of the results. Textures are better recovered in our image. The numbers in parantheses denote PSNR in dB and SSIM [35].
Figure 4.13: *Lama.* Textures on the fur, and on rocks in the background are much better reconstructed in our results as compared to the BM3D-SsSR and BM3D-ScSR baselines. The numbers in parantheses denote PSNR in dB and SSIM [35].
Figure 4.14: Baby. The woolen cap in our results is significantly richer in texture as compared to the NLM-SsSR and NLM-ScSR baselines. The numbers in parantheses denote PSNR in dB and SSIM [35].
noising algorithm, followed by super-resolving with the specified SR algorithm. We indicate the results of our method as “Ours (NLM, SsSR)”, etc., which indicates that our algorithm uses the NLM denoising algorithm and the SsSR super-resolution algorithm, etc. Since we use two denoising algorithms and two SR algorithms, we have a total of four variants of our algorithm, along with four corresponding baselines to compare with.

Both NLM and BM3D denoising algorithms require the noise variance as an input. Although most of our noisy images are simulated by adding noise of known variance, we use the algorithm of [58] to estimate noise variance from the noisy images. This is then fed to the denoising algorithms. In all our images, we found the estimated variance to be within ±5% of the true variance. We try several different noise levels in our experiments.

We use $S = 4$ scales in the steerable pyramid decomposition. Scale levels of 5 or more required much larger images. For each scale, we compute decompositions along $R = 16$ orientation bands, which is the maximum allowable in the available implementation by Simoncelli. We set the band energy threshold parameter $\lambda = 0.6$ in most cases, but we also study the effects of changing it in Section 4.8.3.

4.8.2 Qualitative Results

We first show our result on the Lenna image in Fig. 4.12. We plot the kurtosis values of our result across all frequency bands, and compare it to those of the denoised HR image (red markers) and the noisy HR image (green markers). Higher component numbers correspond to higher frequency bands. Due to noise, the kurtosis values in the higher frequencies of the noisy HR image are low, whereas they are very high for the denoised HR image. Subject to our constraints, our algorithm yields kurtosis values as shown by the black markers. Figure 4.12 also shows our resulting image. Textural details are better preserved as compared to the denoised HR image (BM3D-SsSR), both visually and quantitatively. Noise variance was $\sigma = 20$ in this experiment.

Figure 4.13 shows the results of the (BM3D, SsSR) and (BM3D, ScSR) variants of our algorithm on the Lama image. We see that textural details such as the fur, and the rocks behind are significantly well preserved as compared to the BM3D-SsSR and BM3D-ScSR baselines.

Figure 4.14 shows the results of the (NLM, SsSR) and (NLM, ScSR) variants
of our algorithm on the Baby image. As compared to the NLM-SsSR and NLM-ScSR baselines, we recover significantly more texture in the woolen cap.

Figure 4.15 shows the results of all the variants of our algorithm on the Horse image. Textures like the grass and the horse fur are visually and quantitatively better recovered by our approach, using either NLM or BM3D, with either SsSR or ScSR. We observe that using BM3D as the denoising algorithm gives slightly better results.

Figure 4.16 shows similar results on the Dog image, where we try all four variants of our algorithm. Our results improve over the corresponding baseline in all four cases. Textures on the dog fur, grass, and the wooden pole are better in our results.

4.8.3 Quantitative Analysis

We use 50 natural images from the Berkeley segmentation database [53] for quantitative analysis. We run our algorithm(s) on these images and compute PSNR and SSIM [35]. For the sake of clarity in presentation, we show quantitative results for only two variants of our algorithm, viz., (BM3D, SsSR) and (NLM, SsSR). We have observed similar trends for the other two variants as well.

We first analyze the quantitative performance of our algorithm(s) with different noise levels. Figure 4.17 plots our results. We observe that our algorithms consistently improve over the conventional methods, across different noise levels. Figure 4.18 shows the visual results of varying noise on a test image. All our images are visually better as well.

In our algorithm, we have introduced a parameter \( \lambda \in (0, 1) \) that controls the fraction (in terms of energy) of the orientation bands that are involved in the blending procedure. A very low value would combine only the first few (largest) bands, resulting in improvement in texture only along these specific orientations. A higher value would combine a greater number of bands, resulting in better recovery of texture. However, an excessively high value of \( \lambda \) (e.g. close to 1), would tend to copy the noisy HR image “as is”, and may introduce noisy components in the resulting image. Indeed, this is what we observe quantitatively as well, as shown in Fig. 4.19; both PSNR and SSIM first increase with increasing \( \lambda \), and then drop slightly at around \( \lambda = 0.8 \). Nevertheless, throughout the range, our performance remains significantly higher than the baselines, as can be seen from...
Figure 4.15: *Horse.* For both NLM and BM3D, our algorithms significantly improve over the respective baselines, both visually and numerically. The grass, flowers and horse fur show significant visual improvement. The numbers in parentheses denote PSNR in dB and SSIM [35].
Figure 4.16: Dog. Using either self-similarity-based SR (SsSR), or sparse-coding-based SR (ScSR), our algorithm significantly improves over the respective baselines, both visually and numerically. The fur, grass and tree trunk show the most improvement visually. The numbers in parantheses denote PSNR in dB and SSIM [35].
Figure 4.17: The plots show average SSIM (left) and PSNR (right) as functions of noise variance. Our algorithm(s) consistently improve over their corresponding baselines for all noise levels tested.

the plots. Figure 4.20 shows the visual result of varying $\lambda$ on a test image.

4.8.4 Real-World Example

We demonstrate the practicality of our algorithm on a real world denoising+SR problem. We use our algorithm to enlarge a part of an image captured with a DSLR camera on a high ISO setting, resulting significant sensor noise in the image. Figure 4.21 shows the input image, the result of the BM3D-SsSR baseline and our result. While in smooth regions our result looks similar to the conventionally obtained result, in textured areas our image appears better.
Figure 4.18: Procupine. Our algorithm can be seen to be both visually and quantitatively better than the conventional approach for a range of noise levels.

4.8.5 Limitation

We have demonstrated that our algorithm outperforms the conventional processing strategy in regions containing stochastic textures. A limitation of our algorithm is in handling more regular (non-stochastic) textures (such as in Fig. 4.22). On close observation, we find that the regions containing regular, structural textures (red box) are still quite noisy in our result as compared to the baseline. Indeed, our approach of exploiting noise for texture enhancement using bandpass filtering fails for regular textures. A possible solution would be to use more elaborate texture segmentation or recognition techniques to systematically incorporate this high level information in our constraints. However, here we have restricted ourselves to testing the basic idea wherein the weighting parameters can be easily estimated.

4.9 Conclusion

We have discussed how noise and denoising algorithms affect the single image super-resolution problem. We have argued that although both denoising and SR
Figure 4.19: The figures plot average SSIM (left) and PSNR (right) as a function of the band energy threshold parameter $\lambda$. For a wide range of $\lambda$ values, our performance remains significantly higher than the corresponding baselines.

Figure 4.20: *Fur*. A small value of $\lambda$ results in relatively smaller (but still noticeable) improvement in results. A very high value recovers more texture but may also yield a relatively “noisier” image.
Figure 4.21: *Real-world example.* Although the estimated noise variance (5.3) in this real-world image is quite a bit lower than in any of our simulations, our result still shows perceptible improvement in visual quality as compared to the BM3D-SsSR baseline.
follow similar patch replacement strategies, they are geared towards different and mutually conflicting objectives. Although denoising seeks to remove just noise, some high frequency content of the underlying image is also invariably lost in the process. SR algorithms seek to synthesize higher frequency content for the image, based on the information that is available in the input LR image. The loss of high frequency content caused by denoising algorithms directly hinders the performance of the SR algorithm. In this chapter we have presented a framework that allows for performing noise removal and super-resolution in harmony. Our simple idea is to carefully utilize the high-frequency content from the noisy image (which is ordinarily removed by denoising algorithms) for the benefit of the SR process. Our algorithm, in part, reduces the denoising loss caused by conventionally preprocessing the image using denoising algorithms, and at the same time exploits noise present in the image as a substrate for synthesizing textures, thereby enhancing the image. Overall, our super-resolved images contain richer textural content, and appear more natural than those obtained conventionally.

Figure 4.22: Failure case. While our algorithm yields better results in regions of irregular/stochastic texture (green box), our approach does not do as well in regions containing regular textures (red box), where our result appears slightly more noisy than the baseline.
CHAPTER 5

SUPER-RESOLUTION USING TRANSFORMED SELF-EXEMPLARS

5.1 Introduction

While internal statistics have been successfully exploited for SR, in most algorithms the LR-HR patch pairs are found by searching only for “translated” versions of patches in the scaled-down images. This effectively assumes that an HR version of a patch appears in the same image at the desired scale, orientation and illumination. This amounts to assuming that the patch is planar and the images of the different assumed occurrences of the patch are taken by a camera translating parallel to the plane of the patch. This fronto-parallel imaging assumption is often violated due to non-planar shape of the patch surface, common in both natural and man-made scenes, as well as perspective distortion. Figure 5.1 shows three examples of such violations, where self-similarity across scales will hold better if suitable geometric transformation of patches is allowed.

Figure 5.1: Examples of self-similar patterns deformed due to local shape variation, orientation change, or perspective distortion.

In this chapter, we propose a self-similarity driven SR algorithm that expands the internal patch search space. First, we explicitly incorporate the 3D scene geometry by localizing planes, and use the plane parameters to estimate the perspective deformation of recurring patches. Second, we expand the patch search space to include affine transformation to accommodate potential patch deforma-
tion due to local shape variations. We propose a compositional transformation model to simultaneously handle these two types of transformations. We modify the PatchMatch algorithm [59] to efficiently solve the nearest neighbor field estimation problem. We validate our algorithm through a large number of qualitative and quantitative comparisons against state-of-the-art SR algorithms on a variety of scenes. We achieve significantly better results for man-made scenes containing regular structures. For natural scenes, our results are comparable with current state-of-the-art algorithms.

Our contributions in this chapter can be summarized in the following:

1. Our method effectively increases the size of the limited internal dictionary by allowing geometric transformation of patches. We achieve state-of-the-art results without using any external training images.

2. We propose a decomposition of the geometric patch transformation model into (i) perspective distortion for handling structured scenes and (ii) additional affine transformation for modeling local shape deformation.

3. We use and make available a new dataset of urban images containing structured scenes as a benchmark for SR evaluation.

5.2 Related Work

**Expanding patch search space:** Since internal dictionaries are constructed using only the given LR image, they tend to contain a much smaller number of LR-HR patch pairs compared to external dictionaries which can be as large as desired. In Chapters 2 and 3, we proposed orientation selective sub-band energies for better matching textural patterns [43] and also proposed to reduce the self-similarity based SR into a set of problems of matching simpler sub-bands of the image, amounting to an exponential increase in the effective size of the internal dictionary [44]. Zhu et al. [60] proposed to enhance the expressiveness of the dictionary by optical flow-based patch deformation during searching, to match the deformed patch with images in external databases. We use projective transformation to model the deformation common in urban scenes to better exploit internal self-similarity. Fernandez-Granda and Candès [61] super-resolved a planar regions by factoring out perspective distortion and imposing group-sparse regularization over
image gradients. Our method also incorporates 3D scene geometry for SR, but we can handle multiple planes and recover regular textural patterns beyond orthogonal edges through self-similarity matching. In addition, our method is a generic SR algorithm that handles both man-made and natural scenes in one framework. In the absence of any detected planar structures, our algorithm automatically falls back to searching only affine transformed self-exemplars for SR.

Our work is also related to several recent approaches that solve other low-level vision problems using over-parameterized (expanded) patch search spaces. Although more difficult to optimize than 2D translation, such over-parametrization often better utilizes the available patch samples by allowing transformations. Examples include stereo [62], depth upsampling [63], optical flow [64], image completion [65], and patch-based synthesis [66]. Such expansion of the search space is particularly suited for the SR problem due to the limited size of internal dictionaries.

5.3 Overview

Super-resolution scheme: Given an LR image \( I \), we first blur and subsample it to obtain its downsampled version \( I_D \). Using \( I \) and \( I_D \), our algorithm to obtain an HR image \( I_H \) consists of the following steps:

1. For each patch \( P \) (target patch) in the LR image \( I \), we compute a transformation matrix \( T \) (homography) that warps \( P \) to its best matching patch \( Q \) (source patch) in the downsampled image \( I_D \), as illustrated in Fig. 5.2 (c). To obtain the parameters of such a transformation, we estimate a nearest neighbor field between \( I \) and \( I_D \) using a modified PatchMatch algorithm [59] (details given in Section 5.4).

2. We then extract \( Q_H \) from the image \( I \), which is the HR version of the source patch \( Q \).

3. We use the inverse of the computed transformation matrix \( T \) to “unwarp” the HR patch \( Q_H \), to obtain the self-exemplar \( P_H \), which is our estimated HR version of the target patch \( P \). We paste \( P_H \) in the HR image \( I_H \) at the location corresponding to the LR patch \( P \).
Figure 5.2: Comparison with external dictionary and internal dictionary (self-similarity) approaches. Middle row: Given LR image $I$. Our method allows for geometrically transforming the target patch from the input image, while searching for its nearest neighbor in the downsampled image. The HR version of the best match found is then pasted on to the HR image. This is repeated for all patches in the input image $I$.

4. We repeat steps 1-3 for all target patches to obtain an estimate of the HR image $I_H$.

5. We run the iterative backprojection algorithm [31] to ensure that the estimated $I_H$ satisfies the reconstruction constraint with the given LR observation $I$.

Figure 5.2 schematically illustrates the important steps in our algorithm, and compares it with other frameworks.

**Motivation for using transformed self-exemplars:** The key step in our algorithm is the use of the transformation matrix $T$ that allows for geometric deformation of patches, instead of simply searching for the best patches under translation. We justify the use of transformed self-exemplars with two illustrative examples in Fig. 5.3. Matching using the affine transformation and planar perspective transformation achieves both lower matching errors and more accurate prediction of the HR content than that from matching patches under translation.
Figure 5.3: Examples demonstrating the need for using transformed self-exemplars in our self-similarity based SR. Red boxes indicate a selected target patch (to be matched) in the input LR image $I$. We take the selected target patch, remove its mean, and find its nearest neighbor in the downsampled image $I_D$. We show the error found while matching patches in $I_D$ in the second column. Blue boxes indicate the nearest neighbor (best matched) patch found among only translational patches, and green boxes indicate the nearest neighbor found under the proposed (a) affine transformation and (b) planar perspective transformation. In the third and fourth columns we show the matched patches $Q$ in the downsampled images $I_D$ and their HR version $Q_H$ in the input image $I$.

5.4 Nearest Neighbor Field Estimation

5.4.1 Objective Function

Let $\Omega$ be the set of pixel indices of the input LR image $I$. For each target patch $P(t_i)$ centered at position $t_i = (t_i^x, t_i^y)^\top$ in $I$, our goal is to estimate a transformation matrix $T_i$ that maps the target patch $P(t_i)$ to its nearest neighbor in the downsampled image $I_D$. A dense nearest neighbor patch search forms a nearest neighbor field (NNF) estimation problem. In contrast to the conventional 2D translation (or offsets) field, here we have a field of transformations parametrized by $\theta_i$ for the $i^{th}$ pixel in the input LR image. Our objective function for this NNF estimation problem takes the form

$$\min_{\{\theta_i\}} \sum_{i \in \Omega} E_{\text{app}}(t_i, \theta_i) + E_{\text{plane}}(t_i, \theta_i) + E_{\text{scale}}(t_i, \theta_i)$$ (5.1)

where $\theta_i$ is the unknown set of parameters for constructing the transformation matrix $T_i$ that we need to estimate (in a way explained later in this section). Our objective function includes three costs: (1) appearance cost, (2) plane cost, and
(3) scale cost. In the following we first describe each of these costs.

**Appearance cost** $E_{\text{app}}$: This cost measures similarity between the sampled target and source patches. We use Gaussian-weighted sum-of-squared distance in the RGB space as our metric:

$$E_{\text{app}}(t_i, \theta_i) = ||W_i (P(t_i) - Q(t_i, \theta_i)) ||^2_2$$ (5.2)

where the matrix $W_i$ is the Gaussian weights with $\sigma^2 = 3$, $Q(t_i, \theta_i)$ denotes the sampled patch from $I_D$ using the transformation $T_i$ with parameter $\theta_i$.

We now present how we design and construct the transformation matrix $T_i$ from the estimated parameter $\theta_i$ for sampling the source patch $Q(t_i, \theta_i)$. The geometric transformation of a patch in general can have up to 8 degrees of freedom (i.e., a projective transformation). One way to estimate the patch geometric transformation is to explicitly search in the additional patch space (e.g., scale, rotation) [67, 68, 66] beyond translation. However, perspective distortion can only be approximated by scaling, rotation and shearing of affine transformations. Therefore, affine transformations by themselves are less effective in modeling the appearance variations in man-made, structured scenes. Huang et al. [65] addressed this problem by detecting planes (and their parameters) and using them to determine the perspective transformation between the target and source patch. In Fig. 5.4, we show a visualization of vanishing point detection and posterior probability map for detection of planes, as yielded by [65].

We combine the explicit search strategy of [67, 68, 66], along with the perspective deformation estimation approach of [65]. Using the algorithm of [65], we detect and localize planes and compute the planar parameters, as shown by the example in Fig. 5.4. We propose to parameterize $T_i$ by $\theta_i = (s_i, m_i)$, where
\( s_i = (s_{ix}, s_{iy}, s_{is}, s_{i\theta}, s_{i\alpha}, s_{i\beta}) \) is the 6-dimensional affine motion parameter of the source patch and \( m_i \) is the detected plane index (using [65]). We propose a factored geometric transformation model \( T_i(\theta_i) \) of the form:

\[
T_i(\theta_i) = H(t_i, s_{ix}, s_{iy}, m_i) S(s_{is}, s_{i\theta}) A(s_{i\alpha}, s_{i\beta})
\]

where the matrix \( H \) captures the perspective deformation given the target and source patch positions and the planar parameters (as described in [65]). The matrix

\[
S(s_{is}, s_{i\theta}) = \begin{bmatrix}
    s_{is} R(s_{i\theta}) & 0 \\
    0^\top & 1
\end{bmatrix}
\]

(5.4)
captures the similarity transformation through a scaling parameter \( s_{is} \) and a \( 2 \times 2 \) rotation matrix \( R(s_{i\theta}) \), and the matrix

\[
A(s_{i\alpha}, s_{i\beta}) = \begin{bmatrix}
    1 & s_{i\alpha} & 0 \\
    s_{i\beta} & 1 & 0 \\
    0 & 0 & 1
\end{bmatrix}
\]

(5.5)
captures the shearing mapping in the affine transformation.

The proposed compositional transformation model resembles the classical decomposition of a projective transformation matrix into a concatenation of three unique matrices: similarity, affine, and pure perspective transformation [69]. Yet, our goal here is to “synthesize”, rather than “analyze” the transformation \( T_i \) for sampling source patches. The proposed formulation allows us to effectively factor out the dependency of the positions of the target \( t_i \) and source patch \( (s_{ix}, s_{iy}) \) for estimating the perspective deformation in \( H(t_i, s_{ix}, s_{iy}, m_i) \) from estimating affine shape deformation parameters using \( (s_{is}, s_{i\theta}, s_{i\alpha}, s_{i\beta}) \) for matrices \( S \) and \( A \). This is crucial because we can then exploit piecewise smoothness characteristics in natural images for efficient nearest neighbor field estimation.

**Plane compatibility cost** \( E_{\text{plane}} \): For man-made images, we can often reliably localize planes in the scene using standard vanishing point detection techniques. The detected 3D scene geometry can be used to guide the patch search space. We modify the plane localization code in [65] and add a plane compatibility cost to encourage the search over the more probable plane labels for source and target.
patches.

\[ E_{\text{plane}} = -\lambda_{\text{plane}} \log \left( \Pr[m_i|(s^x_i, s^y_i)] \times \Pr[m_i|(t^x_i, t^y_i)] \right) \quad (5.6) \]

where the \( \Pr[m_i|(x, y)] \) is the posterior probability of assigning label \( m_i \) at pixel position \( (x, y) \) (see Fig. 5.4 (b) for an example).

**Scale cost** \( E_{\text{scale}} \): Since we allow continuous geometric transformations, we observed that the nearest neighbor field often converged to the trivial solution, i.e., matching target patches to itself in the downsampled image \( I_D \). Such a match has small appearance cost. This trivial solution leads to the conventional bicubic interpolation for SR. We avoid such trivial solutions by introducing the scale cost \( E_{\text{scale}} \):

\[ E_{\text{scale}} = \lambda_{\text{scale}} \min(0, SRF - \text{Scale}(T_i)) \quad (5.7) \]

where SRF indicates the desired SR factor, e.g., 2x, 3x, or 4x, and the function \( \text{Scale}(\cdot) \) indicates the scale estimation of a projective transformation matrix. We approximately estimate the scale of the source patch sampled using \( T_i \) with the first-order Taylor expansion [70]:

\[
\text{Scale}(T_i) = \sqrt{\det \begin{bmatrix} T_{1,1} - T_{1,3}T_{3,1} & T_{1,2} - T_{1,3}T_{3,2} \\ T_{2,1} - T_{2,3}T_{3,1} & T_{3,1} - T_{2,3}T_{3,2} \end{bmatrix}}
\]

where \( T_{u,v} \) indicates the value of \( u^{th} \) row and \( v^{th} \) column in the transformation matrix \( T_i \) with \( T_{3,3} \) normalized to one. Intuitively, we penalize if the scale of the source patches is too small. Therefore, we encourage the algorithm to search for source patches that are similar to the target patch and at the same time to have larger scale in the input LR image space, and therefore are able to provide more high-frequency details for SR. We soft-threshold the penalty to zero when the scale of the source patch is sufficiently large.

5.4.2 Inference

We need to estimate the 7-dimensional \( (\theta_i \in \mathbb{R}^7) \) nearest neighbor field solutions over all overlapping target patches. Unlike the conventional self-exemplar-based methods [24, 25], where only a 2D translation field needs to be estimated, the
solution space in our formulation is much more difficult to search. We modify the PatchMatch [59] algorithm for this task with the following detailed steps.

**Initialization:** Instead of the random initialization done in PatchMatch [59]. We initialize the nearest neighbor field with zero displacements and scales equal to the desired SR factor. This is inspired by [25, 71], suggesting that good self-exemplars can often be found in a localized neighborhood. We found that this initialization strategy provides a good start for faster convergence.

**Propagation:** This step efficiently propagates good matches to neighbors. In contrast to propagating the transformation matrix $T_i$ directly, we propagate the parameter $\theta_i = (s_i, m_i)$ instead so that the affine shape transformation is invariant to the source patch position.

**Randomization:** After propagation in each iteration, we perform a randomized search to refine the current solution. We simultaneously draw random samples of the plane index based on the posterior probability distribution, randomly perturb the affine transformation and randomly sample position (in a coarse-to-fine manner) to search for the optimal geometric transformation of source patches and reduce the matching errors.

### 5.5 Experiments

**Datasets:** Yang et al. [72] recently proposed a benchmark for evaluating single image SR methods. Most images therein consist of natural scenes such as landscapes, animals, and faces. Images that contain indoor, urban, architectural scenes etc., rarely appear in this benchmark. However, such images feature prominently in consumer photographs. We therefore have created a new dataset *Urban 100* containing 100 HR images with a variety of real-world structures. We constructed this dataset using images from Flickr (under CC license) using keywords such as urban, city, architecture, and structure.

In addition, we also evaluate our algorithm on the *BSD 100* dataset, which consists of 100 test images of natural scenes taken from the Berkeley segmentation dataset [53]. For this dataset, we evaluate for SR factors of 2x, 3x, and 4x.

**Methods evaluated:** We compare our results against several state-of-the-art SR algorithms. Specifically, we choose four SR algorithms trained using a large number of external LR-HR patches for training. The algorithms we use are: Kernel rigid regression (Kim) [32], sparse coding (ScSR) [14], adjusted anchored neigh-
bor neighbor regression (A+) [17], and convolutional neural networks (SRCNN) [18]. We also compare our results with those of the internal dictionary-based approach (Glasner) [24] and the sub-band self-similarity SR algorithm (Sub-Band) [44] that we proposed in Chapter 2.

**Implementation details:** We use $5 \times 5$ patches and perform SR in multiple steps. We achieve 2x, 3x, 4x SR factors in three, five and six upscaling steps, respectively. At the end of each step, we run 20 iterations of the backprojection algorithm [31] with a $5 \times 5$ Gaussian filter with $\sigma^2 = 1.2$. The NNF solution from a coarse level is upsampled and used as an initialization for the next-finer level. We empirically set the parameters $\lambda_{\text{plane}} = 10^{-3}$ and $\lambda_{\text{scale}} = 10^{-3}$. The parameters are kept fixed for all our experiments.

**Qualitative evaluation:** In Figs. 5.5 and 5.6, we show visual results on images from the Urban 100 dataset. We find that our method is capable of recovering structured details that were missing in the LR image by properly exploiting the internal similarity in the LR input. Other approaches, using external images for training, often fail to recover these structured details. Our algorithm well exploits the detected 3D scene geometry and the internal natural image statistics to super-resolve the missing high-frequency contents. In Figs. 5.7 and 5.8, we demonstrate that our algorithm is not restricted to images of a single plane scene. We are able to automatically search for multiple planes and estimate their perspective and affine transformations to robustly predict the HR image.

In Figs. 5.9 and 5.10, we show two results on natural images where no regular structures can be detected. In such cases, our algorithm reduces to searching for affine transformations only in the nearest neighbor field, similar to [67]. On natural images without any particular geometric regularity, our method performs well as the recent, state-of-the-art methods such as [18, 17], although, as can be seen in both examples, our results contain slightly sharper edges and fewer artifacts such as ringing.

**Quantitative evaluation:** We also perform quantitative evaluation of our method in terms of PSNR (dB) and structural similarity (SSIM) index [35] (computed using luminance channel only). Since such quantitative metrics may not correlate well with visual perception, we invite the reader to examine the visual quality of our results for better evaluation of our method.

Table 5.1 shows the quantitative results on the Urban 100 and BSD 100 datasets. Numbers in red indicate the best performance and those in blue indicate the second-best performance. Our algorithm yields the best quantitative results for this dataset,
Table 5.1: Quantitative evaluation on *Urban 100* and *BSD 100* datasets. Red indicates the best and blue indicates the second-best performances.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (2x)</td>
<td>2x</td>
<td>26.66</td>
<td>28.26</td>
<td>28.74</td>
<td>28.65</td>
<td>28.87</td>
<td>28.34</td>
<td>28.15</td>
<td>29.05</td>
</tr>
<tr>
<td></td>
<td>4x</td>
<td>23.14</td>
<td>24.02</td>
<td>24.20</td>
<td>24.14</td>
<td>24.34</td>
<td>24.21</td>
<td>23.79</td>
<td>24.67</td>
</tr>
<tr>
<td>SSIM (2x)</td>
<td>2x</td>
<td>0.8408</td>
<td>0.8828</td>
<td>0.8940</td>
<td>0.8909</td>
<td>0.8957</td>
<td>0.8820</td>
<td>0.8743</td>
<td>0.8980</td>
</tr>
<tr>
<td></td>
<td>4x</td>
<td>0.6573</td>
<td>0.7024</td>
<td>0.7104</td>
<td>0.7047</td>
<td>0.7195</td>
<td>0.7115</td>
<td>0.6838</td>
<td>0.7314</td>
</tr>
<tr>
<td>PSNR (BSD)</td>
<td>2x</td>
<td>29.55</td>
<td>30.77</td>
<td>31.11</td>
<td>31.11</td>
<td>31.22</td>
<td>30.73</td>
<td>30.56</td>
<td>31.12</td>
</tr>
<tr>
<td></td>
<td>3x</td>
<td>27.20</td>
<td>27.72</td>
<td>28.17</td>
<td>28.20</td>
<td>28.30</td>
<td>27.88</td>
<td>27.36</td>
<td>28.20</td>
</tr>
<tr>
<td>SSIM (BSD)</td>
<td>2x</td>
<td>0.8425</td>
<td>0.8744</td>
<td>0.8840</td>
<td>0.8835</td>
<td>0.8862</td>
<td>0.8774</td>
<td>0.8675</td>
<td>0.8835</td>
</tr>
<tr>
<td></td>
<td>3x</td>
<td>0.7382</td>
<td>0.7647</td>
<td>0.7788</td>
<td>0.7794</td>
<td>0.7836</td>
<td>0.7714</td>
<td>0.7490</td>
<td>0.7778</td>
</tr>
<tr>
<td></td>
<td>4x</td>
<td>0.6672</td>
<td>0.6983</td>
<td>0.7027</td>
<td>0.7018</td>
<td>0.7089</td>
<td>0.7021</td>
<td>0.6842</td>
<td>0.7064</td>
</tr>
</tbody>
</table>
Figure 5.5: Visual comparison for 4x SR. Our method is able to explicitly identify perspective geometry to better super-resolve details of regular structures occurring in various urban scenes.

0.2-0.3 dB PSNR better than the second-best method (A+) [17] and 0.4-0.5 dB better than the recently proposed SRCNN [18]. We are able to achieve these results without any training databases, while both [17] and [18] require millions of external training patches. Our method also outperforms the self-similarity approaches of [24] and [44], validating our claim of being able to extract better
Figure 5.6: Visual comparison for 4x SR. Our method is able to explicitly identify perspective geometry to better super-resolve details of regular structures occurring in various urban scenes.

internal statistics through the expanded internal search space. In the BSD 100 dataset our results are comparable to those obtained by other approaches on this dataset, with ≈ 0.1 dB lower PSNR than the results of A+ [17]. Our quantitative results are slightly worse than the state-of-the-art in this dataset since it is difficult to find geometric regularity in such natural images, which our algorithm seeks to exploit. Also A+ [17] is trained on patches that contain natural textures quite
Figure 5.7: Visual comparison for 4x SR. Our algorithm is able to super-resolve images containing multiple planar structures.
Figure 5.8: Visual comparison for 4x SR. Our algorithm is able to better exploit the regularity present in urban scenes than other methods.
Figure 5.9: Visual comparison for 3x SR. Our result produces sharper edges than other methods. Shapes of fine structures (such as the horse’s ears) are reproduced more faithfully in our result.

suitable for super-resolving the BSD100 images. While we achieve slightly worse quantitative performance on BSD100, our results are often visually more pleasing compared to others and do not have artifacts.

**Effect of the number of NNF iterations:** We investigate the effect of the num-
Figure 5.10: Visual comparison for 3x SR. Our result shows slightly sharper reconstruction of the beaks.
Figure 5.11: Effect of iterations. First row: HR and the SR results on 1, 2, and 5 iterations. Second row: the visualization of the nearest neighbor field. Third row: the patch-matching cost.

ber of iterations for NNF estimation using our algorithm in Fig. 5.11, for one step 2x SR. We show the intermediate results after one, two and five iterations. The second row shows a visualization of the source patch positions in the NNF and the matching cost in each stage. The in-place initialization (zero iterations) already provides good matches for smooth regions. We can see a significant reduction in the matching cost even with one iteration. We use 10 iterations for generating all our results.

**Effect of patch size:** Patch size is an important parameter for example-based SR algorithms. Larger patches may be difficult to map to HR since they may contain complex structural details. Very small patches may not contain sufficient information to accurately predict their HR versions. In Fig. 5.12, we plot PSNR/SSIM for patch sizes ranging from $3 \times 3$ to $15 \times 15$. We obtain these plots by averaging over 25 images. We observe that there is a wide range of patch sizes for which our algorithm is able to perform consistently.

**Limitations:** Our method has difficulty dealing with fine details when the planes are not accurately detected. We show one such case in Fig. 5.13 where we fail to recover the regular structures. Another limitation of our approach is processing time. While external SR methods require time-consuming training procedures, they run quite fast during test time [17, 16]. While our algorithm
5.6 Conclusion

We have presented a self-similarity-based image SR algorithm that uses transformed self-exemplars. Our algorithm uses a factored patch transformation representation for simultaneously accounting for both planar perspective distortion and affine shape deformation of image patches. We exploit the 3D scene geometry and patch search space expansion for improving the self-exemplar search. In the absence of regular structures, our algorithm reverts to searching affine transformed patches. We have demonstrated that even without using external training samples, our method outperforms state-of-the-art SR algorithms on a variety of man-made scenes while maintaining comparable performance on natural scenes.
6.1 Introduction

SR algorithms can be broadly classified into patch-based and edge-based methods, depending on the domains in which they operate. Perhaps the simplest patch-based methods are those which assume simple models for image smoothness (such as linear or cubic). Super-resolution then simply amounts to interpolation of the patch pixels according to the chosen model to obtain the sub-pixel values [3, 4, 5]. However, such methods tend to produce overly smooth results, and tend to produces artifacts such as chessboard effect along edges. A popular class of patch-based methods that are aimed at preserving sharpness are those which impose constraints on the marginal distributions of filterbank responses of the image [6, 7]. Studies on statistical properties of natural images have found that these distributions are well modeled as Laplacians [6] or generalized Gaussians [7]. The constraints therefore occur in terms of fits of these distribution types to the data at hand. These priors, however, are used as a global constraint over the entire image. Spatial localization is incorporated only weakly at best [8]. More recently, learning-based approaches have aimed at estimating the relationship between LR and HR patches using a training database [9, 14, 12, 73, 11]. These learned mappings are then used to predict the HR version of each patch of the given LR image.

While patch-based methods have demonstrated success, they do have certain shortcomings. Image patches, depending on their size, can exhibit high degrees of complexity and variability and it is not clear how many samples are sufficient to adequately model the variability seen in generic image patches, and for effectively learning their mappings across resolutions. The choice of patch size in learning-based methods is itself rather heuristic, and has a significant effect on the number of patches required for learning and on the SR result as well. Patch-based approaches also tend to be susceptible to spurious artifacts near sharp edges, since
patches containing sharp transitions in intensities may be difficult to model using a limited number of training patches, unless a very similar patch exists in the training set.

*Edge-based* methods attempt to overcome some of the limitations of patch-based approaches described above. *Edge smoothness* priors [74, 75, 76] favor smooth contours (or isophotes) in the image, and are motivated by human perceptual preferences for smooth image boundaries [74, 75, 76]. These priors have been effective in minimizing artifacts along high-contrast boundaries while producing geometrically smooth contours. However, they do not directly consider the sharpness of intensities across edges in the estimated HR image. *Edge-profile-based* methods address this issue by modeling 1D edge profiles of the image and learning (using training data) how these profiles transform across resolutions [49, 33, 39, 77, 78, 79]. In [39], statistics of 1D edge profiles are obtained by computing moments of the profile shape, and their transformation across resolutions is learned. The gradient profile prior (GPP) approach of [49, 33], fits a generalized Gaussian distribution to edge profiles, and uses a sharpness parameter to transform them across resolutions. The aforementioned approaches try to reap advantages of 1D modeling over 2D patch modeling. In general, 1D profiles are of lower dimensionality than rectangular patches, and can be described by one [49] or a few [39] parameters.

While existing edge-profile methods extract 1D profiles, these profiles are obtained using gradients, which still involve 2D processing using a predefined filter. Computing gradients using predefined 2D filters requires making strict assumptions about the geometry and scale of structures being detected [80]. Any choice of filter size and coefficients of the gradient operator essentially restricts the type of structures that can be detected, in terms of their scale and geometry. Such restrictions have a particularly detrimental effect on the SR problem, wherein all structures in the image, irrespective of their scale/geometry, need to be upscaled by learning correspondences between LR structures and their HR counterparts. Restrictions and assumptions on the scale/geometry introduced by 2D linear preprocessing causes distortions in learning this mapping, and therefore the advantage of subsequently using 1D profiles diminishes.

In addition, imposing priors on gradients does not impose constraints on the absolute brightness values of the image. This sometimes leads to deviations in the brightness levels of the HR image relative to the given LR image in such methods [39, 49]. We show this by an example presented in Fig. 6.1, which shows an
upsampling result obtained by the edge profile-based method of [39].

In this chapter we propose a new edge profile-based prior for the SR problem, that overcomes some of the limitations of existing edge-based methods as described above. We propose a method which avoids 2D preprocesing for obtaining edge information. We do so by adopting the definition of image structure as proposed in [81, 80]. We treat the image as a layout of homogeneous regions partitioning the image, each surrounded by ramp edges [81, 80], as shown in Fig. 6.2(b). Such a layout is obtained by the simple requirement that the intensity variation within a region interior be strictly less than those in the ramps surrounding it. Ramps are characterized by the property that any path through a ramp pixel, monotonically leading from one to the other side, has monotonically increasing (or decreasing) intensity values along it. Such a ramp profile thus consists of a sharp intensity transition over a relatively narrow area between the interior intensities on the two sides, and thus captures the large contrast between the two regions.

Since ramps correspond to areas most affected by a change in resolution (as illustrated by the example in Fig. 6.3), we propose a prior for the SR problem that learns how ramps transform across resolutions. We model the 1D ramp profiles using sigmoidal functions, adequately parameterized to allow the variability seen in ramp profiles extracted from natural images. We learn functions that map the ramp profiles from LR images to their HR counterparts using a set of training images. As we demonstrate in Section 6.2, ramp profiles are more robust descriptors of edge profiles as compared to gradient profiles. We do not use fixed size and fixed coefficient filters or templates for edge extraction. Like other edge-based methods, the remaining non-ramp, homogeneous region interiors are super-resolved using the simple intensity conservation constraint [31].

Unlike gradient-based edge profile priors, ramp profile modeling allows us to formulate our prior in the intensity domain. Gradient domain constraints are unable to accurately preserve brightness [39, 33]. Our prior enforces sharpness directly in the image domain, thus avoiding deviations from the original intensities/colors.

Current edge profile-based methods assume only a one-to-one transformation between an LR edge profile and its HR counterpart. However, we show that edge profiles across thin regions/structures also exhibit a non-trivial inter-dependency since the distance between two ramps (separated by a thin region) in an HR image may be small enough to cause an overlap between their domains of support in the
Figure 6.1: (a) Bicubic upsampled image. (b) Result of edge-based upsampling using [39]. Although edge sharpness is restored, the original brightness level is not maintained.

Figure 6.2: (a) Original image. (b) Areas of the image containing ramps. (c) Edges contained within the ramps, like axes of the ramps.

corresponding LR image. We model such an inter-edge profile relationship for better recovery of contrast across thin regions and structures.

In Section 6.2, we describe the ramp-based representation of image structure. In Section 6.3 we present an overview of the steps involved in our algorithm. Sections 6.4, 6.5, 6.6 and 6.7 describe our proposed algorithm in detail. Section 6.8 shows our results.

6.2 Ramp Models

We now briefly review some of the key ideas of image modeling presented in [81] and the references therein.

A ramp profile $R(x, \theta)$ at a location $x$ in an image is defined as the longest sequence (ordered set) of monotonically increasing (or decreasing) pixel values
Figure 6.3: (a) LR image. (b) LR image after bicubic interpolation to original image size. (c) Original ground truth HR image. (d) Intensity profiles drawn along the white line from (b) and (c). The greatest difference between the pixel profiles occurs at the ramps, which are the sharp transitions between two relatively homogenous interiors. Notice the difference in slopes of the LR and HR ramps, and also the difference in ramp heights, if two ramps are closely separated. Our algorithm learns how these ramps transform from the (bicubic interpolated) LR image to the HR image.
along a path passing through $x$ in a particular direction $\theta$. Ramp profiles, computed over a sufficiently large number of directions, quantify the intensity variations around a given edge location and capture the local edge structure. Ramp profiles are detected without using any predefined filters (e.g., along horizontal and vertical directions), and are fully adaptive to structures of any width or height and they result from a bottom-up process, without any prior assumptions [81]. We elaborate a bit more on these advantages in the context of the SR problem by a real-world example in Section 6.3.

Using ramp profiles, Akbas and Ahuja [81] developed a low-level segmentation algorithm that realizes the properties targeted in [80]: (i) it uses a realistic model of the segments with each region having a smoothly varying interior intensity profile, surrounded by a relatively steep intensity ramps; (ii) it provides a segmentation with quantitatively and qualitatively demonstrated accuracy that does not assume any priors on region geometry (shape, size), and region topology (how many regions neighbor a given region), but rather lets the segmentation structure emerge in a bottom-up fashion; (iii) it provides regions with closed contours as well as the hierarchy of their spatial embedding, the latter not being used in this work; and (iv) its results are perceptually valid.

To summarize, for an input image, the algorithm of [81] provides us with the following: (i) a binary-valued edge map $E$, containing closed, single-pixel wide boundaries of smooth, homogeneous regions, with $E = \{e : E(e) = 1\}$, denoting the set of these edge pixels; and (ii) for each edge pixel $e \in E$, a set of $D$ ramp profiles $R(e, \theta_i), i = 1, 2, ..., D$, along $D$ different directions $\theta_1, ..., \theta_D$ passing through $e$. These ramp profiles characterize the local image structure around the edge pixel $e$. In our work, we compute ramps along $D = 4$ different directions in the 2D image plane – horizontal, vertical and the two diagonals. Also, we deem a ramp profile valid only if it causes a specified minimum change in the intensity level across it, which is above the sensor noise level. In our application, we set this threshold to be 15. Therefore, we deem a ramp profile to be valid only if it causes a gray level intensity change of at least 15 across it.

Figure 6.2(c) shows the edge map obtained via such a segmentation. Figure 6.2(b) shows the ramp areas, i.e., areas that are populated by the ramp profiles at the edge pixels, which can be seen as forming a thin border around the edges.
6.3 Overview of Proposed Algorithm

To super-resolve a given image $I_{LR}$ defined in the LR domain $\Omega_{LR}$, our algorithm consists of the following steps:

6.3.1 Algorithm Summary

1. We first upscale $I_{LR}$ to the HR domain $\Omega_{HR}$ by a simple bicubic interpolation to yield $I_U$.

2. We then use the algorithm of [81] to obtain the low-level edge map $E$ of $I_U$, and the four ramp profiles $R(e, \theta_i), i = 1, 2, 3, 4$ for each edge pixel $e$.

3. To each ramp profile, we fit a sigmoid function parameterized by $Z_l$, as detailed in Section 6.4.

4. We then transform $Z_l$ to its HR counterpart $Z_h$, using a set of transformation functions that we learn from training images, as described in Section 6.5.

5. The transformed ramp profiles (parameterized by $Z_h$) are then used to create a prior image $I_p$ in the HR domain $\Omega_{HR}$, as described in Section 6.6. This prior image $I_p$ essentially contains the ramp-based structural information that the HR image is expected to have.

6. We estimate our final HR image $\hat{I}_{HR}$ using a regularization framework, by using $I_p$ as a prior constraint, along with the classical backprojection formulation [31] as the data term. This step is described in Section 6.7.

6.4 Parametric Model of Ramp Profiles

Consider a ramp profile $R(e, \theta)$ that is monotonically increasing along a direction $\theta$ though the edge pixel $e$ of an image. To fit a parametric model to this ramp profile, we first consider a 1D spatial domain $t \in (-\infty, \infty)$, centered at $e$, and along the direction $\theta$. We can assume the ramp profile intensities to be a 1D function $r(t)$ in this domain, defined at discrete locations $t = -N, -N+1, ..., 0, ..., M - 1, M$. 
Figure 6.4: (a) Parameterization of a ramp in terms of the sharpness $S$, and the intensity levels at either end, $A$ and $B$. (b) An example image from which we extract a few sample ramp profiles and fit a sigmoid model. (c) A few ramp profiles extracted from the sub-images shown in the colored boxes. The ramp profiles are denoted by the white lines in the sub-images. The plots show the parametric model (red curve) fitted to the ramps extracted (blue dots) from the sub-images. The sigmoid function models ramps well.
Figure 6.5: We extract pixel intensity profiles from an HR image and its upscaled (bicubic) LR version, along the cross section denoted by the green line. We compute gradients along both the HR and LR profiles using various linear filters as shown. The HR gradient is quite sensitive to the filter used. Finding the location of the edge pixels, and subsequently learning correspondences between gradient profiles (as done in [49, 33, 39]) can be difficult given the volatility of the gradient estimation process. On the other hand, ramp profiles are able to obtain a more stable and correct localization of the edge, as shown in red in the rightmost column.

We model this ramp profile using a continuous sigmoid function defined as,

\[ f(t; A, B, S) = A + \frac{B - A}{1 + \exp(-St)} \]  

(6.1)

A and B denote the intensity values at the ends of the ramp, and \( H = B - A \) denotes the height of the ramp profile. \( S \) controls the sharpness or steepness of the ramp profile. Figure 6.4 (a) shows these parameters schematically.

We set \( A = r(-N) \), and \( B = r(M) \) as the intensity levels at the end of the ramps. The least-squares estimate for \( S \) can be analytically obtained as \( S = [t^T t]^{-1} [t^T r] \), where \( t = [N, N - 1, \ldots, 0, \ldots, -M + 1, -M]^T \) and \( r = [r(-N), r(-N + 1), \ldots, r(0), \ldots, r(M - 1), r(M)]^T \).

Since this parameter estimation is simple and non-iterative, we are able to parameterize a large number of ramps relatively fast. We show a few examples of ramps extracted an image, superimposed with the above sigmoidal fit in Fig. 6.4 (c).

To summarize, the above modeling procedure parameterizes the shape of a ramp profile \( R(x, \theta) \) with a parameter vector \( Z = [A, B, S] \).

### 6.5 Learning Ramp Transformations

In order to determine how an LR ramp profile, parameterized by \( Z_l \), transforms to its HR counterpart \( Z_h \), we need to learn functions that map \( Z_l \) to \( Z_h \) using a set
of known pairs of LR and HR ramp profiles \( \{(Z_l^{(i)}, Z_h^{(i)})\}_{i=1}^T \). For this, we collect a set of HR images (of around 1000 \( \times \) 1000 pixels), covering a variety of scenes, and then generate the LR images by downsampling using a filter \( f_{psf} \).

6.5.1 Creating Training Data

For obtaining a pair \( (Z_l^{(i)}, Z_h^{(i)}) \), we need to extract a ramp profile from an LR image and find its corresponding ramp in the HR image. We use the segmentation algorithm [81] to obtain the edge pixels and the associated ramp profiles for all the HR images, and the LR images after upscaling (using bicubic interpolation) to the HR domain. We perform this upscaling step to have both the LR and HR image defined over the same resolution domain, as this would facilitate finding correspondences.

For an LR ramp profile \( R_l(e_l, \theta) \) through the edge pixel \( e_l \) along direction \( \theta \) in an LR image, the corresponding ramp profile \( R_h(e_h, \theta) \) in the HR image is found at a location \( e_h \) given by,

\[
e_h = \arg\min_{e \in \mathcal{N}(e_l)} \left( |H_h(e, \theta) - H_l(e_l, \theta)| \right)
\]

where \( \mathcal{N}(e_l) \) is the set of edge pixels of the HR image, in a 5 \( \times \) 5 neighborhood around \( e_l \). The function \( H(x, \theta) \) quantifies the height of the ramp profile through the pixel \( x \), along the direction \( \theta \), in the image.

Ramp profiles allow for more accurate correspondences to be found, as compared to using gradient profiles. This is illustrated in Fig. 6.5. We show a cross section of an image from its HR version and its upsampled LR version. The gradient profile is quite sensitive to the type of gradient filter used, and it is difficult to infer the edge location in the HR image and establish correspondence to LR. Making any decision on the type of filter to use imposes strong assumptions of the expected geometry and scale of the image profile. On the other hand, by definition, ramp profiles avoids any such assumptions on scale, and detects structure bottom-up, adaptively. It is able to correctly identify the HR edge in the example shown in Fig. 6.5.

We collect \( T \) pairs of LR-HR ramp profiles from the training images, using the matching criterion in (6.2).
Figure 6.6: Given a ramp $R$ (in red), we denote $R^-$ and $R^+$ to be the neighboring ramps (in green) on either side of the ramp $R$. These neighbors are detected by simply proceeding outward from either end of the ramp $R$, along the same cross section. We denote the heights of the ramps as the difference between the intensity levels at the ends of the ramps. Therefore, $H = A - B$, $H^- = A^- - B^-$ and $H^+ = A^+ - B^+$. 

6.5.2 Effect of Neighboring Ramps

To learn a regression function from $Z_l$ to $Z_h$, we also need to account for the dependency between an HR ramp profile and the closely spaced neighbors (across thin structures etc.) of the corresponding LR ramp profile. Figure 6.6 shows an example of a ramp profile $R$ (in red), along with its two neighboring ramp profiles $R^-$ and $R^+$ (in green) in either direction. Without loss of generality, we denote the left neighboring ramp profile with the superscript "\(^-\)" and the right neighbor with a superscript "\(^+\)". The intensity values at the ends of the ramps are respectively denoted by $A$ and $B$, with the appropriate superscript as shown in Fig. 6.6. Therefore, we also can denote the heights of the neighboring ramps as $H^- = B^- - A^-$ and $H^+ = B^+ - A^+$ respectively.

The distance between two neighboring ramps along the same cross section in an HR image may be small enough to cause an overlap between the spatial supports of the corresponding ramps in the LR image. This effect of this dependency is illustrated through a simple example in Fig. 6.7. In both cases of Fig. 6.7, as expected, the filtering operation causes a change in sharpness of the ramps. However, in case of Fig. 6.7(a), due to the presence of a close neighboring ramp, along with sharpness, the height of the filtered ramp $R_{f1}$ is also affected. The height remains unaffected if the ramp does not have neighboring ramps such as in Fig. 6.7(b).
Figure 6.7: Example illustrating the dependency of neighboring ramps across thin structures in the downsampling process. $R_1$ and $R_2$ are identical ramps across the red line in the two images in (a) and (b), but $R_1$ has a neighboring ramp separated by a thin region. After using a downsampling filter, the filtered ramps $R_{f1}$ and $R_{f2}$ are not identical, due to the influence of the neighboring ramp $R_{1}^+$ on $R_1$ during the filtering process.

To incorporate this dependency of neighboring ramps in our model, we formulate our regression function to be,

$$\hat{Z}_h = E (Z_h | Z_l, Z_l^+, Z_l^-)$$

where $Z_l^+$ and $Z_l^-$ denote the parameters of the neighboring ramps on either side of the ramp $Z_l$, along the same cross section of the image. Essentially, the parameters of the HR ramp $Z_h$ are predicted not only by its corresponding LR ramp $Z_l$, but also by the LR ramp’s neighbors, $Z_l^+$ and $Z_l^-$ if they are close enough. We make use of the ramp map (such as in Fig. 6.2(b)) to determine if the ramps are close enough to require modeling using (6.3). Neighboring ramp profiles are deemed to be close and dependent on each other if there are no non-ramp pixels between them. In the example shown in Fig. 6.7(a), the neighboring LR ramps profiles $R_{f1}$ and $R_{f1}^+$ do not have any non-ramp pixels between them. Therefore, we use the dependency model of (6.3) to relate them to the HR ramp profiles $R_1$ and $R_1^+$. For all other ramp profiles, we drop the dependency on $Z_l^+$ and $Z_l^-$ and simply assume a one-to-one function.

6.5.3 Estimating Prediction Functions

We make some simplifying independence assumptions among the variables in (6.3) in order to make the estimation tractable: By comparing the filtered ramps
Figure 6.8: Graphical model illustrating the dependency assumed among the various variables in our model. Bold circles indicate the observed variables. Thin circles denote the latent variables to be estimated.

$R_{f_1}$ and $R_{f_2}$ in Fig. 6.7, we notice that the presence of a neighboring ramp essentially reduces the height of the ramp during the filtering process. Therefore, our regression function must be aimed at compensating for this attenuation in the ramp height. Furthermore, we notice that the attenuation in height is caused by change in intensity level at only one end of the ramp, that is closer to the neighboring ramp. We can therefore incorporate the neighborhood dependency by modeling $A_h$ and $B_h$ as functions of the neighboring LR ramp profiles, in the respective directions. Therefore,

$$\hat{A}_h = E \left( A_h | A_l, H_l^- \right)$$  \hspace{1cm} (6.4)

$$\hat{B}_h = E \left( B_h | B_l, H_l^+ \right)$$  \hspace{1cm} (6.5)

We assume the sharpness parameter of the HR ramp profile $S_h$ to always be independent of the neighborhood ramps. We model $S_h$ as a function of the height and sharpness of corresponding LR ramp profile, without any neighborhood dependency.

$$\hat{S}_h = E \left( S_h | S_l, H_l \right)$$  \hspace{1cm} (6.6)

Figure 6.8 shows a graphical representation of the dependence and independence relationship assumed in our model.

To estimate the prediction functions (6.4), (6.5) and (6.6), we take a discriminative modeling approach. We approximate $\hat{S}_h = E \left( S_h | S_l, H_l \right)$ using support vector regression with a polynomial kernel. We choose the polynomial order to be 3 based on $k$-fold ($k = 10$) cross validation by partitioning the training data. Figure 6.9(a) shows plots of the learned $\hat{S}_h$ as a function of the LR sharpness $S_l$, for different values of LR ramp heights $H_l$. Clearly, there is a significant dependence of the HR ramp sharpness on not just the LR ramp sharpness, but also the
LR ramp height $H_l$.

We use a linear model to estimate $\hat{A}_h$ and $\hat{B}_h$ in (6.4) and (6.5) in terms of $A_l, H_l^-$ and $B_l, H_l^+$ respectively. Figure 6.9(b) shows the learned $\hat{B}_h$ as a function of $B_l$, for different values of $H_l^+$. $\hat{A}_h$ behaves similarly, and we do not show it here.

Clearly, the estimation of $\hat{B}_h$ is dependent on the neighboring ramp height $H_l^+$. To better understand the plots in Fig. 6.9(b), let us first focus on the $\hat{B}_h = B_l$ line that is shown in the plot for reference. This line shows the case when the intensity level at the end of the HR ramp $B_h$ is the same as the intensity level $B_l$ at the end of its corresponding LR ramp. Indeed, this is the case if there are no neighboring ramps present. However, due to the presence of a neighboring ramp, the predicted intensity level $\hat{B}_h$ of the HR ramp deviates from $B_l$. This deviation is dependent on the height $H_l^+$ of the neighboring ramp. For example, let us focus on the red curve, which corresponds to the presence of a neighboring ramp of height $H_l^+ = 100$. Qualitatively, we show such an example in Fig. 6.10 (b), where the red colored ramp is to be super-resolved, and the green colored ramp is the neighboring ramp of height $H_l^+$. Figure 6.10 (c) shows the transformed ramp, without incorporating the neighboring ramp dependency. In this case, while the sharpness of the ramp is appropriately transformed, the height of the ramp remains the same as in the LR ramp of Fig. 6.10 (b). However, this height is lower than the ground truth HR ramp height as in Fig. 6.10 (a). To compensate for this smaller height as compared to the ground truth, our neighboring ramp dependency model predicts a lower intensity value $\hat{B}_h$ at the ramp end, as compared to the LR ramp end intensity $B_l$, in Fig. 6.10 (d). This is evident by the plots of Fig. 6.9(b), such as the red curve. Due to this lower predicted intensity level of the ramp end, the resultant ramp in Fig. 6.10 (d) is similar in height to the ground truth HR ramp of Fig. 6.10 (a).

In case of real-world images, the effect of incorporating our neighboring ramp dependency on our SR results is demonstrated in Fig. 6.11. Thin structures like the bird’s beak show better contrast, owing to the correction provided by the learned function in Fig. 6.9(b).
Figure 6.9: (a) The learned HR ramp sharpness parameter $S_h$, shown as a function of the corresponding LR ramp sharpness $S_l$, at a few discretely sampled values of the LR ramp height $H_l$. The HR ramp sharpness depends not only on the LR ramp sharpness but also on the LR ramp height. (b) Learned linear function for predicting $\hat{B}_h$ as a function of $B_l$, for different values of $H_l^+$. We show the $\hat{B}_h = B_l$ line for reference. The intensity level of the HR ramp end, $\hat{B}_h$, is clearly dependent on the neighboring LR ramp height $H_l^+$. 

(a)

(b)
6.6 Ramp-Based Prior

Given an LR image $I_{LR}$ (and its bicubic-upsampled version $I_U$) to be super-resolved, each ramp profile in $I_U$ is transformed using the prediction functions learned above. The transformed sigmoids are then resampled at the positions where the corresponding LR ramp profiles were defined, and the intensities thus obtained are placed in a new image $I_p$. Let $\Omega_R \subset \Omega_{HR}$ denote the set of pixels of $I_U$ or $I_p$ that is populated by ramp profiles (as shown in the example of Fig. 6.2(b)).

Unlike previous gradient-based approaches that model gradient profiles in only one direction (in the direction of the gradient), we extract and transform ramp profiles in four directions. As a result, the value $I_p(x)$ of a particular ramp pixel $x \in \Omega_R$ is typically predicted by multiple transformed ramp profiles. We perform a weighted average of all these predictions, to get the final predicted value of $I_p(x)$.

$$I_p(x) = \frac{\sum_j S_h(x, \theta_j) H_h(x, \theta_j) I_{p\theta_j}(x)}{\sum_j S_h(x, \theta_j) H_h(x, \theta_j)}, \quad x \in \Omega_R$$  

(6.7)
Figure 6.11: Incorporating dependency between neighboring ramps across edges allows for better recovery of contrast across thin structures such as the beaks in the above images.

\( S_h(x, \theta_j) \) and \( H_h(x, \theta_j) \) are the sharpness and height of the (transformed) ramp profiles through \( x \) in direction \( \theta_j \), and \( I_{p\theta_j}(x) \) is the intensity predicted at \( x \) by the (transformed) ramp profile along direction \( \theta_j \) alone. Intensities predicted by high contrast and sharp ramp profiles have higher weight.

Due to this averaging, smoothness is automatically achieved between neighboring pixels in \( I_p \), without an explicit Markov chain-based inference [39, 73].

For the non-ramp locations in \( I_p \), we simply retain the bicubic interpolated values from \( I_U \). This new image \( I_p \) serves as our ramp-based prior constraint.

Figure 6.12 shows an example where we extract ramps from an upsampled LR image, transform the ramps using the learned functions, and obtain the prior image \( I_p \).

### 6.7 SR Reconstruction

We use the prior \( I_p \) together with the intensity conservation constraint in the LR domain to estimate the HR image. The cost function to minimize is therefore,

\[
J(\hat{I}_{HR}) = \left\| (1 - \Lambda) \downarrow \circ \left[ \left( \hat{I}_{HR} \ast f_{psf} \right) \downarrow - I_{LR} \right] \right\|^2 \\
+ \left\| \Lambda \circ (\hat{I}_{HR} - I_p) \right\|^2
\]  

(6.8)

Here “\( \circ \)” denotes the Hadamard (entry-wise) product between matrices. \( \Lambda \) is a matrix containing spatially adaptive regularization parameters, defined as, \( \Lambda = \lambda M \ast g \), where \( M(x) = 1 \) if \( x \in \Omega_R \), and 0 everywhere else. \( M \) is therefore a binary-valued matrix used as a map to indicate the ramp regions. \( \Lambda \) is obtained by smoothing out the map \( M \) using Gaussian filter \( g \) and rescaling it by \( \lambda \).
Figure 6.12: (a) Bicubic upsampled LR image. (b) Ramps extracted from LR image. (c) Transformed ramps using learned transformation functions. (d) Prior image $I_p$ comprising the transformed ramps and the bicubic interpolated values from (a) in the non-ramp regions. (e) HR image estimated using the prior constraint image (d) and the backprojection constraint.

$\Lambda$ gives high weight to our prior in the ramp areas. In smooth regions, $J(\hat{I}_{HR})$ defaults to the backprojection formulation [31]. We minimize (6.8) using gradient descent.

6.8 Experimental Results

6.8.1 Implementation Details

We used a training set of 10 LR-HR image pairs, as shown in Fig. 6.13. We used a $7 \times 7$ Gaussian filter of width 1.6 as $f_{psf}$ for downsampling the HR training images by a factor of 4, to create the LR versions. The same $f_{psf}$ is used for the reconstruction in (6.8) as well. We extracted around $T = 200,000$ pairs of LR-HR ramp profile pairs for learning. We found that learning with even $T = 20,000$ produced comparable results. We trained our ramp transformations for an upscaling factor of 4X. We choose $\lambda = 0.8$ as the regularization parameter in (6.8), as this yielded the most visually pleasing results. For processing color images, we apply the proposed SR method only on the luma component. The chroma channels are upscaled using bicubic interpolation.

6.8.2 Evaluation Strategy

Despite the growing interest in the single image SR problem within the image processing and vision communities, it still lacks a common benchmark for objective evaluation and comparisons. Accurate numerical and quantitative evaluation is also an open problem. Peak signal-to-noise ratio (PSNR) and structural similarity
measure (SSIM) [35] have been two commonly used numerical measures in the past. However, their correlation with human perception of image quality (which tends to be sharpness-driven) is debatable. Numerical measures like PSNR often tend to favor smoother images, whereas the challenge in the SR problem is to recover adequate sharpness. Sharp reconstructed edges yielded by SR algorithms are susceptible to high numerical errors due to the ambiguity in edge localization in the HR domain. Perhaps for this reason, several recent state-of-the-art SR methods (such as [24, 25, 39, 34]) emphasize visual quality rather than quantitative comparisons, and provide results without any ground truth HR versions.

For evaluation, we show most of our results on images that have ground truth HR versions. For these images we report both PSNR and SSIM measures. Although we report these numbers, for better evaluation we encourage the reader to examine the visual similarity of our results with the ground truth images, as compared to competing methods. In some cases where our numerical results are close to those of others, visually our results appear better and bear closer resemblance to the ground truth images when compared to them. In some results, we also extract a few 1D pixel profiles, in order to visually examine the quality of the reconstructed ramps of our method vs. the others, and compare to the ground truth.

We also show some results on LR images that do not have ground truth HR versions, but have been used by several recent methods and have been provided online [24, 25, 39]. This allows us to visually compare our results with these methods, albeit without any ground truth.
Figure 6.14: *Leaves* (4X). Thin structures (e.g., the stem) are well reconstructed in our result, and sharpness along the leaves is also better maintained as compared to other methods.
Figure 6.15: Leaves 1D profiles. We extract 1D intensity profiles (along the white lines) from the results of the Leaves image of Fig. 6.14, and plot them. As evident from the plots, our algorithm is able to better reconstruct the ramp transitions, and our profile bears closer resemblance to the ground truth. We can see that thin/narrow structures are flattened out by other methods, while our result does a better job of preserving such details.

6.8.3 Comparison with Learning-Based Methods

We first compare our results to the gradient profile prior (GPP) [49, 33] method, which is a learning-based approach, most related to our method. We also compare to the patch-based dictionary-learning approach of Yang et al. [14]. In addition, as baselines we also compare to the classical bicubic interpolation as well as the backprojection algorithm [31], which is equivalent to setting $\lambda = 0$ in (6.8).

Figure 6.14 shows our comparisons on the Leaves image. Both the GPP-based approach [49] as well as the dictionary-based approach [14] do not produce sufficiently sharp results. Our edges are reconstructed better and are most similar to the ground truth image shown in Fig. 6.14(a). In Fig. 6.15, we show a 1D profile extracted from the Leaves image. We can see that the ramp transitions are better reconstructed by our algorithm as compared to the others. Thin/fine structures are blurred and flattened out by other methods, but our algorithm is able to better preserve such structures.

Figure 6.16 compares results on the Bird image. Here we highlight an important drawback of gradient-based edge priors such as GPP [49]: As shown in Fig. 6.16(d), the GPP result fails to accurately reproduce the color of the sky, as compared to the ground truth figure of Fig. 6.16(a). The gradient domain constraint in [49] does not enforce brightness consistency. Edges are also not as sharp as those obtained by our method in Fig. 6.16(f). Although sharper than GPP,
Figure 6.16: *Bird* (4X). GPP [49] is unable to preserve the color of the sky. Yang et al. [14] exhibits ringing artifacts along the bird’s beak. Our results are sharp, preserve the original image color, and show little or no ringing artifacts.
Figure 6.17: Monarch image (4X). Yang et al. [14] (e) produces considerable ringing around high-contrast edges. GPP [49] (d) is less sharp and the image color is slightly off compared to the original. Our result is closest to the original, with sharp edges and no visible ringing effect.
Figure 6.18: *Stripes* (4X). The stripes in our result are sharp, without ringing effect, and bears close visual resemblance to the original image.

Figure 6.19: *Stripes* 1D profiles. We extract 1D intensity profiles (along the red lines) from the results of the *Stripes* image of Fig. 6.18, and plot them. Our algorithm is able to produce ramps that bear closer resemblance to the ground truth. The ramp transitions produced by other methods are not as sharp. Also, the GPP algorithm [49] is not able to reproduce the correct intensity level for the black stripes.
the dictionary-based method by Yang et al. [14] produces ringing artifacts along edges, such as along the bird’s beak in Fig. 6.16(e). Our result is free of such artifacts.

Figures 6.17 and 6.18 show two more similar comparisons. Our results are consistently sharper than other methods, and bear closer visual resemblance to the ground truth, with almost no ringing artifacts. Figure 6.19 shows 1D profiles extracted from the results of Fig. 6.18. Our ramps bear the closest resemblance to the ground truth. Other methods produce ramp transitions that are less sharp.

For these images, we tabulate the PSNR (db) and SSIM [35] values in Table 6.1. Our numerical results are significantly better than GPP [49] and also the bicubic and backprojection [31] baselines. Our values are quite close to those of Yang et al. [14], although visually our results appear better, particularly along strong edges.

We also compare our results with the edge statistics driven method of Fattal [39] in Figs. 6.20 and 6.21. Note that no ground truths are available for these images. In the Wheel image of Fig. 6.20, the result of Fattal [39] exhibits more ringing artifacts than our result, and it also appears overly smooth in places. In the Sculpture image of Fig. 6.21, Fattal [39] exhibits disparity in overall brightness. Our method, while also being edge driven, is fundamentally free from this drawback.

Table 6.1: Comparison of PSNR (db) and structural similarity measure (SSIM) [35] for the Leaves, Bird, Monarch and Stripes images.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaves</td>
<td>28.88</td>
<td>29.78</td>
<td>30.94</td>
<td>26.98</td>
<td>30.96</td>
</tr>
<tr>
<td></td>
<td>.9197</td>
<td>.9148</td>
<td>.9316</td>
<td>.9164</td>
<td>.9344</td>
</tr>
<tr>
<td>Bird</td>
<td>27.29</td>
<td>28.08</td>
<td>29.43</td>
<td>24.16</td>
<td>29.84</td>
</tr>
<tr>
<td></td>
<td>.8677</td>
<td>.8718</td>
<td>.8917</td>
<td>.8364</td>
<td>.8938</td>
</tr>
<tr>
<td>Monarch</td>
<td>24.23</td>
<td>24.97</td>
<td>25.81</td>
<td>22.54</td>
<td>25.67</td>
</tr>
<tr>
<td></td>
<td>.8426</td>
<td>.8451</td>
<td>.8656</td>
<td>.8261</td>
<td>.8661</td>
</tr>
<tr>
<td>Stripes</td>
<td>18.58</td>
<td>19.85</td>
<td>20.77</td>
<td>16.04</td>
<td>20.31</td>
</tr>
<tr>
<td></td>
<td>.7203</td>
<td>.7585</td>
<td>.8191</td>
<td>.6620</td>
<td>.8146</td>
</tr>
</tbody>
</table>
Figure 6.20: Steering Wheel image (4X). Our result shows less ringing than Fattal [39] and Yang et al. [14]. The self-similarity approach of Freedman and Fattal [25] does not look very photo-realistic (e.g. the circular rim in the center of the wheel), and some textural details are smoothed out. Our result seems well balanced overall. No ground truth is available for this image.
Figure 6.21: Sculpture (4X). Fattal [39] does not maintain the original brightness level of the image. Our result for this image looks better than the edge-based methods of GPP [49] and Fattal [39]. Glasner et al. [24] appears slightly sharper than ours. No ground truth is available for this image.

6.8.4 Comparisons with Interpolation and Reconstruction-Based Methods

We now compare our results to some methods that do not utilize any learning-based prior knowledge. In particular, we compare our results to those of Shan et al. [34] that is-based on iterative feedback-based filtering. We also compare to the edge-directed interpolation method called NEDI [3], and non-local (NL) backprojection [82] which is an edge-aware extension to the classical backprojection algorithm [31].

Figure 6.22 shows our comparison on the Penguin image. NL backprojection [82] overcomes some of the excessive ringing effect seen in the classical backprojection algorithm [31], but also introduces other artifacts along the edges. The edge directed interpolation method in NEDI [3] produces overly smooth results. Shan et al. [34] produces good results, but shows slightly more ringing artifacts as compared to our result, such as along the beak of the penguin. We again show 1D profiles in Fig. 6.23, extracted from the Penguin results. Our algorithm is able to accurately reconstruct the sharpness of the ground truth ramps. Other methods
Figure 6.22: Penguin image (4X). NEDI [3] tends to oversmooth the image. NL backprojection [82] shows less ringing than the classical backprojection algorithm [31], but introduces other artifacts along edges. Shan et al. [34] produces good results, but also shows some ringing artifacts along the beaks. Our result is sharp, without ringing effect, and match the original image closely.
Figure 6.23: Penguin 1D profiles. We extract 1D intensity profiles (along the red lines) from the results of the Penguin image of Fig. 6.22, and plot them. Most other algorithms are not able to produce ramp transitions as sharp as ours. Our profile bears the closest resemblance to the ground truth.

typically are not that sharp.

In the Zebra image of Fig. 6.24, Shan et al. [34] exhibits ringing or zig-zag-like effects along the stripe edges. NL backprojection [82], although sharp, creates unwanted dark streaks in the interior of the white stripes. This is better visualized by zooming in on Fig. 6.22(e). Our result is free of such artifacts.

Figure 6.25 shows another set of results. Although the ringing effect of Shan et al. [34] is not very evident in this result, fine textural details (such as in the interior of leaves) seem to be slightly washed out as compared to our result.

We show quantitative results of these three images in Table 6.2. Overall, our results are better quantitatively as well.

Table 6.2: Comparison of PSNR (db) and structural similarity measure (SSIM) [35] for the Penguin, Zebra, and Flower images.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Penguin</td>
<td>26.15</td>
<td>.8576</td>
<td>26.15</td>
<td>23.54</td>
<td>25.46</td>
</tr>
<tr>
<td></td>
<td>.8576</td>
<td>.8565</td>
<td>.8766</td>
<td>.8170</td>
<td>.8799</td>
</tr>
<tr>
<td>Zebra</td>
<td>20.06</td>
<td>.6755</td>
<td>20.20</td>
<td>19.68</td>
<td>20.23</td>
</tr>
<tr>
<td></td>
<td>.6755</td>
<td>.6539</td>
<td>.7123</td>
<td>.6750</td>
<td>.7145</td>
</tr>
<tr>
<td>Flower</td>
<td>29.61</td>
<td>.8574</td>
<td>29.99</td>
<td>26.03</td>
<td>30.08</td>
</tr>
<tr>
<td></td>
<td>.8574</td>
<td>.8185</td>
<td>.8554</td>
<td>.7734</td>
<td>.8591</td>
</tr>
</tbody>
</table>
Figure 6.24: Zebra image (4X). The stripes in our result are sharp, without any ringing effect such as in Shan et al. [34], and match the original image closely. NL backprojection [82] maintains sharpness but introduces spurious black artifacts in the interior of the white stripes. Best viewed when zoomed in.
Figure 6.25: *Flower* image (4X). NL backprojection [82] maintains sharpness but introduces artifacts near edges. Shan et al. [34] yields results comparable to ours, but fine textural details (such as in the interior of leaves) appear slightly faded.
6.8.5 Comparison with Self-Similarity-Based Methods

Freedman [25] and Glasner [24] have proposed SR methods that exploit self-similarity within images. While they produce visually pleasing results with sharp edges, they often tend to smooth out fine details.

Figure 6.20(e) shows the result of the method by Freedman and Fattal [25] on the Wheel image. The circular rim in the center of the steering wheel appears unnatural as compared to the other results, and fine textural details also appear to be smoothed out as compared to our result in Fig. 6.20(f). Figure 6.21(e) shows the result of Glasner et al. [24] on the Sculpture image. Objective evaluation on this image is difficult since the ground truth HR image is not available.

6.9 Discussion and Conclusion

Quantifying structure accurately is a fundamental problem in several low-level vision tasks. Conventional methods are based on using pre-defined filters and linear convolution. In this chapter, we have shown some drawbacks of such a formulation, for the single image SR problem. We have presented a ramp profile-based model of structure around image edges for learning a prior for SR, that overcomes the important drawbacks of edge-based priors using gradients. In addition, we have also proposed in our model a simple, but novel idea of incorporating dependency between closely spaced edges, while recovering the HR image. Our method is based on characterizing structure around edges, as detected by a low-level segmentation procedure. For dealing with region interiors, our algorithm simply defaults to the classical backprojection algorithm. We have obtained better results than several state-of-the-art techniques. Our primary improvement and contribution lies in the better reconstruction of edges as compared to other methods.

A limitation of our algorithm is the lack of robustness to noise. Our definition of ramps as described here assumes a relatively noise-free image. In the presence of noise, spurious or distorted ramps may be detected which may hinder performance in the subsequent steps, particularly during the learning phase. In such noisy scenarios, we preprocess the images with a denoising algorithm [40, 23] before using the proposed SR method.

Note that noise sensitivity is an issue for other edge-based SR methods as well.
[49, 33, 39]. Presence of noise significantly affects gradient computations on which algorithms such as GPP [49] are based. Patch-based methods generally tend to perform relatively better in presence of noise [14].

In our algorithm, the relatively smooth region interiors are super-resolved using the backprojection constraint alone. Although from a perceptual standpoint, we are justified in focusing on edges and the structures around them for the SR problem, improvement in results may be expected through better modeling of region interiors as well. One approach of doing so could be the integration of both edge-based and patch-based priors, in order to reap the advantages of each. This would also lend some noise robustness to our approach. We would be exploring these ideas in the future.
CHAPTER 7

CONCLUSION

In this dissertation, we have addressed a number of important limitations in existing SR algorithms. First, we presented two algorithms that were able to better super-resolve fine textural details in the scene. The first of these algorithms used the self-similarity principle on the different sub-bands of the image, with a result that different sub-band components of a patch could find their matches independently, thereby increasing the variety of possible textures that could be synthesized in the HR image. The second algorithm augmented the $L_2$ criterion used during patch matching, with constraints based on sub-band energies of the patches. This helped restore the fine textures that are typically lost during conventional $L_2$ distance-based matching.

Second, we addressed the problem of jointly denoising and super-resolving a noisy LR image. Not much attention has been given to this more challenging extension of the SR problem in the literature so far. We presented an algorithm that effectively exploited the noise present in the LR image to synthesize perceptually valid textures in the super-resolved image. Our “texture from noise” approach was based on careful analysis of local textural properties using sub-band decomposition.

Third, we presented a self-similarity-based SR method that made use of mid-level cues from the image. Specifically, we detected and localized planar surfaces in the scene and used the knowledge of these planes to better guide the self-similarity search process. This idea turned out to be particularly useful for urban and man-made scenes containing buildings etc., that exhibit geometric regularity. Here, we achieved results that were considerably better than the state of the art.

Lastly, we also presented a novel edge-based SR algorithm. While edge-profile-based approaches are not as popular as patch-based approaches, they do offer some advantages, particularly in the simplicity of learning their transformations across resolutions. We presented an edge-based algorithm that extracted edge
profiles in a novel manner, after a detailed and explicit examination of local image structure. We showed the advantages of such an approach when compared to using conventional gradient-based edge profiles.

Super-resolution from a single image remains a challenging and interesting problem. The past few years have seen substantial progress largely due to new example-based approaches. In the years to come, we are likely to see consistent improvements in results largely owing to better computational resources that can process more training examples, and improvements in large-scale learning algorithms such as convolutional networks. We predict further improvements will also come by algorithms that are able to extract more information from the given image itself, thereby enriching and strengthening the self-similarity prior as we understand today. We expect that larger gains might also come by not treating all images (or all regions of an image) homogeneously, but by incorporating higher-level reasoning about the scene(s), and using these higher-level cues for learning better image-specific or region-specific functions that map across resolutions.
REFERENCES


[18] C. Dong, C. Loy, K. He, and X. Tang, “Learning a deep convolutional net-


[22] M. Ebrahimi and E. Vrscay, “Solving the inverse problem of image zooming


examples,” ACM Trans. Graph., 2010.

[26] T. Michaeli and M. Irani, “Nonparametric blind super-resolution,” in Pro-
cedings of ICCV, 2013.

[27] A. Singh and N. Ahuja, “Learning ramp transformation for single image

[28] E. Simoncelli and W. Freeman, “The steerable pyramid: A flexible architec-

[29] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger, “Shiftable multi-


