Abstract of “The Devil is in the Details: Example-based Image Restoration and Detail Synthesis” by Libin Sun, Ph.D., Brown University, May 2016.

In our modern digital age, camera-equipped gadgets such as smart-phones and tablet devices let people capture and share increasing amounts of digital image data. However, we often find our daily snapshots plagued by various forms undesirable features such as blur, low resolution and sensor noise. Undoing such artifacts is typically ill-posed and involves filling in missing image details. But how can we get more from less? How can we go beyond the limits of what can be unambiguously restored? Fortunately, the vast digital imagery available on the Internet provides a dense sampling of our visual world through time and scale, and ushers in fundamental changes in the way we approach these problems.

In this dissertation, we develop data-driven example-based image restoration methods in the context of super-resolution and deblurring. Recent advancements in modern methods have been very successful at restoring salient image structures and edges, but fall short in terms of recovering details in the highest spatial frequency bands. Furthermore, much of these details are beyond the recoverable limits given a single degraded image. Motivated by this observation, this dissertation places special emphasis on exploring methods and representations that are capable of hallucinating and synthesizing novel image details in an image restoration framework.

First, we investigate how to remove motion blur from photos due to camera shake. We present patch-based image priors specifically tailored towards modeling image edges and show its superior performance compared to current leading methods in blind deblurring. To further understand how image priors might hinder inserting image details, we compare generic vs specific image priors for non-blind deblurring, and investigate their ability to insert image details given example images of varying levels of similarity.

Second, we study how to best utilize example images for single image super-resolution. Motivated by recent success in scene matching methods, we examine how big-data can benefit super-resolution systems and show improved ability to hallucinate image details. To mitigate painterly artifacts common to patch-based methods, we draw inspiration from both traditional texture synthesis and recent deep learning based approaches to image synthesis. By moving the image representation from intensity domain to a highly expressive feature space within a Convolutional Neural Network, combined with a sparse image correspondence to better guide synthesis of local image content, we show how convincing image details can be hallucinated far beyond capabilities of existing methods from super-resolution.
The Devil is in the Details: Example-based Image Restoration and Detail Synthesis

by

Libin Sun

B.A., Swarthmore College, 2010
Sc. M., Brown University, 2012

A dissertation submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2016
© Copyright 2016 by Libin Sun
This dissertation by Libin Sun is accepted in its present form by
the Department of Computer Science as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.

Date ____________________ 

James Hays, Advisor

Recommended to the Graduate Council

Date ____________________ 

John Hughes, Reader

Date ____________________ 

Erik Sudderth, Reader

Date ____________________ 

Jue Wang, Reader
(Adobe Systems)

Approved by the Graduate Council

Date ____________________ 

Peter M. Weber
Dean of the Graduate School
Vita

Libin Sun was born on August 1, 1985 in Wuhan, China.

Education

- Ph.D. in Computer Science, Brown University, Providence, RI, May 2016.
- M.Sc. in Computer Science, Brown University, Providence, RI, May 2012.
- B.A. in Computer Science and Mathematics, Swarthmore College, Swarthmore, PA, May 2010.

Internship and Experiences

- Computational Imaging Intern, Light, Palo Alto, CA, Fall 2015.
- Research Intern, Microsoft Research, Redmond, WA, Summer 2014.
- Research Intern, Adobe Research, Seattle, WA, Summer 2012.

Publications

- Libin Sun, James Hays. Super-resolution From Internet-scale Scene Matching. International Conference on Computational Photography (ICCP), 2012
Acknowledgements

This dissertation would not have been possible without the intellectual guidance and constant support from my advisor, James Hays, who has shown tremendous amount of dedication and inspiration to his students. Throughout the years in my Ph.D program, his comments and insightful ‘big picture’ ideas have always helped me zero in on promising directions and reflect on my research. Many a times when I feel stuck about progress, I can walk out of his office full of motivation and ideas for experiments. James has been a great friend to all of his students, sharing his experiences and humor, also playing soccer for the CS department team in the Brown Intramural League. I am grateful to be his first Ph.D student.

My sincere gratitude also goes to my thesis committee members for their interest in my research and insightful comments along the way, which also challenged and enriched my ideas and understanding. I would like to thank Professors John Hughes and Erik Sudderth for providing excellently taught courses at Brown, Jue Wang from Adobe Research for guiding me through an amazing internship experience. I also want to thank Sunghyun Cho (Adobe, Samsung), Brian Guenter (Microsoft Research) and Neel Joshi (Microsoft Research) for collaborating on various exciting projects and helping me expand my knowledge in computer vision and graphics.

I would also like to thank all the administrative staff and technical staff in the CS department. A special thanks goes to Lauren Clarke for taking care of all the graduate students so well. Thanks to Donald Johwa for helping me set up the machine on which I remotely ran most of my final experiments.

This long journey would not have been possible without the support and friendship of my peers at Brown, and the memories we had will be cherished forever. I would like to thank my office mates: Genevieve Patterson, John Oberlin, Ryan Cabeen and Hua Guo; friends from Professor Erik Sudderth’s lab: Soumya Ghosh,
Mike Hughes, Dae il Kim, Jason Pacheco and Zhile Ren; friends from Professor Michael Black’s lab: Peng Guan and Deqing Sun. I want to pay special thanks to Mingming Jiang, Kefei Lei, Lu Lu, Chaolun Song and Shutong Wang for always being there and the good times in Providence. I am fortunate to have a group of friends that started out together in WFLS/Singapore and finally wound up in the Boston area, we are magically always by each other’s side for years (we go way back): Mu Chen, Yin Chen, Yang Du, Shi Fang, Yang Liu and Hengfeng Tian, good luck to all your endeavors.

Reflecting on my two-decade long education path, I owe my deepest gratitude to my parents, who understood the value of education and gave me a good start since my early age. Nothing compares to their patience, encouragement, understanding and love. This dissertation is dedicated to my father, who, through my ups and downs, has been undoubtedly the most important source of strength. Finally and most importantly, I want to thank my lovely wife Yifan Xu, who also recently embarked on a long journey towards her Ph.D.. Without her unwavering love and understanding, the last miles of my Ph.D would not have been possible.
Dedicated to my father, Youxin Sun
# Contents

List of Tables xii

List of Figures xiii

1 Introduction 1
   1.1 Summary of Original Content of This Dissertation 2
      1.1.1 Chapter 2 Related Work 2
      1.1.2 Chapter 3: Edge-based Blur Kernel Estimation Using Patch Priors 2
      1.1.3 Chapter 4: Good Image Priors for Non-blind Deconvolution: Generic vs Specific 3
      1.1.4 Chapter 5: Super-resolution From Internet Scale Scene Matching 3
      1.1.5 Chapter 6: Texture Transfer for Super-resolution 3
      1.1.6 Chapter 7: Conclusions and Recommendations 3

2 Related Work 4
   2.1 Image Prior Models 5
      2.1.1 Compact Image Priors 6
      2.1.2 More Complex Image Priors 6
   2.2 Single Image Super-resolution 7
      2.2.1 Edge Based Methods 8
      2.2.2 Image Statistics Based Methods 9
      2.2.3 External Patch Based Methods 9
      2.2.4 Internal Patch Based Methods 10
   2.3 Deblurring 10
      2.3.1 PSF Estimation and Blind Deblurring 11
2.3.2 Non-Blind Deblurring .................................................. 12
2.3.3 Deblurring for Specific Image Types .............................. 13
2.4 Texture Synthesis and Image Manipulation .......................... 14
  2.4.1 Texture synthesis ...................................................... 14
  2.4.2 Image Editing and Image Correspondence ...................... 15
  2.4.3 Style Transfer ....................................................... 15
  2.4.4 Recent CNN Based Methods ................................. 16

3 Edge-based Blur Kernel Estimation Using Patch Priors 18
  3.1 Introduction .......................................................... 18
  3.2 Algorithm Overview .................................................. 20
  3.3 Nonparametric Patch Priors ....................................... 21
    3.3.1 Learning a Natural Edge Patch Prior ......................... 21
    3.3.2 A Synthetic Edge Patch Prior ................................ 23
    3.3.3 Behavior Comparison ........................................... 23
  3.4 Kernel Estimation using Patch Priors ............................. 24
    3.4.1 x-step ............................................................ 24
    3.4.2 k-step ............................................................ 27
  3.5 Experimental Results .............................................. 27
    3.5.1 Existing Test Set from Levin et al. ......................... 28
    3.5.2 A New Synthetic Test Set of 640 Images .................. 30
    3.5.3 Deblurring Real Photographs ................................. 31
  3.6 Conclusion .......................................................... 32

4 Good Image Priors for Non-blind Deconvolution: Generic vs Specific 33
  4.1 Introduction .......................................................... 33
  4.2 Overview ............................................................. 35
  4.3 Patch-pyramid Prior ................................................ 36
    4.3.1 Optimization .................................................... 37
    4.3.2 Z-Step ........................................................... 38
    4.3.3 X-step ........................................................... 38
  4.4 Locally Adapted Priors ............................................. 38
4.5 How Do Example Images Help? ........................................... 40
4.6 Comparison to Leading Methods ....................................... 44
  4.6.1 Synthetically Blurred Images .................................... 44
  4.6.2 Real Photos with Unknown Blur .................................. 45
  4.6.3 Limitations .............................................................. 45
4.7 Conclusion ................................................................. 48

5 Super-resolution From Internet Scale Scene Matching 49
  5.1 Introduction .............................................................. 49
    5.1.1 Repairing Image Blur ........................................... 51
    5.1.2 Super-resolution .................................................. 51
    5.1.3 Super-resolution Goals and Evaluation ....................... 53
  5.2 Algorithm Overview .................................................. 54
  5.3 Scene Matching .......................................................... 55
    5.3.1 Understanding the Quality of Scene Matches ................. 56
  5.4 Super-resolution Method .............................................. 59
    5.4.1 Segmentation and Texture Correspondence .................... 60
    5.4.2 Segment-level Synthesis of Coherent Textures ............... 61
  5.5 Results ................................................................. 63
  5.6 Discussion .............................................................. 65

6 Constrained Texture Transfer and Synthesis for Super-resolution 68
  6.1 Background and Motivation .......................................... 68
    6.1.1 Traditional Image Priors Tend to Over-smooth ............ 69
    6.1.2 Complex Priors Cannot Insert Textures ...................... 69
    6.1.3 Texture Related Techniques from Graphics .................. 70
    6.1.4 Deep Learning Related Approaches ............................ 71
  6.2 Baseline Methods ..................................................... 71
  6.3 Method ................................................................. 74
    6.3.1 Basic Adaptation to SR ....................................... 76
    6.3.2 Local Texture Transfer via Masked Gram Matrices ........... 76
  6.4 Experimental Results ................................................ 79
6.4.1 Test Data ................................................................. 79
6.4.2 Black and White Patterns ............................................ 79
6.4.3 Textures ................................................................. 79
6.4.4 Natural Scenes .......................................................... 82
6.4.5 Face Images .............................................................. 86
6.5 Conclusions ............................................................... 86

7 Conclusions and Recommendations 89

Bibliography 92
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Quantitative comparison for each method: mean PSNR, mean SSIM, and geometric mean for error ratio, computed over the 32 test images from [80, 81].</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>Quantitative comparison on our synthetic test set of 640 images. Our method significantly outperforms existing methods.</td>
<td>30</td>
</tr>
<tr>
<td>4.1</td>
<td>Quantitative evaluation against existing methods. Methods [76, 69, 154, 104] utilize universally learned image information for deconvolution, while [50] and our method focus on by-example deblurring. For fair comparison, our results in the last column are produced with the estimated PSF from [50]. Both methods make use of example images.</td>
<td>44</td>
</tr>
</tbody>
</table>
List of Figures

3.1 Algorithm pipeline. Our algorithm iterates between $x$-step and $k$-step with the help of a patch prior for edge refinement process. In particular, we coerce edges to become sharp and increase local contrast for edge patches. The blur kernel is then updated using the strong gradients from the restored latent image. After kernel estimation, the method of [154] is used for final non-blind deconvolution. ................................................................. 20

3.2 The generative process from our natural prior (left) and synthetic patch prior (right). (a) 32 of the 2560 learned centroids, with decreasing cluster size in scan-line order. (b) four basic structure seed patches we use as bases for our synthetic prior. (c) 64 random patch samples generated from $\{Z_{nat}\}$. (d) 64 random patch samples generated from $\{Z_{synth}\}$. .................. 22

3.3 The empirical distribution of local constrasts from the 220k patches collected from BSDS500 [3]. The distribution is asymmetric and heavy-tailed, indicating a fair amount contrast in image primitives should be large. ................................................................. 23

3.4 Restoring edges using heuristic image filtering [21] is sensitive to noise and can lead to ringing artifacts, whereas our patch-based optimization can avoid such issues by modeling larger neighborhoods. ................................................................. 24

3.5 Comparison of results on one test image from [80]. Our kernels are less noisy and better resemble the ground truth (leftmost column). Sparse deconvolution with identical parameters are applied to all compared methods except Cho and Lee [21]. .................. 27
3.6 Performance comparison using the error ratio measure as in Levin et al. [80, 81]. The geometric mean of error ratios is shown in the legend for each algorithm. A ratio larger than 3 is deemed visually unacceptable, whereas a ratio less than 1 means the estimated kernel can do better than the ground truth kernel under the given sparse deconvolution method. It is worth mentioning that several estimated kernels from [21] appear better than the ground truth kernels. This could be due to the fact that Cho and Lee used a different deconvolution method to produce the final latent image $x$.

3.7 Performance on our synthetic test set of 640 images. Top: comparison of success rate vs error ratio for all competing methods. Our methods (thicker lines) significantly outperform all others on this test set. Bottom: a bar plot for the success rates at an error ratio of 3, which is deemed as the threshold for visually plausible deblurred results by Levin et al. [81].

3.8 Qualitative comparison on four image regions from our synthetic data set. Note that previous methods tend to introduce oriented trailing noises in the estimated kernels, which could lead to low-frequency ringing artifacts in the deconvolved latent image $x$. The method of Zoran and Weiss [154] is used as the final non-blind step for all algorithms.

3.9 An example photo with unknown camera shake. Our method produces competitive results.

3.10 An example photo with unknown camera shake taken from Xu and Jia [140]. Our method produces sharper edges around the texts and less ringing in densely textured regions.

4.1 The synthetically blurred input and sharp example images show different views of downtown Seattle. Even when given the ground truth input image, the core correspondence algorithm in [49, 50] returns partial (22%) correspondence from example 1 and zero matches from example 2. Our algorithm is able to establish meaningful region level correspondences, and locally adapt the prior to produce significantly more details than state-of-the-art non-blind deconvolution methods.

4.2 (a) Input blurred image with known PSF and sharp example images, (b) initial latent image, (c) best matching example image crops for several query crops from the input, (d) visualization of the nearest neighbor crops overlaid on the input image. The initial latent image is very noisy, the nearest neighbor crops are misaligned and incoherent. Neither alone is a satisfactory image restoration, but we will use the information from both sources to restore blurry photos.
4.3 (a) Using patch-pyramids from nearest neighbor crops for the bottom query crop in Fig. 4.2(c), we train a $7 \times 7 \times 2$ local GMM and compare its random samples (left) against patches drawn directly from training data (right). The prior captures intricate coupling in different frequency bands. (b) The global objective function in Eqn. (4.3) converges over iterations with a fixed schedule for $\beta$, while the PSNR of the latent image increases. Locally trained $7 \times 7 \times 2$ priors are used to restore the input image in Fig. 4.2. 

4.4 Comparing various baselines across example scenarios and prior configurations. From top to bottom: various scenarios of example images, from the best possible (groundtruth) to similar scenes, to irrelevant images (random scenes); averaged overlay of 20 nearest neighbor crops; output using globally trained priors and locally adapted priors. Results obtained using $7 \times 7 \times 1$, $5 \times 5 \times 2$ and $5 \times 5 \times 3$ GMM priors are shown in row (a), (b) and (c) respectively. Better image details can be recovered by (1) using better example images and (2) local training of patch-pyramid priors. 

4.5 Quantitative evaluation of different image priors across example images at various levels of similarity. The six groups of example images are the same visualized in Figure 4.4. Both PSNR and SSIM scores are reported. Each point is obtained by averaging scores from 20 test images. 

4.6 Comparison on uniformly blurred synthetic test images. Groundtruth PSF’s are assumed known and used by all competing methods. 

4.7 Test image from HaCohen et al. [50] with spatially varying PSF estimates. Our approach is highly competitive without requiring dense correspondence. 

4.8 Comparisons against the state-of-the-art by-example method of HaCohen et al. [50] on our uniformly blurred synthetic test images. Four examples are shown. Within each example, the first row shows (from left to right): dense correspondence found by [50], output of [50] with estimated PSF (top-left) and groundtruth PSF (top-right), close-up of [50]. The second row shows (from left to right): our nearest neighbor example crop overlay, our output, our close-up. The PSF estimates are supplied by the authors of [50]. All results are generated using the same input blurry images and PSF estimates, hence directly comparable. The last example shows a failure case due to inaccurate PSF estimate.
4.9 Except for the third row, the core correspondence algorithm at the heart of [50] yields zero successful matches. For the third test image, it cannot explain more than $70\%$ of the image. All input images are real photos with unknown blur. We estimate the blur kernel using [21].

4.10 An example where our method produces convincing textures but also inappropriate high frequency content in background smooth regions (bottom crop).

5.1 Super-resolution results for 8x upsampling. The input image is 128 pixels wide. We compare our results to those of Sun and Tappan [119] and Glasner et al. [45].

5.2 SSIM scores calculated with respect to the reference patch on the left. The middle patch, cropped from the same texture, scores poorly while the patch on the right, a blurred version of the reference, scores very highly. Because SSIM and other reconstruction measures favor blur over texture misalignment, they favor conservative algorithms which do not insert texture details.

5.3 Our proposed pipeline. From left to right, for a low-resolution input we find most similar scenes from a large database. Each input segment is corresponded with best matching segments in these similar scenes. Then a patch-based super-resolution algorithm is used to insert detail from the matched scene segments.

5.4 For four low-resolution query scenes, we show six of the top twenty scene matches that our algorithm will use to insert high-frequency detail. The last row shows an example of scene match failure. For a small portion of test cases the scene matching finds some instance-level matches, as in the Venice image, but generally this is not the case. We will explicitly indicate when a result was generated using instance-level matches.

5.5 Comparison of expressiveness of internal vs external databases. Using up to 20 scene matches, the expressiveness of external database can be significantly better than internal. The “limited” internal database is the low frequencies of the input image that would be usable for a super-resolution algorithm. 150,000 query patches from 80 query images were sampled to generate the plots.

5.6 Comparison of prediction error and uncertainty of internal vs external databases. A total of 180,000 query patches sampled uniformly from our 80 test cases are used for this experiment.
5.7 Counter-clockwise from upper left: Input image, top 20 scene matches, and the top 5 matching segments for the largest input segments. Each input segment is restricted to draw texture from slightly expanded versions of these matched segments. ......................... 60

5.8 Results on man-made scenes. Appropriate textures/materials can be observed among the trees in (c) and surfaces in (a). Edges appear realistic and detailed in (b). ......................... 61

5.9 Results on natural scenes. Our results show successful hallucination of details in water, grass and sand. Some of the details might actually violate the downsampling reconstruction to some extent, but they certainly appear reasonable and appropriate. ......................... 62

5.10 Results where we have at least one instance level scene match. Our algorithm is able to hallucinate salient image structures. For example, the ferry and arches in (c) are successfully hallucinated. In this case, they also approximate the ground truth. ......................... 64

5.11 Results which contain noticeable artifacts. ......................... 65

5.12 From top to bottom: super-resolution results using random scenes rather than matching scenes, zoomed in crops, and the corresponding crops from our algorithm using matched scenes. ......................... 66

5.13 The breakdown of votes when participants compared our results to those of Sun and Tappen [115]. For scenes where the scene matches offered very similar textures to the input (left), participants favor our results. For scenes where the scene matches are spurious or mismatched in scale neither algorithm is favored. ......................... 66

5.14 Failure example with excellent scene matches. Top row: input image (left) and scene matches (right). Bottom row: close-up view of output result at locations indicated by the blue squares. 67

6.1 A sample comparison of various algorithms applied to upsampling texture images for a factor of ×3. Two examples per test image are provided for example-based approaches. It can be seen that the example image has significant impact on the appearance of the hallucinated details in the output images, indicating effectiveness of the transfer. ......................... 72

6.2 Sample images and their corresponding masks, each one is manually generated. ......................... 78

6.3 Visualization of the masks automatically generated using the PatchMatch algorithm. PatchMatch is applied to the low resolution grayscale input and example images to compute a dense correspondence. The HR output image is divided into cells, and all correspondences contained in the input cell are aggregated to form the example image mask. ......................... 78
6.4 Example comparisons on a Chinese text image (top) and black and white pattern image (bottom). Example based methods can hallucinate edges in interesting ways, but also produce biases in background intensity, copied from the example image. Other artifacts are also present. Best viewed electronically and zoomed in. 80

6.5 Example comparisons on regular textures. Best viewed electronically and zoomed in. 80

6.6 Example comparisons on various types of textures. Best viewed electronically and zoomed in. 81

6.7 Example comparisons on simple natural images. Best viewed electronically and zoomed in. 82

6.8 Example comparisons on moderately complex natural images. CNNMRF, Gatys and ‘our local’ consistently synthesize more high frequencies appropriate to the scene. CNNMRF and Gatys suffer from color artifacts due to mismatching colors between the example and the input image. CNNMRF also produces significant amount of color artifacts when viewed more closely, especially in smooth regions and near image borders. Gram matrix based methods such as Gatys and ‘our local’ outperform other methods in terms of hallucinating image details, however also produce more artifacts in a few test cases. Best viewed electronically and zoomed in. 83

6.9 Example comparisons on natural scenes with manually supplied masks. Best viewed electronically and zoomed in. 84

6.10 Example comparisons on a portrait image. Our method is able to hallucinate appropriate details given the well-matched image statistics. Most noticeably, plausible details are successfully introduced to the eyebrows, hair, and eyes. CNNMRF produces decent amount of details as well, however, it makes the output image less recognizable as the person in the input image. Best viewed electronically. 87

6.11 Example comparisons on a face image. Our method fails due to mismatch in global image statistics. It is interesting to note that CNNMRF works extremely well for face images, however, it cannot insert image details not present in the example image. In this case, it cannot synthesize a closed mouth of the baby. Best viewed electronically. 88
Chapter 1

Introduction

Everything you can imagine is real.

Pablo Picasso

In our modern digital age, camera-equipped gadgets such as smart-phones and tablet devices let people capture and share increasing amounts of digital image data. However, we often find our daily snapshots plagued by various forms of image degradation and artifacts such as motion blur, defocus blur and sensor noise. Undoing such artifacts is typically ill-posed and involves filling in missing image details. But how can we get more image content from the partially missing or corrupted observed image signals? How can we go beyond the limits of what can be unambiguously restored? Fortunately, the plethora of digital imagery available provides a dense sampling of our visual world through time and scale, and ushers in fundamental changes in the way we approach these problems. Recent advancements in single image super-resolution have shown that learning coupled image statistics between low and high resolution images from large training set provides consistent and leading performance. Example-based deconvolution methods are starting to emerge to further improve image quality of deblurred images. In addition, texture synthesis and inpainting techniques have long enjoyed the success of leveraging on examples to transfer image details.

In this dissertation work, we develop data-driven example-based methods in the context of super-resolution...
and deblurring, with an emphasis on hallucinating and synthesizing novel textural details. First, we examine how big-data can benefit super-resolution systems and show improved ability to hallucinate image details. Second, we present patch-based image priors tailored to modeling image edges and show its superior performance in blind deblurring. As a follow-up, we compare generic vs specific image priors for non-blind deblurring, and investigate their ability to insert image details given example images of varying levels of similarity. Finally, building on recent advancement of texture synthesis work, we propose to develop a general constrained texture transfer framework that would be useful to example-based image restoration applications. Traditional restoration methods are conservative in recovering signals due to PSNR considerations and lack the ability to insert image details beyond recoverable limits. By careful relaxation of image reconstruction constraints and selective transferring of example image details, our proposed system aims to achieve a high level of photo-realistic appearance while still being perceptually faithful to the input signal constraints. We hope our proposed work could offer a new perspective on image restoration tasks.

1.1 Summary of Original Content of This Dissertation

The work in this dissertation explores several areas of image restoration and draws techniques from computer vision, computational photography, and graphics. We outline the major contributions below:

1.1.1 Chapter 2 Related Work

This chapter presents survey on recent advances in several topics relevant to the broader aspects of image restoration. Classic problems such as super-resolution and deblurring will be discussed. Commonly used forms of image priors and models will be introduced. Furthermore, relevant works in image editing, texture synthesis as well as image quality will be briefly presented as well since they could potentially provide useful techniques to enable novel restoration framework from a graphics perspective.

1.1.2 Chapter 3: Edge-based Blur Kernel Estimation Using Patch Priors

In this chapter, we develop novel patch-based image priors specifically designed to better recover image edges, and show that it achieves state-of-the-art performance for blur kernel estimation assuming spatially invariant motion blur.
1.1.3 Chapter 4: Good Image Priors for Non-blind Deconvolution: Generic vs Specific

In this chapter, we examine the performance of generically trained and specifically trained image priors for non-blind deconvolution, using a broad range of training images at varying levels of similarity to the input scene. We show that our patch-pyramid prior, coupled with local training, is able to best harness the richness of image details in the exemplars, and significantly outperforms other formulations.

1.1.4 Chapter 5: Super-resolution From Internet Scale Scene Matching

In this chapter, we present a big-data approach to single image super-resolution. Our system makes use of the most relevant exemplar image content distilled from a large Internet scale database of six million images. We discuss the richness of external image statistics in comparison to internal image statistics. Our method is capable of inserting convincing image details and is shown to provide competitive results for large factors of upsampling.

1.1.5 Chapter 6: Texture Transfer for Super-resolution

In this chapter, we build on recent advancement in deep learning based texture synthesis work and propose several variants to allow better hallucination in the upsampling process. We place special focus on synthesizing appropriate texture details. Our method compares favorably against traditional SR baselines and recent texture transfer approaches. We hope our proposed work could offer a new perspective on single image super-resolution as well as the broader field of image restoration.

1.1.6 Chapter 7: Conclusions and Recommendations

In this chapter, we summarize the strengths and weakness of our approaches and draw conclusions for this dissertations. We present a forward looking perspective on exciting future directions to explore.
Chapter 2

Related Work

A photograph is a click away. A good photograph is a thousand clicks away and a better one, a million clicks away.

Kowtham Kumar K

The literature on image restoration is vast, including classic topics such as super-resolution, deblurring, denoising and inpainting, where the goal is to restore a corrupted image under certain assumptions regarding a well-defined image formation model. Successful frameworks typically involve (1) a linear image formation model, (2) an objective function that is a linear sum of a data fidelity term and regularization term(s), (3) an iterative optimization strategy to minimize the objective function. There are scenarios where the image is not strictly corrupted but our goal is to enhance it in certain ways, be it aesthetically (color grading, style transfer) or semantically (attribute manipulation). Relevant work in these areas provide insights and techniques that would allow improvement in image restoration research.

In the following discussion, we will first present popular image prior models that are essential to modern day image restoration methods, followed by typical stand-alone problems in image restoration and their respective state-of-the-art methods in recent literature. Finally, we discuss related work from the graphics
community on topics such as texture synthesis and image retargetting, which offer practical techniques and representations suitable for image restoration applications.

2.1 Image Prior Models

The most common image formation model in image restoration work is to assume a linear degradation process with additive Gaussian noise:

\[ y = Ax + n \]  

(2.1)

where \( y \) and \( x \) are the observed (corrupted) and the clean latent image respectively. \( A \) is a matrix representing the linear degradation process such as convolution (blur) and downsampling (reduced resolution). Finally, \( n \) is an additive noise term, typically treated as i.i.d. zero mean Gaussian for simplicity. In many situations such as super-resolution (SR), denoising, and non-blind deconvolution, the goal is to solve for \( x \) given \( A \) and \( y \).

Assuming i.i.d. zero mean Gaussian noise, then it is natural to seek a latent image estimate \( \hat{x} \) such that:

\[ \hat{x} = \arg \min ||y - Ax||^2 \]  

(2.2)

Unfortunately, due to the ill-posed nature of many image restoration tasks (deblurring, SR), simply minimizing the objective in Eq 2.2 is insufficient, leading to unstable results, noise, or ringing artifacts. As a result, it is common in modern image restoration literature to take a Bayesian approach to the problem and constrain the latent image \( x \) via various forms of natural image priors \( p(x) \). Now, instead of solving for the likelihood term alone (Eq 2.2), we seek to maximize the posterior likelihood to obtain a maximum a posteriori (MAP) solution:

\[ \hat{x} = \arg \max p(x|y) = \arg \max p(y|x)p(x) \]  

(2.3)

Image priors are challenging to model statistically because images are complex, high dimensional, and non-Gaussian. Nonetheless, a large variety of natural image priors for \( p(x) \) have been proposed in the past several decades with great success, we will discuss representative works below according to their complexity and expressiveness.
2.1.1 Compact Image Priors

One of the earliest working image prior models was pioneered by Geman and Geman [44]. They proposed to model images via the Gibbs distribution and paved ways for many Markov Random Field (MRF) based priors later [151, 101, 100, 134] (see Sec 2.1.2). This prior models the interaction between neighboring pixels and penalizes image gradients. When each clique involves only two adjacent pixels, it relates to the family of priors that directly model image gradients ($p(x) \propto e^{-k|x|^\alpha}$), such as [38, 76, 80, 69]. Fergus et al. [38] learns a mixture of Gaussians to fit the distribution of natural image gradients and shows its effectiveness for deblurring. Levin et al. [76, 80] considers both Gaussian ($\alpha = 2$) and Hyper-Laplacian priors ($\alpha \in [0.5, 0.8]$) for deconvolution and shows that the latter produces sharper results due to its heavy-tailed nature. A mixture of Laplacian is also used in [79] to separate reflections in a single image. These sparsity based priors are effective but still blurry edges (low gradient value) to sharper edges (large gradient value). To address this issue, Krishnan et al. [70] proposes the normalized sparsity prior ($l_1$ norm divided by $l_2$ norm) which is shown to favor sharp images and works well for deblurring. Other parametric forms have also been explored. Sun et al. [117] considers a Generalized Gaussian Distribution (GGD) to model image gradient profiles (sharpness) and applies it to super-resolution. Cho et al. [23] locally adapts the parameters of the GGD per image segment to form a content-aware image prior.

2.1.2 More Complex Image Priors

Over the past decade, image priors have evolved tremendously to become more expressive. Many of the more complex priors model image neighborhoods or patches instead of adjacent pixels. Zhu et al. [152] proposed the FRAME model to allow MRF priors to be learned from training images. However, the filters were hand selected instead of being learned. Inspired by FRAME, Roth and Black introduced the Fields of Experts (FoE) model [101], which is a MRF image model with fully learned parameters, including learned filters. Combining ideas from steerable pyramid [111], Roth and Black further proposed Steerable Random Fields [100] to model image gradients by locally adapting to edge structures. Clique potentials are defined over steered filter responses using a Gaussian scale mixture model and show improved performance for denoising and inpainting.

To investigate the effectiveness of these filter-based MRF models, Weiss and Freeman [134] analyze the lower and upper bounds of the partition functions and find that, due to the Power Law fall-off of spatial frequencies in natural images, ‘good filters’ in such learned MRF models should fire rarely on natural images.
and have high concentration of high spatial frequencies in their power spectrum. As a result, these seemingly complex higher order models essentially boil down to learning fancy low-pass filters and encoding basic principles such as ‘images should be smooth’ in various ways. No matter how these filter-based MRF priors are formulated, they work under the same principle by penalizing high frequency image content, thereby inappropriate for recovering high frequency details when considering tasks such as super-resolution and deblurring (going to low frequencies to higher frequencies). This could be why examples in FoE and Steerable Random Fields mainly focus on denoising and inpainting.

Perhaps the most complex and expressive model in recent literature is the EPLL-GMM model from Zoran and Weiss [154], where a Gaussian Mixture Model (GMM) is learned over millions of natural image patches, and the log likelihood of an image ($p(x)$) is computed as the expected patch log likelihood summed over all patches in the image. This simple and elegant setup is able to soft partition patch space learn useful representations through the covariance matrices of the Gaussian components. As the authors show, the principal components of the covariance matrices exhibit clear structures from natural images, such as edges at various orientations and texture boundaries, and makes an elegant connection to the classical Dead Leaves Model [156].

Finally, many approaches ignore the underlying statistical representation of natural images, and simply resample from the input image or other images in a non-parametric fashion. This approach is first popularized by works in texture synthesis [35, 34], and have been quickly adopted by works in image restoration, such as super-resolution [41, 45, 48, 122], deblurring [120, 89, 50], denoising [13], and inpainting [27, 51].

### 2.2 Single Image Super-resolution

Classic super-resolution (SR) is defined in a multi-frame setting, where a sequence of low-resolution images of the *same scene* aligned to sub-pixel shifts reveal some high frequency detail signals, and the goal is to super-resolve the scene to obtain higher resolution. This problem is well studied and is beyond the scope of this work. Instead, most of the recent literature in super-resolution tries to tackle the more challenging case of single image super-resolution (SISR), where the high frequency content must be hallucinated in the high resolution output. This is the type of super-resolution we will be focusing on as well.

In SISR, we try to recover a high-resolution image ($\mathbf{x}$) given a single low-resolution input ($\mathbf{y}$) such that
the following image formation model is satisfied:

\[ y = (g * x) \downarrow_s + n \]  

where \( g \) is a Gaussian filter, \( * \) is the convolution operator, \( \downarrow_s \) is the downsampling operator with a factor of \( s \), and \( n \) is an additive noise term, typically i.i.d. Gaussian.

Unlike traditional multi-frame SR, it is impossible to unambiguously restore high frequencies in a SISR framework. Single image super-resolution is an extremely under-constrained problem: there are many plausible natural images that would downsample exactly to a given low-resolution input, because of the downsampling operation. As a result, existing works in the field present intelligent ways to hallucinate plausible image content instead of recovering the ground truth, which is simply impossible. However, the hope is that with the correct constraints and optimization process, our estimate of \( x \) can be visually and semantically as close to the ground truth image as possible.

Over the past decade, SR methods have evolved from interpolation based and edge oriented methods to learning based approaches. Most leading methods involve mapping low resolution (LR) patches to high resolution (HR) patches, either by establishing learned statistical models such as regressors [143, 106, 33], or by searching for image patch candidates internally [45, 61] or externally [41, 119, 122] in a non-parametric fashion, popularized by works from the graphics community [35, 8].

### 2.2.1 Edge Based Methods

Edges are the most important image primitives and play a crucial role in understanding structures and objects in a scene. A group of methods have been proposed to focus on restoring image edges during the upsampling process. [118] combines a learned primal sketch prior over image primitives (edges, ridges, corners) with a Markov-chain based inference algorithm to hallucinate image structures. [37] models the upsampling process by imposing statistics over edge dependencies and intensity constraints to output HR images that are smooth and free of grid artifacts. [117] models image edges via the gradient profile prior, in the form a Generalized Gaussian distribution learned over natural images, and shows its effectiveness for SISR. While these methods work well to produce sharp edges in the high resolution image, high frequency image details and textures are completely left out, usually handled using basic interpolation methods such as bicubic upsampling. As a result, the HR image does not appear sufficiently natural and lacks critical elements of realism.
2.2.2 Image Statistics Based Methods

Natural image statistics is a reliable property that can be exploited for various image restoration tasks including SISR. The most basic heuristic is that image gradients should be sparse, and this has been successfully adopted in the works of [67, 124]. However, such priors only constitute a weak constraint for the optimization process when applied globally to the whole image. Many works utilize these image priors in conjunction with various other constraints in their objective function, as done in [67, 119, 108].

2.2.3 External Patch Based Methods

Many modern SISR algorithms operate over image patches to achieve better results. Due to the limited resolution of the LR input image, it is natural to rely on an external training set of images to learn mappings from LR to HR patches. Leading methods in this space differ on how such mappings are learned and represented. In the pioneering work of Freeman et al. [41], nearest neighbor (NN) LR/HR patch pairs are searched from a set of training images and a Markov Random Field (MRF) model is used to choose patch candidates that satisfy both the downsampling constraint and minimize misalignment in overlapping pixels. Other strategies include manifold embedding [16], kernel ridge regressions [67], sparse coding [145, 1]. To improve performance in terms of both computational speed and mapping accuracy, several recent methods perform clustering in patch space and learn simple regressors locally for each partition [143, 106, 125, 126]. Since the complexity of local subspace is substantially lower than that of natural images as a whole, these locally adapted methods are able to learn better mappings than others. A recent leading method – SRCNN – based on Convolutional Neural Networks (CNN) is introduced by Dong et al. [33], where a 3-layer CNN is trained end-to-end to map LR to HR patches. The methods from [143, 106, 126, 33] represent the current state-of-the-art, measured in terms of PSNR/SSIM performance.

While these methods are able to obtain sharp edges and some amount of image details, high frequency details are still challenging to reconstruct and hallucinate. There are several recent works that attempt to specifically address this issue by leveraging higher level image statistics to allow more content-aware upsampling. Sun and Tappen [119] propose a system where image segments are matched using filter responses and retrieved from a database to help hallucinate high frequency image content. HaCohen et al. [48] introduces a texture database and a segment matching mechanism to allow better patch-based synthesis of texture details. Sun and Hays [122] utilizes an Internet-scale scene matching system to find semantically similar scenes via
global image statistics from the LR image, and constrain the texture insertion process using matched seg-
ments from retrieved exemplar scenes. Landmark super-resolution is also explored in [149] via retrieving
instance level scenes from the Internet. These methods are able to hallucinate more high frequency details
than the aforementioned learning based methods, but usually fall short in terms of PSNR/SSIM performance.

### 2.2.4 Internal Patch Based Methods

There is an interesting body of work that does not rely on training images or patches externally, but rather
solely makes use of internal image statistics for the upsampling process. These methods work because natural
images typically exhibit self-similarity and redundancy within and across scale, the LR/HR patch mappings
are inherent within the Gaussian pyramid of the input image itself. [45] is the first successful method to unify
traditional SR techniques with self-similarity and patch recurrence. The method is surprisingly simple and
fast, yet produces competing results with sharp edges. [39] presents an extremely upsampling method that
relies on local self-similarity of the input image, and iteratively upsample the image with small factors using
a set of dedicated novel non-dyadic filter banks. [144] further pursues this direction of local self-similarity
and learns a first-order regression model using in-place examples. [88] rely on self-similarity across scales to
jointly estimate the blur kernel for the downsampling process and the HR image for blind SR. Finally, [61]
enriches the search space for self-similarity based methods by detecting perspective geometry and introducing
transformed self examples.

### 2.3 Deblurring

There is an enormous body of research aimed at alleviating the effects of blur-inducing imaging phenomena
– defocus, motion, and aberrations to name a few. Photographic blur cannot be unambiguously inverted in
realistic imaging conditions [7], therefore “...the central challenge ... is to develop methods to disambiguate
solutions and bias the processes toward more likely results given some prior information”[64]. In this thesis
we focus only on motion blur, which is caused by camera shake during exposing for a static scene.

Deblurring algorithms tend to use relatively compact, parametric image priors, often learned from natural
image statistics, that encode principles such as “edges should be sharp”, “gradients should be rare”, “colors
should be locally smooth” [80, 81, 69, 21, 140]. These parametric models are helpful but limited. Their
assumptions, such as a heavy-tailed gradient distribution, are not universally true [23]. In general, these
models can sharpen edges but will not enhance texture, material, or object detail because these phenomena
are too complex for the models.

Image blur caused by camera shake is a common problem in consumer photography. A motion blurred image $y$ capturing a static scene is often modeled as:

$$y = k \ast x + n$$

(2.5)

where $k$ is the blur kernel (could be uniform or spatially varying), also known as the point spread function (PSF), $x$ is the latent image, $n$ is an additive noise term, and $\ast$ is the convolution operator. Blind deconvolution is the problem of jointly estimating $k$ and $x$ given $y$; non-blind deconvolution is better conditioned, in which $k$ is the only unknown. Either way deblurring is an ill-posed problem with unique challenges.

### 2.3.1 PSF Estimation and Blind Deblurring

To overcome the ill-posed nature of blind deconvolution, previous works place strong assumptions or prior knowledge on $k$ and $x$. Regarding $k$, it is often assumed that $k$ should be sparse [38, 107] and continuous [17]. For $x$, it is often assumed that image gradients are heavy-tailed [38, 107, 80, 81, 69, 70]. More extreme form of sparsity such as the $L_0$ norm has also been explored by [142]. One crucial finding in recent literature shows that sparsity priors prefer blurry images to sharp ones [80], because blur reduces overall gradient magnitude. As a result, under a MAP$_{k,x}$ framework, sparsity priors have limited capacity to steer the latent image towards a sharp solution. A common failure mode is the degenerate solution, where the blur kernel is a delta function, and the latent image $x$ is the same as the blurred image $y$. In order to drive the solution away from the no-blur solution, leading methods often employ careful engineering and optimization smarts, such as maximizing marginal likelihood [81], optimizing $k$ and $x$ iteratively [21, 140, 120]. The recent work of [94] re-examined the analysis of [80] and shows that implementation details such as the delayed normalization of the blur kernel could have huge impacts on the solution.

One common trait of leading methods for blind deconvolution is to explicitly target image edges for reliable kernel estimation, since image edges contain the most information of blur. Joshi et al. [65] and Cho et al. [24] directly predict sharp edges from blurry edges and rely on them to estimate the blur kernel. While these methods work well for small scale blur, they have difficulty dealing with large blur kernels, as directly restoring sharp edges from a severely blurred image is non-trivial. To handle large blur kernels, Cho and Lee [21] introduce an edge-based approach, which alternates between restoring sharp edges and estimating the blur kernel in a coarse-to-fine fashion. This framework has been further extended by Xu and
Jia [140] and has proven to be effective in recent benchmark studies [68]. However, these approaches heavily rely on heuristic image filters such as shock and bilateral filtering for restoring sharp edges, which are often unstable.

To better model and recover image edges, several works adopt a patch-based approach, inspired by works in super-resolution and texture synthesis. [120] learns a patch-based prior over sharp edges from an external database and specifically constrain image edges during the deblurring process. [89] adopted the internal patch recurrence property for estimation of the blur kernel. These patch-based approaches resulted in a significant improvement in performance of blind deblurring, especially on robustness, according to the recent comparative study of [60].

For simplicity, the aforementioned methods usually assume a uniform blur over the whole image. Spatially varying blurs is more realistic and can be extended from the uniform case. Several leading methods [135, 47, 55] represent state-of-the-art performance at handling non-uniform blurs. Following recent advancements in deep convolutional neural networks, [116] estimates a distribution over per patch motion blur via a CNN; a MRF is then used to infer a smoothly varying field of blur kernels.

Another realistic consideration is noise. [150] combines directional low-pass filters with Radon transform to recover the blur kernel in the presence of high noise. [59] targets light streaks present in low-light images and night photography to estimate the blur kernel via a non-linear blur model.

Unlike all existing methods, the idea of using a reference image for blind deblurring is explored in [50]. They estimate a dense correspondence via the NRDC algorithm [49] in the inner loop to constrain blur kernel estimation and the non-blind deconvolution step.

### 2.3.2 Non-Blind Deblurring

In contrast to blind deblurring, non-blind deblurring/deconvolution is a less ill-posed problem. Since there are more known variables than unknowns, there is no degenerate solution. The challenge is to produce a natural image that is sharp, realistic, and better yet, contains plausible high frequency details.

Early deconvolution methods can be traced back half century ago, and are still relevant today. Wiener deconvolution [137] imposes Gaussian assumptions for both image noise and gradients. Richardson-Lucy deconvolution [98] assumes Poisson image noise and provides an iterative algorithm. However, these classic methods tend to over-smooth image edges and introduce ringing artifacts.

Modern methods employing natural image statistics based on sparsity [76] show improved handling of image gradients and can suppress ringing artifacts. Carefully engineered methods [148] specifically focus
on reducing ringing artifacts. Other parametric image gradient prior models also exist, such as Hyper-Laplacian [69] and Gaussian mixtures [38, 154]. It is worth noting that the EPLL-GMM framework of [154] learns a highly expressive GMM model over large number of training patches, is able to outperform compact models based on image gradient statistics. Instead of modeling image patches generatively, discriminative deconvolution [104] is also proposed to directly learn a cascade of Gaussian CRF based on regression tree fields.

Several works place special attention to the amount of high frequency details that can be restored or hallucinated. The works of Cho et al. [23, 25] also try to insert more details in the restored image, via a locally adapted content aware prior over image gradients, and an optimization process to locally coerce gradient histograms to match sharp images, respectively. Sun et al. [121] provides an analysis of generic vs. specific image priors that are adapted to local image content. A GMM-based patch-pyramid prior is used to successfully insert high frequency details.

There are recent methods that address severe image noise and outliers in the presence of blur. [22] presents an EM-based framework to jointly mask out saturated pixels/outliers and estimate the sharp latent image. [136] augments Richardson-Lucy with an auxiliary variable to obtain improved results.

Another significant trend in the literature is the use of deep neural networks. [105] considers a two-phase approach where the observed image is first denoised then deconvolved using a learned Multi-Layer Perceptron. [141] introduces a CNN architecture consisting of a deconvolution subnetwork and an outlier rejection subnetwork inspired by traditional methods.

2.3.3 Deblurring for Specific Image Types

With recent advancements in deblurring, researchers have started to examine deblurring specific types of images. Domain knowledge or class-specific priors can be applied to better constrain the optimization process, leading to superior results compared to generic deblurring methods. [92] presents a face deblurring framework exploiting known structures of human facial landmarks and matching from a large database of face exemplars. [20, 93] utilizes text-specific properties for deblurring text images. More recently, [57] uses a 15 layer CNN trained over artificially blurred text patches to deblur text images and outperform existing methods for both motion blur and defocus blur. Finally, [2] explores the notion of class-specific deblurring via a prior based on the class-specific subspace of image responses to band-pass filters. By training on existing datasets such as cars, pedestrians and cats, significant performance gain can be achieved over generic deblurring algorithms.
2.4 Texture Synthesis and Image Manipulation

2.4.1 Texture synthesis

Texture Synthesis is a well studied problem in the graphics and vision community. Given a sample texture image input, the goal is to create an output image that matches the textural appearance of the input so that perceptually they cannot be distinguished by human perception. This problem is first introduced by Heeger and Bergen [53], and quickly received a lot of research attention. Texture synthesis is a relevant topic to image restoration because the highest frequency details usually cannot be reconstructed after the degradation process in problems such as SR, deblurring and denoising. The only way to make the restored image realistic is to hallucinate and synthesize plausible high frequency details that are appropriate given the existing frequencies in the image; and modern texture synthesis methods can provide such insights and techniques. We now highlight a number of leading methods for texture synthesis and refer the readers to [131] for a more comprehensive survey on this topic.

In general, texture synthesis methods can be categorized into parametric and non-parametric models. Parametric models require a statistical representation of textures, which then can be used in the generative process. On the other hand, non-parametric methods simple resample the input texture itself by matching pixels and patches without having any understanding or representation of the texture. Recently, deep learning based methods have emerged as well, we discuss several relevant approaches in Section 2.4.4.

**Parametric Methods** such as [53, 96] are based on matching statistical constraints in a [111]. Heeger and Bergen [53] applies histogram matching in the sub-bands of the steerable pyramid. Portilla and Simoncelli [96] considers a more complex model by constraining the correlation and marginal statistics in the wavelet domain. Both methods are able to turn noise images into the target texture, however, they often fail to more structured and regular textures that exhibit less stochastic nature.

**Non-Parametric Methods** were first pioneered by [12, 35] to completely sidestep statistical representation for textures, because texture is indeed challenging to model in faithfully. Instead, the new texture is synthesized pixel by pixel by matching neighborhoods between the input and output in a nearest neighbor fashion. This worked surprisingly well to inspire follow-up works [34, 71, 72, 132], most of which are patch-based resampling methods. To further allow better synthesis of textures and images, Ashikhmin introduces a greedy synthesis approach to encourage continuous copying of pixels [6]. Hertzmann et al. extends these approaches and propose the ‘Image Analogies’ framework [54], which can transfer textures and image styles to enable artistic filters, texture-by-numbers, and even super-resolution. However, the SR examples in [54]
require training pairs to be extremely similar to the target image (c.r. Figure 6 in [54]).

2.4.2 Image Editing and Image Correspondence

Finding nearest neighbor image patches and establish some form of correspondence between two images is a shared component in various modern graphics and vision tasks, such as image editing, image retargeting, and texture synthesis. Naive approaches include brute force nearest neighbors (NN) and approximate NN via tree structures [127]. However, these methods treat image patches as independent data points in feature space and fail to exploit the spatial relationship among them. Inspired by the greedy techniques from texture synthesis methods [6, 54], the PatchMatch algorithm from Barnes et al. [8, 9] seeks to solve the image correspondence problem by propagating local matchings in a continuous fashion to boost coherence. The method is highly efficient to allow interactive image editing applications. To improve robustness to geometric and photometric variations, HaCohen et al. [49] proposes Non-Rigid Dense Correspondence (NRDC) framework and show its applications to image enhancement tasks, including deconvolution. Other follow-up works such as PatchNet [58] and PatchTable [10] attempt to scale up the algorithm for larger datasets by exploring more efficient representation and data structures.

There are works that focus on manipulating object level features as well. Lalonde et al. [74] proposes a system to allow user-friendly image recomposition by querying and inserting objects into a photograph. Similarly, Chen et al. [18] presents the ‘Sketch2Photo’ system to allow complete recomposition of an image, in which a user provided sketch is turned into a realistic photograph. Image parts and objects are retrieved from the Internet and seamlessly stitched together via a novel blending algorithm. Going further, Chen et al. [19] proposes an interactive image editing tool to allow easy extraction and manipulation of 3D shapes and objects in a photograph.

2.4.3 Style Transfer

While texture models capture local statistics and operate on small parts of an image, global statistics transfer have also been explored, such as color transfer [97, 77, 62], tonal adjustment [14], lighting [110], semantic attributes [73], and artistic style for portraits [109].

Several of these methods draw inspirations from both low level and high level vision techniques. Johnson et al. [63] enhances the realism of computer generated scenes by transferring color and texture details from real photographs. Shih et al. [110] considers the problem of hallucinating time of day for a single photo
by learning local affine transforms in a database of time-lapse videos. Retrieving exemplar frames from the database relies on recent advancements from scene matching [139], while the optimization centers on a MRF setup similar to [40, 41]. Laffont et al. [73] utilizes crowdsourcing to establish an annotated webcam database to facilitate transferring high level transient attributes among different scenes. Pools of transforms are first precomputed from the database and applied to matched superpixels to provide more consistent results. Style transfer for specific image types such as portraits is also explored by Shih et al. [109], in which multi-scale local transforms in a Laplacian pyramid are used to transfer contrast and color styling from exemplar professional portraits. Domain knowledge allows the system to combine detected facial landmarks with SIFT Flow [85] to ensure reliable dense correspondence.

### 2.4.4 Recent CNN Based Methods

With recent advancements in deep learning and Convolutional Neural Networks (CNN), many aspects in texture synthesis have started to witness revisits and re-invention. The Google Deep Dream [90] project examines interesting properties of very deep neural nets by maximizing response in certain layers or classes. It is shown that images can exhibit ‘dreamy’ components of objects from training classes such as birds, dogs and temples, turning images of a certain class into something completely unrelated. Details can be continuously hallucinated to elicit a zoom effect as well. This generative aspect of deep networks is well explored in many recent works.

Gatys et al. [42] propose a style transfer system using the 19-layer VGG network [112], which was trained for image classification, a completely different task. The key constraint is to match the Gram matrix of numerous feature layers between the output image and a style image, while high level features of the output is matched that of a content image. In this way, textures of the style image is transferred to the output image as if painted over the content image, similar to Image Quilting [34]. They apply the same Gram matrix constraints and drop the content image constraint to demonstrate Deep Texture Synthesis [43] with impressive realism of results. This model shares many connections with earlier parametric methods such as [53, 96], but contains orders of magnitudes more parameters.

Drawing inspirations from texture synthesis methods, Li and Wand propose to combine a MRF with CNN for image synthesis [82]. This CNNMRF model adds additional layers in the network to enable resampling ‘neural patches’, namely, each local window of the output image should be similar to some patch in the style image in feature space in a nearest neighbor sense. This has the benefit of more coherent details should the style image be sufficiently representative of the content image. However, this copy-paste resampling
mechanism is unable to synthesize new content. In addition, this method is prone to produce ‘washed out’ artifacts due the blending/averaging of neural patches. This is a common problem to patch-based synthesis methods [34, 41, 71], if some of what averaging is used to handle disagreement in patch overlapping regions. Recently, the a deep image analogies framework [15] is proposed to enable semantic style transfer. It augments the CNNMRF framework [82] by introducing semantic maps to constrain the patch sampling process. In particular, these manually supplied maps define regions and segments so that patches can only be sampled within matching segments to synthesize more coherent textures. However, the examples presented in [15] only focus on paintings instead of natural images.
Chapter 3

Edge-based Blur Kernel Estimation

Using Patch Priors

You don’t take a photograph, you make it.

Ansel Adams

3.1 Introduction

Image blur caused by camera shake is a common problem in consumer photography. A motion blurred image y capturing a static scene is often modeled as:

\[ y = k * x + n, \]

where \( k \) is the blur kernel, \( x \) is the latent image, \( n \) is noise, and \( * \) is the convolution operator. For blind deconvolution, the goal is to recover both \( x \) and \( k \) from \( y \), which is an ill-posed problem.

To overcome the ill-posedness of blind deconvolution, previous works place strong assumptions or prior knowledge on \( k \) and \( x \). Regarding \( k \), it is often assumed that \( k \) should be sparse [38, 107] and continuous [17].
For $x$, it is often assumed that image gradients are heavy-tailed [38, 107, 80, 81, 70]. However, we argue that this popular family of sparsity priors is not suitable for the task of kernel estimation for the following reasons. First, sparsity priors prefer blurry images to sharp ones [80], because blur reduces overall gradient magnitude. Hence, sparsity priors have limited capacity to steer the latent image towards a sharp solution. Second, they fundamentally suffer from the fact that the unit of representation is extremely limited: gradient filters often consider two or three pixels, hence ignoring longer-range dependencies which give rise to the most salient image structures and geometry. This is also why state-of-the-art image restoration methods often involve larger neighborhoods or image patches [13, 102, 134, 154].

Another family of blind deconvolution algorithms explicitly exploits edges for kernel estimation. Joshi et al. [65] and Cho et al. [24] directly restore sharp edges from blurry edges and rely on them to estimate the blur kernel. While these methods work well for small scale blur, they have difficulty dealing with large blur kernels, as directly restoring sharp edges from a severely blurred image is non-trivial. To handle large blur kernels, Cho and Lee [21] introduce an edge-based approach, which alternates between restoring sharp edges and estimating the blur kernel in a coarse-to-fine fashion. This framework has been further extended by Xu and Jia [140] and has proven to be effective [68]. However, these approaches heavily rely on heuristic image filters such as shock and bilateral filtering for restoring sharp edges, which are often unstable, as we will show in Sec. 3.3.3.

In this paper, we propose a new edge-based approach using patch priors on edges of the latent image $x$. Patches can model image structures better than filter responses. In our approach, we estimate a “trusted” subset of $x$ by imposing patch priors specifically tailored towards modeling the appearance of image edge and corner primitives. We only restore these primitives since other image regions, e.g. flat or highly-textured ones, do not carry much useful blur information for kernel estimation. Furthermore, restoring textures often results in hallucinated high frequency content, which corrupts the subsequent kernel estimation steps. We illustrate how to incorporate the patch prior into an edge-based iterative blind deconvolution framework through an optimization process, where we iteratively recover the partial latent image $x$ and the blur kernel $k$. Experimental results show that our approach achieves state-of-the-art results (Sec. 5.5).

The main question we address in this paper is what is the right prior for $x$ in blind deconvolution. Intuitively, the image prior used for blind deconvolution should not be too expressive, *i.e.*, the prior should not be allowed to express a wide range of visual phenomena, such as motion blur and defocus blur, which are natural and frequent in professional photographs. A prior that is overly expressive, such as the GMM model proposed by Zoran and Weiss [154], will inevitably accommodate blur and hinder the convergence of the
Figure 3.1: Algorithm pipeline. Our algorithm iterates between $x$-step and $k$-step with the help of a patch prior for edge refinement process. In particular, we coerce edges to become sharp and increase local contrast for edge patches. The blur kernel is then updated using the strong gradients from the restored latent image. After kernel estimation, the method of [154] is used for final non-blind deconvolution.

solution pair. With this in mind, our patch-based prior is specifically tailored towards modeling particular image primitives [31]– atomic elements that form the structural part of the image, namely, edges, corners, T-junctions, etc. To choose proper patch priors, we examine both statistical priors learned from a natural image dataset and a simple patch prior from synthetic structures. Experimental results show that, surprisingly, the simple synthetic patch prior can generate the same quality or even better results than the learned statistical prior.

3.2 Algorithm Overview

Our algorithm builds upon the coarse-to-fine, iterative optimization framework commonly used in recent methods [107, 21, 140, 70], as shown in Fig. 3.1. In particular, at each level $s$, we initialize $k_s$ by upsampling $k_{s-1}$ from the previous level, followed by a latent recovery step to solve for $x_s$ ($x$-step). However, our $x$-step only attempts to partially restore $x_s$, i.e. edge primitives, which we deem as the only reliable image regions for both $x$-step and $k$-step. Given a newly updated $x_s$, $k_s$ is updated ($k$-step) by comparing restored gradients and the blurred image. This procedure is carried out iteratively until convergence and the resulting $k_s$ is propagated to the next level to initialize $k_{s+1}$. Unlike [21], we do not rely on heuristic image filtering steps to “guess” the ground truth gradients in the $x$-step. Instead, we introduce a nonparametric patch prior to coerce image primitives in $x$ to be sharp and noise-free. This also avoids the problem of falsely penalizing large gradients, which may happen if a sparsity prior is applied over image gradients [80].

We will first introduce our patch prior formulation in Sec. 3.3, then describe how to incorporate the patch prior for kernel estimation in Sec. 3.4.
3.3 Nonparametric Patch Priors

Most iterative deblurring methods are carefully engineered to drive the latent image towards a sharper solution during the optimization process, and avoid the degenerate case of a $\delta$ PSF and a blurry $x$. To achieve this, we introduce two sets of independent auxiliary variables, namely $\{Z^i\}$ and $\{\sigma^i\}$, for each pixel location $i$ considered as an edge primitive. $Z^i$ is a particular example patch assigned to location $i$, and $\sigma^i$ is the target local contrast we wish the latent image patch to have. Together they provide strong constraints over desired structural shapes and sharpness for primitive patches in the latent image. Note that we model patches in the normalized space: subtracting its mean and dividing by standard deviation.

We would like our patch prior to be sufficiently expressive, i.e., any edge patch $P$ in natural images can be approximated by $P = \sigma Z + \mu + \epsilon$, where $\sigma$ is the patch contrast, $\mu$ is the patch intensity, $\epsilon$ is a small error term. However, this prior cannot be overly expressive, i.e., it should not be allowed to express high frequency textures or gradual changes in image gradients, otherwise it will start to accommodate blur and noise in the latent image, hence losing its power to restore sharpness. Since our prior is only applied to a subset of pixels in the latent image, it should be distinguished from existing generic natural image priors such as the simple sparsity prior [80] and the complex GMM-based patch prior from Zoran and Weiss [154], which are applied over the whole image.

We will first examine how we learn a set of representative natural edge patches $\{Z_{nat}\}$. We then introduce a set of synthetic patches $\{Z_{synth}\}$ as an alternative solution.

3.3.1 Learning a Natural Edge Patch Prior

To model image primitives, a natural approach is to learn a prior from a training set of patches extracted along image contours. We first downsample all 500 images (grayscale) from the BSDS500 dataset [3] by half in each dimension to reduce noise and compression artifacts, then compute a mask (see Sec. 3.4 for more details) based on gradient magnitude. The mask is then intersected (AND operator) with human annotated ground truth contours provided by the dataset to produce a final mask. We extract all $5 \times 5$ patches centered within the mask, resulting in approximately 220k patches from 500 images. We normalize each patch by removing its DC and dividing by its standard deviation (local contrast). Finally, we apply a standard k-means algorithm to learn the representative primitive patches: $\{Z_{nat}\}$ is the set of centroids. For our experiments, we use $k = 2560$. In Fig. 3.2(a), we show 32 such centroids, which are regularly sampled from the 2560 centroids after sorting them by cluster size. Note that we expect to have enough training patches such that the
Figure 3.2: The generative process from our natural prior (left) and synthetic patch prior (right). (a) 32 of the 2560 learned centroids, with decreasing cluster size in scan-line order. (b) four basic structure seed patches we use as bases for our synthetic prior. (c) 64 random patch samples generated from \( \{ Z_{nat} \} \). (d) 64 random patch samples generated from \( \{ Z_{synth} \} \).

Cluster centroids encode the appropriate variations in orientation and translation, hence we do not apply such transformations to \( \{ Z_{nat} \} \) during the optimization process.

We also learn the distribution of local contrast encoded by \( \sigma \) for such primitive patches. The empirical distribution of \( \sigma \) is shown in Fig. 3.3, which will be used in our framework to restore diminished/blurred image gradients in Sec. 3.4.

We make the following observations about our learned patch priors:

1. Patch complexity is correlated with cluster size. In particular, simple edge structures such as horizontal/vertical step edges make up the largest clusters, and the patch samples within these clusters exhibit little variation. On the other hand, the smallest clusters capture complex textures and noisy structures, and the patch samples within these clusters can be significantly different from each other. This is consistent with recent findings from Levin et al. [78].

2. The overall complexity of image primitives is surprisingly limited. Using \( k = 2560 \), there is already a good amount of redundancy in the clusters: some centroids appear almost identical to each other. Furthermore, most of the clusters capture simple step edges with varying profiles, orientations and
3.3.2 A Synthetic Edge Patch Prior

The learned natural edge patches are relatively simple, but contain slight blur and noise as a result of averaging real image patches with slight misalignment. For this reason, we design a synthetic patch prior that is hopefully comparable in terms of expressiveness, while being cleaner and sharper.

To generate a set of synthetic patches \( \{ Z_{\text{synth}} \} \), we use four seed patches as shown in Fig. 3.2(b). These seed patches are one step edge, one corner and two bars of different widths. We consider two kinds of transformations: rotations (every 3 degrees from 0 to 360), and translations (up to \( \pm 1, 2 \) pixels in each direction). The set of example patches \( \{ Z_{\text{synth}} \} \) is generated by applying all possible combinations of these transformations to each seed patch. All patches are then normalized by subtracting its mean and dividing by its standard deviation. Fig. 3.2(d) provides a visualization of 64 random samples from this synthetic patch prior.

In this work we demonstrate that this synthetic prior \( \{ Z_{\text{synth}} \} \) works roughly as well as \( \{ Z_{\text{nat}} \} \), indicating that the complexity of 5 × 5 edge patches is limited, and that most properties of image primitives are intuitive: sharp, noise-free and geometrically simple. Nonetheless, they are still a powerful prior for deblurring.

3.3.3 Behavior Comparison

Cho and Lee [21] restore sharpness via a so called prediction step involving heuristic image filtering. Specifically, bilateral filtering is used to reduce noise and shock filtering is used to sharpen edges, both are applied after solving for \( x \) under a Gaussian prior over image gradients. However, such filtering procedures often fail in the presence of dense image structures and noise along edges. In contrast, we replace these steps by a
more principled formulation that is robust to noise (more so than bilateral filtering) and produces sharp edges (more so than shock filtering).

In Fig. 3.4 we provide an illustration of how our approach can efficiently remove blur via restoring both the patch shape ($Z$) and patch contrast ($\sigma$). It shows the before/after comparison for an intermediate latent image $x$ under one iteration of optimization using Cho and Lee’s method, and ours. The comparison suggests that our method generates higher quality intermediate latent image $x$, which naturally leads to a higher quality kernel estimation in the $k$-step.

## 3.4 Kernel Estimation using Patch Priors

Like the deblurring framework from Cho and Lee [21], our kernel estimation approach iterates between $x$- and $k$-steps, in a coarse-to-fine manner. We will discuss these two steps separately. At the coarsest level, we initialize $k$ by a small $3 \times 3$ Gaussian.

### 3.4.1 $x$-step

Given the current estimate of the blur kernel $k$, the goal of $x$-step is to produce a latent image $x$ (or some trusted subset) that is sharp and free of artifacts (ringing, halos, etc). To estimate $x$ we minimize the following energy function:
\[ f_x(x) = \sum_{D_x} \omega_x \| KD_x x - D_y y \|^2 \]
\[ + \alpha \| D_h x \|^2 + \alpha \| D_v x \|^2 \]
\[ + \beta \frac{1}{|M|} \sum_{i \in M} \rho(P_i x - q_i) \]
\[ + \gamma \sum_{i \in M} \left( \sigma_i - F^{-1}_{ref} (F_{\sigma,x}(\sigma_i)) \right)^2 , \]
(3.2)

where \( K \), \( y \) and \( x \) represent the matrix form of the blur kernel \( k \), the blurred input image \( y \), and the latent image \( x \), respectively. \( D_x \) is the matrix form of the partial derivative operator in different directions and \( \omega_x \) represents the corresponding scalar weight. As done in [107, 21], we also use zero-th, first, and second order derivatives for \( D_x \). We borrow the weights used in [107] for \( \omega_x \). \( D_h \) and \( D_v \) are the first order differentiation operators along the horizontal and vertical axes. \( P_i \) is a binary matrix extraction operator, extracting the patch at location \( i \) in the latent image \( x \). In this work, we fix the patch size at \( 5 \times 5 \). \( q_i \) is defined as \( q_i = \sigma_i Z_i + \mu_i \)

where \( Z_i \) is a vector representing \( Z_i \), the example patch assigned to location \( i \).

We maintain a binary mask \( M \) to indicate pixel locations classified as edge primitives, and only apply the patch prior to such locations to encourage sharpness. The other regions in \( x \) are weakly regularized by a Gaussian prior over image derivatives. The use of such “edge” masks is adopted in [65] as well. \( |M| \) is the number of non-zero elements in \( M \). \( \rho() \) is a robust penalty function. In particular, we use the Lorentzian loss function, defined as \( \rho(r) = \log \left( 1 + \frac{r^2}{2\theta^2} \right) \). Finally, \( F_{\sigma,x} \) is the empirical cumulative distribution of \( \{\sigma_i\} \) in the current latent image \( x \). \( F_{ref} \) is the reference cumulative distribution of local contrasts, based on the learned distribution in Fig. 3.3.

Intuitively, the first term is the data term, enforcing the blur model. The second and third term yield a weak Gaussian prior to regularize image smoothness. Note that while regions outside the mask \( M \) do not participate in the \( k \)-step, we only use this term to weakly regularize the energy function so that optimization becomes stable. The last two terms encode our patch prior involving \( Z_i \) and \( \sigma_i \), providing two strong constraints: (1) edge primitive patches in \( x \) should be similar to some example patch (after normalization), (2) the distribution of \( \sigma \)'s in \( x \) should be similar to a reference distribution.

Directly optimizing Eqn. (3.2) is hard. We present an iterative approximation procedure to update the variables \( M, \{Z_i\}, \{\sigma_i\} \) and \( x \) below. For the first iteration, we set \( M = \emptyset \), meaning that we use only the Gaussian prior to initialize an intermediate latent image \( x \), then the following procedures are applied until
convergence:

1. Update $M$: We first obtain a binary mask by keeping the top 2% of pixel locations with the largest 
filter responses from a filter bank consisting of derivatives of elongated Gaussians in eight orientations. 
We morphologically thin this mask and then remove small isolated components. This step chooses the 
right locations to apply our edge patch prior.

2. Update $\sigma^i$: Fixing $M$, $Z^i$, and $x$, we use an iterative reweighted least squares (IRLS) method to optimize 
Eqn. (3.2) with respect to $\sigma$. We present the full derivation in the supplementary material\(^1\), but the main 
steps for IRLS are as follows:

a. Let $r^i = P_i x - q^i$, compute weights $w_i$ given current $\sigma^i$ by:

$$w_i = (2\epsilon^2 + r_i^T r_i)^{-1}, \quad (3.3)$$

b. Update $\sigma^i$ by solving a weighted least square problem:

$$\sigma^i \leftarrow \frac{w_i}{|M|} Z^i (P_i x - \mu^i) - \gamma F_{ref}^{-1}(F_{\sigma^i x}(\sigma^i)) Z^i - \gamma. \quad (3.4)$$

3. Update $Z^i$: Holding other variables constant, we find example patch $Z^i$ that is most similar to $\frac{P_i x - \mu^i}{\sigma^i}$, 
for each location $i$.

4. Update $x$: Holding other variables constant, $x$ is updated by solving the following for $x$:

$$F^{-1}(A \odot F(x)) = \frac{\beta}{|M|} \sum_{i \in M} \frac{2}{2\epsilon^2 + r_i^T r_i} P_i^T P_i x$$

$$= F^{-1}(B) + \frac{\beta}{|M|} \sum_{i \in M} \frac{2}{2\epsilon^2 + r_i^T r_i} P_i^T (\sigma^i Z^i + \mu^i),$$

where

$$A = \left( \sum_{\delta_x} \omega_x F(\delta_x) \odot F(\delta_x) \right) \odot F(k) \odot F(k)$$

$$+ \alpha \sum_{\delta_x, \delta_y} F(\delta_x) \odot F(\delta_y),$$

\(^1\)Please refer to our project page: http://cs.brown.edu/~lbsun/deblur2013iccp.html
Figure 3.5: Comparison of results on one test image from [80]. Our kernels are less noisy and better resemble the ground truth (leftmost column). Sparse deconvolution with identical parameters are applied to all compared methods except Cho and Lee [21].

\[
B = \left( \sum_{\delta_x} \omega_x \mathcal{F}(\delta_x) \circ \mathcal{F}(\delta_x) \right) \circ \mathcal{F}(k) \circ \mathcal{F}(y).
\]

We use \( \mathcal{F} \) to represent the Fourier transform and \( \mathcal{F}^* \) for its complex conjugate. \( \circ \) is the element-wise multiply operator. Note that this equation is no longer linear in \( x \) because \( r_i \) involves \( x \) as well.

Hence, we use IRLS to iteratively optimize \( x \), where the diagonal weighting matrix has entries \( w_{ii} = \frac{2}{2\epsilon^2 + r_i^2} \forall i \in M \).

### 3.4.2 \( k \)-step

In this step, we hold \( x \) constant and optimize with respect to \( k \) only. We adopt the method of [21], with the following objective function:

\[
f_k(k) = \sum_{\delta_x} \omega_x \| k \ast \delta_x x - \delta_x y \|^2 + \beta \| k \|^2,
\]

where \( \omega_x \) are as introduced in Sec. 4.3.3, and \( \delta_x \) represent partial derivatives corresponding to \( D_x \). To speed up computation, FFT is used as derived in [21]. One important difference is that we set the gradients \( \delta_x x \) outside of \( M \) to zero since we only allow edges to participate in the kernel estimation process.

### 3.5 Experimental Results

We first test our algorithm on the widely used 32-image test set introduced in [80], and further establish a synthetically blurred test set of 640 images of our own. Finally we show comparisons on blurred photographs with real unknown camera shakes.
To ensure fair comparison for evaluating estimated kernels from competing methods, we standardize the final non-blind deconvolution step by using sparse deconvolution\(^2\) for Sec. 3.5.1, and the state-of-the-art method of Zoran and Weiss [154]\(^3\) for Sec. 3.5.2. Note that the use of [154] as the final step is a strict improvement compared to the default non-blind deconvolution step from all of the methods considered.

We make use of three measurements for quantitative analysis: mean PSNR, mean SSIM, and the geometric mean of error ratios, which is introduced in [80].

### 3.5.1 Existing Test Set from Levin et al.

The 32 test images in [80] were produced from 4 images and 8 different kernels. The blurred images and ground truth kernels were captured simultaneously by carefully controlling the camera shake and locking the Z-axis rotation handle of the tripod. We test our algorithm using both the natural and synthetic priors on this test set and compare with results provided by [81].

![Figure 3.6: Performance comparison using the error ratio measure as in Levin et al. [80, 81]. The geometric mean of error ratios is shown in the legend for each algorithm. A ratio larger than 3 is deemed visually unacceptable, whereas a ratio less than 1 means the estimated kernel can do better than the ground truth kernel under the given sparse deconvolution method. It is worth mentioning that several estimated kernels from [21] appear better than the ground truth kernels. This could be due to the fact that Cho and Lee used a different deconvolution method to produce the final latent image \(x\).

Fig. 3.5 shows a test image from this dataset deblurred by various methods. Note that kernels estimated by previous methods may contain trailing noise in certain directions. In contrast, our method produces a kernel that is most similar to the ground truth in shape, and the recovered latent images contain the least amount of artifacts.

In Fig. 3.6, we report the cumulative error ratio performance, which suggests that our method outperforms

\(^2\)We use the MATLAB code provided by [81] at: http://www.wisdom.weizmann.ac.il/~levina/papers/LevinEtalCVPR2011Code.zip

\(^3\)We use the MATLAB code from: http://www.cs.huji.ac.il/~daniez/epllcode.zip
Table 3.1: Quantitative comparison for each method: mean PSNR, mean SSIM, and geometric mean for error ratio, computed over the 32 test images from [80, 81].

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>Error Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known k</td>
<td>33.8197</td>
<td>0.9286</td>
<td>1.0000</td>
</tr>
<tr>
<td>Levin et al. [81]</td>
<td>31.1372</td>
<td>0.8960</td>
<td>1.8546</td>
</tr>
<tr>
<td>Fergus et al. [38]</td>
<td>29.4629</td>
<td>0.8451</td>
<td>2.7270</td>
</tr>
<tr>
<td>Cho &amp; Lee [21]</td>
<td>30.7927</td>
<td>0.8837</td>
<td>2.0077</td>
</tr>
<tr>
<td>Our-Nat</td>
<td><strong>32.3842</strong></td>
<td><strong>0.9108</strong></td>
<td><strong>1.3917</strong></td>
</tr>
<tr>
<td>Our-Synth</td>
<td>32.1042</td>
<td>0.9033</td>
<td>1.4844</td>
</tr>
</tbody>
</table>

Figure 3.7: Performance on our synthetic test set of 640 images. Top: comparison of success rate vs error ratio for all competing methods. Our methods (thicker lines) significantly outperform all others on this test set. Bottom: a bar plot for the success rates at an error ratio of 3, which is deemed as the threshold for visually plausible deblurred results by Levin et al. [81].

It is interesting to note the level of saturation in performance: our method is able to achieve an error ratio of 2 for approximately 90% of the images, and the method of Cho and Lee [21] can produce several kernels that are better than the ground truth. The reasons are two-fold. First, the images are small and easy to deblur, because they contain plenty simple step edges, e.g., one of the images is an illustration and contains only clean and uninterrupted step edges. Second, the final non-blind deconvolution method—sparse deconvolution—can only achieve limited restoration performance in terms of PSNR. As a result, it’s relatively easy to obtain low error ratios on these test images.
Figure 3.8: Qualitative comparison on four image regions from our synthetic data set. Note that previous methods tend to introduce oriented trailing noises in the estimated kernels, which could lead to low-frequency ringing artifacts in the deconvolved latent image $x$. The method of Zoran and Weiss [154] is used as the final non-blind step for all algorithms.

3.5.2 A New Synthetic Test Set of 640 Images

The 32 test images in [80] are only $255 \times 255$ in size, and limited in terms of diversity. To further test the limits of leading algorithms, and characterize the performance gap among them, we develop a synthetic test set of 640 high-resolution natural images of diverse scenes. We start with the 80 high quality natural images used in [122], and synthetically blur each of them with the 8 blur kernels from [80]. Finally, we add 1% additive Gaussian noise to the blurred images to model sensor noise. The same noise level is used in [69, 80, 154]. We present a comprehensive evaluation via both qualitative and quantitative comparisons for the methods from [21, 24, 81, 70, 140], as well as our results using the natural and synthetic priors.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>Error Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>24.7822</td>
<td>0.6429</td>
<td>5.8598</td>
</tr>
<tr>
<td>Known k</td>
<td>32.4610</td>
<td>0.8820</td>
<td>1.0000</td>
</tr>
<tr>
<td>Cho &amp; Lee [21]</td>
<td>26.2353</td>
<td>0.8138</td>
<td>4.1934</td>
</tr>
<tr>
<td>Cho et al. [24]</td>
<td>20.1700</td>
<td>0.5453</td>
<td>18.1437</td>
</tr>
<tr>
<td>Krishnan et al. [70]</td>
<td>23.2158</td>
<td>0.7554</td>
<td>8.3673</td>
</tr>
<tr>
<td>Levin et al. [81]</td>
<td>24.9410</td>
<td>0.7952</td>
<td>5.6493</td>
</tr>
<tr>
<td>Xu &amp; Jia [140]</td>
<td>28.3135</td>
<td>0.8492</td>
<td>2.5987</td>
</tr>
<tr>
<td>Our-Nat</td>
<td>29.5279</td>
<td>0.8533</td>
<td>1.9647</td>
</tr>
<tr>
<td>Our-Synth</td>
<td><strong>29.5585</strong></td>
<td><strong>0.8546</strong></td>
<td><strong>1.9510</strong></td>
</tr>
</tbody>
</table>

Table 3.2: Quantitative comparison on our synthetic test set of 640 images. Our method significantly outperforms existing methods.
Fig. 3.7 presents the cumulative distribution of error ratios. Our method outperforms all other methods, with [21, 140] being the most competitive. However, even with some parameter tuning, we are unable to obtain good results with the online MATLAB code packages from [81, 70, 24] on this test dataset. Fig. 3.8 shows a qualitative comparison of cropped results from different algorithms. Table 3.2 shows a quantitative evaluation of each method.

### 3.5.3 Deblurring Real Photographs

In Fig. 3.9 and Fig. 3.10 we show two comparisons on real photos with unknown camera shakes. Again for fair comparison we use different algorithms– [70, 21, 140] and ours–to estimate the kernel, followed by Zoran and Weiss [154] for the final non-blind deconvolution. However, such images often exhibit spatially varying blur, so a kernel might only explain (hence sharpen) some regions of the image. Overall our method is robust and generates results that are comparable to, if not better, than the state-of-the-art methods.

Figure 3.9: An example photos with unknown camera shake. Our method produces competitive results.

Figure 3.10: An example photo with unknown camera shake taken from Xu and Jia [140]. Our method produces sharper edges around the texts and less ringing in densely textured regions.
3.6 Conclusion

We explore a new approach for kernel estimation from a single image via modeling image edge primitives using patch priors. Two type of priors are proposed, including a statistical prior learned from natural images, and a simple synthetic prior. We further show how to incorporate the priors into a deblurring objective to significantly improve the state-of-the-art performance. As future work, we would like to extend our image formation model to handle more severe noise and other outliers to make it more robust on low quality inputs. We would also like to improve the computational efficiency to make the system more practical.
Chapter 4

Good Image Priors for Non-blind Deconvolution: Generic vs Specific

There are no rules for good photographs, there are only good photographs.

Ansel Adams

4.1 Introduction

Deblurring is a long-standing challenge in the field of computer vision and computational photography because of its ill-posed nature. In non-blind deconvolution, even though the point spread function (PSF) is known, restoring coherent high frequency image details can still be very difficult. In this paper, we address the problem of non-blind deconvolution with the help of similar (but not identical) example images, and explore deblurring performance across a spectrum of example image scenarios. For each type of training data, we evaluate various strategies for learning image priors from these examples. In contrast to popular methods that apply a single universal image prior to all pixels in the image [76, 80, 69, 140, 81], we adapt the prior to local image content and introduce a multi-scale patch modeling strategy to fully take advantage of the example images and show improved recovery of image details. Unlike the recent instance-level deblurring method of [50], we do not require accurate dense correspondence between image pairs and hence generalize better to a wide variety of example image scenarios.
In a typical deblurring framework, a blurry image $y$ is often modeled as a convolution between a PSF $k$ and a sharp image $x$, with additive noise $n$:

$$y = k * x + n.$$  \hspace{1cm} (4.1)

In non-blind deconvolution, both $y$ and $k$ are given, and $n$ is often assumed to be i.i.d Gaussian with known variance. A typical choice of image prior is to encode the heavy-tailed characteristics on image gradients [76, 80, 69], and regularize the deconvolution process via some form of sparsity constraints on image gradients:

$$x = \arg \min_x ||y - k * x||^2 + \lambda (||D_x x||^\alpha + ||D_y x||^\alpha)$$  \hspace{1cm} (4.2)

where $\lambda$ is proportional to the noise variance. For Gaussian priors ($\alpha = 2$), there exist fast closed-form solutions via Fourier transform [76, 21]. However, Gaussian priors are not appropriate for capturing the heavy-tailedness of natural images, hence produce oversmoothed image gradients. Sparsity priors based on Laplace distribution ($\alpha = 1$) [76] and hyper-Laplacian distributions ($0.5 \leq \alpha \leq 0.8$) [69] have been shown to work well. Other forms of parameterization have also been introduced, such as the generalized Gaussian distribution [23] and mixture of Laplacians [79]. Constraints on image gradients alone are usually insufficient and methods that are able to reason about larger neighborhoods lead to state-of-the-art performance [101, 154, 147, 104, 105]. In particular, Zoran and Weiss [154] model image patches via a simple Gaussian mixture model (GMM). This prior turns out to be extremely powerful for removing blur and noise. More recently, discriminative methods trained on corrupted/sharp patch pairs [104, 105] have shown impressive performance without specifically modeling the image prior. However, a common problem for these generic methods is that restoring coherent high frequency details remains a challenging task. Deblurred results often contain artifacts such as broken lines and painterly structure details (see Fig. 4.1).

One likely cause is that given only very local image evidence based on a few adjacent pixels [76, 69, 23] or image patches [101, 134, 154, 78, 147], there is insufficient contextual information to drive the solution away from the conservative smooth state. In addition, most existing methods apply a single image prior to the whole image, which will inevitably introduce a bias towards smooth solutions, since natural images are dominated by smooth gradients.

To combat the tendency to oversmooth details, several recent works consider a content-aware formulation of image priors to accommodate the spatially-varying statistical properties in natural images [23, 26, 157]. While such content-aware approaches are promising, it is difficult to choose the right prior in the presence of
blur and noise. For example, [23, 26] estimate content-aware parametric priors based on the downsampled input image. The power of such internal statistics can be rather limited when faced with limited resolution or large blur. However, constructing expressive, content-aware image priors becomes feasible if we have access to sharp example images that are similar to the input.

In the digital age, photographers are likely to take many photos of the same physical scene over time, and this is the type of context we exploit to restore an image and enable content-aware adaptation of image priors. As an experiment, we randomly picked 100 query photos on Flickr and found instance level scene matches right next to the query in their respective photostream 42% of the time. This is probably a conservative estimate because photographers are exercising editorial restraint and tend to only publish good and unique photos. For photos where the shutter count was visible, 29% of the time the photographer had taken additional (non-uploaded) photos between instance level matching scenes. It is frustrating for photographers that restoring a blurry photo, even when they can often provide sharp photos of the same scene, remains a problem seldom considered by the research community, with the exception of [50], which requires a dense correspondence between the input and the example. However, in the presence of blur and noise, such dense correspondence is unreliable and cannot handle occlusions (see Fig. 4.1).

Given the recent advances in blur kernel estimation [38, 80, 21, 140, 120, 68] and the fact that non-blind deconvolution can be regarded as separate step in the deblurring process, we consider the stand-alone problem of by-example non-blind deconvolution: given a blurry input image, a known PSF, and one or more sharp images with shared content, how can we reliably remove blur and restore coherent image details?

4.2 Overview

In order to explore non-blind deconvolution performance over a broad range of example image scenarios, we need to define a general deconvolution framework. We extend the EPLL/GMM framework from Zoran and Weiss [154] by augmenting the single-scale patch priors to a multi-scale formulation (Sec. 4.3). Once the form of image prior and deconvolution method is defined, we consider two training strategies: global training using data from example images, or local training using specific subsets of example data based on a region level correspondence (Sec. 4.4). Based on this setup, we can investigate various baseline methods that incorporate (1) different parameters in the prior configuration, and (2) different training strategies. We evaluate the performance of these baselines for each example image scenario (Sec. 4.5) and discover a set of key strategies that show significant benefit from having better example images. Finally, we compare
Figure 4.1: The synthetically blurred input and sharp example images show different views of downtown Seattle. Even when given the groundtruth input image, the core correspondence algorithm in [49, 50] returns partial (22%) correspondence from example 1 and zero matches from example 2. Our algorithm is able to establish meaningful region level correspondences, and locally adapt the prior to produce significantly more details than state-of-the-art non-blind deconvolution methods.

experimental results (Sec. 5.5) using both synthetically blurred and real photos, against leading methods in generic non-deconvolution as well as by-example deblurring.

4.3 Patch-pyramid Prior

Our work builds on Zoran and Weiss [154] in which a single-scale patch prior is trained from DC-removed patches. Natural images exhibit diverse yet structured content in different frequency bands that are tightly coupled. A single-scale patch model lacks the ability to learn such statistical dependencies. We propose to jointly model multi-scale concentric patches extracted from an image pyramid, which we call patch-pyramids. This naturally extends the spatial scale of the patches without a geometric increase in dimensionality as would happen at a single scale. Furthermore, by capturing how mid and high frequency details covary, image details can be restored more coherently to remove common artifacts such as smudged-out structures, zigzag edges,
Consider an image $x_1$ and its Gaussian pyramid layers $\{x_1, \ldots, x_m\}$. Given a fixed patch width $w$, we denote a patch-pyramid by $[x_1^i]^1$, meaning a collection of $m$ patches centered at the same relative coordinates $i$ in each layer of the Gaussian pyramid. For conciseness, we use $[x]^i$ to denote patch-pyramid at relative location $i$ with some fixed size. We use bold fonts to indicate matrices. $[x]^i \in \mathbb{R}^{mw^2}$ is formed by concatenating patches in each layer of the pyramid.

We treat patch-pyramids with DC removed per layer as random variables and model the joint occurrence of these $m$ layers via a Gaussian Mixture Model (GMM). For simplicity, a $w \times w \times m$ GMM prior means that the model is trained using patch size $w$ with $m$ layers.

Let $x$ and $y$ be the latent and observed image. We follow the EPLL framework of [154] to minimize:

$$f_p(x|y) = \frac{\lambda}{2} ||Ax - y||^2 - \sum_i \log p([x]^i)$$

where $A$ represents the blur operator, $\lambda = \frac{mw^2}{\sigma^2}$, $\sigma^2$ is the noise variance in the image formation process, and $p([x]^i) \sim \sum_k \pi_k N([x]^i; \mu_k, \Sigma_k)$ is the density function of the GMM prior for patch-pyramids. $\{\pi_k, \mu_k, \Sigma_k\}$ are the mixture weight, mean, and covariance of the $k^{th}$ Gaussian component, respectively. The single-scale patch model in [154] is a special case when $m = 1$ and $\mu = 0$.

### 4.3.1 Optimization

To optimize Eqn. (4.3) directly is challenging. A common strategy is to introduce auxiliary variables to assist the optimization process via half quadratic split [69, 154]. To achieve this, we introduce auxiliary patch-pyramids $[z]^i$ to each location $i$ and minimize the following global objective:

$$c_{p, \beta}(x, \{[z]^i\}|y) = \frac{\lambda}{2} ||Ax - y||^2 + \sum_i \frac{\beta}{2} (||[x]^i - [z]^i||^2 \Sigma_{noise}^{-1} ([x]^i - [z]^i)) - \log p([z]^i)$$

The diagonal matrix $\Sigma_{noise}$ reflects the varying relative noise level in each layer, with diagonal entries $\sigma^2_j, j \in \{1, 2, \ldots, m\}$, each repeating $w^2$ times. However, the noise across layers is correlated due to the effect of filtering and downsampling in the Gaussian pyramid. We empirically found the relationship $\sigma^2_{j+1} = \sigma^2_j / 2$ to work well in our experiments. We set $\sigma^2_1 = 1$.

The optimization iterates between updating the auxiliary variables $[z]^i$ (Sec. 4.3.2) and solving the latent
image \( x \) (Sec. 4.3.3). Over iterations, \( \beta \) increases to tighten the coupling of \( [z]^i \) and \( [x]^i \) via the second term, which enables convergence. We empirically found the schedule \( \beta = 60 \cdot [1, 2, 4, \ldots] \) to work well, typically converging within 8 iterations as shown in Fig. 4.4.

### 4.3.2 Z-Step

Given the current estimate for \( x \), finding \( [z]^i \) amounts to solving for the MAP estimate, but computing the exact MAP solution is intractable. We follow the approximation procedure from [154] to obtain a Wiener filtering solution:

\[
[z]^i = (\Sigma_{k_{\text{max}}} + \beta \Sigma_{\text{noise}})^{-1} (\Sigma_{k_{\text{max}}} [x]^i + \beta \Sigma_{\text{noise}} \mu_{k_{\text{max}}})
\] (4.5)

where \( k_{\text{max}} \) is the index of the Gaussian component with the highest responsibility.

### 4.3.3 X-step

Keeping \( [z]^i \) fixed, we solve for \( x \) by the following update:

\[
\hat{x} = \left( \lambda A^T A + \sum_i \sum_j \beta_j (P_{ij} H_j)^T (P_{ij} H_j) \right)^{-1} \left( \lambda A^T y + \sum_i \sum_j \beta_j (P_{ij} H_j)^T [z]^i_j \right)
\] (4.6)

where \( j \) indexes over layers in the pyramid, \( H_j \) is the Toeplitz matrix representation of the Gaussian filtering and downsampling operators associated with layer \( j \), \( \beta_j = \beta / \sigma_j^2 \), \( [z]^i_j \) is the \( j^{th} \) layer patch in \( [z]^i \), and \( P_{ij} \) is the matrix operator extracting the patch at location \( i \) in layer \( j \).

### 4.4 Locally Adapted Priors

Clearly, the prior in Eqn. (4.3) plays a central role in the deblurring process. But how much can the prior benefit from example images? One way is to learn the GMM parameters globally using training data collected from the example images. Unfortunately, globally trained priors do not seem to benefit from having better example images, as we will show in Sec. 4.5. This may be because image statistics vary significantly across image locations and using a single global image prior for all image content inevitably compromises image details for smoothness. Instead, we show that priors can be adapted to local image content to provide significantly better recovery of image details.
Figure 4.2: (a) Input blurred image with known PSF and sharp example images, (b) initial latent image, (c) best matching example image crops for several query crops from the input, (d) visualization of the nearest neighbor crops overlaid on the input image. The initial latent image is very noisy, the nearest neighbor crops are misaligned and incoherent. Neither alone is a satisfactory image restoration, but we will use the information from both sources to restore blurry photos.

Figure 4.3: (a) Using patch-pyramids from nearest neighbor crops for the bottom query crop in Fig. 4.2(c), we train a $7 \times 7 \times 2$ local GMM and compare its random samples (left) against patches drawn directly from training data (right). The prior captures intricate coupling in different frequency bands. (b) The global objective function in Eqn. (4.3) converges over iterations with a fixed schedule for $\beta$, while the PSNR of the latent image increases. Locally trained $7 \times 7 \times 2$ priors are used to restore the input image in Fig. 4.2.

To construct locally adapted priors, we operate on a half-overlapping grid of image crops and seek local correspondence as shown in Fig. 4.2. First, a fast $L_2$-based deconvolution is performed to provide a rough estimate of the latent image, which is then divided into half-overlapping $64 \times 64$ crops. For each crop, a HOG descriptor [28] is computed and compared against a database of crops extracted from the sharp example images. We apply scale (factor of 1, 0.9, 0.8) and rotation ($-3, 0, 3$ degrees) adjustments to each example image to better fit query image content. To reduce noise, we downsample the image by 0.5 in each dimension and apply Gaussian blur before computing the HOG features. A visualization of the nearest neighbor (NN) crops overlay is shown in Fig. 4.2, where salient image content is matched to reasonable example crops in the presence of noise. Additional visualizations across various example image scenarios can be found in Fig. 4.4.

Given the above crop-level correspondences, we train independent local GMM priors using patch data
collected from 20 nearest neighbors for each query. For each query crop $q_i$, we adaptively choose the number of Gaussian components $K_i \in [K_{\min}, K_{\max}]$ according to gradient complexity in the training data. Specifically, we first run canny edge detection on the sharp example images, and the total count of edge pixels $N_i$ in the 20-NN crops for each $q_i$ is recorded. We linearly scale $K_i$’s by $K_i = K_{\min} + (K_{\max} - K_{\min})(N_i - N_{\min})/(N_{\max} - N_{\min})$, where $N_{\min}$ and $N_{\max}$ are the smallest and largest count among all queries. We set $K_{\min} = 5$, $K_{\max} = 50$ and learn the GMM via the Expectation-Maximization (EM) algorithm.

Due to the overlapping structure, each pixel is governed by at most four different local GMM priors. To be consistent with the overall objective in Eqn. (4.3), we choose the solution that gives the highest posterior log likelihood during the MAP approximation of $|z|^i$ (see Sec. 4.3.2).

### 4.5 How Do Example Images Help?

In order to answer this question, we consider how performance is affected by (1) various example image scenarios and (2) different parameters in our prior. Since the state-of-the-art by-example deblurring method of [50] requires instance-level examples, it is hard to evaluate its performance across a wide spectrum of examples. In this section only, we use the groundtruth image content for retrieving similar scenes as well as finding crop-level correspondences so that we can more accurately experimentally manipulate the quality of training data.

We consider a number of scenarios of example images: oracle, instance-level, scene matches, and random scenes. The test images are synthetically formed based on the landmark dataset of [149] (see Sec. 4.6.1 for details). The oracle scenario assumes that the groundtruth image is available for training the GMM priors. The instance-level examples come directly from the dataset of [149]. Scene matches are computed using the method and database described in [52]. The “good” scene matches (rank 1 to 3) are very similar scenes at similar scale under similar illumination, but typically not instance-level matches. The “fair” scene matches (rank 10-12) are usually less similar but still reasonable. The “bad” scene matches (rank 1998-2000) might only be of the same broad scene category. Finally, we select three random scenes from the database of [52] to act as the worst case scenario. See Figure 4.4 for examples of each set of training images.

For each example scenario, we consider six alternative prior configurations: (1) the prior can be either globally or locally trained, and (2) the patch-pyramid dimensions can be $7 \times 7 \times 1$, $5 \times 5 \times 2$, $5 \times 5 \times 3$. For the globally trained priors, we randomly sample $2 \times 10^6$ patch-pyramids from the example images (with scale and rotation adjustments) and learn a 50-component GMM via mini-batch EM.
Figure 4.4: Comparing various baselines across example scenarios and prior configurations. From top to bottom: various scenarios of example images, from the best possible (groundtruth) to similar scenes, to irrelevant images (random scenes); averaged overlay of 20 nearest neighbor crops; output using globally trained priors and locally adapted priors. Results obtained using $7 \times 7 \times 1$, $5 \times 5 \times 2$ and $5 \times 5 \times 3$ GMM priors are shown in row (a), (b) and (c) respectively. Better image details can be recovered by (1) using better example images and (2) local training of patch-pyramid priors.
So how do example images help? Our experiments show that the answer is rather subtle: it depends on the priors. In Fig. 4.4, we show how the deblurring results change as the training examples become less similar to the blurry input. Using a test set of 20 images (see Sec. 4.6.1), we present quantitative evaluation of these baselines in Fig. 4.5.

We summarize several key observations below:

1. Better example images do help, but it also depends on the priors being used. Locally adapted priors appear to be very sensitive to example images, whereas global priors are not.

2. Given instance-level example images, local priors significantly outperform global priors. This is because local priors can provide fine-grained content-aware constraints whereas global priors apply a universal treatment to all image content, often introducing a bias towards smoothness.

3. Given sufficiently similar examples (not necessarily instance-level), multi-scale priors outperform single-scale priors. Quantitatively, the $5 \times 5 \times 2$ prior consistently performs the best (both global and local). In Fig. 4.4, better connected edges and structured details become much more visible under multi-scale priors.
Figure 4.6: Comparison on uniformly blurred synthetic test images. Groundtruth PSF’s are assumed known and used by all competing methods.
Table 4.1: Quantitative evaluation against existing methods. Methods [76, 69, 154, 104] utilize universally learned image information for deconvolution, while [50] and our method focus on by-example deblurring. For fair comparison, our results in the last column are produced with the estimated PSF from [50]. Both methods make use of example images.

4.6 Comparison to Leading Methods

With the above analysis and observations, we combine local training and multi-scale patch-pyramid modeling, and report our results using $7 \times 7 \times 2$ local priors for subsequent comparisons. For comprehensive evaluation, we consider a wide range of test images, containing both synthetic uniform blur and real unknown camera shake. We present quantitative and qualitative comparisons against leading methods in both generic and by-example deblurring methods.

4.6.1 Synthetically Blurred Images

For quantitative evaluation, we generate 20 synthetically blurred test images using four kernels (number 2, 4, 6, 8) from Levin et al. [75] and five color images with examples taken from [149]. 1% i.i.d Gaussian noise is added to the luminance channel. Evaluation is based on only the gray scale output images with the outer ring of 30 pixels removed. Color information is only used to assist the correspondence step in [50] and our pipeline (see Sec. 4.4). In Table 4.1, we show quantitative comparisons based on PSNR and SSIM scores. For comparisons against non-blind deconvolution methods [76, 69, 154, 104], we assume the groundtruth PSF is known. In this case, our performance is better than the compared methods 100% of the time. A visual comparison of deblurred results can be found in Fig. 4.6.

When comparing to the recent by-example blind deblurring method of [50], we assume the groundtruth PSF is unknown, and run our system with the estimated blur kernels provided by the authors of [50] to ensure fair comparison. We report PSNR and SSIM performance in Table 4.1. In this case, we outperform the method of HaCohen et al. [50] 85% of the time. A qualitative comparison is shown in Fig. 4.8. Please note that a single example image is manually selected by the authors of [50] (out of all the examples we supplied) to generate their results since their system does not support multiple example images.

Our method clearly outperforms existing methods in terms of PSNR and SSIM scores, and is capable...
Our approach is highly competitive without requiring dense correspondence.

of restoring coherent mid to high level frequencies such as straight lines and structured details. The recent methods of [154, 104] are very competitive without using context-specific example images, but can be quite limited in terms of recovering high frequency details, as shown in Fig. 4.1 and Fig. 4.6.

### 4.6.2 Real Photos with Unknown Blur

In Fig. 4.7, we show comparison on a test image from [50], where the input image exhibits unknown and spatially varying blur. Our latent image is produced with the PSF estimates from [50], and shows competitive restoration of details. In Fig. 4.9, we present additional results with unknown blur. All images are taken with the same camera. For most of the test cases, we were unable to obtain successful dense correspondences using the online code provided by the NRDC algorithm [49], which is at the heart of [50].

### 4.6.3 Limitations

While our system achieves competitive restoration of details, it requires heavy computation especially in the training stage. Using our unoptimized MATLAB implementation, training a 50-component $5 \times 5 \times 2$ GMM global prior takes roughly 5 hours on an Intel Xeon E5-2650 CPU, whereas training its local prior counterpart requires 12 minutes over a compute grid using 120 cores. However, we find that simply changing the stopping criteria for EM lets us speed up training by a factor of 100 at the expense of a 0.03 drop in PSNR on average. We speculate that further speedup can be obtained by reducing the number of parameters to learn via PCA and by optimizing our code. Finally, incorrect synthesis of details can occur near texture transitions, as shown in Fig. 4.10.
Figure 4.8: Comparisons against the state-of-the-art by-example method of HaCohen et al. [50] on our uniformly blurred synthetic test images. Four examples are shown. Within each example, the first row shows (from left to right): dense correspondence found by [50], output of [50] with estimated PSF (top-left) and groundtruth PSF (top-right), close-up of [50]. The second row shows (from left to right): our nearest neighbor example crop overlay, our output, our close-up. The PSF estimates are supplied by the authors of [50]. All results are generated using the same input blurry images and PSF estimates, hence directly comparable. The last example shows a failure case due to inaccurate PSF estimate.
Figure 4.9: Except for the third row, the core correspondence algorithm at the heart of [50] yields zero successful matches. For the third test image, it cannot explain more than 70% of the image. All input images are real photos with unknown blur. We estimate the blur kernel using [21].

Figure 4.10: An example where our method produces convincing textures but also inappropriate high frequency content in background smooth regions (bottom crop).
4.7 Conclusion

In this work, we have provided a novel analysis for by-example non-blind deconvolution by comparing performance against quality of example images for various scenarios using patch-based priors. In particular, we show that locally adapted priors with multi-scale patch-pyramid modeling leads to significant performance gains. We propose a method relying on mid-level correspondence of image crops that does not require dense correspondence at the pixel level. By modeling local image content using multi-scale patch-pyramids, our approach can efficiently take advantage of the sharp example images to restore coherent mid to high frequency image details. We conduct extensive evaluation based on images with both synthetic and real blur, comparing against leading methods in non-blind deconvolution as well as the state-of-the-art by-example deblurring method. By-example deblurring is a promising direction to alleviate the fundamental difficulty of existing algorithms to restore coherent high frequency details, and our method is one step closer to achieving high quality deblur results. For future work, we would like to investigate how our approach can be extended to utilize non-instance level (but still similar) example images and explore ways to improve blind deconvolution via examples.
Chapter 5

Super-resolution From Internet Scale

Scene Matching

Color is my day-long obsession, joy and torment.

__________________________
Claude Monet

5.1 Introduction

Single image super-resolution is a well-studied problem where one tries to estimate a high-resolution image from a single low-resolution input. Unlike multi-frame super-resolution where a sequence of low-resolution images of the same scene aligned to subpixel shifts reveal some high frequency detail signals, it is impossible to unambiguously restore high frequencies in a single image super-resolution framework. Single image super-resolution is an extremely under-constrained problem: there are many plausible natural images that would downsample exactly to a given low-resolution input. As a result, existing works in the field present intelligent ways to hallucinate plausible image content instead of recovering the ground truth.

Over the past decades there has been impressive work on single image super-resolution, but no method is
Figure 5.1: Super-resolution results for 8x upsampling. The input image is 128 pixels wide. We compare our results to those of Sun and Tappen [119] and Glasner et al. [45].

able to address the fundamental problem of synthesizing novel object and texture detail. Many methods can sharpen edges, but the long term challenge in this field is to produce realistic, context-appropriate texture, material, and object detail. This limitation becomes more apparent as the desired magnification factor increases. State-of-the-art methods [45, 119] can sometimes synthesize convincing detail for blurs equivalent to a 2x or 3x loss of resolution, but we compare algorithms with a more challenging task – 8x super-resolution. While 8x magnification might seem extreme, the equivalent amount of detail loss is commonly caused by imaging artifacts such as defocus or motion blur.

How can an algorithm synthesize appropriate detail for an arbitrary, low-resolution scene? Our visual experience is extraordinarily varied and complex – photos depict a huge variety of scene types with different viewpoints, scales, illuminations, and materials. How can an algorithm have image appearance models specific to each possible scene configuration? Recent works [29, 51, 52, 63] show that with diverse, “Internet scale” photo collections containing millions of scenes, for most query photos there exist numerous examples of very similar scenes. A key insight of this paper is that research in scene representation and matching has advanced such that one can find similar enough scenes even when a query is very low-resolution and then use these matching scenes as a context-specific high-resolution appearance model to enhance the blurry scene. This lets us convincingly enhance image detail at magnification factors beyond previous super-resolution
methods.

Our primary contributions are that: (1) We examine scene matching in a low-resolution regime that has rarely been studied. The notable exception is “Tiny Images” [128] which limited experiments to an intentionally impoverished representation. (2) We quantify the expressiveness and predictive power of matched scenes and show that they are competitive with single-image priors. This contrasts with and expands upon the findings of Zontak and Irani [153]. (3) We produce super-resolution results with plausible image detail beyond the capabilities of existing super-resolution method for diverse photographic scenes. Compared to previous work, our results are especially convincing for texture transitions which challenge previous region-matching super-resolution methods.

5.1.1 Repairing Image Blur

There is an enormous body of research aimed at alleviating the effects of blur-inducing imaging phenomena – defocus, motion, and scattering to name a few. Photographic blur can not be unambiguously inverted in realistic imaging conditions [7], therefore “...the central challenge ... is to develop methods to disambiguate solutions and bias the processes toward more likely results given some prior information”[64]. Deblurring algorithms tend to use relatively compact, parametric image priors, often learned from natural image statistics, that encode principles such as “edges should be sharp”, “gradients should be rare”, “colors should be locally smooth” [151, 99, 100, 11, 133, 64, 23]. These parametric models are helpful but limited. Their assumptions, such as a heavy-tailed gradient distribution, are not universally true [23]. In general, these models can sharpen edges but will not enhance texture, material, or object detail because these phenomena are too complex for the models.

5.1.2 Super-resolution

Unlike the previous causes of blur in which inverting a point spread function can sometimes yield useful detail, with single image super-resolution it is clearer which detail can be “recovered” (none of it) and what detail must be “hallucinated” or “synthesized” (all of it). This ambiguity makes super-resolution a demanding application for statistical image models or priors.

While some recent super-resolution methods use parametric image priors similar to those used in deblurring applications (e.g. [37, 86]), many super-resolution methods in the last decade utilize data-driven image priors, starting with the seminal work of Freeman et al. [40, 41]. Such data-driven methods implicitly or
explicitly “learn” the mapping between low and high-resolution image patches [67, 48, 123, 115]. A data-driven prior does not make the super-resolution problem any less ambiguous – it is simply a more expressive model for proposing high-frequency versions of low-resolution image content.

Consider a hypothetical, ideal super-resolution algorithm. When presented with a low-resolution mountain, it would insert details only appropriate to mountains. When presented with a face, it would insert details specific to faces. This idea led to the development of very effective domain specific face super-resolution algorithms [7, 83]. But for real scenes, to insert the most plausible detail, one must first recognize the context. The seminal work of Baker and Kanade [7] refers to this process as “recognition”, a portmanteau of “recognition” and “reconstruction”.

To achieve “recognition” one needs to go beyond compact, parametric models or data-driven models trained from tiny image patches that are not expressive enough for recognition or reconstruction. Recent works [48, 115] add explicit or implicit material/texture recognition to help alleviate the limits of these local, compact representations. In both methods, low-resolution input images are segmented and each segment is constrained to synthesize details by drawing patches from matched material or texture regions which are hopefully semantically and visually similar. These methods are very promising, but in both cases the material matching is not reliable – material recognition is very hard [84] and it is even harder at low-resolution. [48] alleviates this difficulty with manual intervention. Another difficulty with these approaches is handling boundaries between texture regions. [48] resorts to self-similarity for edge refinement because they do not have training examples of texture transitions. In [115], the segments do not capture the diverse texture transition scenarios either, and their algorithm relies on an edge smoothness prior to produce sharp edges. Our algorithm requires no such special case because our matched scenes typically contain the same texture transitions as our query scene.

One complementary and surprisingly effective way to synthesize scene-appropriate detail is to build a statistical image model from the low-resolution input image itself [45]. This only works when a scene exhibits self-similarity across scales, but this is common because perspective projection causes surfaces to span many scales. More recently, Zontak and Irani [153] argue that these “internal” image statistics are often a better prior than “external” image databases for image restoration tasks. One of our key results is to use the evaluation protocol of [153] to show that it is possible to compete with single-image internal statistics by intelligently leveraging a large image database (Section 5.3.1).

\(^1\)The “recognition” does not need to be explicit – an algorithm needs to establish correspondence among visually and semantically similar image content, whether that involves explicit classification or not.
The methods by Sun and Tappen [115] and Glasner et al. [45] are representative of the state-of-the-art in automatic super-resolution, but they still do not reliably insert texture or object detail into photographs. More often than not the results show sharper edges but are not convincing beyond magnification factors of 2 or 3. We compare our results to these algorithms in Section 5.5.

Beyond single image super-resolution, there is ongoing research for which the input is multiple photographs of the same physical scene. For instance, “Photozoom” [36] relates photographs with a hierarchy of homographies and then transfers details. Lastly, image enhancement methods such as “CG2Real” [63] can be modified to perform super-resolution by inputting blurry scenes. However, CG2Real assumes that the input is corrupted in some way and thus is not faithful to the input image, as is desirable in super-resolution.

5.1.3 Super-resolution Goals and Evaluation

In typical image restoration and super-resolution literature (e.g. [64, 23]) the formal goal is to recover “clean” scene \( x \) given blurred scene \( y \), a known PSF (or blur kernel) \( k \), and a known downsampling function \( D \). These variables have the following relationship: \( y = D(x \otimes k) + n \) where \( \otimes \) is the convolution operator and \( n \) is a noise term. One can then evaluate a result by comparing the estimated \( x \) to the known, “ground truth” \( x \) which generated \( y \).

This evaluation makes sense when (1) \( k \) is small or invertible and (2) you are interested in forensically accurate reconstructions. But when either \( k \) or the downsampling factor becomes large, the possible values for \( x \) grow enormously. For 8x super-resolution, the output space is 64 times higher-dimension than the observed low-resolution input. There is an enormous space of detailed and plausible output images that are faithful to the low-resolution input. Why should one penalize a convincing result just because it doesn’t resemble the observed “ground truth”? Recognizing this problem, recent work has adopted a more forgiving comparison between the estimated \( x \) and “ground truth” – SSIM [130] – which rewards local structural similarity rather than exact pixel to pixel correspondence. However, SSIM and other existing measurements of reconstruction error penalize texture hallucination (See Figure 5.2 for an example).

Rather than evaluating reconstruction error, an alternative is to perform human perceptual studies [86]. Such experiments are difficult, though, because of the subjective biases of individual, non-expert observers. In Section 5.5 we perform such a study. However, we think the most diagnostic results are qualitative in nature – in Section 5.5 we show that our approach is able to insert edge and texture detail in diverse scenes where previous methods could not.

\(^2\text{For super-resolution a Gaussian blur of appropriate width can be used as the PSF [45].}\)
Figure 5.2: SSIM scores calculated with respect to the reference patch on the left. The middle patch, cropped from the same texture, scores poorly while the patch on the right, a blurred version of the reference, scores very highly. Because SSIM and other reconstruction measures favor blur over texture misalignment, they favor conservative algorithms which do not insert texture details.

Figure 5.3: Our proposed pipeline. From left to right, for a low-resolution input we find most similar scenes from a large database. Each input segment is corresponded with best matching segments in these similar scenes. Then a patch-based super-resolution algorithm is used to insert detail from the matched scene segments.

5.2 Algorithm Overview

Our algorithm (Figure 5.3) first finds matching scenes from a large Internet database (Section 5.3). The input image and each matching scene are segmented and a correspondence is found between each segment in the input and several best matching segments from the similar scenes (Section 5.4.1). Finally, each input segment is upsampled by matching low-resolution patches and transferring in high-resolution details from its corresponding segments (Section 5.4.2).

The local patch matching at the heart of most data-driven super-resolution algorithms is fundamentally ambiguous – it is hard to match to semantically similar texture based on local image evidence regardless of the size of the training database. Our pipeline follows a coarse-to-fine structure not just to reduce computational complexity, but also to alleviate this ambiguity by constraining matching to segments from scenes which are hopefully semantically similar. Instead of making decisions entirely locally, we make easier decisions at the scene and segment level first. Constraining the synthesis process to a small number of regions from similar
scenes also increases perceived texture coherence.

5.3 Scene Matching

Our proposed detail synthesis pipeline can be thought of as taking the data-driven super-resolution trend to its extreme by using a massive, “Internet-scale” photo collection as an extremely detailed statistical image model. While the state-of-the-art method of [115] uses a training set of four thousand images, the largest to date, our algorithm uses more than six million images. We follow in the footsteps of several successful massively data-driven scene matching algorithms, e.g. [51, 114, 29, 103, 63], which sample the space of scenes so densely that for most query scenes one can find semantically and structurally similar scenes.

A key insight for this paper is that while other super-resolution representations and models can not understand the context presented by low-resolution scenes, scene matching can succeed even in the presence of extreme blur. If we can find very similar scenes for a low-resolution query then those scenes provide an ideal set of context-appropriate textures of similar scale, illumination, and viewpoint to use for detail synthesis.

However, our application of scene matching is especially difficult because the input images are low-resolution and thus have degraded textures, which are the most discriminative scene features [138]. To make the most of what scene statistics remain we use a combination of scene descriptors – color histograms, tiny images [128], gist descriptors [91], dense texton histograms [87], sparse bags-of-visual-words [113] built with “soft assignment” [95], geometric layout [56], and surface-specific color and texton histograms [138]. The distances in each feature space are weighted such that each feature contributes roughly equally to the ranking of top scene matches.

For accurate scene matching, the scene features in our photo collection need to be computed at the same resolution as a query scene. However, the query scene can be of arbitrarily low resolution and recomputing features for an entire database is computationally expensive. Therefore we use a hierarchical scene matching process where initial matches are found at a low, fixed resolution, then for each initial match the scene descriptors are recomputed at the query resolution and the matches are re-ranked. Figure 5.4 shows examples of scene matches for several queries where each input image is only 128 pixels wide.

To find similar scenes we need a diverse photo collection with millions of example scenes. We use the Flickr-derived database of [52] which contains over 6 million high resolution photographs. Because we use this photo database to learn the relationship between low-resolution scenes and high-frequency details, it is important that all scenes actually contain high-frequency details. Therefore we filter out all blurry
photographs using the “blur” classifier of [66]. This disqualifies about 1% of photographs. We use the top 20 matches for each input image as a scene-specific training database for detail enhancement.

![Figure 5.4](image)

Figure 5.4: For four low-resolution query scenes, we show six of the top twenty scene matches that our algorithm will use to insert high-frequency detail. The last row shows an example of scene match failure. For a small portion of test cases the scene matching finds some instance-level matches, as in the Venice image, but generally this is not the case. We will explicitly indicate when a result was generated using instance-level matches.

5.3.1 Understanding the Quality of Scene Matches

Data-driven super-resolution methods estimate a high-resolution image by matching to a database of low and high resolution pairs of patches. In our case, the database is a set of query-specific scene matches. Recently, Zontak and Irani [153] proposed criteria to assess the value of training databases for image restoration tasks. First, **expressiveness** quantifies how well patch matches from a database could possibly reconstruct the ground truth. Second, **predictive power** quantifies how effective patch matches from a database are at constraining the solution toward the ground truth. Expressiveness is similar to the “reconstruction error” examined in [5] for image databases with trillions of patches.

In the following subsections, we analyze two “external” databases: (1) our query-specific scene matches
and (2) random scenes from the Berkeley Segmentation Dataset (BSD) [87], and two “internal” databases: (1) a database of all scales of the full resolution ground truth, except for a 21x21 window around the current patch under consideration and (2) a limited internal database of all scales of the input image. Of the two internal databases, only the “limited” variant is applicable to the task of super-resolution because one does not have access to the full-resolution ground truth during super-resolution. Internal databases include patches at scales of $0.8^i, i = \{0, 1, 2, 3, 4, 5, 6\}$ while external databases are not multi-scale.

We use a test set of 80 diverse scenes and evaluate expressiveness and predictive power for the task of 2x super-resolution. We analyze 2x super-resolution to be consistent with [153] even though we show results for 8x super-resolution in Section 5.5. At higher levels of magnification the internal image statistics are increasingly unhelpful for super-resolution. Even though the task is 2x super-resolution, scene matches are found from input images at 1/8 resolution. We resize all images to a maximum dimension of 512 pixels and convert to grayscale. Query patches are sampled uniformly across all gradient magnitudes from input images.

**Expressiveness**

Expressiveness provides an upper-bound for image restoration tasks if there were an oracle guiding selection of high-resolution patches out of a database. An infinite database of random patches would have perfect expressiveness (but poor predictive power). Expressiveness is defined by the average $L_2$ distance between each ground truth patch and its nearest neighbor in a database. Patch comparisons are made with $5 \times 5$ patches with DC removed. Figure 5.5 compares the expressiveness of the ground truth high resolution image, the limited internal scales derived from the input image itself, 20 random images from BSD [87], and the 20 best scene matches for each query.

Zontak and Irani [153] show that it is favorable to exploit the stability of single image statistics for tasks such as denoising and super-resolution because the same level of expressiveness can only be achieved by external databases with hundreds of random images. Indeed, the “internal (all scales)” database outperforms 20 random images and 20 scene matches. But the “internal (limited)” scenario which simulates the super-resolution task is less expressive than both external databases. The 20 scene matches are only slightly more expressive than 20 random images. We believe this is because expressiveness favors variety. However, this variety causes the random BSD images to have less predictive power. Overall, this analysis shows that, compared to other approaches, our scene matches contain slightly more relevant appearance statistics to drive a super-resolution algorithm.
Figure 5.5: Comparison of expressiveness of internal vs external databases. Using up to 20 scene matches, the expressiveness of external database can be significantly better than internal. The “limited” internal database is the low frequencies of the input image that would be usable for a super-resolution algorithm. 150,000 query patches from 80 query images were sampled to generate the plots.

Predictive Power

Figure 5.6: Comparison of prediction error and uncertainty of internal vs external databases. A total of 180,000 query patches sampled uniformly from our 80 test cases are used for this experiment.

The predictive power involves two measurements: (i) prediction error and (ii) prediction uncertainty. For each $5 \times 5$ low-resolution query patch $l$ (DC removed), we find the 9 most similar low-res patches $\{l_i\}_9$ and set the predicted high-res patch to $\hat{h} = \sum_i w_i h_i$, where $h_i$ is the high-res patch corresponding to $l_i$, and $w_i$ is a similarity score defined by $w_i = \exp\left\{-\frac{||l - l_i||^2}{2\sigma^2}\right\}$. Then, prediction error is simply the SSD between the ground truth and estimated high-res patch: $||h_{GT} - \hat{h}||^2$; and prediction uncertainty is approximated by $\text{trace}(\text{cov}_w(h_i, h_j))$, using the same weighting scheme. In our experiments, we set $\sigma^2 = 50$.

Figure 5.6 plots the prediction error (left) and prediction uncertainty (right) against the mean gradient.
magnitude per patch. Prediction error is arguably the most important metric for a super-resolution database, and here our scene matches outperform the internal and random external super-resolution databases. In fact, the “internal (all scales)” condition which is something of an upper-bound for this task is only slightly more predictive than the scene matches.

In the prediction uncertainty evaluation, an unexpected observation is that toward high gradient magnitude the curve starts to drop. We speculate the reason is that (1) high gradient patches contain sufficient information (even at low-res) to make the matching unambiguous, and (2) high gradient patches are rare, thus there are fewer patches to possibly match to.

Overall, our external database of scene matches is more expressive, has lower prediction error, and comparable prediction uncertainty compared with single image statistics (the “limited” scenario which corresponds to super-resolution).

However, the relative expressiveness and prediction power of these strategies can change depending on which transformations are considered and which representation is used for the matching. For instance, expressiveness can be improved significantly by considering transformations of each database such as rotations, scalings, mirroring, and contrast scaling. However, enriching a database in this manner tends to decrease predictive power. Therefore we did not apply these transformations to our external databases. In [153] the internal database includes rotated versions of the input, but adding rotations did not significantly impact our evaluations. Also note that while these plots are a valuable quantitative evaluation of the training database, they are not a direct predictor of synthesis quality. For instance, a good patch-based synthesis algorithm will overcome prediction uncertainty by considering spatial overlap between patches and this analysis intentionally ignores that.

### 5.4 Super-resolution Method

Our detail synthesis algorithm is similar to the method proposed in [115]. The significant difference is that our synthesis method is constrained to sample from a small set of scene matches while [115] uses a universal database of image segments. We also differ from [115] in that we use a greedy texture transfer method which considers high frequency coherence instead of picking candidate patches independently.
5.4.1 Segmentation and Texture Correspondence

While our scene matches provide expressive, context-specific image content for hallucination, we want to constrain the local patch matching further. An exhaustive search over all scene matches while synthesizing textures is inefficient, but more importantly it leads to texture incoherence as each local patch could potentially draw textures from very different sources. Constraining the local texture search by first matching at the region level significantly reduces output incoherence and helps push back against the prediction uncertainty observed in Figure 5.6.

We use a recent hierarchical segmentation algorithm [4] to segment our input and matched scenes. Extremely small segments are merged to nearby ones to provide more stable segment matching results. Each segment is represented by color histograms and texton histograms, and the top 5 most similar scene match segments for each input segment are found using chi-square distance. These segments provide a relevant yet highly constrained search space for detail insertion. An example segment-level correspondence is shown in Figure 5.7.

Figure 5.7: Counter-clockwise from upper left: Input image, top 20 scene matches, and the top 5 matching segments for the largest input segments. Each input segment is restricted to draw texture from slightly expanded versions of these matched segments.

Using non-overlapping segments presents a problem at segment transitions. By definition, segments tend not to contain these transition zones. Such transitions are also hard to find in a universal database of image segments [48, 115]. e.g., even if each region is correctly matched to brick, vegetation, sky, etc., there may be no examples of the transitions between those regions. For this reason, previous methods rely on single-image self-similarity [48] or parametric priors [115] to handle segment boundaries. Alternatively, scene matches allow us to elegantly handle texture transitions and boundaries because our scene matches often contain the same transitions (e.g. building to grass, tree to sky) as a query scene. We simply expand each segmented region to include the transition region of textures and boundaries. Thus our segmentations are
Figure 5.8: Results on man-made scenes. Appropriate textures/materials can be observed among the trees in (c) and surfaces in (a). Edges appear realistic and detailed in (b).

actually overlapping and not a strict partitioning of the images.

5.4.2 Segment-level Synthesis of Coherent Textures

As shown in [153] and Figure 5.6, the under-constrained nature of super-resolution causes large uncertainty in the missing high frequencies in the image. When the upsampling factor is large, i.e. 8x, finding appropriate patches based on local evidence alone is fundamentally ambiguous [7].

We use a greedy tiling procedure similar to the “single-pass” algorithm of [41], allowing each subsequent patch choice to be conditioned on existing high frequencies and thus providing a well-constrained environment for synthesizing details. We do not expect this step to generate perfect textures, but allow for
opportunistic insertion of details while remaining faithful to the low frequencies. Let $P^l$ be a low-resolution input patch with DC removed and let $I^h_x, I^h_y$ be the existing image gradient of the output image in the $x$ and $y$ direction respectively. Initially $I^h_x, I^h_y$ are set to 0. Let $S_P$ be the segment containing patch $P$, and $S(S_P)$ be the top 5 most similar example segments to $S_P$. We seek to find among $S(S_P)$ a patch $Q$ (with DC removed) that is both faithful and coherent:

$$Q = \arg \min_{Q \in S(S_P)} D_f(P^l, Q^l) + \beta D_c(I^h_x, I^h_y, Q^h)$$  \hspace{1cm} (5.1)
where

\[ D_f(P^l, Q^l) = \sum_i |P^l(i) - Q^l(i)| \]  

(5.2)

\[ D_c(I^h_x, I^h_y, Q^h) = \sum_{j \in \text{overlap}} |I^h_x(j) - \nabla Q^h_x(j)| \]

\[ + |I^h_y(j) - \nabla Q^h_y(j)| \]  

(5.3)

Then we update the existing high frequencies by a weighted average of the gradients copied from \( Q^h \), with the weights for each \( Q \) defined by 

\[ w = (D_f + \beta D_c)^{-0.8} \]

Query patches are sampled over a half-overlapping grid while database patches are densely sampled.

After we have our set of overlapped patches, we carry out the super-resolution optimization described in [115], using the set of patches \( \{Q\} \) to generate the pixel candidates for the hallucination term, so that neighboring pixels in the output image will be collectively constrained by a group of coherent pixel values. Similar to other super-resolution methods, we find it advantageous to incrementally upsample the image, so we upsample the input image by a factor 2 three times to achieve 8x magnification. We make no effort to optimize the running time of our algorithm so it is quite slow – roughly four hours per image, most of which is spent on the last 2x upsampling.

### 5.5 Results

We compare our algorithm against two recent methods which we consider exemplary of the state-of-the-art in super-resolution – [115] uses segment-level matching to a database of thousands of images, and [45] uses internal image statistics. We also show bicubic interpolation as a baseline. Figure 5.8 and figure 5.9 show results on man-made scenes and natural scenes, respectively. Figure 5.10 shows results where some of our scene matches are instance-level matches. Finally, figure 5.11 shows cases in which our algorithm produces undesirable artifacts. To help visualize the level of detail achieved by each method we zoom in on three crops from each result. In general, our results exhibit sharp edges, natural transition of textures and distinctive details.

Figure 5.12 compares results from our algorithm, which draws texture from matched scenes, against a baseline which instead uses random scenes from our database. This “random scene” baseline is similar to the early data-driven super-resolution method of [41] in which patches were matched in a small, universal
Figure 5.10: Results where we have at least one instance level scene match. Our algorithm is able to hallucinate salient image structures. For example, the ferry and arches in (c) are successfully hallucinated. In this case, they also approximate the ground truth.

To help evaluate the quality of our results and to further evaluate the contribution of scene matching, we perform a comparative perceptual study with 22 participants. We use 20 test cases for the study – 10 which have good scene matches and 10 which have bad scene matches, as evaluated by the authors. As in [86], we show participants pairs of super-resolution outputs from the same input image but different algorithms and ask them to select “the image they think has better quality”. We also allow a participant to indicate that the images are equally good. The left/right placement of outputs is randomized. In a pilot study we found that that our results and those of Sun and Tappen [115] were almost universally favored over [45] and bicubic, so we exclude them from the main study. Figure 5.13 shows the preference of participants towards each algorithm for the test cases with “good” and “bad” scene matches. While participants seem to favor...
our algorithm when the scene matching is successful, the task is quite subjective – a few users preferred our algorithm on almost all outputs while some users exclusively preferred [115]. We believe this discrepancy arises because our results tend to have more detail but also more artifacts and individual participants weigh these factors differently. We experimented with study designs which ask users about “detail” and “realism” separately, but we find that observers have trouble disentangling these factors.

5.6 Discussion

Our algorithm is somewhat more likely to introduce artifacts than other state-of-the-art algorithms because it is more aggressive about inserting texture detail. Most algorithms err on the side of caution and avoid committing to texture details because a single bad patch-level correspondence can produce a glaring artifact.
Figure 5.12: From top to bottom: super-resolution results using random scenes rather than matching scenes, zoomed in crops, and the corresponding crops from our algorithm using matched scenes.

Figure 5.13: The breakdown of votes when participants compared our results to those of Sun and Tappen [115]. For scenes where the scene matches offered very similar textures to the input (left), participants favor our results. For scenes where the scene matches are spurious or mismatched in scale neither algorithm is favored.

which ruins a result. Only with a large database and our scene matching pipeline can we safely insert textures for many low-resolution images. We do not claim that our algorithm represents the unambiguous state-of-the-art in super-resolution. The algorithms we compare against perform well, especially at lower levels of magnification. Existing algorithms are less likely to make mistakes for inputs where our scene matching algorithm may have trouble, such as indoor or rarely photographed scenes. However, we expect scene matching to perform more reliably as better image descriptors are developed and larger image databases become commonplace. We also think that our approach is complementary to prior methods which make use of internal image statistics, but we think that the quality of “external” databases is likely to increase faster than the quality of “internal” databases which are fundamentally quite limited.

While scene matching quality is important, we believe that the quality of our results is strongly bottlenecked by the well-studied texture transfer problem. Even for scenes with excellent scene matches our algorithm can produce surprisingly poor results. For example, when the scene consists of highly intricate
textures without dominant structure it is hard to synthesize coherent textures, as shown in Figure 5.14. Such
difficulties persist with alternative texture transfer schemes such as those based on Markov Random fields [40]
or texture optimization [48]. While the super-resolution task is certainly “vision hard”, it seems as if there is
much progress to be made by improving relatively low-level texture transfer optimizations.

Figure 5.14: Failure example with excellent scene matches. Top row: input image (left) and scene matches
(right). Bottom row: close-up view of output result at locations indicated by the blue squares.
Chapter 6

Constrained Texture Transfer and Synthesis for Super-resolution

Life is like a camera. Just focus on what’s important and capture the good times, develop from the negatives and if things don’t work out, just take another shot.

- Unknown

6.1 Background and Motivation

In this chapter, we build on traditional texture synthesis work and recent advancement of deep learning approaches to experiment with new texture transfer methods that would be useful to example-based image restoration applications, specifically, single image super-resolution and hallucination. Traditional SR methods are conservative in recovering signals due to PSNR considerations and lack the ability to insert image details beyond recoverable limits. The goal of this final chapter is to examine practical ways towards achieving a higher level of photo-realistic appearance while still being perceptually faithful to the input signal.
6.1.1 Traditional Image Priors Tend to Over-smooth

Natural image content spans a broad range of spatial frequencies, and it is typically easy to constrain the restoration process to reliably recover information in the low frequency bands. These typically include smoothly varying regions without large gradients (textures and edges). In fact, a Gaussian or Laplacian prior would suit well for most image restoration tasks. This family of image priors have been shown to work in a variety of settings, in [38, 79, 80, 21, 115, 140], to name a few. More advanced prior models have also been developed such as FRAME [152], the Fields of Experts model and its extensions [102]. It is known that the filters learned in these higher order models are essentially tuned low high-pass filters [134]. As a result, no matter how these priors are formulated, they work under the same principle by penalizing high frequency image content, imposing the constraint that “images should be smooth” unless required by the image reconstruction constraint. When these priors are universally applied to every pixel location in the image, it is bound to yield over-smoothed output. For image regions where strong signals are present in the (corrupted) observed image (strong edges, salient structures and boundaries), we can expect the image reconstruction constraints to push the solution away from a flat image. But for regions where meaningful signals are almost completely lost due to blur, downsampling and noise, there is nothing for the reconstruction term to latch onto, and the priors would then favor smooth solutions. But smoothness is just another form of blur, which is exactly what we are trying to restore in the first place (such as SR).

6.1.2 Complex Priors Cannot Insert Textures

Another problem with current image restoration frameworks is even the most complex and expressive models cannot insert image details and textures. Image priors have evolved from basic two-tap filter responses to statistics on image patches. One of the most complex (hence expressive) model is the GMM patch model from Zoran and Weiss [155], which works extremely well for deconvolution and denoising. However, as shown in 4, the GMM model does not benefit from having better exemplars, nor does it perform well for restoring high frequency image details. In single image SR, one of the most recent leading methods is based on a deep convolutional network [33], which consists of millions of learned parameters. While it outperforms competing methods slightly in term of PSNR and SSIM scores, the improvement in visual quality is minimal.
As we will show in Sec 6.4, the recent state-of-the-art SRCNN method produces nearly visually identical results compared to ScSR [145], a widely cited sparse coding based approach from 2008. With considerable amount of active effort and progress in the recent years, hallucinating and synthesizing visually plausible image details in restored results remains a challenging problem, even for the most successful SR methods. In order to allow better detail synthesis and insertion, maybe it is helpful to consider a different route, and cast the restoration problem as a constrained detail synthesis problem.

### 6.1.3 Texture Related Techniques from Graphics

Outside of the realm of image restoration, works in the graphics community have examined the task of texture synthesis and manipulation with great success. By taking on a non-parametric approach, simple copy-paste inspired techniques are able to produce visually pleasant image details in various application settings [35, 8, 9, 30]. But why do results from these systems look so much more photo-realistic than typical output images in the image restoration literature? We believe there are three fundamental reasons. First, the constraints are different. Almost all image restoration methods require a well-defined image formation model (e.g. linear operator with additive Gaussian noise), which is typically regularized by a carefully chosen prior term. But for texture synthesis, there is little constraint on the global appearance of the output image. In image editing and retargetting, there can often be highly specific constraints involving salient structures, straight lines, and mask, but these are typically local constraints, and user-defined. Second, the starting point is different. Image restoration problems usually start from globally corrupted signals, namely, every local patch in the input image requires restoration. However, texture transfer/synthesis work always start with clean and sharp images free of artifacts, hence the algorithm has a better source image to manipulate, and high quality outputs can be expected. Finally, the end goals are different. The quality of restored images have been traditionally measured against a ground truth image by PSNR and SSIM scores, which penalizes any signal that deviates from the ground truth, even if it represents plausible and appropriate image details. However, there are many situations where there exists no single best output image, and any plausible restoration or hallucination of image details should be favored, instead of being penalized due to having high frequency content.

We believe that in order to further advance the field of image restoration research, it is beneficial to consider a different direction, where texture transfer is used as the driving force in the restoration process.
6.1.4 Deep Learning Related Approaches

With recent advancements in deep learning and Convolutional Neural Networks (CNN), many aspects in image restoration and texture synthesis have started to witness revisits and re-invention. Deep learning architecture have been adapted to develop new approaches to deblurring [141, 116] and SR [33, 129], numerous on texture hallucination and transfer emerged [90, 42, 43, 82, 15], while principled deep learning approaches begin to drive image synthesis applications as well [46, 32]. The methods from [141, 116, 33, 129] typically involve a supervised training step where image patch pairs are used to learn the network parameters, and the objective function is typically PSNR. As discussed before, this general direction cannot be expected to recover or hallucinate the highest frequency bands in natural images. We have also done experiments where the SRCNN training procedure is carried over a dataset of grass images, and did not outperform the standard SRCNN model over grass test images either qualitatively or quantitatively. On the other hand, the generative adversarial network (GAN) line of work [46, 32] is interesting. It works by training a pair of networks: a generative network G that is responsible for generating samples, and a discriminative network D that detects whether the generated sample is real or fake. Over time, the accuracy of both G and D increase, hence allowing better generative quality of the whole system. Impressive image synthesis has been shown in [32], but the system takes an extremely long time to train, and the generated images are typically quite small due to computational cost. As a result, we do not consider this line of work for our experiments. Instead, we explore adaptations from the works of Gatys et al. [42, 43] and introduce our formulation in the following sections.

6.2 Baseline Methods

For comparison, we first describe several baseline methods from recent literature on super-resolution and texture transfer, and introduce our adaptations in the next section. These baseline methods are representative of state-of-the-art performance in their respective tasks, and form the basis of comparison for Section 6.4.

ScSR by Yang et al. [145, 146] is one of the most widely used methods for comparison in recent SR literature. It is a sparse coding based approach, using a dictionary of 1024 atoms learned over a training set of 91 natural images. Sparse coding is a well studied framework for image reconstruction and restoration, in which the output signal is assumed to be a sparse linear activation of atoms from a learned dictionary. We use the Matlab implementation provided by the authors \(^1\) as a baseline method for comparison.

\(^1\)We use the Matlab ScSR code package from http://www.ifp.illinois.edu/~jyang29/codes/ScSR.rar
Figure 6.1: A sample comparison of various algorithms applied to upsampling texture images for a factor of \( \times 3 \). Two examples per test image are provided for example-based approaches. It can be seen that the example image has significant impact on the appearance of the hallucinated details in the output images, indicating effectiveness of the transfer.
**SRCNN** [33] is a CNN based method for single image super-resolution that is shown to produce state-of-the-art performance on natural images among recent methods. It combines insights from sparse coding approaches and findings in deep learning. A 3-layer CNN architecture is proposed as an end-to-end system. We can view this representation as a giant non-linear regression system in neural space, mapping LR to HR image patches. For subsequent comparisons, we use the version of SRCNN learned from 5 million of 33 × 33 subimages randomly sampled from ImageNet. The Matlab code package can be found on the author’s website².

Gatys et al. [42, 43] first consider reformulating the texture synthesis problem within a CNN framework. In both work, the VGG network is used for feature representation and modeling image space, and the correlation of feature maps at each layer is the key component in encoding textures and structures across spatial frequencies. In [42], a linear combination of content loss and style loss is minimized to produce an output image that is induced by the textures of the style image while maintaining visual and semantic similarity to the content image. The Gram matrix representation is compact and extremely effective at synthesizing a wide variety of textures [43]. The style transfer framework in [42] is mainly used to transfer artistic style to render an artistic impression of an image, however, it is also possible to apply it to the task of constrained texture synthesis and super-resolution. We use a Lasagne and Theano based implementation of [42] as a baseline method for comparison³. We refer to this baseline as **Gatys**.

Following the success of [42, 43], Li and Wand propose **CNNMRF** [82] to address the loss of spatial information due to the Gram matrix representation. A MRF style layer is added on top of the VGG hidden layers to constrain local similarity of neural patches, where each local window in the output image feature map is constrained to be similar to the nearest neighbor in the corresponding layer of the style image feature maps. We use the torch based implementation from the authors⁴.

To adapt the code from Gatys et al. and CNNMRF for our experiments, we upsample the LR input image bicubicly to serve as the content image. All other processing remain identical to their respective implementation.

We show a sample comparison of these methods in Figure 6.1, where a low resolution texture image is upsampled by a factor of 3. For the example based methods [42, 82] and ours, we provide two example images to test the algorithm’s ability in transferring textures. Some initial observations can be made:

- ScSR [145] and SRCNN [33] produce nearly identical results qualitatively, even though their model

---

²We use the SRCNN code package from [http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html](http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html)

³Our implementation is adapted from the art style transfer recipe from Lasagne: [https://github.com/chuanli11/CNNMRF](https://github.com/chuanli11/CNNMRF)

⁴Chuan Li’s CNNMRF implementation is available at: [https://github.com/chuanli11/CNNMRF](https://github.com/chuanli11/CNNMRF)
complexity is orders of magnitude apart. This represents half a decade of progress in the single image SR literature.

- CNNMRF [82] produces painterly artifacts due to averaging in neural space. The highest frequencies among different color channels can be misaligned and appear as colorful halos when zoomed in.

- Our method produces convincing high frequency details while being faithful to the LR input. The effect of the example image can be clearly seen in the output image.

6.3 Method

Our method is based on the works of Gatys et al. [42, 43], which encodes feature correlations of an image in the VGG network via the Gram matrix. The VGG-Network is a 19-layer CNN that rivals human performance for the task of object recognition. This network consists of 16 convolutional layers, 5 pooling layers, and a series of fully connected layers for softmax classification. To ease the task of learning the large number of free parameters, each filter is constrained to be $3 \times 3$ in size. Due to the pooling operations that reduce feature map sizes as layer positions increase, these small filters are able to consider increasingly larger neighborhoods in the image without having to deal with higher dimensions that natural arise when actually using larger image patches.

A latent image $x$ is to be estimated given constraints, such as content similarity, and style similarity. We assume a style or example image $s$ is available for the transfer of appropriate textures from $s$ to $x$, and that $x$ should stay similar to a content image $c$ in terms of mid to high level image content. The CNN convolved feature space representations are $X$, $S$ and $C$ respectively. At each layer $l$, a non-linear filter bank of $N_l$ filters is convolved with the previous layer’s feature map to produce an encoding in the current layer, which can be stored in a feature matrix $X^l \in \mathbb{R}^{N_l \times M_l}$, where $M_l$ is the number of elements in the feature map (height times width). We use $X^l_{ij}$ to denote the activation of the $i^{th}$ filter at position $j$ in layer $l$ generated by image $x$.

In the work of Gatys’s et al. [42], the goal is to solve for an image $x$ that is similar to a content image $c$ but takes on the style or textures of $s$. Specifically, the following objective function minimized via gradient descent to solve for $x$:

$$x = \arg\min_x (\alpha E_{\text{content}}(c, x) + \beta E_{\text{style}}(s, x))$$  \hspace{1cm} (6.1)
where $E_{\text{content}}$ is defined as:

$$E_{\text{content}}(c, x) = \frac{1}{2} \sum_l \sum_{ij} (C_l^{ij} - X_l^{ij})^2 \quad (6.2)$$

The content similarity term is simply a $L_2$ loss given the difference between the feature map of the latent image in layer $l$ and the corresponding feature map from the content image.

The definition of $E_{\text{style}}$ is based on the the $L_2$ loss between the Gram matrix of the latent image and the style image in a set of chosen layers. The Gram matrix encodes the correlations between the filter responses via the inner product of vectorized feature maps. Given a feature map $X^l$ for image $x$ in layer $l$, the Gram matrix $G(X^l) \in \mathbb{R}^{N_l \times M_l}$ has entries $G_{ij}^l = \sum_k X_{ik}^l X_{jk}^l$, where $i, j$ index through pairs of feature maps, and $k$ indexes through positions in each vectorized feature map. Then the style similarity component of the objective function is defined as:

$$E_{\text{style}}(s, x) = \sum_l w_l \frac{w_l}{4N_l^2M_l^2} \left( \sum_{i,j} (G(S^l)^{ij} - G(X^l)^{ij})^2 \right) \quad (6.3)$$

where $w_l$ is a relative weight given to a particular layer $l$. The derivatives of the above energy terms can be found in [42]. To achieve best effect, the energy components are typically enforced over a set of layers in the network. For example, the content layer can be a single conv4_2 layer, while the style layers can be over a larger set \{conv1_1, conv2_1, conv3_1, conv4_1, conv5_1\} to allow consistent texture appearances across all spatial frequencies.

In [42], this setup is applied to style transfer where $s$ is typically a painting and $c$ is a photograph, so that we obtain an image that appears to be painted by the same artist in the same style as $s$. However, when applied to pairs of photographs of natural scenes, the output would still retain a painterly look as if it was an artistic rendering. However, Gatys et al. also use the same Gram matrix constraints for texture synthesis in [43], where a wide range of natural stochastic textures are tested and it is shown to work extremely well to synthesize similar textures almost indistinguishable from the source, except for regular textures such as brick walls and other structured (non-stochastic) contents. Clearly, the Gram matrix encodes valuable information about image details, and this representation is capable of inducing natural textures. We introduce a few adaptations of this method to the task of single image super-resolution, and specifically examine its effectiveness in terms of transferring and synthesizing natural textures.
6.3.1 Basic Adaptation to SR

The objective function in Equation 6.1 consists of a content similarity term and a style term. The content term is analogous to the faithfulness term in single image SR frameworks. The style term can be seen as a high specific natural image prior derived from a single example image, which is assumed to represent the desired image statistics. A first step in our experiments is to replace the content similarity term $E_{content}$ with a faithfulness term $E_{faithfulness} = |G * x \downarrow f - c|^2$, where $f$ is the downsampling factor, $G$ is a Gaussian lowpass filter, and $c$ is low resolution input image that we would like to upsample. These variables associated with the downsampling process are assumed known a-priori (non-blind SR). In the subsequent discussion, we refer to this basic adaptation as our global, since the Gram matrix constraint is globally applied to the whole image. Formally, the our global method solves the following objective via gradient descent:

$$x = \arg \min_x \alpha E_{faithfulness}(c, x) + \beta E_{style}(s, x)$$

(6.4)

We further make the following changes to the original setup:

- All processing is done in gray scale. The original work of Gatys et al. [42] computes the feature maps using RGB images. However, this requires strong similarity among color channel correlations between the example and input image, which is hard to achieve. For transferring artistic styles [42], this is not a problem. We drop the color information to allow better sharing of image statistics between the image pair.

- We use the layers $\{\text{conv1}_1, \text{pool1}_1, \text{pool2}_1, \text{pool3}_1, \text{pool4}_1, \text{pool5}_1\}$ to capture the statistics of the example image for better visual quality, as done in [43].

We show that the above setup, while simple and basic, is capable of transferring texture details reliably for a wide variety of textures, even if the textures are structured and regular. However, for general natural scenes, this adaptation falls short and produces painterly artifacts or inappropriate image details for smooth image regions, because their global image statistics no longer matches each other.

6.3.2 Local Texture Transfer via Masked Gram Matrices

Natural images are complex in nature, usually consisting of a large number of segments and parts, some of which might contain homogeneous and stochastic textures. Clearly, globally applying the Gram matrix constraint for such complex scenes cannot be expected to yield good results, due to mismatched global
statistics. However, with carefully chosen local correspondences, we can selectively transfer image details by pairing image parts of the same or similar textures via two sets of binary masks \( \{ m^k_s \}_1^K \) and \( \{ m^k_x \}_1^K \). To achieve this, we introduce an outer summation to the \( E_{\text{style}} \) term to loop over each corresponding pair of components in the masks.

\[
E_{\text{style local}} = \sum_k E_{\text{style}}(s \otimes m^k_s, x \otimes m^k_x) = \sum_k \sum_l \frac{1}{4N_l^2 |R_{lx}(m^k_x)|^2} \left( \sum_{i,j} \left( G(S_l \otimes R_l(s(m^k_s)))_{ij} - G(X_l \otimes R_l(x(m^k_x)))_{ij} \right)^2 \right)
\]  

(6.5)

where \( R_{lx} \) is an image resizing operator that resamples an image (a binary mask in this case) to the resolution of feature map \( x^l \) using nearest neighbor interpolation. The normalization constant also reflects that we are aggregating image statistics over a subset of pixels in the images. Finally, the parameter \( \beta \) from Eq 6.1 needs to be divided by the number of masks \( K \) to ensure the same relative weight between \( E_{\text{faithfulness}} \) and \( E_{\text{style local}} \). Note that these binary masks are not necessarily exclusive, namely, pixels can be covered or explained by multiple masks if need be.

However, the sparse correspondences are non-trivial to obtain. We examine two cases for the correspondence via masks: manual masks, and automatic masks via the PatchMatch [8] algorithm.

**Manual Masks** For moderately simple scenes with large areas of homogeneous textures such as grass, trees, sky, etc., we manually generate 2 to 3 masks per image at the full resolution to test out the local texture transfer. We refer to this setup as our local manual. A visualization of the images and masks can be found in Figure 6.2.

**PatchMatch Masks** We also test to see if the correspondence part of the algorithm can be automated, for example, through the well-known PatchMatch algorithm introduced by Barnes et al. [8]. The PatchMatch algorithm is applied to the low-resolution input image \( c \) and a low-resolution version of the style image \( s \) after applying the same downsampling process used to generate \( c \). Both images are assumed to be grayscale. Once the nearest-neighbor field (NNF) is computed at the lower resolution, we divide the output image into cells and pool and dilate the interpolated offsets at the full resolution to form the mask pairs. Each \( m^k_x \) contains a square cell of 1’s, and its corresponding mask \( m^k_s \) will be the union of numerous of binary patches. We refer to this variation as our local. A sample visualization is given in Figure 6.3.
Figure 6.2: Sample images and their corresponding masks, each one is manually generated.

Figure 6.3: Visualization of the masks automatically generated using the PatchMatch algorithm. PatchMatch is applied to the low resolution grayscale input and example images to compute a dense correspondence. The HR output image is divided into cells, and all correspondences contained in the input cell are aggregated to form the example image mask.
6.4 Experimental Results

In this section we showcase the performance of the algorithm variants our global, our local (PatchMatch based) and our local manual on a variety of textures and natural images. We also compare against leading methods in single-image super-resolution such as ScSR [145] and SRCNN [33], as well as deep learning based style transfer methods including Gatys et al. [42] and CNNMRF [82] by Li and Wand.

6.4.1 Test Data

We collect a variety of images from the Internet including natural and man-made textures, regular textures, black and white patterns, text images, simple natural scenes consisting of 2 or 3 clearly distinguishable segments, and face images. These test images are collected specifically to test the texture transfer aspect of the algorithms. As a result, we do not attempt to evaluate performance of single image super-resolution in its traditional sense.

6.4.2 Black and White Patterns

The simplest test images are texts and black and white patterns. As shown in Figure 6.4, traditional SR algorithms do a decent job at sharpening strong edges, with SRCNN producing slightly less ringing artifacts than ScSR. As expected, the example based methods produce interesting hallucinated patterns based on the example image. CNNMRF yields considerable amount of artifacts due to averaging patches in neural space. Gatys and our global introduce a bias in background intensity but are capable of keeping the edges crisp and sharp. Much fine details and patterns are hallucinated for the bottom example.

6.4.3 Textures

For homogeneous textures, SR methods simply cannot insert meaningful high frequency content besides edges because they are not designed to handle textures. Most of these methods essentially boil down to learning smart ways to blend/average patches linearly or non-linearly in intensity space, which inevitably produces blur and painterly appearances. On the other hand, we see that the Gram matrix constraint of Gatys et al. [42, 43] works extremely well because it is coercing image statistics across spatial frequencies in neural space, and ensuring that the output image match these statistics. However it is less effective when it comes to non-homogeneous image content such as edges and salient structures, or any type image phenomena that is spatially unexchangeable. Finally, the CNNMRF method of Li and Wand [82] works reasonably well but
Figure 6.4: Example comparisons on a Chinese text image (top) and black and white pattern image (bottom). Example based methods can hallucinate edges in interesting ways, but also produce biases in background intensity, copied from the example image. Other artifacts are also present. Best viewed electronically and zoomed in.

Figure 6.5: Example comparisons on regular textures. Best viewed electronically and zoomed in.
still falls short in terms of being realistic and inserting of high frequency details. This is due to the linear blending of neural patches in the optimization process, some amount of high frequencies are inevitably lost. Another artifact of this method is that this blending process can produce neural patches from the null space of natural image patches, introducing colored halos and tiny rainbows when zoomed in.

The main benefits of the our global method adapted from Gatys et al. [42] is (1) better faithfulness to the input LR image, and (2) less color artifacts. The Gatys transfer baseline operates in RGB color space, hence any correlated color patterns from the style image will remain in the output image. However, the style image might not represent the correct color correlation observed in the input image, e.g., blue vs yellow flowers against a background of green grass. Our global transfer method operates in gray scale, relaxing the correlation among color channels and allowing better sharing of image statistics. This relaxation helps bring out a more realistic output image, as shown in Figure 6.5, 6.6, 6.7.

Comparisons on regular textures are shown in Figure 6.5. our global produces better details and color faithfulness, whereas traditional SR methods do not appear too different from bicubic interpolation. Figure 6.6 shows results on numerous stochastic homogeneous textures. Example based methods can be quite
biased by the example image and produce an image quite different from the input, such as the fur image (third row). However, better details can be consistently observed throughout the examples. **Gatys** can be seen to produce a typical flat appearance in color (e.g., rock, first row), this is because of the color processing constraint.

Going beyond homogeneous textures, we test these algorithms on simple natural images in Figure 6.7. Realistic textures and details can be reasonably well hallucinated by **our global**, especially the roots in the roil (first row) and the patterns on the butterfly wings (bottom row). The pipes (second row) are synthesized well locally, however, the output image when viewed globally becomes too ‘busy’. It is worth pointing out that CNNMRF essentially produces a painting for the forest image (third row), this is a clear example of the disadvantages of averaging/blending patches. It could be argued that such averaging should be avoided both in neural domain and intensity domain to ensure better detail synthesis.

### 6.4.4 Natural Scenes

Natural images exhibit much more complexity than homogeneous textures, here we only consider scenarios where the image can be clearly divided into several types of textures, mostly homogeneous. In this way, we can better test the effectiveness of the algorithm’s performance on synthesizing and hallucinating texture details. One complication that arises here is that texture transitions and borders represent extremely
Figure 6.8: Example comparisons on moderately complex natural images. CNNMRF, Gatys and ‘our local’ consistently synthesize more high frequencies appropriate to the scene. CNNMRF and Gatys suffer from color artifacts due to mismatching colors between the example and the input image. CNNMRF also produces significant amount of color artifacts when viewed more closely, especially in smooth regions and near image borders. Gram matrix based methods such as Gatys and ‘our local’ outperform other methods in terms of hallucinating image details, however also produce more artifacts in a few test cases. Best viewed electronically and zoomed in.
Figure 6.9: Example comparisons on natural scenes with manually supplied masks. Best viewed electronically and zoomed in.
non-homogeneous statistics that is not easily handled by synthesis methods. Since the image now contains different types of statistics, we will apply our masked variants using PatchMatch masks and manual masks to these test images. To better deal with texture transitions, we dilate the manually generated masks slightly to include pixels near texture borders.

In Figure 6.8, all results under our local are generated using our PatchMatch based variant. These test images consist of moderately complex natural scenes. It can be seen that CNNMRF, Gatys and our local consistently synthesize more high frequencies appropriate to the scene, traditional SR methods appear similar to bicubic interpolation. CNNMRF and Gatys suffer from color artifacts due to mismatching colors between the example and the input image. Again, CNNMRF produces significant amount of color artifacts when viewed more closely, especially in smooth regions and near image borders. Gram matrix based methods such as Gatys and our local outperform other methods in terms of hallucinating image details, however also produce more artifacts in a few test cases.

PatchMatch is far from perfect for generating the masks suitable for our application. This can be seen in many regions in the output images. For example, the trees in the pond image (second last row) is hallucinated by water textures towards the left, even the tree on the far left shows much water-like textures, clearly due to bad correspondences generated by PatchMatch. Similar artifacts can be seen in the crater lake image (last row). For natural scenes, our method is capable of opportunistically inserting appropriate textures, but cannot produce a perfect flaw-free output.

One would expect manually generated masks to be more suitable than PatchMatch masks. Although there are two drawbacks:

- The entire masked example region would participate in the Gram matrix computation, forcing the output image to take on the exemplar statistics, even though it might be undesirable. For example, when match sky with slow intensity gradient with a flat sky region. PatchMatch offers more freedom in this regard, allowing certain regions to be completely discarded (in the example image).

- Texture transitions are hard to account for. Even though we dilate the masks hoping to include the borders, the pyramid nature of the CNN architecture and pooling operations will eventually introduce boundary conditions.

Figure 6.9 show comparisons using our manually generally masks (c.r. 6.2). Clearly, there is less low frequency artifacts in color biases. However, ringing artifacts become more prominent near texture transitions and image borders.
6.4.5 Face Images

Another interesting scenario is to test the algorithms on face images. When the example image is sufficiently close the input, such as in Figure 6.10, our method works well for hallucinating image details. In this particular example, the facial features in the output image remain similar to the input, and it is almost impossible to tell who the example image is given just the output. However, CNNMRF lacks the ability to synthesize new content (copy-paste in neural space) and its output is more of a blend between the input and example. The final output image somewhat falls into the ‘uncanny valley’, and is almost unrecognizable as De Niro.

However, the CNNMRF method works extremely well on faces. In Figure 6.11, CNNMRF is able to produce a natural looking output with decent high frequency details except for the mouth region, since the example image does not contain the best source patches. On the other hand, our Gram matrix based method (our global setup) fails completely for the face region, only synthesizing details on parts of the hat, which happens to be homogeneous textures. This is because human faces are highly structured and far from textures.

6.5 Conclusions

In this final chapter, we have examined recent advances in deep learning based texture transfer and proposed a CNN based detail synthesis algorithm that can perform competitively at the task of single image super-resolution and texture transfer. We compare our method variants against a wide range of modern SR algorithms and texture transfer methods, and show that it is capable of hallucinating image details far beyond traditional SR methods on texture images and simple natural scenes. Localized synthesis is achieved through a sparse correspondence between the example and input image. While our method is still prone to producing artifacts, it offers a promising direction for future research in texture transfer and image hallucination.
Figure 6.10: Example comparisons on a portrait image. Our method is able to hallucinate appropriate details given the well-matched image statistics. Most noticeably, plausible details are successfully introduced to the eyebrows, hair, and eyes. CNNMRF produces decent amount of details as well, however, it makes the output image less recognizable as the person in the input image. Best viewed electronically.
Figure 6.11: Example comparisons on a face image. Our method fails due to mismatch in global image statistics. It is interesting to note that CNNMRF works extremely well for face images, however, it cannot insert image details not present in the example image. In this case, it cannot synthesize a closed mouth of the baby. Best viewed electronically.
Chapter 7

Conclusions and Recommendations

Which of my photographs is my favorite? The one I’m going to take tomorrow.

Imogen Cunningham

In this dissertation, we have investigated two main problems in image restoration: single image deblurring and single image super-resolution. Our approaches to both problems are built on data-driven strategies. We learn natural image statistics from large image datasets and Internet scale imagery. We examine ways to augment and push forward the capability of modern image priors, and explore novel image representations that could allow better synthesis of image details. In both problems, our approaches have proved effective and excel at recovering and inserting image details. However, our methods are also dependent on the existence of large datasets and similar example images. In terms of image quality, our approaches can sometimes render a less faithful result compared to leading methods, which are typically conservative by design. Hence, the various efforts in this dissertation work complements traditional approaches and push the envelope of current research in the field. To summarize and draw conclusions, we would like to attribute the effectiveness of our approaches to the following factors:

- Big Data. With Internet-scale imagery and state-of-the-art scene matching algorithms, we can craft a highly content aware image prior that is specific to the input image, disregarding 99.9% of the space
of natural images. As a result, the constraints drawn from similar scenes/examples can provide an extremely strong yet appropriate influence to the latent image. In a sense, we can suddenly cheat by coming up with a content aware image prior that is not too far from the correct posterior distribution. Our experiments also show that such external image statistics can be extremely useful when extracted from only scene matches.

- **Localized restoration.** Throughout our experiments, a key strategy to success is to constrain the patch synthesis process as locally as possible, from millions of images down to a few and finally down to segment/crop level correspondences. Such correspondences are be reliably obtained using the existing low frequency information present in the input image, hence provide appropriate extra constraints to work with during optimization. This allows different image content (sky v.s. grass) to be treated differently as they should.

- **Application specific image priors.** The way the image priors are devised and formulated play a central role in an algorithm. We carefully designed our image priors to target specific applications and even substeps in optimization processes. For example, when the application needs to move a latent image from a smooth state to a sharp state, the image prior must favor sharp image content over blurred ones, and vice versa. Universal image priors such as sparsity works both ways and are suitable for many tasks but struggle to outperform application specific image priors.

- **Expressive image representation.** It is no secret to the computer vision community that the feature space in which an algorithm operates plays a vital role in its performance. Many image restoration algorithms succeed by operating in gradient domain due to known image properties of image gradients. Successful representations can also be obtained by implicitly projecting image signals onto learned bases and further partitioning the feature space. Recent advancements in neural networks also offer a natural and extremely expressive feature space via Convolutional Neural Networks. It is interesting to that a network trained for object recognition such as the VGG network can also work extremely well for completely different tasks such as texture transfer and synthesis.

Modern methods in deblurring and SR have come a long way, offering continued new insights and understanding into such long standing problems. However, the performance in various applications such as SR, deblurring and denoising are certainly approaching an upper bound in terms of what can be unambiguously recovered. And the next generation of image restoration algorithm should emphasize hallucinating and synthesizing image details beyond such recoverable limit. To this end, we make the following recommendation
to facilitate exciting future research directions.

• Relaxation of the data term. Almost all data term for SR, deblurring and denoising are based on minimizing the $L_2$ norm of the noise residual. However, even under a Gaussian noise model, this is only appropriate for the MAP solution. One direction would be to consider minimizing the KL divergence between the distribution of the residual layer and the (known) noise distribution. Another important consideration would be to learn distance metrics that are closer to measuring perception based image quality, instead of fighting for imperceptible PSNR gains.

• Explore deeper architectures. Modern hardware excel at deep CNN architectures and the overhead of going deeper is getting lower and lower. Such deep representations have been the standard for object recognition and many other tasks. Many current CNN-based image restoration methods have yet to caught up in terms of deepness, hence, going deeper could be a direction to explore as a low hanging fruit for performance gain.

• Moving state-of-the-art statistical priors into CNN space. [82, 15] are good examples moving known patch-based techniques from image intensity space to CNN feature space. Similar efforts can prove effective for leading image priors. For example, we can learn generative GMM models for neural patches, learn smooth manifolds that characterize the Gram matrix to interpolate and manipulate image textures.

• Human interaction. The end goal of single image restoration methods is to fix photographs so that they can be used. However, the quality of restored outputs are typically far below what is considered usable, and often times, blurry or noisy photos are discarded right away. Considering human interaction in the restoration process can bring several benefits. First, when example images are present, human annotations can help provide more accurate correspondences hence better content aware priors and localized restoration quality. Second, user can interact with the algorithm to convey what needs to be done, such as disambiguating defocus blur v.s. motion blur, realism of texture details (synthesized) v.s. level of faithfulness. Developing such interactive systems and understanding user needs will result in more practical algorithms and attract more attention to the field.
Bibliography


[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierar-

[4] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierar-

[5] Sean Arietta, Jason Lawrence, and Jason Lawrence. Building and using a database of one trillion


[8] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A random-

[9] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized Patch-

patch queries for large datasets and applications. *ACM Trans. Graph.*, 2015.


2016.


[19] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 3-sweep: extracting editable


[73] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. ACM Trans. Graph., 2014.


[152] Song Chun Zhu, Ying Nian Wu, and David Mumford. Filters, random fields and maximum entropy
(frame): Towards a unified theory for texture modeling. In *International booktitle of Computer Vision*,
1998.

2011.

[154] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restora-

[155] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restora-


[157] Wangmeng Zuo, Lei Zhang, Chunwei Song, and David Zhang. Texture enhanced image denoising via