



2009-03-11

Super-Resolution via Image Recapture and Bayesian Effect Modeling

Neil B. Toronto

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Computer Sciences Commons](#)

BYU ScholarsArchive Citation

Toronto, Neil B., "Super-Resolution via Image Recapture and Bayesian Effect Modeling" (2009). *All Theses and Dissertations*. 1839.
<https://scholarsarchive.byu.edu/etd/1839>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

SUPER-RESOLUTION VIA IMAGE RECAPTURE AND BAYESIAN
EFFECT MODELING

by
Neil Toronto

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science
Brigham Young University
April 2009

Copyright © 2009 Neil Toronto

All Rights Reserved

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Neil Toronto

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date

Dan Ventura, Chair

Date

Bryan S. Morse

Date

Kevin Seppi

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Neil Toronto in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Dan Ventura
Chair, Graduate Committee

Accepted for the
Department

Kent E. Seamons
Graduate Coordinator

Accepted for the
College

Thomas W. Sederberg
Associate Dean, College of Physical and Mathematical
Sciences

ABSTRACT

SUPER-RESOLUTION VIA IMAGE RECAPTURE AND BAYESIAN EFFECT MODELING

Neil Toronto

Department of Computer Science

Master of Science

The goal of super-resolution is to increase not only the size of an image, but also its apparent resolution, making the result more plausible to human viewers. Many super-resolution methods do well at modest magnification factors, but even the best suffer from boundary and gradient artifacts at high magnification factors. This thesis presents Bayesian edge inference (BEI), a novel method grounded in Bayesian inference that does not suffer from these artifacts and remains competitive in published objective quality measures. BEI works by modeling the image capture process explicitly, including any downsampling, and modeling a fictional *recapture* process, which together allow principled control over blur. Scene modeling requires noncausal modeling within a causal framework, and an intuitive technique for that is given. Finally, BEI with trivial changes is shown to perform well on two tasks outside of its original domain—CCD demosaicing and inpainting—suggesting that the model generalizes well.

Table of Contents

1	Introduction to Super-Resolution	1
1.1	Nonadaptive Methods	1
1.1.1	Function Fitting Methods	1
1.1.2	Frequency Domain Methods	2
1.1.3	Qualitative Analysis	2
1.2	Adaptive Methods	3
1.2.1	Edge-Preserving	3
1.2.2	Training-Based	4
1.2.3	Optimization	5
1.2.4	Qualitative Analysis	6
1.3	Bayesian Edge Inference	6
2	Graphical Models	9
2.1	Introduction	9
2.1.1	Notation and Terminology	9
2.2	Bayesian Networks	10
2.2.1	Formulation	11
2.2.2	Inference	13
2.3	Markov Random Fields	14
2.3.1	Formulation	15
2.3.2	Inference	16
2.4	Factor Graphs	17

3	Super-Resolution via Recapture and Bayesian Effect Modeling	19
3.1	Introduction	19
3.2	Reconstruction by Recapture	22
3.3	Effect Modeling in Bayesian Networks	23
3.4	Super-Resolution Model	25
3.4.1	Facets	26
3.4.2	Scene Model	28
3.4.3	Capture and Recapture	30
3.4.4	Minimum Blur	31
3.4.5	Decimation Blur	31
3.4.6	Inference	32
3.5	Results	33
3.6	Other Applications	34
3.7	Limitations and Future Work	36
4	Conclusion	39

List of Figures

2.1	Example Bayesian networks	11
2.2	Example Markov random fields	15
2.3	Factor graphs vs. MRFs	17
3.1	“Peppers” compared at 2x and 4x	20
3.2	Super-resolution framework	21
3.3	The recapture framework	23
3.4	Bayesian effect modeling	25
3.5	Spatially varying point-spread function	27
3.6	Step edges	28
3.7	Utility of compatibility	29
3.8	Compatibility keeps boundary coherence	34
3.9	Smooth gradients and sharp edges	36
3.10	CCD demosaicing	37
3.11	Inpainting	37

List of Tables

3.1 Objective measures 35

Chapter 1

Introduction to Super-Resolution

In many image processing tasks, it is essential to get good estimates of between-pixel values. Photographers and other content creators use interpolation both in hardware and software to rescale images and parts of images. Good interpolation is vital in rotating and warping (visual effects or lens distortion correction, for example), where source locations do not correspond one-to-one with pixel locations. It is also used in presenting low-resolution signals on high-resolution displays, such as in up-converting a DVD signal to HDTV, or in preparing an image for printing. This research is primarily concerned with super-resolution: resizing a digital image to larger than its original size.

Interpolation methods may be split into two broad categories: nonadaptive and adaptive.

1.1 Nonadaptive Methods

Nonadaptive methods are those that make no assumptions about how an image is created. They are the most well-known and widely implemented. There are two significant families of nonadaptive methods: function fitting methods and frequency domain methods.

1.1.1 Function Fitting Methods

Function fitting methods [1] fit basis functions to image samples either exactly, or inexactly by goodness-of-fit criteria. They come in linear, quadratic, and many different cubic spline varieties. Because image samples usually have a rectilinear grid structure, they are usually

implemented as convolution with single, separable kernels. Image interpolation with these methods is often followed by some kind of sharpening to reduce blurry artifacts.

1.1.2 Frequency Domain Methods

Frequency domain methods [2, 3] differ primarily in how they are motivated: an attempt to approximate perfect reconstruction of a bandlimited signal. They are also implemented as convolution. Because the ideal filter in the spatial domain (a sinc function) is a convolution kernel of infinite width, finite approximations must be made, and much of the literature for these methods deals with finding a filter that minimizes artifacts while approximating the ideal filter well.

1.1.3 Qualitative Analysis

A comprehensive chronology of interpolation is given in [4], from as early as 300 B.C. through 2001, but with special emphasis on the development of the two aforementioned families.

The best of each of these families tend to produce results of comparable quality, though some results indicate that B-spline interpolation, a function fitting method, is superior under certain reasonable criteria [3, 5]. For many real-world tasks such as image rotation, one can almost always find a method that gives acceptable trade-offs between aliasing and blurring while maintaining high accuracy. Room for improvement in this well-studied area is limited.

However, super-resolution requires more than just perfect reconstruction to look subjectively correct. Because upscaling widens the apparent point spread function, even a resampled “perfectly reconstructed” natural image would appear to have blurry-looking artifacts: fuzzy edges and too-smooth textures. Although natural images are bandlimited due to the point spread of a capturing device, natural *scenes* are not.

1.2 Adaptive Methods

Adaptive interpolation methods are those that use local context to preserve natural image characteristics. They make strong assumptions about how images are produced and captured in an attempt to overcome the subjective incorrectness and finite limitations of perfect reconstruction. These fall into three significant families: edge-preserving, training-based and optimization methods.

1.2.1 Edge-Preserving

Many adaptive methods regard natural images primarily as projections of solid objects with clear discontinuities, and thus attempt to preserve step edges in the upscaled result. One of the earliest, subpixel edge localization, fits a rigid sigmoidal surface to overlapping 3×3 windows and averages surface values in the output image whenever a goodness-of-fit threshold is met, falling back on bilinear interpolation [6, 7]. The authors of directional interpolation [8], noting that subpixel edge localization can only handle step edges, fit a planar model to small neighborhoods where a simple gradient and Laplacian test pass, and also fall back on bilinear interpolation.

Edge-directed interpolation [9] seems to have popularized edge-preserving methods. Its basic interpolant is bilinear, but it detects edges using zero-crossings from the output of a Laplacian-like filter and refuses to interpolate over them. It is one of the few edge-preserving methods to incorporate a reconstruction constraint, which iteratively reduces the disparity between the low-resolution image and downsampled high-resolution output. More recently, “new” edge-directed interpolation [10] takes a softer approach: adapting interpolation based on estimated covariances from low-resolution neighborhoods. Edge inference [11] is another soft approach, which models edges with neural network regression over overlapping neighborhoods and combines network outputs with standard nonadaptive interpolants.

1.2.2 Training-Based

Training-based methods take a nonparametric approach, preferring instead to discover natural image characteristics or a transfer function from low-resolution to high-resolution from a training set comprised of low-resolution and high-resolution pairs. These most often address inventing plausible details.

Rather than trying to preserve edges alone, local correlation [12] and resolution synthesis [13] try to discover features that should be preserved from a training set. Local filters or kernels are learned and applied to each neighborhood of the low-resolution input based on a clustering scheme. In [14], neural networks learn a transfer function from low-resolution patches to high-resolution patches; training also incorporates a differentiable measure of perceptual quality.

Other methods focus on inventing, or “hallucinating” plausible high-frequency detail. Freeman’s work [15, 16] is similar in structure to the model presented in this thesis. In it, a training set both represents the distribution of images and serves as primitives for reconstruction. The scene is modeled as a Markov random field: a grid of indexes into the training set, with soft constraints ensuring that nearby primitives are compatible. Low-resolution primitives are constrained to be compatible with low-resolution input neighborhoods. A MAP estimate finds a best match, and high-resolution primitives are selected for the final output. All data is bandpass-filtered and contrast-normalized to focus inference and generalize the training set. The work is extended later to incorporate priors based on the characteristic distribution of natural images’ directional derivatives and to demosaic CCD output [17]. While the details are good, this method often fails to infer plausibly smooth object boundaries.

The image analogies algorithm [18], when used for super-resolution, is very similar to Freeman’s, but eschews probabilistic modeling in favor of straight matching to training set instances based on reasonable measures of local similarity and compatibility with neighbors. Baker motivates a “reconstruction” method [19] by noting that all methods that

incorporate smoothness priors on the output image produce overly smooth results at some magnification factor no matter how many input images are available, and instead chooses a prior calculated from training set data. The training data is also used to hallucinate details for faces and text.

Face hallucination [20] takes a two-step approach to face super-resolution: first, a global linear model learns a transfer function from low-resolution faces to high-resolution faces, then a Markov network locally reconstructs high-frequency content. A fully Bayesian approach to text super-resolution [21] models generation of binary textual glyphs with an explicit noise model and represents likelihood functions nonparametrically with grayscale, antialiased training instances.

Finally, texture synthesis methods [22, 23, 24] (and also the image analogies in [18]) create novel textures from training data. This is related in that super-resolution, at least the problem of inventing plausible details, can be regarded as guided or semi-supervised texture synthesis.

1.2.3 Optimization

Optimization methods assume that an original high-resolution image existed and was degraded by a known process to produce the input image. They are either motivated or formulated explicitly in Bayesian terms. A regularizing prior is chosen for the high-resolution output, a transition function (often called a *reconstruction constraint* [19], *back-projection* [25] or *sensor model* [26]) is chosen to represent degradation, and inference is performed by finding a MAP estimate. This is very similar to many of the training-based methods; the difference is that here, the random variable to be estimated is usually the high-resolution output itself rather than a higher-level representation of it.

Level-set-based reconstruction [27] uses a regularizing prior that promotes correct level-set topology while penalizing jagged artifacts. A sharpening model is added in [28], and in [29] it is extended to automatically select regularization parameters and to handle

other forms of image manipulation besides super-resolution. In data-dependent triangulation [30] the problem is cast as finding the optimal triangulation of the image surface, where the regularization constraint is one that minimizes jagged artifacts. Another recent local geometry approach, inspired by the natural distribution of directional derivatives mentioned above, puts a prior distribution on gradient profiles [31], which is learned from a training set.

1.2.4 Qualitative Analysis

While most papers on adaptive super-resolution methods compare against nonadaptive methods and a few compare against other adaptive methods, the most comprehensive qualitative analysis to date is Ouwerkerk’s recent survey [32], which compares some methods mentioned here. The survey includes a discussion of general techniques and presentation of methods to be tested. A set of test images was downsampled and then restored, and for each method tested, three objective measures were applied to pairs of original and restored images.

While many methods performed well at 2x magnification factors, all showed artifacts by 4x. The best methods had issues with boundary coherency and false edges in gradients at those scales. A primary goal of this research is to surpass those methods in subjective quality with respect to those artifacts, while remaining competitive on objective measures.

1.3 Bayesian Edge Inference

The method presented here, Bayesian edge inference (BEI), has many similarities to all three families of adaptive methods, but also significant differences.

Edge-preserving methods tend to model step edges directly, and so does BEI—but BEI models them as scene primitives rather than as image primitives. This is a subtle but important difference: it allows reasoning about blur due to downsampling as part of a

capture process. Further, to obtain a result, BEI uses image *recapture*, a fictional higher-resolution capture process, rather than simply evaluating a function.

Training-based methods tend to model the scene using primitives and recreate an image from high-resolution versions of the same. The main differences here are that BEI is parametric rather than training-based, and that it formally and explicitly models image recapture. Again, this allows reasoning about blur due to downsampling. BEI also shares the notion of compatibility among scene elements that some training-based methods use.

Optimization methods and BEI are both built on Bayesian inference. Because regularization is conceptually equivalent to compatibility, BEI shares this concept with optimization methods as well. Yet rather than infer a new image, BEI infers a scene and recaptures it. MAP estimates are used almost exclusively in optimization methods, but BEI samples a posterior predictive density.

In Bayesian inference, the better a model represents the true process that produced the data, the better it can infer the causes. Thus, another primary goal of this research is to model image capture and recapture more explicitly than has been previously done. Besides increasing accuracy, explicit modeling exposes assumptions and approximations, and allows greater control over outcome.

Explicit modeling also poses a problem. Capture is a causal process, which Bayesian models express naturally. However, compatibility is not: it models the effects of unknown causes, which Bayesian models generally cannot express. A technique for incorporating effect models in Bayesian models is necessary, which this research also contributes.

Chapter 2

Graphical Models

2.1 Introduction

Stochastic processes are processes with output or behavior characterized by probability distributions. “Graphical models” is the somewhat unfortunate name given to a class of modeling techniques that use graphs (in the nodes-and-edges sense) to represent stochastic processes.

There are at least two good reasons that graphical models are fitting candidates for modeling image reconstruction tasks. First, image capture is in part a physical process that involves optical and quantum properties that can be modeled by probability distributions, such as the trajectory of a light wave as it passes through a lens and the proportion of photons that are localized and detected by a CCD element. (These are often modeled by *point-spread functions* (PSFs) and *white noise*, respectively.) Second, it is often necessary to model the scene that generated an image. The scene is not completely unknown, only uncertain, and we can characterize our beliefs about scenes using probability distributions.

2.1.1 Notation and Terminology

This chapter assumes some knowledge of probability, including the terms *random variable*, *probability mass*, *probability density*, *joint distribution*, and *conditional distribution*.

Random variables are in capital letters; lowercase letters denote their values (such as samples, observations and integration variables). Bold letters denote a collection of some kind, such as vectors, arrays or sets, as in $\mathbf{Z} = \{Z_1, Z_2, Z_3\}$, $\mathbf{z} = \{z_1, z_2, z_3\}$ and $\mathbf{z} \in \mathbf{Z}$.

Subscripting, such as X_i and x_i , denotes indexing into these collections. Specified density and mass functions are subscripted with their random variable, as in $f_X(x|y)$. A bare p , such as $p(y|x)$, denotes a derived density or mass function. As always, $P(X)$ is thoroughly abused as a stand-in for a probability density, mass or query as the occasion requires.

2.2 Bayesian Networks

Bayesian networks [33] are often used to model *generative stochastic processes*: those that can be readily modeled as a noisy function, or as a function whose true nature is uncertain. They are often called *generative*, *causal* or *conditional* models. The basic modeling unit is the random variable: an entity that represents a set of events that might happen and the probabilities with which they will. These probabilities are expressed as conditional distributions. It is important to note that in Bayesian modeling, a probability distribution can represent belief as well as frequency of occurrence.

After a causal process is modeled it may be *sampled* or *simulated*: executed top-down like a program, replacing random variables with samples from their conditional distributions. The result is a sample from the joint distribution of all random variables in the model.

Each random variable may be optionally *observed*, or given a fixed value, which alters the distribution of the remaining variables. Reasoning about this altered model is performing *inference*. Inference is generally most useful when modeling the process in reverse would be difficult; hence process outputs are usually observed, and likely inputs or future outputs are recovered.

The process most immediately relevant to this thesis is image capture. Here a device such as a digital camera records proportions of photons localized on a CCD array, which can be modeled as a generative process. An image (the output) is observed, and a scene (the likely input) is recovered. From this, a new image can be generated as if it had been produced by a higher-resolution device.



(a) A directed acyclic graph (DAG) that represents the decomposition $P(X_1) P(X_2|X_1) P(X_3|X_1, X_2)$.

(b) Independence is asserted by leaving edges out. This graph represents $P(X_1) P(X_2) P(X_3|X_1, X_2)$.

Figure 2.1: Bayesian networks represent joint distributions decomposed into conditional distributions via the chain rule. Both of these represent $P(X_1, X_2, X_3)$.

2.2.1 Formulation

The basic structure of a Bayesian network, the directed acyclic graph (DAG), is due to decomposition according to the chain rule:

$$P(X_1, X_2, \dots, X_n) = P(X_1) P(X_2|X_1) P(X_3|X_1, X_2) \dots P(X_n|X_1, X_2, \dots, X_{n-1}) \quad (2.1)$$

This may be performed with variables in any order.

Any decomposition of n variables can be represented as a fully connected DAG, and every fully connected DAG uniquely represents a decomposition. To create the DAG, first create a node for each random variable. Then for each conditional distribution, add an arrow in the DAG from each variable on the right side pointing to the variable on the left. See Figure 2.2(a) for a demonstration with $n = 3$. The arrow is read “generates”, “causes” or “is conditioned on” as in “ X_1 generates X_2 ”, “ X_1 causes X_2 ” or “ X_2 is conditioned on X_1 ”. The direction of the arrows favors and emphasizes the first two readings.

Specifying all the conditional distributions is more work than specifying the joint distribution. Simplification comes from the notion of *independence*, which means that the value of a random variable has no *direct* effect on the distribution of another. That is, if X_1 and X_2 are independent, then

$$P(X_2|X_1) = P(X_2) \quad (2.2)$$

Thus, as in Figure 2.2(b), the arrow from X_1 to X_2 is left out, and the distribution of X_2 is specified more simply. Any number of direct dependencies may be removed. Every possible DAG may be generated this way; thus, every DAG represents a decomposition that may include independence.

Modeling with Bayesian networks consists of 1) picking a topology that represents the forward process, which involves making independence assumptions; and 2) assigning conditional distributions.

Much of the appeal of Bayesian networks comes from their close relationship to the chain rule. Important laws of probability, such as independence, Bayes' Law, and the multiplication rule (the inverse of the chain rule) can be understood as graph transformations. Important properties of a stochastic model, such as direct and indirect dependence, can be found by inspecting the DAG.

Another desirable property is that *unnormalized complete conditionals* are easy to compute. These are the density or mass functions of each random variable given that every other in the model has been observed:

$$p(x_i | \mathbf{x}_{\{-i\}}) \propto f_{X_i}(x_i | \mathbf{x}_{\text{par}(i)}) \prod_{j \in \text{ch}(i)} f_{X_j}(x_j | \mathbf{x}_{\text{par}(j)}) \quad (2.3)$$

where $x_{\{-i\}}$ represents the values of all random variables except for that of X_i , f_{X_i} is the conditional density or mass function of X_i , $\text{par}(i)$ yields the parents of X_i (the variables that generate it), and $\text{ch}(i)$ yields the children of X_i (the variables it generates). This can be read “the probability of x_i given its parents times the probabilities of the children of x_i given their parents”. The variables referred to are often called the *Markov blanket* of x_i . Note that the complete conditional is comprised only of distributions specified during modeling.

Computing complete conditionals is central to the primary tools currently used for approximate inference.

2.2.2 Inference

Forward processes are usually modeled in order to run them in reverse; that is, to assert that an outcome has been observed and recover the likely inputs. (The process can optionally be run forward using the recovered inputs to obtain likely future outputs.) Considering image reconstruction sheds some light on the reason for this. Human experience tells us something about how scenes are composed. Optical physics and engineering embody knowledge about how scenes cause images—how they are projected and captured. However, what it means for an image to cause a scene, or for an image to cause a reconstructed image, is less than clear. It makes more sense to model known processes and do inference than to try to model unknown, reverse processes.

Inference is generally concerned with deriving or computing a conditional distribution from the model. Let $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ and $\mathbf{Y} = \{Y_1, Y_2, \dots, Y_m\}$ be sets of random variables such that \mathbf{X} generates \mathbf{Y} . (In reconstruction, \mathbf{X} is the scene and \mathbf{Y} is the image.) If \mathbf{Y} is observed, the distribution of most interest for \mathbf{X} becomes its distribution conditioned on \mathbf{Y} , which is given by Bayes' Law:

$$\begin{aligned} P(\mathbf{X}|\mathbf{Y}) &= \frac{P(\mathbf{X}) P(\mathbf{Y}|\mathbf{X})}{P(\mathbf{Y})} \\ &= \frac{P(\mathbf{X}) P(\mathbf{Y}|\mathbf{X})}{\int_{\mathbf{x} \in \mathbf{X}} P(\mathbf{X} = \mathbf{x}) P(\mathbf{Y}|\mathbf{X} = \mathbf{x}) d\mathbf{x}} \end{aligned} \tag{2.4}$$

This may be the reason for the name “Bayesian networks”. Notice that the right-hand side of the law is comprised of distributions that have been specified during modeling. Recovering inputs \mathbf{X} from outputs \mathbf{Y} is equivalent to deriving $\mathbf{X}|\mathbf{Y}$ from known distributions \mathbf{X} and $\mathbf{Y}|\mathbf{X}$, and from observations of \mathbf{Y} .

Special cases have analytic solutions for the integral in the denominator; they are said to be *conjugate*. Most do not, so we turn to approximate inference.

Often, all that is required is the mode, or *maximum a posteriori* (MAP) value of $\mathbf{X}|\mathbf{Y}$. In this case, inspecting Equation 2.4 makes the solution clear. The denominator

is a constant, so finding the value of \mathbf{X} that maximizes the numerator is equivalent to finding the value of \mathbf{X} that maximizes the entire fraction. Hill-climbing algorithms are certainly capable. Stochastic relaxation is often used. In this technique, samplers used to approximate distributions of $\mathbf{X}|\mathbf{Y}$ are modified to have a temperature term that increases concentration of probability near the modes on a schedule [34].

In other cases, the actual distribution of $\mathbf{X}|\mathbf{Y}$ is required. Approximate inference algorithms almost always yield samples from this distribution because samples have convenient properties. For example, approximate expected values, such as utilities, means and variances, can be computed using summation. “Integrating out” a random variable to get marginal distributions is done by simply not including it in the sample vector. Higher accuracy can always be obtained by sampling more.

One of the most effective approximate inference algorithms is the *Gibbs sampler*. It works on a surprising principle: under certain common conditions, sampling from each unobserved random variable’s complete conditional in turn yields samples from $\mathbf{X}|\mathbf{Y}$. When a complete conditional is not conjugate, which is the common case, it may be sampled using a Markov chain sampler called *Metropolis-Hastings*. The combination of these samplers is called *Markov chain Monte Carlo* (MCMC).

2.3 Markov Random Fields

Sometimes the forward stochastic process is either unknown or too difficult to model. This is often the case in image reconstruction, where a complete causal model would contain the entire three-dimensional scene or an approximation of it. In such cases, it is better to model effects rather than causes. For example, while it is difficult to model object boundaries that cause edge primitives, it is relatively easy to model the fact that edge primitives tend to line up with neighbors.

Markov random fields (MRFs) [35] represent processes that are often called *generative* or *noncausal*. The basic modeling unit is still the random variable, but relationships



(a) A simple Markov random field with two cliques: $\{X_1, X_2, X_3\}$ (indicated) and $\{X_1, X_3, X_4\}$. These have associated clique potentials $\Phi_1(x_1, x_2, x_3)$ and $\Phi_2(x_1, x_3, x_4)$.

(b) A Markov random field with four-connected neighborhoods. This topology is used often in image reconstruction.

Figure 2.2: Examples of Markov random fields.

are specified using *clique potentials* rather than conditional distributions. Sampling or simulating the process cannot be done top-down, but generally must be done by samplers used for approximate inference. Observation works in the same way as in Bayesian networks, and reasoning about a model with observations is likewise called inference.

2.3.1 Formulation

An undirected graph may be viewed as a set of nodes and a set of cliques (maximal fully connected subgraphs). Each node in an MRF represents a random variable, and each clique has an associated clique potential that describes the interaction of its members. Let $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ be the set of random variables, and $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_m\}$ be the set of clique potentials. Let \mathbf{x} be instances of \mathbf{X} , with $\mathbf{x}_{\{i\}}$, $i \in 1..m$ denoting an indexed subset. Each Φ_j is a mapping from clique values $\mathbf{x}_{\{j\}}$ to \mathbb{R}^+ . Note that there is no requirement that clique potentials sum to one. Figure 2.3(a) shows a simple MRF with four nodes and two cliques.

Define the *potential* as

$$\Phi(\mathbf{x}) \equiv \prod_{j=1}^m \Phi_j(\mathbf{x}_{\{j\}}) \quad (2.5)$$

The joint density or mass is the normalized potential

$$p(\mathbf{x}) \equiv \frac{\Phi(\mathbf{x})}{\int_{\mathbf{x}' \in \mathbf{X}} \Phi(\mathbf{x}') d\mathbf{x}'} \quad (2.6)$$

Normalization is almost never necessary in practice.

Certain topologies are popular in image reconstruction: grids of four-connected neighborhoods, shown in Figure 2.3(b), and less commonly, grids of eight-connected neighborhoods.

While the relationship between probability laws and MRFs is not so close as between probability laws and Bayesian networks, direct and indirect dependence can still be found by inspecting the graph. The other desirable property of Bayesian networks, that unnormalized complete conditionals are easy to compute, holds for MRFs. It is the product of the potentials of every clique that a random variable participates in:

$$p(x_i | \mathbf{x}_{\{-i\}}) \propto \prod_{j \in \text{cl}(i)} \Phi_j(\mathbf{x}_{\{j\}}) \quad (2.7)$$

where $\text{cl}(i)$ returns the indexes of cliques that X_i belongs to.

2.3.2 Inference

As in Bayesian networks, inference is concerned with deriving or computing a conditional distribution from the model, given some observed variables. Also as before, few cases have analytic solutions. Again we turn to approximate inference, and find that both MAP estimates and samples from the posterior distribution can be computed using exactly the same algorithms.

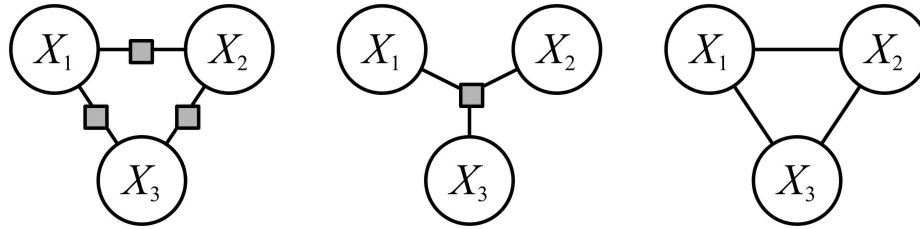


Figure 2.3: Two factor graphs and the equivalent Markov random field topology. Shaded boxes represent factors, or relationships among connected variables. By being formulated in terms of clique potentials, MRFs hide important information.

2.4 Factor Graphs

Factor graphs [36] are a general way of representing functions of multiple variables that can be factored into a product of “local” functions. Bayesian networks whose distributions are specified as mass or density functions, and all MRFs, can be represented as factor graphs.

The formulation is almost identical to that of MRFs. Let $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ be the set of variables, and $\mathbf{f} = \{f_1, f_2, \dots, f_m\}$ be the set of factors. Each f_j is a mapping from $\mathbf{x}_{\{j\}} \subseteq \mathbf{x}$ to \mathbb{R} . The main differences are that factors do not have to be associated with cliques, and they range over the entire real line (which is restricted to \mathbb{R}^+ for stochastic models). The product of factors represents a function,

$$f(\mathbf{x}) \equiv \prod_{j=1}^m f_j(\mathbf{x}_{\{j\}}) \quad (2.8)$$

Because factor graphs were developed without probability in mind, normalization is not part of their definition.

Even without a probabilistic pedigree, factor graphs may be a better choice for noncausal modeling than MRFs. Figure 2.3 shows two factor graphs and the result of converting each of them into an MRF. Though the factor graphs’ direct dependencies are different, inspecting the MRF gives no hint of such, since all factors were multiplied into the same clique potential.

Though factor graphs are more expressive, MRFs have more mindshare in computer vision and image processing. Part of this is simply due to social momentum. Also, though MRFs are defined in terms of *maximal* cliques, they are often specified in terms of *subcliques*. In this case, the only difference between MRFs and factor graphs—as used in stochastic modeling—is implied normalization (rarely carried out) and graphical representation.

Chapter 3

Super-Resolution via Recapture and Bayesian Effect Modeling

An earlier version of this chapter was accepted for publication at the *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2009*.

3.1 Introduction

Many image processing tasks, such as scaling, rotating and warping, require good estimates of between-pixel values. Though this research may be applied to interpolation in any task, we restrict our attention to single-frame super-resolution, which we define as scaling a digital image to larger than its original size.

While recent methods give excellent results at moderate scaling factors [32], all show significant artifacts by scaling factors of 4x (Figure 3.1). We contribute a novel method grounded in Bayesian inference that preserves high-quality edges in a manner agnostic to scale. Central to this is an image reconstruction framework adapted from supervised machine learning. Certain aspects of BEI require modeling unknown causes with known effects, which we show can be incorporated fairly easily into an otherwise causal model.

The simplest existing super-resolution methods are nonadaptive, which make the fewest assumptions and are the easiest to implement. Function fitting methods regard the image as samples from a continuous function and fit basis functions to approximate it [1]. Frequency-domain methods regard the image as a sample of a bandlimited signal

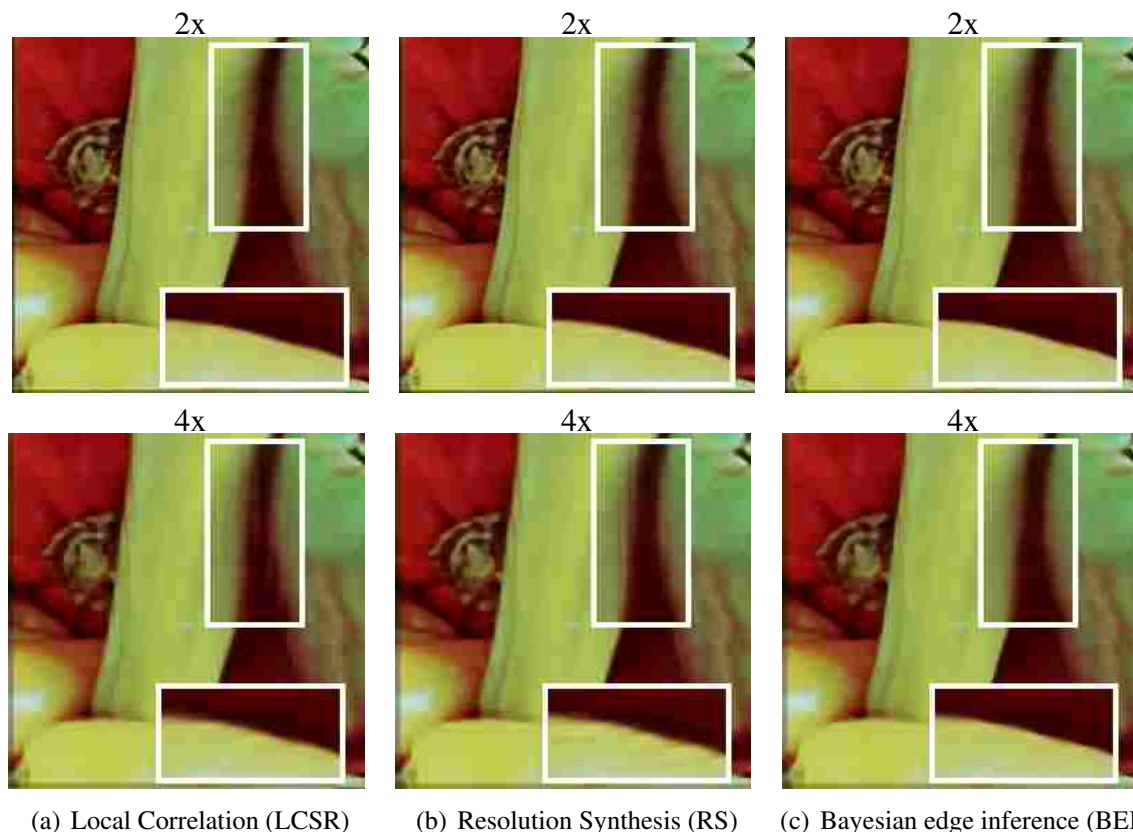


Figure 3.1: Comparison of three super-resolution methods on a region of “Peppers” at factors of 2x and 4x. Insets are subregions magnified an additional 2x using bilinear interpolation for display only. LCSR [12] (a) and RS [13] (b) are arguably the best published methods, as measured in [32]. Note that artifacts are almost entirely absent in 2x but show up clearly in 4x, namely steps in steep gradient areas (upper inset) and boundary incoherence (lower inset). BEI (c) does not exhibit these artifacts even at 4x.

and attempt perfect reconstruction [2, 3]. All of these suffer from blockiness or blurring at moderate magnification factors. The reason is simple: the scene itself is not bandlimited.

Adaptive methods make strong assumptions about scenes in general to obtain more plausible results. Parametric methods attempt to preserve strong edges by fitting edges [6, 9] or adapting basis functions [10, 37]. Nonparametric methods discover features that should be preserved using training images [12, 13] or use training images both as samples from the distribution of all images and as primitives for reconstruction [17, 18].

Ouwerkerk recently surveyed adaptive methods [32] and applied objective measures to their outputs on test images at 2x and 4x magnification factors. Methods that give excellent results on 2x super-resolution tend to show artifacts at higher factors. For exam-

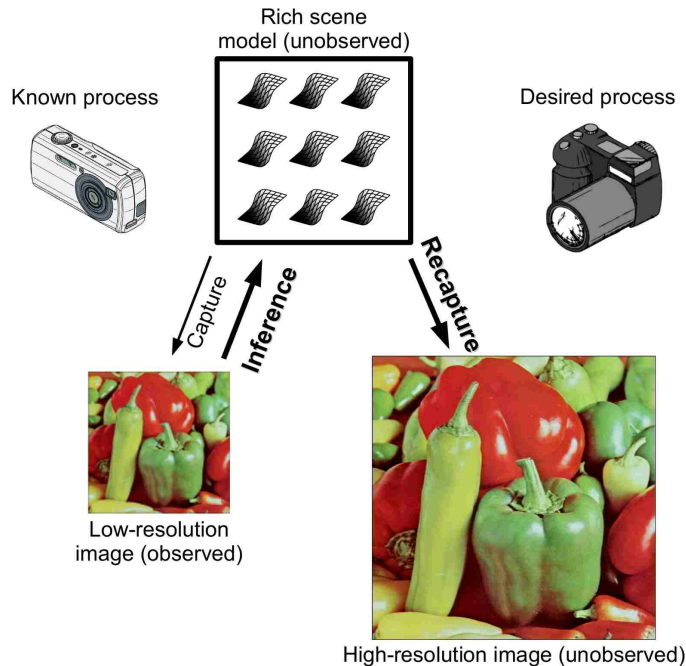


Figure 3.2: The recapture framework applied to super-resolution. The original low-resolution image is assumed to have been generated by a capture process operating on an unobserved scene. Inference recovers the scene, which is used to capture a new image.

ple, Figure 3.1 shows the results of applying the two methods found to be best to a region of “Peppers”. Artifacts that are nearly absent at 2x become noticeable at 4x, namely steps in gradient areas and boundary incoherence. Because artifacts show up so well at those scales, our research focuses on factors of 4x or more.

Optimization methods [27, 30, 31] formulate desirable characteristics of images as penalties or priors and combine this with a reconstruction constraint to obtain an objective function. The reconstruction constraint ensures that the result, when downsampled, matches the input image. Though BEI is similar in many respects, it does not model downsampling of a high-resolution image, but models image capture of a detailed scene and *recapture* with a fictional, higher-resolution process (Figure 3.2). For this we adapt a Bayesian framework from supervised machine learning [38].

For scale-invariance we model a projection of the scene as a piecewise continuous function, much like a facet model [39]. To address blur analytically, we construct it such

that it approximates the continuous blurring of step edges with a spatially varying PSF. We address stepping in gradients by carefully modeling minimum blur.

Outside of modeling the scene hierarchically, some notion of compatibility among scene primitives [17, 18] is required to ensure that object boundaries are coherent. We show that Markov random field compatibility functions can be incorporated into Bayesian networks in a way that is direct, intuitive, and preserves independence relationships, and then incorporate compatibility into our model.

3.2 Reconstruction by Recapture

Optimization methods apply to reconstruction tasks in general. These assume that an original image I' existed, which was degraded to produce I . They are either motivated or formulated explicitly in Bayesian terms, in two parts. First is a prior on I' , which encodes knowledge about images such as gradient profiles [31] or isophote curvature [27]. The second part is often called a *reconstruction constraint* [19], *back-projection* [25] or *sensor model* [26]: a conditional distribution $I|I'$ that favors instances of I' that, when degraded, match I . The result is usually found by maximizing the joint probability $P(I|I')P(I')$ to obtain the most probable $I'|I$. Figure 3.4(a) shows the framework as a simple Bayesian network.

Consider two related tasks that have been addressed in the optimization framework. First is a super-resolution task: scaling up a full-resolution digital photo for printing. Second is CCD demosaicing: a consumer-grade camera filtered light before detecting it with a single-chip sensor. Two-thirds of the color data is missing and must be inferred. These tasks violate assumptions made by optimization methods. There was no pristine original image that was degraded, and the only thing that can possibly be reconstructed is the scene. In these cases and many others, *the true objective is to produce a novel image of the same scene as if it had been captured using a better process.*

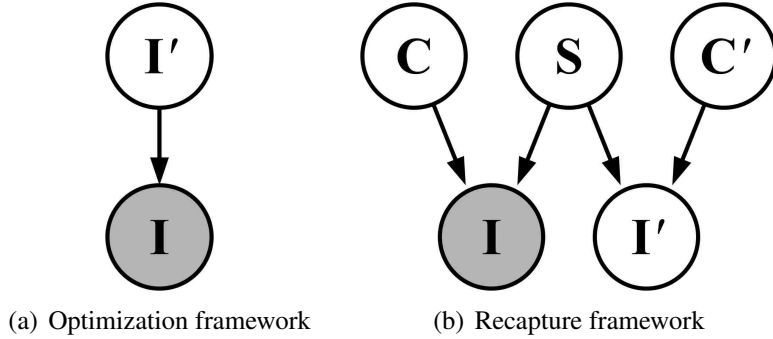


Figure 3.3: Bayesian frameworks for image reconstruction. Shaded nodes are observed. Optimization methods (a) model an assumed high-resolution original \mathbf{I}' generating the input image \mathbf{I} . This requires prior knowledge about \mathbf{I}' and a degradation process $\mathbf{I}|\mathbf{I}'$. The proposed framework (b) models *capture* and *recapture* rather than (or including) degradation. This requires a rich scene model \mathbf{S} and capture process models $\mathbf{I}|\mathbf{C}, \mathbf{S}$ and $\mathbf{I}'|\mathbf{C}', \mathbf{S}$.

Based on this objective, we propose the more general recapture framework shown in Figure 3.4(b). Here, a process (*e.g.* a camera) with parameters \mathbf{C} is assumed to have captured the scene \mathbf{S} as the original image \mathbf{I} . This process may include degradation. A fictional process (*e.g.* a better camera) with parameters \mathbf{C}' *recaptures* the same scene as the result \mathbf{I}' . As in optimization methods, \mathbf{I} is observed and inference recovers \mathbf{I}' , but through \mathbf{S} rather than directly. This requires a scene model rich enough to reconstruct an image.

There is also a practical advantage to recapture. With the right scene model, if only recapture parameters are changed, the result can be rerendered at interactive speeds.

3.3 Effect Modeling in Bayesian Networks

Our super-resolution method models the scene using overlapping primitives, which must be kept locally coherent. This has been done using ad-hoc compatibility [18] and Markov random field (MRF) clique potentials [17]. However, converting recapture models to MRFs would hide independence relationships and the notion of causality—and image capture is obviously causal in nature. Graphical models rarely mix causal and noncausal dependence. Chain graphs do [40] but are less well-known and more complex than Bayesian networks and MRFs.

It is worth noting that MRFs are not used in reconstruction to model noncausal dependence for its own sake, but to *model unknown causes that have known effects* on an image or scene, which are usually symmetric. This is appropriate when inferring causes is cumbersome or intractable.

Fortunately, modeling unknown causes with known effects in a Bayesian network is fairly simple (Figure 3.4). In the interest of saving space, we note without giving details that motivation for the following comes from the conversion of MRFs to factor graphs to Bayesian networks [41]. Let $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ be the set of random variables in a Bayesian network, and $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_m\}$ be a set of functions that specify an effect (such as compatibility). Let \mathbf{x} be instances of \mathbf{X} , with $\mathbf{x}_{\{i\}}$, $i \in 1..m$ denoting an indexed subset. Each Φ_i is a mapping from $\mathbf{x}_{\{i\}}$ to \mathbb{R}^+ . For each Φ_i , add to the network a new real-valued observed variable Z_i with density f_{Z_i} such that

$$f_{Z_i}(z_i = 0 | \mathbf{x}_{\{i\}}) = \Phi_i(\mathbf{x}_{\{i\}}) \quad (3.1)$$

Because Z_i is real-valued, f_{Z_i} does not have to be normalized. Because it will remain observed, its density does not have to be specified except at 0. (There are uncountably infinite candidates for f_{Z_i} ; we will assume one of them.) Adding this new observed variable cannot create cycles or introduce unwanted first-order dependence.

Inference may proceed on joint density $p'(\mathbf{x})$:

$$p'(\mathbf{x}) \equiv p(\mathbf{x} | \mathbf{z} = \mathbf{0}) = \frac{p(\mathbf{x})p(\mathbf{z} = \mathbf{0} | \mathbf{x})}{p(\mathbf{z} = \mathbf{0})} \propto p(\mathbf{x})p(\mathbf{z} = \mathbf{0} | \mathbf{x}) = p(\mathbf{x}) \prod_{i=1}^m \Phi_i(\mathbf{x}_{\{i\}}) \quad (3.2)$$

For Gibbs sampling [34], Markov blanket conditionals are

$$p'(x_j | \mathbf{x}_{\{-j\}}) \equiv p(x_j | \mathbf{x}_{\{-j\}}, \mathbf{z} = \mathbf{0}) \propto f_{X_j}(x_j | x_{\text{par}(j)}) \prod_{k \in \text{ch}(\mathbf{X}, j)} f_{X_k}(x_k | x_{\text{par}(k)}) \prod_{i \in \text{ch}(\mathbf{Z}, j)} \Phi_i(\mathbf{x}_{\{i\}}) \quad (3.3)$$

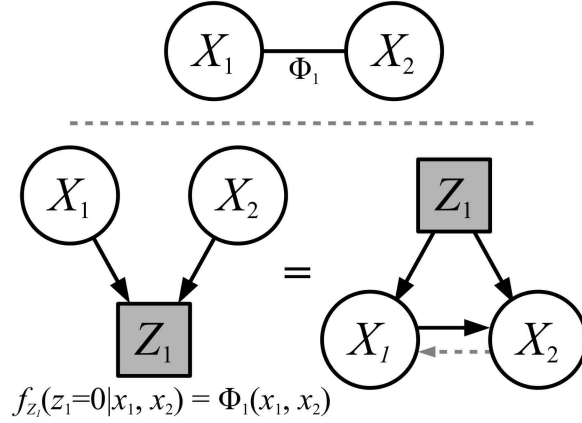


Figure 3.4: Bayesian effect modeling. X_1 and X_2 share an unknown cause with known effect Φ_1 . This is modeled as an observed node in the network. An equivalent network exists in which X_1 and X_2 are directly dependent and the joint distribution X_1, X_2 is symmetric if Φ_1 is symmetric and $X_1 \sim X_2$.

where $\text{par}(j)$ yields the indexes of the parents of X_j and $\text{ch}(\mathbf{A}, j)$ yields the indexes of the children of X_j within \mathbf{A} .

3.4 Super-Resolution Model

The steps to using the recapture framework for reconstruction are: 1) define the scene model, expressing knowledge about the scene or scenes in general as priors; 2) define the capture and recapture processes; and 3) observe \mathbf{I} and report \mathbf{I}' . Because the objective is to generate an image as if it had been captured by a fictional process, the proper report is a sample from (rather than say, a MAP estimate of) the *posterior predictive* distribution $\mathbf{I}'|\mathbf{I}$. This may be done by running a sampler such as Gibbs or MCMC on $\mathbf{S}|\mathbf{I}$, followed by sampling $\mathbf{I}'|\mathbf{S}$ once.

Definitions. An image \mathbf{I} , which is an $m \times n$ array of RGB triples normalized to $[0, 1]$, is observed. A real-valued scaling factor s is selected and an $\lfloor sm \rfloor \times \lfloor sn \rfloor$ image \mathbf{I}' is reconstructed through an $m \times n$ scene model \mathbf{S} . Coordinates of triples, which are parameters

of the capture process, are

$$\begin{aligned} \mathbf{C}_{i,j}^x &\equiv i + \frac{1}{2} & i \in 0..m - 1 \\ \mathbf{C}_{i,j}^y &\equiv j + \frac{1}{2} & j \in 0..n - 1 \end{aligned} \tag{3.4}$$

where i, j are image indexes.

The scene and capture models use the following to convert an integer- or real-valued coordinate to the set of its nine nearest neighbors:

$$\begin{aligned} N_9(x, y) &\equiv \{i \in \mathbb{Z} \mid -1 \leq i - \lfloor x \rfloor \leq 1\} \\ &\quad \times \{j \in \mathbb{Z} \mid -1 \leq j - \lfloor y \rfloor \leq 1\} \end{aligned} \tag{3.5}$$

For clarity we omit here treatment of image borders.

The model uses an approximating quadratic B-spline kernel [42] to weight facet outputs, which we denote as w_q .

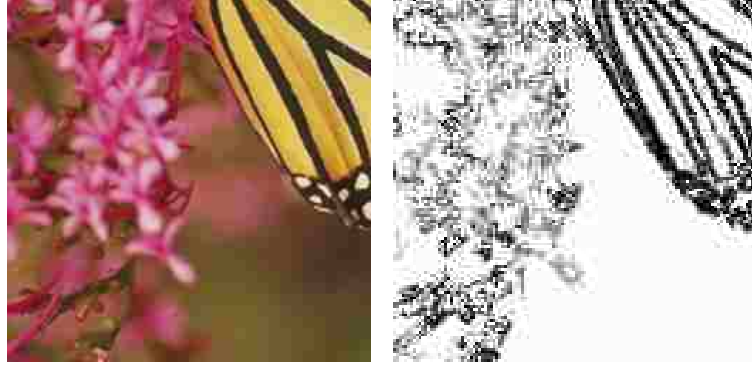
3.4.1 Facets

S is similar to a facet model [39] in that it uses overlapping geometric primitives to represent a continuous function. It differs in these fundamental ways: 1) facets are blurred step edges, not polynomials; 2) it represents a scene rather than an image; 3) the *combined* output of the primitives is fit to the data through a capture model; and 4) facets are made compatible with neighbors where they overlap.

The assumption that the most important places to model well are object boundaries determines the shape of the facets. Each is based on an implicit line:

$$\text{dist}(x, y, \theta, d) \equiv x \cos \theta + y \sin \theta - d \tag{3.6}$$

To approximate blurring with a spatially varying PSF, we assign each facet a Gaussian PSF and convolve each analytically before combining outputs. For simplicity, PSFs are



(a) A region of “Monarch,” reduced 2x and reconstructed. (b) The inferred PSF standard deviation S^σ . Darker is narrower.

Figure 3.5: BEI’s spatially varying PSF. It has correctly inferred a wider PSF for the flower petals, which are blurry due to shallow depth-of-field. These are still blurry in the final output.

symmetric and only vary in standard deviation. The usefulness of a spatially varying PSF is shown in Figure 3.5.

Convolving a discontinuity with a Gaussian kernel gives the profile of the step edge:

$$\begin{aligned} \text{prof}(d, \sigma, v^+, v^-) &\equiv v^+ \int_0^\infty G(d-t, \sigma) dt + v^- \int_{-\infty}^0 G(d-t, \sigma) dt \\ &= \frac{v^+ - v^-}{2} \text{erf}\left(\frac{d}{\sqrt{2}\sigma}\right) + \frac{v^+ + v^-}{2} \end{aligned} \quad (3.7)$$

where erf is the error function and v^+ and v^- are intensities on the positive and negative sides of the edge. Because of the PSFs’ radial symmetry, a facet can be defined in terms of its profile:

$$\text{edge}(x, y, \theta, d, v^+, v^-, \sigma) \equiv \text{prof}(\text{dist}(x, y, \theta, d), \sigma, v^+, v^-) \quad (3.8)$$

An example step edge is shown in Figure 3.6.

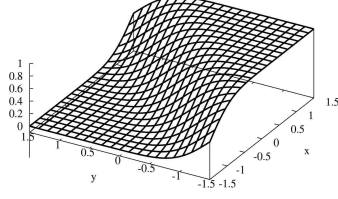


Figure 3.6: Scene facets are blurred step edges, or linear discontinuities convolved with blurring kernels. This has $\theta = -\frac{\pi}{4}$, $d = 0$ as the line parameters and a Gaussian kernel with $\sigma = \frac{1}{3}$.

3.4.2 Scene Model

The scene model random variables are a tuple of $m \times n$ arrays sufficient to parameterize an array of facets:

$$\mathbf{S} \equiv (\mathbf{S}^\theta, \mathbf{S}^d, \mathbf{S}^{v^+}, \mathbf{S}^{v^-}, \mathbf{S}^\sigma) \quad (3.9)$$

We regard the scene as an array of facet functions. Let

$$\mathbf{S}_{i,j}^{\text{edge}}(x, y) \equiv \text{edge}(x - \mathbf{C}_{i,j}^x, y - \mathbf{C}_{i,j}^y, \mathbf{S}_{i,j}^\theta, \mathbf{S}_{i,j}^d, \mathbf{S}_{i,j}^{v^+}, \mathbf{S}_{i,j}^{v^-}, \mathbf{S}_{i,j}^\sigma) \quad (3.10)$$

be an array of facet functions centered at \mathbf{C}^x , \mathbf{C}^y and parameterized on the variables in \mathbf{S} .

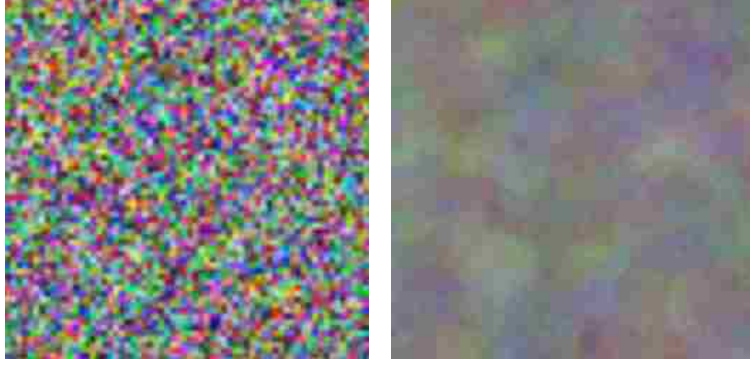
A generalization of weighted facet output, *weighted expected scene value*, is also useful:

$$\mathbb{E}[h(\mathbf{S}_{x,y})] \equiv \sum_{k,l \in \mathcal{N}_9(x,y)} w_q(x - \mathbf{C}_{k,l}^x, y - \mathbf{C}_{k,l}^y) h(\mathbf{S}_{k,l}^{\text{edge}}(x, y)) \quad (3.11)$$

When $h(x) = x$, this is simply weighted output. Weighted scene variance will be defined later using $h(x) = x^2$.

Priors. It seems reasonable to believe that, for each facet considered alone,

1. No geometry is more likely than any other.
2. No intensity is more likely than any other.
3. There are proportionally few strong edges [17].



(a) Two samples from the prior predictive \mathbf{I}' (*i.e.* no data).



(b) Two samples from the posterior predictive $\mathbf{I}'|\mathbf{I}$ (*i.e.* with data).

Figure 3.7: The utility of compatibility. The right images include compatibility. The samples without data (a) show that it biases the prior toward contiguous regions. The samples with data (b) show that it makes coherent boundaries more probable.

The priors are chosen to represent those beliefs:

$$\begin{aligned}
 \mathbf{S}_{i,j}^{\theta} &\sim \text{Uniform}(-\pi, \pi) & \mathbf{S}_{i,j}^{v^+} &\sim \text{Uniform}(0, 1) \\
 \mathbf{S}_{i,j}^d &\sim \text{Uniform}(-3, 3) & \mathbf{S}_{i,j}^{v^-} &\sim \text{Uniform}(0, 1) \\
 \mathbf{S}_{i,j}^{\sigma} &\sim \text{Beta}(1.6, 1)
 \end{aligned}
 \tag{3.12}$$

Compatibility. It seems reasonable to believe that scenes are comprised mostly of regions of similar color, and that neighboring edges tend to line up. We claim that both can be represented by giving high probability to low variance in facet output. (Figure 3.7 demon-

strates that this is the case.) Recalling that $E[\mathbf{S}_{i,j}^2] - E[\mathbf{S}_{i,j}]^2 = \text{Var}[\mathbf{S}_{i,j}]$, define

$$\Phi_{i,j}(\mathbf{S}_{N_9(i,j)}) \equiv \exp\left(-\frac{\text{Var}[\mathbf{S}_{i,j}]}{2\gamma^2}\right) \quad (3.13)$$

as the compatibility of the neighborhood centered at i, j , where γ is a standard-deviation-like parameter that controls the relative strength of compatibility. At values near ω (defined in the capture model as standard deviation of the assumed white noise), compatibility tends to favor very smooth boundaries at the expense of detail. We use $\gamma = 3\omega = 0.015$, which is relatively weak.

In image processing, compatibility is usually defined in terms of pairwise potentials. We found it more difficult to control its strength relative to the capture model that way, and more difficult to reason about weighting. Weighting seems important, as it gives consistently better results than not weighting. This may be because weighted compatibility has a measure of freedom from the pixel grid.

3.4.3 Capture and Recapture

The capture and recapture processes assume uniform white noise approximated by a narrow Normal distribution centered at the weighted output value:

$$\mathbf{I}_{i,j} | \mathbf{S}_{N_9(i,j)} \sim \text{Normal}(E[\mathbf{S}_{i,j}], \omega) \quad (3.14)$$

where i, j are real-valued coordinates and ω is the standard deviation of the assumed white noise. We use $\omega = 0.005$.

Recapture differs from capture in treatment of \mathbf{S}^σ (defined in the following section) and in using a bilinear kernel to combine facet outputs. The bilinear kernel gives better results, possibly because it makes up for blur inadvertently introduced by the quadratic kernel w_q .

3.4.4 Minimum Blur

To make the recaptured image look sharp, we assume the original capture process had a minimum PSF width C^σ and give the recapture process a narrower minimum PSF width $C^{\sigma'}$. Because variance sums over convolution of blurring kernels, these are accounted for in the capture model by adding variances. That is, rather than computing \mathbf{S}^{edge} using \mathbf{S}^σ , $\mathbf{I}|\mathbf{S}$ uses

$$\mathbf{S}_{k,l}^{\sigma * } \equiv \sqrt{(\mathbf{S}_{k,l}^\sigma)^2 + (C^\sigma)^2} \quad (3.15)$$

The value of C^σ depends on the actual capture process. For recapture, we have found that $C^{\sigma'} \equiv C^\sigma/s$ tends to give plausible results. (Recall that s is the scaling factor.)

3.4.5 Decimation Blur

In [32], as is commonly done, images were downsampled by decimation: convolving with a 2×2 uniform kernel followed by nearest-neighbor sampling. When decimation blur has taken place, regardless of how many times, the capture process can model it as a constant minimum PSF.

With image coordinates relative to \mathbf{I} , decimation is approximable as application of ever-shrinking uniform kernels. The last kernel was one unit wide, the kernel previous to that was a half unit wide, and so on. Let u_w be a uniform kernel with width w . Assuming no upper bound on the number of decimations, the upper bound on variance is

$$\begin{aligned} u &\equiv u_1 * u_{\frac{1}{2}} * u_{\frac{1}{4}} * u_{\frac{1}{8}} * u_{\frac{1}{16}} * \dots \\ \text{Var}[u] &= \text{Var}[u_1] + \text{Var}[u_{\frac{1}{2}}] + \text{Var}[u_{\frac{1}{4}}] + \dots \\ &= \sum_{n=0}^{\infty} \frac{(1/2)^{2n}}{12} = \frac{1}{12} \left(\frac{4}{3}\right) = \frac{1}{9} \end{aligned} \quad (3.16)$$

The series converges so quickly that $C^\sigma = \frac{1}{3}$ is a fairly good estimate for any number of decimations.

3.4.6 Inference

The Markov blanket for $S_{i,j}$ includes its nine children in \mathbf{I} and Φ , and their nine parents each in \mathbf{S} .

Only the time to convergence seems to be affected by choice of initial values. We use the following:

$$\begin{aligned} \mathbf{S}^\theta &= \tan^{-1}((\nabla \mathbf{I})_y / (\nabla \mathbf{I})_x) & \mathbf{S}^d &= 0 \\ \mathbf{S}^{v^+} &= \mathbf{S}^{v^-} = \mathbf{I} & \mathbf{S}^\sigma &= \frac{1}{2} \end{aligned} \quad (3.17)$$

In this model, posterior density in \mathbf{S} is so concentrated near the modes that samples of \mathbf{S} after convergence are virtually indistinguishable. Therefore we find a MAP estimate of $\mathbf{S}|\mathbf{I}$ and sample $\mathbf{I}|\mathbf{S}$ to approximate sampling $\mathbf{I}|\mathbf{I}$.

Gibbs with stochastic relaxation [34] finds a MAP estimate quickly, but a deterministic variant of it is faster. It proceeds as Gibbs sampling, except that for each random variable X , it evaluates the Markov blanket conditional at x , $x + \sigma_X$ and $x - \sigma_X$, and keeps the argmax value.

Tuning σ_X online results in fast convergence. Every iteration, it is set to an *exponential moving standard deviation* of the values seen so far. This is computed by tracking an exponential moving mean and moving squared mean separately and using $\text{Var}[X] = \text{E}[X^2] - \text{E}[X]^2$. Let

$$\begin{aligned} \sigma_{X_i}^2 &= v_{X_i} - m_{X_i}^2 \\ m_{X_0} &= x_0 & m_{X_i} &= \alpha m_{X_{i-1}} + (1 - \alpha) x_i \\ v_{X_0} &= x_0^2 + \sigma_{X_0}^2 & v_{X_i} &= \alpha v_{X_{i-1}} + (1 - \alpha) x_i^2 \end{aligned} \quad (3.18)$$

where σ_{X_0} is the initial standard deviation. The value of α denotes how much weight is given to previous values. Using $\sigma_{X_0} = 0.05$ for all X and $\alpha = 0.5$, we found acceptable convergence within 100 iterations on all test images.

3.5 Results

Ouwerkerk [32] chose three objective measures that tend to indicate subjective success better than mean squared error, and gave results for nine single-frame super-resolution methods on seven test images chosen for representative diversity. The original images were decimated once or twice, reconstructed using each method, and compared. Therefore we set minimum blur $C^\sigma = \frac{1}{3}$ as derived in Section 3.4.5.

Figure 3.8 shows that BEI keeps boundaries coherent even in difficult neighborhoods because of compatibility. Note the boundaries of the narrow black veins, which are especially easy to get wrong. Figure 3.9 is a comparison of some methods with BEI on a region of “Lena,” which shows that BEI preserves gradients and sharpens edges. Note gradients on the nose and on the shadow on the forehead, and the crisp boundaries of the shoulder and brim of the hat.

Table 3.1 gives measures for BEI in 4x super-resolution along with linear interpolation for a baseline and the top two, resolution synthesis (RS) [13] and local correlation (LCSR) [12], for comparison.

Unfortunately, a bug in computing ESMSE was not caught before publication of [32], making this measure suspect [43]. Further, it is questionable whether it measures *edge* stability, as the edge detector used falsely reports smooth, contiguous regions as edges. Therefore, Table 3.1 includes a corrected ESMSE measure using the same edge detector with its minimum threshold raised from 10% to 20%.

We give numeric results for a noiseless recapture process because the objective measures are somewhat sensitive to small amounts of noise. In practice, though, a little noise usually increases plausibility.

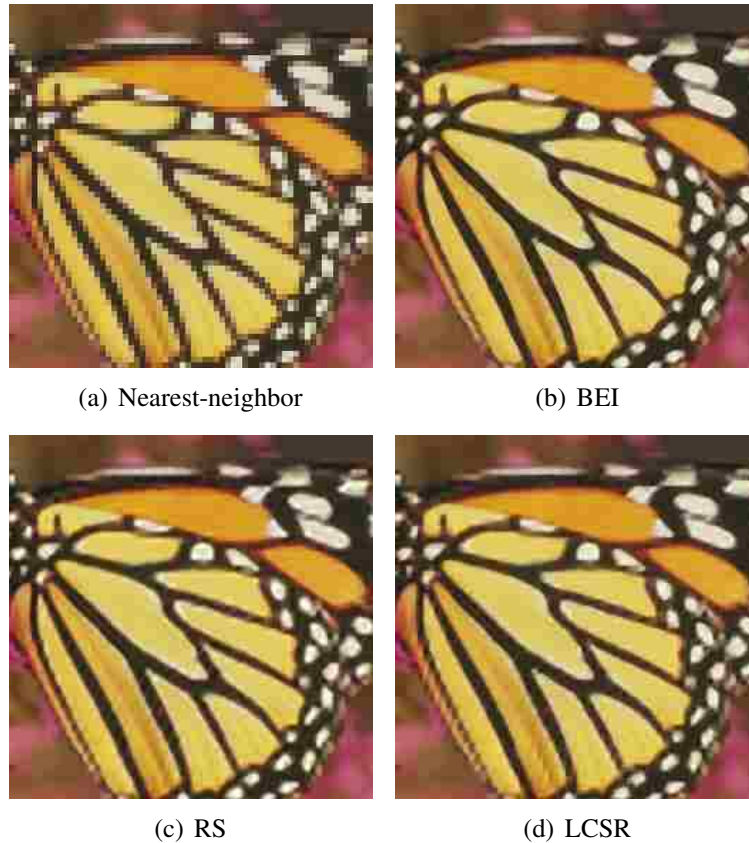


Figure 3.8: A difficult region of “Monarch”. Most 3×3 neighborhoods within the black veins include part of the boundary on each side. While RS and LCSR have done well at avoiding artifacts here (much better than the others compared in [32]), BEI eliminates them almost entirely because of compatibility.

3.6 Other Applications

One advantage to Bayesian inference is that missing data is easy to deal with: simply do not include it.

In CCD demosaicing [17], a Bayer filter, which is a checkerboard-like pattern of red, green, and blue, is assumed overlaid on the capture device’s CCD array [44]. It could be said that the filtered two-thirds is missing data. We implemented this easily in BEI by not computing densities at missing values. The result of simulating a Bayer filter is shown in Figure 3.10. We also found it helpful to change the prior on S^σ to $\text{Uniform}(0, 1)$ and set minimum blur to zero.

Image	PSNR, higher is better				MSSIM, higher is better			
	Bilinear	RS	LCSR	BEI	Bilinear	RS	LCSR	BEI
<i>Graphic</i>	17.94	20.19	19.55	20.87	0.775	0.864	0.854	0.898
<i>Lena</i>	27.86	29.57	29.08	29.60	0.778	0.821	0.810	0.820
<i>Mandrill</i>	20.40	20.71	20.63	20.67	0.459	0.536	0.522	0.519
<i>Monarch</i>	23.91	26.41	25.90	26.65	0.848	0.896	0.889	0.902
<i>Peppers</i>	25.31	26.26	25.66	26.27	0.838	0.873	0.864	0.876
<i>Sail</i>	23.54	24.63	24.31	24.55	0.586	0.679	0.657	0.663
<i>Tulips</i>	25.43	28.19	27.56	28.44	0.779	0.843	0.831	0.847

Image	ESMSE, lower is better				ESMSE fixed, 20% thresh.			
	Bilinear	RS	LCSR	BEI	Bilinear	RS	LCSR	BEI
<i>Graphic</i>	3.309	2.998	3.098	2.151	5.871	3.760	4.027	3.571
<i>Lena</i>	5.480	4.718	4.706	4.786	5.212	4.472	4.547	4.556
<i>Mandrill</i>	6.609	6.301	6.278	6.393	6.333	6.097	6.075	6.213
<i>Monarch</i>	5.448	4.518	4.606	4.547	5.260	4.177	4.445	4.214
<i>Peppers</i>	5.531	4.905	4.864	4.889	5.448	5.061	5.061	5.043
<i>Sail</i>	6.211	5.776	5.808	5.893	6.025	5.305	5.418	5.447
<i>Tulips</i>	5.994	5.198	5.286	5.161	5.679	4.569	4.769	4.549

Table 3.1: Comparison of bilinear, BEI, and the top two methods from [32], using objective measures from the same, on 4x magnification. $PSNR = 10 \log_{10}(s^2/MSE)$, where s is the maximum image value and MSE is the mean squared error. MSSIM is the mean of a measure of local neighborhoods that includes mean, variance, and correlation statistics. ESMSE is the average squared difference in maximum number of sequential edges as found by a Canny edge detector with increasing blur. See the text for an explanation of “ESMSE fixed”.

Inpainting can also be regarded as a missing data problem. By not computing densities in defaced regions, BEI returned the image shown in Figure 3.11. Again we flattened the prior on S^σ and set minimum blur to zero. We also set the initial values to the rather blurry output of a simple diffusion-based inpainting algorithm [45], which tends to speed convergence without changing the result.

Super-resolution can be regarded as a missing data problem where the missing data is off the pixel grid. In fact, there is nothing specific to super-resolution in BEI’s scene or capture model at all. Bayesian inference recovers the most probable scene given the scene model, capture process model, and whatever data is available. In this regard, super-resolution, CCD demosaicing, and inpainting are not just related, but are nearly identical.



Figure 3.9: A 256×256 region of “Lena” (a) decimated twice and magnified 4x (b – g). Note the gradient steps in (e) and (f), especially in steep gradients such as on the nose and in the shadow on the forehead. Because BEI can model decimation blur explicitly in the capture and recapture processes, it preserves these gradients at 4x (g) and 8x (h) while keeping boundaries sharp.

3.7 Limitations and Future Work

BEI is computationally inefficient. Though inference is linear in image size, computing Markov blanket log densities for $9mn$ random variables is fairly time-consuming. Our

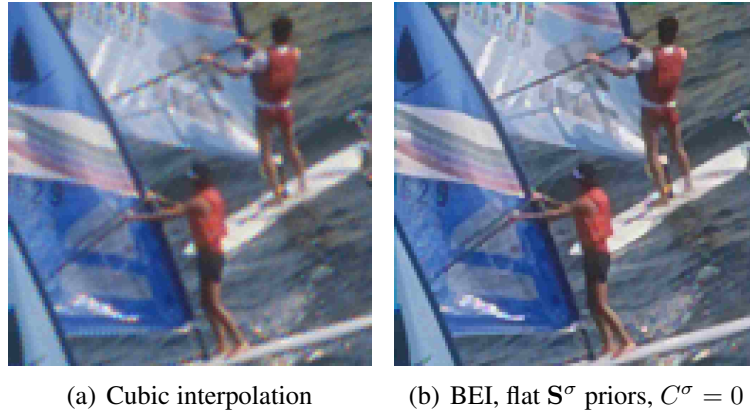


Figure 3.10: CCD demosaicing with a simulated Bayer filter. BEI, which was changed only trivially for this, treats it naturally as a missing data problem. Note the sharp edges and lack of ghosting.

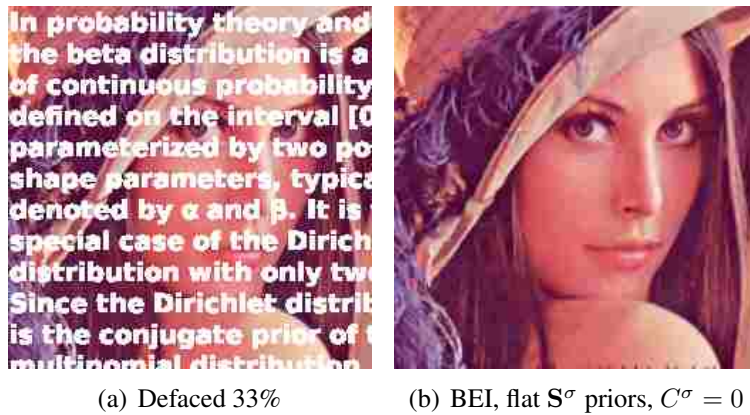


Figure 3.11: Inpainting with BEI. As with CCD demosaicing, this requires only trivial changes to the model. Bayesian inference has recovered the most probable scene given the available data.

highly vectorized Python + NumPy implementation takes about 5 minutes on a 2.4GHz Intel CPU for 128×128 images. However, there is interpreter overhead, vectorization means BEI scales well in parallel, and nearly quadratic speedup could be gained by taking NEDI's hybrid approach [10], which restricts inference to detected edges.

Almost all single-frame super-resolution methods tend to yield overly smooth results. Sharp edges and relative lack of detail combine to create an effect like the uncanny valley [46]. BEI, which does well on object boundaries and gradients, could be combined with methods that invent details like Tappen and Freeman's MRFs [17]. Good objective measures for such methods could be difficult to find. Many are confused by small amounts of noise, and would likely be even more confused by false but plausible details.

Ouwerkerk observed [32] that most methods could benefit from a line model, and we have observed that T-junctions are another good candidate.

Related to CCD demosaicing is undoing *bad* CCD demosaicing in images captured by devices that do not allow access to raw data. This may be as simple as modeling the naïve demosaicing algorithm in the capture process.

Because Bayesian models are composable, any sufficiently rich causal model can model the scene. If not rich enough, it can be used as a scene prior. For example, parameterized shape functions that return oriented discontinuities or functions from region classifications to expected gradients and values can directly condition priors on S . Even compatibility functions can be conditioned on these.

Acknowledgments

We gratefully acknowledge Jos van Ouwerkerk for his time, test images, and code, and Mike Gashler for the exponential moving variance calculation.

Chapter 4

Conclusion

This research is concerned with single-frame super-resolution: increasing the apparent resolution of a digital image. It focuses in particular on magnification factors of 4x and higher, because while the best existing methods give excellent results at moderate scaling factors, they suffer from artifacts by 4x. The two types of artifacts addressed in this research are stepping in steep gradients and boundary incoherence.

The practical result of this research is *Bayesian edge inference* (BEI), which has been shown to adequately address both of these issues while remaining competitive with the best existing methods on objective correctness measures, and surpassing them in many cases.

BEI is built on a Bayesian recapture framework, in which a scene rather than an image is reconstructed, and then recaptured using a better, fictional process. Capture and recapture processes are explicitly modeled in this framework, which allows BEI to model downsampling blur differently in each. This has been shown to sharpen edges in the result while preserving gradients.

Also critical to BEI's success is easy modeling of unknown causes with known effects in Bayesian networks. This allows a prior on scenes that favors both regions of similar color and coherent object boundaries. Without modeling effects, the prior would be highly cumbersome to specify or even make inference intractable.

Effect modeling has been presented as a technique only. In research with broader scope it should be presented as a graphical model in its own right, defined by its transformation into Bayesian networks, with its own set of properties, strengths, and weaknesses.

Finally, BEI with only trivial changes has been shown to have good subjective performance on tasks related to super-resolution: CCD demosaicing and inpainting. Bayesian modeling excels at missing data problems, and all three may be regarded as such. This indicates that BEI and its governing framework should generalize well to even more types of reconstruction tasks.

Bibliography

- [1] T. M. Lehmann, “Survey: Interpolation methods in medical image processing,” *IEEE Transactions on Medical Imaging*, vol. 18, no. 11, pp. 1049–1075, November 1999.
- [2] T. Theussl, H. Hauser, and E. Gröller, “Mastering windows: Improving reconstruction,” in *Proceedings of the IEEE Symposium on Volume Visualization*, 2000, pp. 101–108.
- [3] E. H. W. Meijering, W. J. Niessen, and M. A. Viergever, “Quantitative evaluation of convolution-based methods for medical image interpolation,” *Medical Image Analysis*, vol. 5, pp. 111–126, 2001.
- [4] E. Meijering, “A chronology of interpolation: from ancient astronomy to modern signal and image processing,” in *Proceedings of the IEEE*, no. 3, March 2002, pp. 319–342.
- [5] P. Thevenaz, T. Blu, and M. Unser, “Interpolation revisited,” *IEEE Transactions on Medical Imaging*, vol. 19, no. 7, pp. 739–758, July 2000.
- [6] K. Jensen and D. Anastassiou, “Spatial resolution enhancement of images using non-linear interpolation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, April 1990, pp. 2045–2048.
- [7] ———, “Subpixel edge localization and the interpolation of still images,” *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 285–295, March 1995.
- [8] V. R. Algazi, G. E. Ford, and R. Potharlanka, “Directional interpolation of images based on visual properties and rank order filtering,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, April 1991, pp. 3005–3008.
- [9] J. Allebach and P. W. Wong, “Edge-directed interpolation,” in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, September 1996, pp. 707–710.

- [10] X. Li and M. T. Orchard, “New edge-directed interpolation,” *IEEE Transactions on Image Processing*, vol. 10, pp. 1521–1527, 2001.
- [11] N. Toronto, D. Ventura, and B. S. Morse, “Edge inference for image interpolation,” in *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2005, pp. 1782–1787.
- [12] F. M. Candocia and J. C. Principe, “Super-resolution of images based on local correlations,” *IEEE Transactions on Neural Networks*, vol. 10, pp. 372–380, 1999.
- [13] C. Atkins, C. Bouman, and J. Allebach, “Optimal image scaling using pixel classification,” in *Proceedings of the IEEE International Conference on Image Processing*, 2001, pp. 864–867.
- [14] C. Staelin, D. Greig, M. Fischer, and R. Maurer, “Neural network image scaling using spatial errors,” HP Laboratories Israel, October 2003.
- [15] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, “Learning low-level vision,” *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.
- [16] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, March/April 2002.
- [17] M. Tappen, B. Russell, and W. Freeman, “Exploiting the sparse derivative prior for super-resolution and image demosaicing,” in *IEEE Workshop on Statistical and Computational Theories of Vision*, 2003.
- [18] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, “Image analogies,” *Proceedings of SIGGRAPH 2001*, pp. 327–340.
- [19] S. Baker and T. Kanade, “Limits on super-resolution and how to break them,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, September 2002.
- [20] C. Liu, H. Y. Shum, and W. T. Freeman, “Face hallucination: theory and practice,” *International Journal of Computer Vision*, vol. 75, no. 1, pp. 115–134, October 2007.
- [21] G. Dalley, W. Freeman, and J. Marks, “Single-frame text super-resolution: a Bayesian approach,” in *Proceedings of the IEEE International Conference on Image Processing*, 2004, pp. 3295–3298.

- [22] L. Wei and M. Levoy, “Fast texture synthesis using tree-structured vector quantization,” *Proceedings of SIGGRAPH 2000*, pp. 479–488.
- [23] A. A. Efros and W. T. Freeman, “Image quilting for texture synthesis and transfer,” *Proceedings of SIGGRAPH 2001*, pp. 341–346.
- [24] V. Kwatra, A. Schdl, I. Essa, G. Turk, and A. Bobick, “Graphcut textures: Image and video synthesis using graph cuts,” *Proceedings of SIGGRAPH 2003*, pp. 277–286.
- [25] M. Irani and S. Peleg, “Improving resolution by image registration,” *CVGIP: Graphical Models and Image Proc.*, vol. 53, no. 3, pp. 231–239, 1991.
- [26] T. E. Boult and G. Wolberg, “Local image reconstruction and subpixel restoration algorithms,” *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 1, pp. 63–77, 1993.
- [27] B. S. Morse and D. Schwartzwald, “Image magnification using level-set reconstruction,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 333–340.
- [28] D. Goggins, “Constraint-based interpolation,” Master’s thesis, Brigham Young University, 2005.
- [29] J. Merrell, “Generalized constrained interpolation,” Master’s thesis, Brigham Young University, 2008.
- [30] X. Yu, B. S. Morse, and T. W. Sederberg, “Image reconstruction using data-dependent triangulation,” *IEEE Computer Graphics and Applications*, vol. 21, no. 3, pp. 62–68, 2001.
- [31] J. Sun, Z. Xu, and H. Shum, “Image super-resolution using gradient profile prior,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [32] J. D. van Ouwerkerk, “Image super-resolution survey,” *Image and Vision Computing*, vol. 24, no. 10, pp. 1039–1052, October 2006.
- [33] D. Heckerman, “A tutorial on learning with Bayesian networks,” in *Learning in graphical models*. MIT Press, 1999, pp. 301–354.

- [34] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 721–741, 1984.
- [35] P. Perez, “Markov random fields and images,” *CWI Quarterly*, vol. 11, pp. 413–437, 1998.
- [36] F. R. Kschischang, B. J. Frey, and H. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, 2001.
- [37] S. Lee and J. Paik, “Image interpolation using adaptive fast B-spline filtering,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 1993, pp. 177–180.
- [38] J. L. Carroll and K. D. Seppi, “No-Free-Lunch and Bayesian optimality,” in *IEEE International Joint Conference on Neural Networks Workshop on Meta-Learning*, 2007.
- [39] R. M. Haralick, “Digital step edges from zero crossings of second directional derivatives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, 1984.
- [40] W. L. Buntine, “Chain graphs for learning,” in *Uncertainty in Artificial Intelligence*, 1995, pp. 46–54.
- [41] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Understanding belief propagation and its generalizations,” Mitsubishi Electric Research Laboratories, Tech. Rep., January 2002.
- [42] N. A. Dodgson, “Quadratic interpolation for image resampling,” *IEEE Transactions on Image Processing*, no. 9, pp. 1322–1326, September 1997.
- [43] J. D. van Ouwerkerk, Personal correspondence, Nov. 2008.
- [44] B. E. Bayer, “Color imaging array,” U.S. Patent 3,971,065, 1976.
- [45] M. M. Oliveira, B. Bowen, R. McKenna, and Y. S. Chang, “Fast digital image inpainting,” in *Proceedings of the International Conference on Visualization, Imaging, and Image Processing*, 2001, pp. 261–266.
- [46] M. Mori, “The uncanny valley,” *Energy*, vol. 7, no. 4, pp. 33–35, 1970.