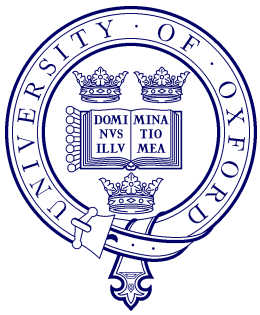
Machine Learning in Multi-frame Image Super-resolution



Lyndsey C. Pickup

Robotics Research Group
Department of Engineering Science
University of Oxford

Michaelmas Term, 2007

Lyndsey C. Pickup
St. Cross College

Doctor of Philosophy
Michaelmas Term 2007

Machine Learning in Multi-frame Image Super-resolution

Abstract

Multi-frame image super-resolution is a procedure which takes several noisy low-resolution images of the same scene, acquired under different conditions, and processes them together to synthesize one or more high-quality *super-resolution* images, with higher spatial frequency, and less noise and image blur than any of the original images. The inputs can take the form of medical images, surveillance footage, digital video, satellite terrain imagery, or images from many other sources.

This thesis focuses on Bayesian methods for multi-frame super-resolution, which use a *prior distribution* over the super-resolution image. The goal is to produce outputs which are as accurate as possible, and this is achieved through three novel super-resolution schemes presented in this thesis.

Previous approaches obtained the super-resolution estimate by first computing and fixing the imaging parameters (such as image registration), and then computing the super-resolution image with this registration. In the first of the approaches taken here, superior results are obtained by optimizing over both the registrations and image pixels, creating a complete *simultaneous* algorithm. Additionally, parameters for the prior distribution are learnt automatically from data, rather than being set by trial and error.

In the second approach, uncertainty in the values of the imaging parameters is dealt with by *marginalization*. In a previous Bayesian image super-resolution approach, the marginalization was over the super-resolution image, necessitating the use of an unfavorable image prior. By integrating over the imaging parameters rather than the image, the novel method presented here allows for more realistic prior distributions, and also reduces the dimension of the integral considerably, removing the main computational bottleneck of the other algorithm.

Finally, a domain-specific image prior, based upon patches sampled from other images, is presented. For certain types of super-resolution problems where it is applicable, this sample-based prior gives a significant improvement in the super-resolution image quality.

Acknowledgements

I owe a lot of thanks to lots of people...

Steve Roberts took me on as an undergrad, and I'd like to thank him for helping me become a researcher, and for all his support in making me better at it.

I'd like to thank Andrew Zisserman for making Computer Vision so awesome all along. He pointed me towards super-resolution when I'd lost my way, and I'm indebted to him for all the time and advice he has offered me ever since.

I'm very grateful to my examiners, Andrew Blake and Amos Storkey, for their constructive observations and discussion of my work; my thesis is much stronger now as a result.

I'd like to thank my parents for taking care of me when I've had a hard time, for letting me go my own way, and for making it possible for me to pursue what I enjoy.

Last (but never least!) I'd like to thank all my friends for keeping me happy and sane! Particular thanks must go to the wonderful members of OUSFG and Taruithorn (the Oxford University Speculative Fiction Group and the Oxford Tolkien Society), and also to the lovely people on my LiveJournal "flist" and elsewhere online, for updating me on the real world once in a while.

This work was funded by a three-year studentship from the Engineering and Physical Sciences Research Council (EPSRC).

Contents

1 Introduction **1**

1.1 How is super-resolution possible? 3

1.2 Challenges 4

1.3 Thesis contributions 6

2 Literature review **10**

2.1 Early super-resolution methods 11

2.1.1 Frequency domain methods 12

2.1.2 The emergence of spatial domain methods 13

2.1.3 Developing spatial-domain approaches 15

2.2 Projection onto convex sets 16

2.3 Methods of solution 18

2.4 *Maximum a posteriori* approaches 19

2.4.1 The rise of MAP super-resolution 20

2.4.2 Non-Gaussian image priors 21

2.4.3 Full posterior distributions 25

2.5 Image registration in super-resolution 26

2.6 Determination of the point-spread function 29
2.6.1 Extensions of the simple blur model 31
2.7 Single-image methods 32
2.8 Super-resolution of video sequences 35
2.9 Colour in super-resolution 36
2.10 Image sampling and texture synthesis 38

3 The anatomy of super-resolution 40
3.1 The generative model 41
3.2 Considerations in the forward model 42
3.2.1 Motion models 43
3.2.2 The point-spread function 44
3.2.3 Constructing $\mathbf{W}^{(k)}$ 45
3.2.4 Related approaches 48
3.3 A probabilistic setting 49
3.3.1 The *Maximum Likelihood* solution 50
3.3.2 The *Maximum a Posteriori* solution 53
3.4 Selected priors used in MAP super-resolution 59
3.4.1 GMRF image priors 59
3.4.2 Image priors with heavier tails 64
3.5 Where super-resolution algorithms go wrong 69
3.5.1 Point-spread function example 69
3.5.2 Photometric registration example 72
3.5.3 Geometric registration example 74
3.5.4 Shortcomings of the model 79

3.6 Super-resolution so far 84

4 The simultaneous approach 86

4.1 Super-resolution with registration 87

4.1.1 The optimization 89

4.2 Learning prior strength parameters from data 90

4.2.1 Gradient descent on α and ν 92

4.3 Considerations and algorithm details 93

4.3.1 Convergence criteria and thresholds 94

4.3.2 Scaling the parameters 94

4.3.3 Boundary handling in the super-resolution image 95

4.3.4 Initialization 97

4.4 Evaluation on synthetic data 100

4.4.1 Measuring image error with registration freedom 100

4.4.2 Registration example 103

4.4.3 Cross-validation example 106

4.5 Tests on real data 109

4.5.1 Surrey library sequence 109

4.5.2 Eye-test card sequence 111

4.5.3 Camera "9" sequence 113

4.5.4 "Lola" sequences 115

4.6 Conclusions 115

5 Integrating over the imaging parameters 119

5.1 Bayesian marginalization 120

5.1.1 Marginalizing over registration parameters 120

5.1.2 Marginalizing over the super-resolution image 126
5.1.3 Implementation notes 128
5.2 Results 130
5.2.1 Butterfly sequence 130
5.2.2 Synthetic eyechart sequence 134
5.2.3 Face sequence 135
5.2.4 Real data 141
5.3 Discussion 142
5.4 Conclusion 145

6 Texture-based image priors for super-resolution 148
6.1 Background and theory 149
6.1.1 Image patch details 153
6.1.2 Optimization details 155
6.2 Experiments 156
6.2.1 Synthetic data 156
6.2.2 Results 158
6.3 Discussion and further considerations 159

7 Conclusions and future research directions 166
7.1 Research contributions 166
7.2 Where super-resolution algorithms go next 168
7.2.1 More on non-parametric image priors 168
7.2.2 Registration models and occlusion 169
7.2.3 Learning the blur 170
7.2.4 Further possibilities with the model 173

7.3 Closing remarks 174

A Marginalizing over the imaging parameters 176

A.1 Differentiating the objective function *w.r.t.* $\mathbf{x}$ 177

A.2 Numerical approximations for $\mathbf{g}$ and $\mathbf{H}$ 183

B Marginalization over the super-resolution image 187

B.1 Basic derivation 188

B.1.1 The objective function 188

B.1.2 The Gaussian distribution over $\mathbf{y}$ 192

B.2 Gradient *w.r.t.* the registration parameters 194

B.2.1 Gradients *w.r.t.* $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ 194

B.2.2 Gradient of the objective function 196

Chapter 1

Introduction

This thesis addresses the problem of image super-resolution, which is the process by which one or more low-resolution images are processed together to create an image (or set of images) with a higher spatial resolution.

Multi-frame image super-resolution refers to the particular case where multiple images of the scene are available. In general, changes in these low-resolution images caused by camera or scene motion, camera zoom, focus and blur mean that we can recover extra data to allow us to reconstruct an output image at a resolution above the limits of the original camera or other imaging device, as shown in Figure 1.1. This means that the super-resolved output image can capture more of the original scene's details than any one of the input images was able to record.

Problems motivating super-resolution arise in a number of imaging fields, such as satellite surveillance pictures and remote monitoring (Figure 1.2), where the size of the CCD array used for imaging may introduce physical limitations on the resolution of the image data. Medical diagnosis may be made more easily or more accuratly if

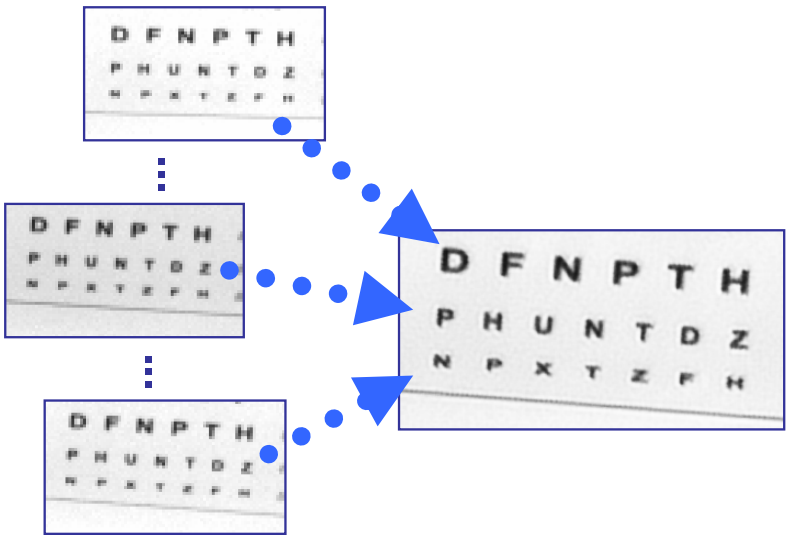


Figure 1.1: **Multi-frame image super-resolution.** The low-resolution frames to the left are processed together and used to create a high-resolution view of the text, seen to the right. Notice that the bottom row of the eyechart letters cannot be read in any of the input images, but is clear in the super-resolved image.

data from a number of scans can be combined into a single more detailed image. A clear, high-quality image of a region of interest in a video sequence may be useful for facial recognition algorithms, car number plate identification, or for producing a "print-quality" picture for the press (Figure 1.3).

Figure 1.2: **An example of super-resolution for surveillance:** several frames such as the low-resolution one shown left can be combined into an output frame (right) where the level of ground detail is significantly increased. Images from [102].

Figure 1.3: **An example of super-resolution in the media:** frame-grabs such as the one of the left can be combined to give a much clearer print-quality image, as on the right. Images from [43].

1.1 How is super-resolution possible?

Reconstruction-based super-resolution is possible because each low-resolution image we have contains pixels that represent subtly different functions of the original scene, due to sub-pixel registration differences or blur differences. We can model these differences, and then treat super-resolution as an inverse problem where we need to reverse the blur and decimation.

Each low-resolution pixel can be treated as the integral of the high-resolution image over a particular blur function, assuming the pixel locations in the high-resolution frame are known, along with the point-spread function that describes how the blur behaves. Since pixels are discrete, this integral in the high-resolution frame is modelled as a weighted sum of high-resolution pixel values, with the *point-spreaad function* (PSF) kernel providing the weights. This image generation process is shown in Figure 1.4.

Each low-resolution pixel provides us with a new constraint on the set of high-resolution pixel values. Given a set of low-resolution images with different sub-pixel registrations with respect to the high-resolution frame, or with different blurs, the set

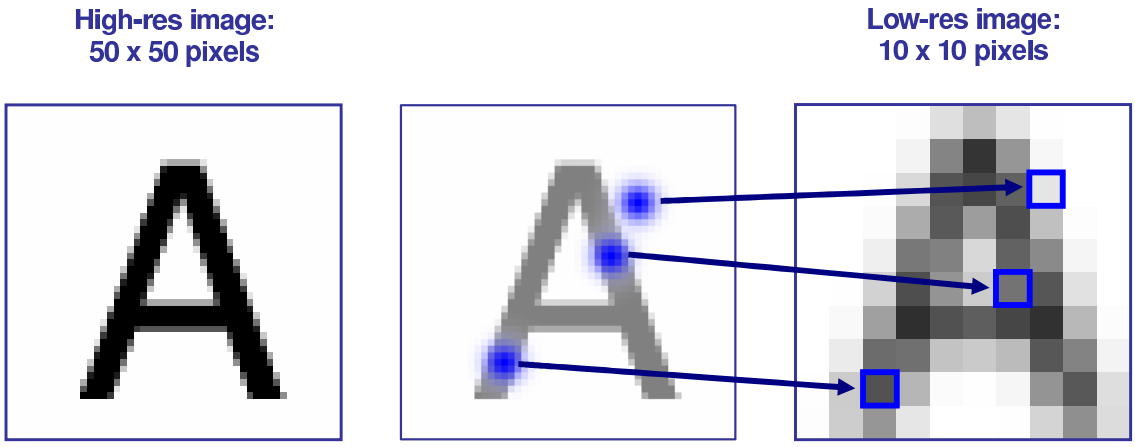


Figure 1.4: **The creation of low-resolution image pixels.** The low-resolution image on the right is created from the high-resolution image on the left one pixel at a time. The locations of each low-resolution pixel are mapped with sub-pixel accuracy into the high-resolution image frame to decide where the blur kernel (plotted as a blue Gaussian in the middle view) should be centred for each calculation.

of constraints will be non-redundant. Each additional image like this will contribute something more to the estimate of the high-resolution image.

In addition to the model of Figure 1.4, however, real sensors also have associated noise in their measurements, and real images can vary in illumination as well as in their relative registrations. These factors must also be accounted for in a super-resolution model, so the full picture of how a scene or high-resolution image generates a low-resolution image set looks more like that of Figure 1.5.

1.2 Challenges

Super-resolution algorithms face a number of challenges in parallel with their main super-resolution task. In addition to being able to compute values for all the super-resolution image pixels intensities given the low-resolution image pixel intensities, a super-resolution system must also be able to handle:

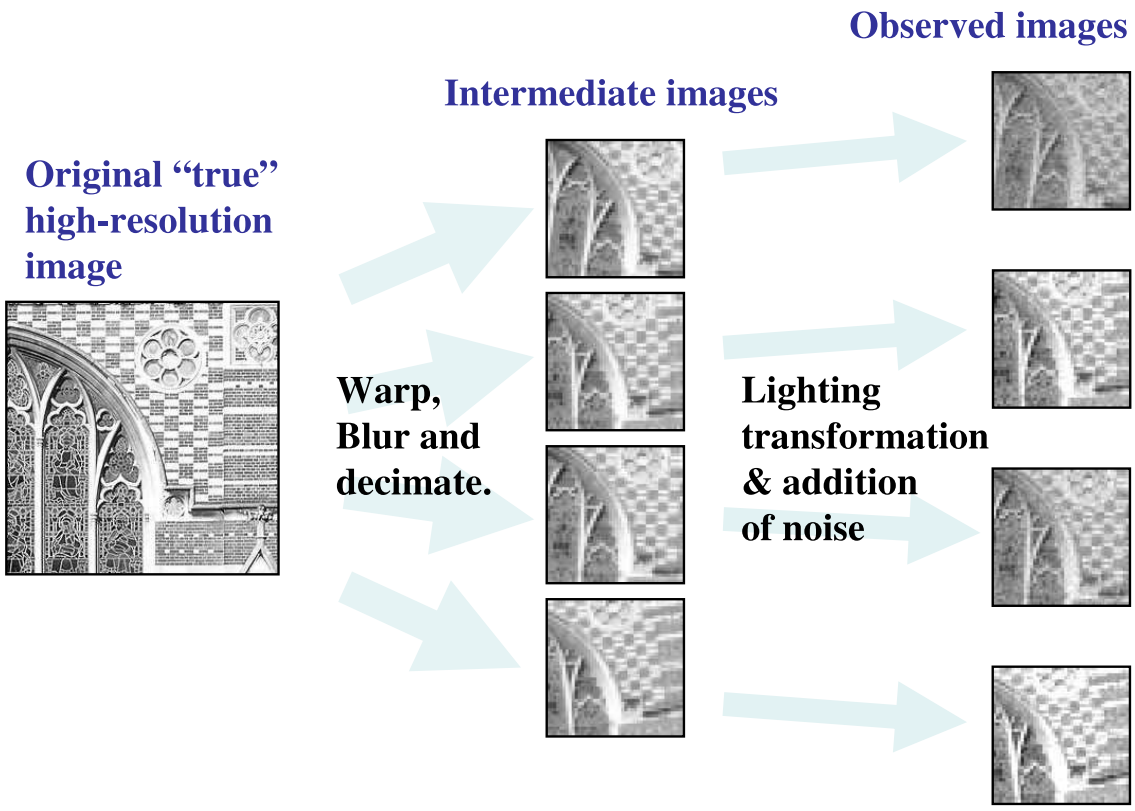


Figure 1.5: **One high-resolution image generates a set of low-resolution images.** Because the images are related by sub-pixel registrations, each observation gives us more additional constraints on the values of the high-resolution image pixel intensities.

- Image registration – small image displacements are crucial for beating the sampling limit of the original camera, but the exact mappings between these images are unknown. To achieve an accurate super-resolution result, they need to be found as accurately as possible.

- Lighting variation – when the images are aligned geometrically, there may still be significant photometric variation, because of different lighting levels or camera exposure settings when the images were captured.

- Blur identification – before the light from a scene reaches the film or camera CCD array, it passes through the camera optics. The blurs introduced in this stage are modelled by a point-spread function. Separating a blur kernel from an image is an extensively-studied and challenging problem known as *Blind Image Deconvolution*. This can be even more challenging if the blur varies spatially across the image.

These three cases are illustrated for some low-resolution image patches in Figure 1.6. While the goal is to compute a high-resolution image (*e.g.* as illustrated in Figure 1.1), the efficacy of any super-resolution approach depends on its handling of these additional considerations as well. Given "good" low-resolution input images, the output from an otherwise successful super-resolution algorithm will still be poor if the registration and blur estimates are inaccurate.

1.3 Thesis contributions

This thesis addresses several aspects of super-resolution by treating it as a machine learning problem.

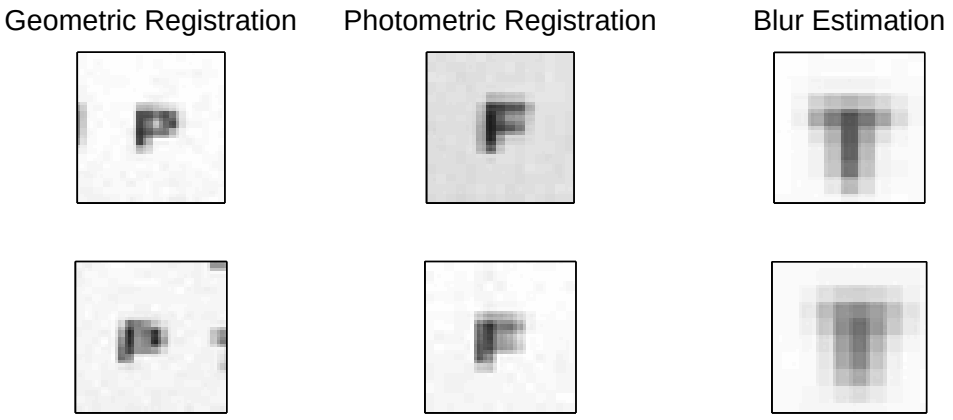


Figure 1.6: **Three of the challenges facing a multi-frame super-resolution algorithm.** Left column: the two instances of the "P" character have different alignments, which we must know accurately for super-resolution to be possible. Middle column: the eyechart in the images has different illumination between two frames, so the photometric registration between the images also has to be estimated. Right column: the image blur varies noticeably between these two input frames, and blur estimation is a difficult challenge.

Chapter 2 gives an overview of the state of super-resolution research, and relevant literature around the field. Notation and a formal framework for reasoning about the super-resolution problem are laid out in Chapter 3, which then goes on to present new intuition about the behaviour of super-resolution approaches, giving particular consideration to uncertainty associated with registration and blur estimation, which is not something that has been addressed to this degree in other discussions of multi-frame image super-resolution.

Chapter 4 describes a *simultaneous* approach for super-resolution, where the geometric and photometric registration challenges are addressed at the same time as the super-resolution image estimate. This allows us to find a more accurate registration than can generally be found between two noisy low-resolution images, and the improvement in the registration estimate carries through to give a more accurate high-resolution image.

A second key point of the simultaneous super-resolution approach, also in Chapter 4, is the automatic tuning of parameters for an image prior at the same time as the super-resolution process. We already know that a well-chosen prior representation can yield a better super-resolution image, so selecting the right shape of prior becomes important, especially because most successful image priors are not easy to normalize, meaning that parameters can be difficult to learn directly.

Chapter 5 goes a step beyond what we show in the preceding chapter; rather than making a point estimate of the unknown registration parameters, we derive a method through which we can integrate them out of the problem in a Bayesian manner. This leads to a direct method of optimizing the super-resolution image pixel values, and this gives improved results which are shown both on synthetic data, where differences are easily quantified with respect to known parameters, and on real image data, which shows that the algorithm is flexible enough to be useful in real-world situations.

Chapter 6 introduces a texture-based prior for super-resolution. Using this prior, *Maximum a Posteriori* super-resolution can achieve much better results on image datasets from highly textured scenes than an algorithm using a more conventional smoothness prior, and this is demonstrated on datasets with a variety of textures. This highlights the importance and power of a well-chosen representation for prior knowledge about the super-resolution image, and also shows how the patch-based methods popular in single-image super-resolution can be brought into the multi-frame setting.

The main contributions are drawn together in Chapter 7, which consolidates the main overarching themes of registration uncertainty and selection of an *appropriate* image prior. This chapter also highlights several promising directions for further

research, and concludes with some final thoughts on the multi-frame image super-resolution problem.

Chapter 2

Literature review

Image super-resolution is a well-studied problem. A comprehensive review was carried out by Borman and Stevenson [12, 13] in 1998, though in the intervening period a wide variety of super-resolution work has contributed to many branches of the super-resolution problem.

There are several popular approaches to super-resolution overall, and a number of different fundamental assumptions to be made, which can in turn lead to very different super-resolution algorithms. Model assumptions such as the type of motion relating the low-resolution input images, or the type of noise that might exist on their pixel values, commonly vary significantly, with many approaches being tuned to particular assumptions about the input scenes and configurations that the authors are interested in. There is also the question of whether the goal is to produce the very best high-resolution image possible, or a passable high-resolution image as quickly as possible.

The formulation of the problem usually falls into one of two main categories:

either *Maximum Likelihood* (ML) methods, *i.e.* those which seek a super-resolution image that maximizes the probability of the observed low-resolution input images under a given model, or *Maximum a Posteriori* (MAP) methods, which make explicit use of prior information, though these are also commonly couched as regularized least-squares problems.

We begin with the historically earliest methods, which tend to represent the simplest forms of the super-resolution ideas. From these grow a wide variety of different approaches, which we then survey, turning our attention back to more detailed model considerations, in particular the ways in which registration and blur estimation are handled in the literature. Finally, we cover a few super-resolution approaches which fit less well into any of the categories or methods of solution listed so far.

2.1 Early super-resolution methods

The field of super-resolution began growing in earnest in the late eighties and early nineties. In the signal processing literature, approaches were suggested for the processing of images in the frequency domain in order to recover lost high-frequency information, generally following on from the work of Tsai and Huang [117] and later Kim *et al.* [67]. Roughly concurrent with this in the Computer Vision arena was the work of Peleg, Keren, Irani and colleagues [58, 59, 66, 87], which favoured the spatial domain exclusively, and proposed novel methods for super-resolution reconstruction.

2.1.1 Frequency domain methods

The basic frequency-domain super-resolution problem of Tsai and Huang [117] or Kim *et al.* [67] looks at the horizontal and vertical sampling periods in the digital image, and relates the continuous Fourier transform of the original scene to the discrete Fourier transforms of the observed low-resolution images. Both these methods rely on the motion being composed purely of horizontal and vertical displacements, but the main motivating problem of processing satellite imagery is amenable to this restriction.

The ability to cope with noise on the input images is added in by Kim *et al.* in [67], and Tekalp *et al.* [110] generalize the technique to cope with both noise and a blur on the inputs due to the imaging process.

Tom and Katsaggelos [113, 115] take a two-phase super-resolution approach, where the first step is to register, deblur and de-noise the low-resolution images, and the second step is to interpolate them together onto a high-resolution image grid. While much of the problem is specified here in the spatial domain, the solution is still posed as a Frequency domain problem. A typical result on a synthetic data sequence from [113] is shown in Figure 2.1, where the zoom factor is 2 and the four input frames are noisy.

Lastly, wavelet models have been applied to the problem, taking a similar overall approach as Fourier-domain methods. Nguyen and Milanfar [83] propose an efficient agorithm based on representing the low-resolution images using wavelet coefficients, and relating these coefficients to those of the desired super-resolution image. Bose *et al.* [16] also propose a method based on *second-generation* wavelets, leading to a fast algorithm. However, while this shows good results even in the presence of high levels of input noise, the outputs still display wavelet-like high frequency artifacts.

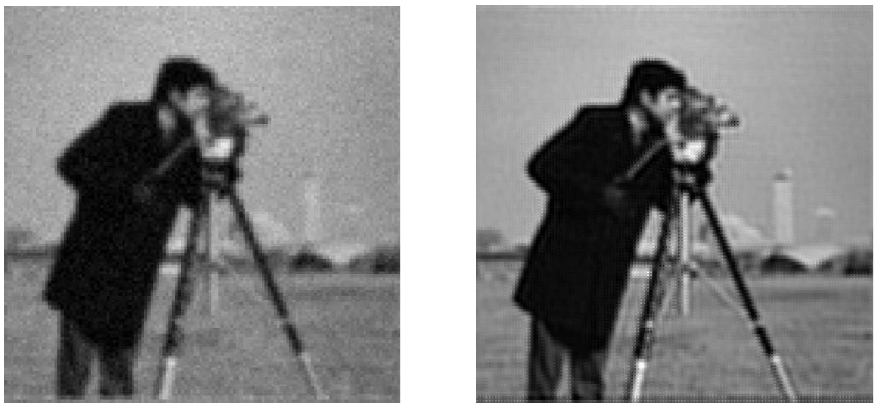


Figure 2.1: **Example of Frequency-domain super-resolution** taken from Tom and Katsaggelos [113]. Left: One of four synthesized low-resolution images. Right: Super-resolved image. Several artifacts are visible, particularly along image edges, but the overall image is improved.

2.1.2 The emergence of spatial domain methods

A second strand in the super-resolution story, developing in parallel to the first, was based purely in the spatial domain. Spatial domain methods enjoy better handling of noise, and a more natural treatment of the image point-spread blur in cases where it cannot be approximated by a single convolution operation on the high-resolution image (*e.g.* when the zoom or shear in the low-resolution image registrations varies across the inputs). They use a model of how each offset low-resolution pixel is derived from the high-resolution image in order to solve for the high-resolution pixel values directly.

The potential for using sub-pixel motion to improve image resolution is highlighted by Peleg *et al.* [87]. They point out that a blurred high-resolution image can be split into 16 low-resolution images at a zoom factor of 4 by taking every 4^{th} pixel in the horizontal and vertical directions at each of the 4×4 different offsets. If all 16 low-resolution images are available, the problem reduces to one of regular image deblurring, but if only a subset of the low-resolution frames are present, there is still

a clear potential for recovering high-frequency information, which they illustrate on a synthetic image sequence.

A method for registering a pair of images is proposed in Keren *et al.* [66]. The registration deals with $2D$ shifts and with rotations within the plane of the image, so is more general than that of [87]. However, their subsequent resampling and interpolation scheme do little to improve the high-frequency information content of the super-resolution image.

Progress was made by Irani *et al.* [58, 59], who used the same registration algorithm, but proposed a more sophisticated method for super-resolution image recovery based on back-projection similar to that used in *Computer Aided Tomography*. They also propose a method for estimating the point-spread function kernel responsible for the image blur, by scanning a small white dot on a black background with the same scanner.

To initialize the super-resolution algorithm, Irani *et al.* take an initial guess at the high resolution image, then apply the *forward model* of the image measurement process to work out what the observed images would be if the starting guess was correct. The difference between these simulated low-resolution images and the real input images is then projected back into the high-resolution space using a *back projection kernel* so that corrections can be made to the estimate, and the process repeated. It is also worth observing that at this point, the algorithms proposed for spatial-domain super-resolution all constitute *Maximum Likelihood* methods.

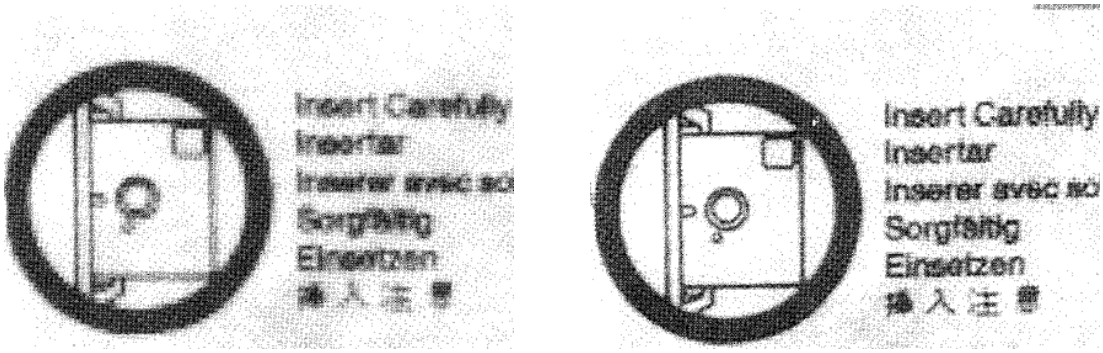


Figure 2.2: **Example of early spatial-domain super-resolution** taken from Irani *et al.* [59] (1991). Left: One of three low-resolution images (70×100 pixels) obtained using a scanner. Right: Super-resolved image. The original images were captured using a scanner.

2.1.3 Developing spatial-domain approaches

Later work by Irani *et al.* [60] builds on their early super-resolution work, though the focus shifts to object tracking and image registration for tracked objects, which allows the super-resolution algorithm to treat some rigid moving objects independently in the same sequence. Another similar method by Zomet *et al.* [123] proposes the use of medians within their super-resolution algorithm to make the process robust to large outliers caused by parallax or moving specularities.

The work of Rudin *et al.* [53, 100] is also based on the warping and resampling approach used by *e.g.* Keren *et al.* or Ur and Gross [66, 118]. Again the transforms are limited to planar translations and rotations. The method performs well, and is simple to understand, but the way the interpolation deals with high frequencies means that the resulting images fail to recreate crisp edges well.

A good example of limiting the imaging model in order to achieve a fast super-resolution algorithm comes from Elad and Hel-Or [33]. The motion is restricted to shifts of integer numbers of high-resolution pixels (though these still represent

sub-pixel shifts in the low-resolution pixels), and each image must have the same noise model, zoom factor, and spatially invariant point-spread function, the latter of which must also be realisable by a block-circulant matrix. These conditions allow the blur to be treated *after* the interpolation onto a common high-resolution frame – this intuition is exactly the same as in [53, 100], but the work of [33] formalizes it and explores more efficient methods of solution in more depth.

2.2 Projection onto convex sets

Projection onto Convex Sets (POCS) is a set-theoretic approach to super-resolution. Most of the published work on this method is posed in an ML framework, though later extensions introduced a super-resolution image prior, making it a MAP method. However, the way in which POCS incorporates prior image constraints, and the forms such constraints take, sets this method apart from later MAP spatial-domain super-resolution methods.

POCS works by finding constraints on the super-resolution image in the form of convex sets of possible values for super-resolution pixels such that each set contains every possible super-resolution image that leads to each low-resolution pixel under the imaging model. Any element in the intersection of all the sets will be entirely consistent with the observed data.

An early ML form of POCS was proposed by Stark and Oskoui in 1989 [105], thus it emerged only just later than the "early" methods covered in Section 2.1. In the same paper that commented on early frequency-domain super-resolution, Tekalp *et al.* [110] also proposed extensions to POCS super-resolution to include a noise model. Later work by Patti *et al.* [85, 86] extended the family of homographies

considered for the image registration, and also allowed for a spatially-varying point-spread function. The linear model relating low- and high-resolution pixels is closely related to that used in later fully-probabilistic super-resolution approaches, and in general POCS it is a strong and successful method.

Like the other super-resolution approaches, the POCS method has been applied and extended in number of different directions: Eren [34] *et al.* extend [85, 86] for cases of multiple rigid objects moving in the scene by using validity or segmentation maps, though it uses pre-registered inputs; Elad and Feuer [29] propose a hybridized ML/MAP/POCS scheme which optimizes a differently-formulated set of convex constraints including the usual ML least-squares formulation along with other POCS-style solution constraints; Patti *et al.*, and also Elad and Feuer [30, 31, 32] use Kalman filtering to pose the problem in a way which is computationally efficient to solve; and Patti and Altunbasak [84] consider a scheme to include a constraint in the POCS method to represent prior belief about the structure of the recovered high-resolution image, making it a MAP estimator.

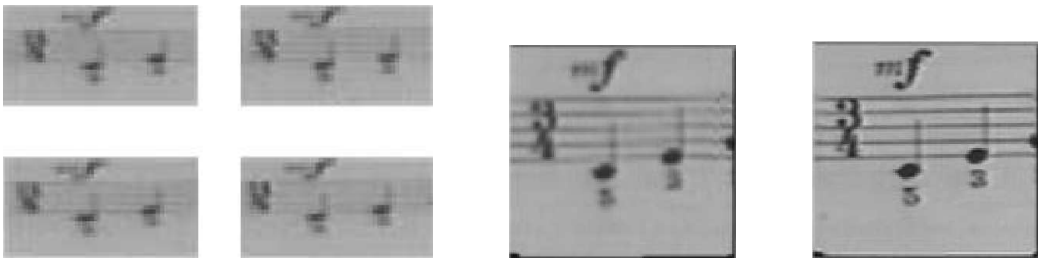


Figure 2.3: **An example of POCS super-resolution** taken from Patti *et al.* [86]. Left: four of twelve low-resolution images, which come from six frames of interlaced low-resolution video captured with a hand-held camcorder. Middle: interpolated approximation to the high-resolution image. Right: super-resolution image found using Patti *et al.*'s POCS-based super-resolution algorithm.

2.3 Methods of solution

One factor motivating the choice of any particular super-resolution method for an application is the cost of the computation associated with the various different methods, and how likely they are to converge to a unique optimal solution.

The early Frequency-domain methods, such as that of Kim *et al.* [67], are posed as simple least-squares problems (*i.e.* of the type $\mathbf{Ax} = \mathbf{b}$), hence are examples of *convex functions* for which a unique global minimum exists. The super-resolution estimate is found using an iterative re-estimation process, *e.g.* gradient descent. In contrast, the projection-based work of Irani *et al.* [58, 59] was reported to yield several solutions depending on the initial image guess, or to oscillate between several solutions rather than converging onto one.

The capabilities of the standard ML estimator are explored by Capel in [21] (see also Section 3 of this thesis), and this is another classic linear system. We can find this spatial ML super-resolution estimate directly by using the pseudoinverse, though this is a computationally expensive operation for large systems of image pixels, so cheaper iterative methods are generally preferred. Since this is another convex problem, the algorithm is guaranteed to converge onto the same global optimum whatever the starting estimate, though a poor initial super-resolution image may mean that many more iterations are needed.

In POCS super-resolution, as in Tekalp *et al.* [110], the high-resolution image is found using another iterative projection technique. The constraints from the low-resolution images form ellipsoids in the space spanned by all the possible high-resolution pixel values. Other constraints, like bounds on acceptable pixel intensity values, can also be added, but will not necessarily be ellipsoids. As long as the

constraints are convex and closed, then projecting onto each of the sets of constraints in turn, cyclically, guarantees convergence to a solution satisfying all the constraints.

2.4 *Maximum a posteriori* approaches

Most super-resolution problems now are either explicit *Maximum a Posteriori* (MAP) approaches, or are approaches that can be re-interpreted as MAP approaches because they use a regularized cost function whose terms can be matched to those of a posterior distribution over a high-resolution image, as the regularizer can be viewed as a type of image prior.

If a prior over the high-resolution image is chosen so that the log prior distribution is convex in the image pixels, and the basic ML solution itself is also convex, then the MAP solution will have a unique optimal super-resolution image. This makes the convexity-preserving priors an attractive choice, though we will touch on various other image priors as well.

In the error-with-regularizer framework, a popular form of convex regularizer is a quadratic function of the image pixels, like $\|\mathbf{Ax}\|_2^2$, for some matrix $\mathbf{A}$ and image pixel vector $\mathbf{x}$. If the objective function is exponentiated, it can be manipulated to give the probabilistic interpretation, because a term like $\exp\left\{-\frac{1}{2}\|\mathbf{Ax}\|_2^2\right\}$ is proportional to a zero-mean Gaussian prior over $\mathbf{x}$ with covariance $\mathbf{A}^T\mathbf{A}$. These are examples of *Gaussian Markov Random Field* (GMRF) priors, because they assume that the distribution of high-resolution pixels can be captured as a multivariate Gaussian. Many types of GMRFs can be formulated, depending on the correlation structure they assume over the high-resolution image.

While simple regularizer/MAP approaches began with GMRF priors, we shall

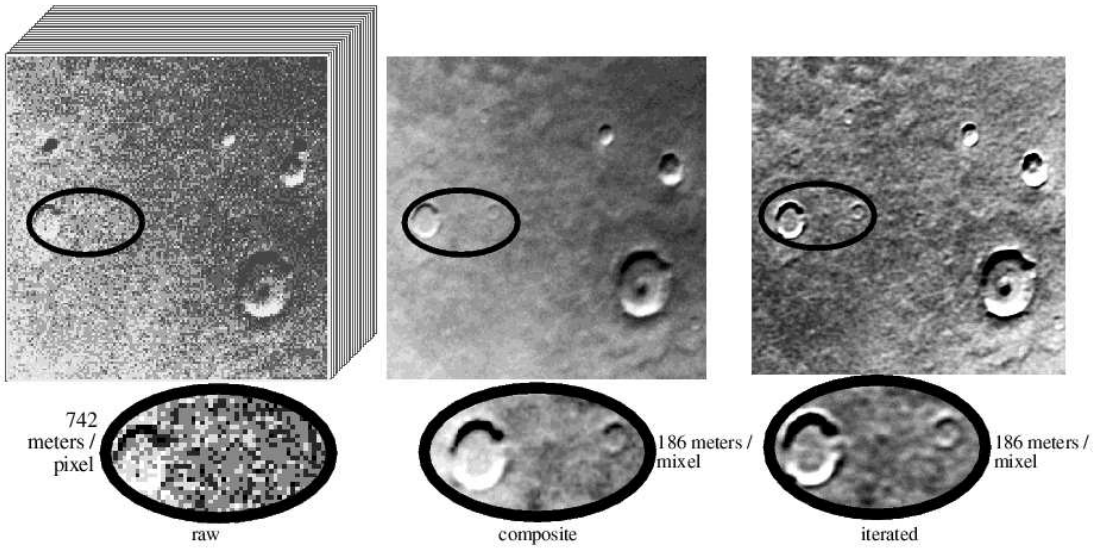


Figure 2.4: **Cheeseman *et al.*'s MAP approach, applied to data from the Viking Orbiter.** Figure taken from [23]. Left: one of 24 input frames. Centre: their "composite image" (same as the *average image* later used by Capel [21]). Right: MAP super-resolution result.

see that these are not always the best choices for image super-resolution, and a number of superior options can be found without having to sacrifice the benefits of having a prior convex in the image pixel values.

2.4.1 The rise of MAP super-resolution

In a powerful early example of MAP super-resolution Cheeseman *et al.* [23] proposed a Bayesian scheme to combine sets of similar satellite images of Mars using a Gaussian prior in a MAP setting. The prior simply assumed each pixel in the high-resolution image was correlated with its four immediate neighbours on the high-resolution grid. Cheeseman *et al.* illustrated that while their prior is a poor model for earth satellite imagery, the Martian images did give rise to intensity difference histograms that are approximately Gaussian. Figure 2.4 shows their result

on Viking Orbiter data.

In another relatively early MAP super-resolution paper, Hardie *et al.* use the L_2 norm of a Laplacian-style filter over the super-resolution image to regularize their MAP reconstruction. Again, this forms a Gaussian-style prior (viewing the Laplacian-style filter as matrix $\mathbf{A}$ in the example above). In addition to finding a MAP estimate for the super-resolution image, Hardie *et al.* also found a maximum likelihood estimate for the shift-only image registration at the same time [56].

2.4.2 Non-Gaussian image priors

Schultz and Stevenson [101, 102] look at video sequences with frames related by a dense correspondence found using a hierarchical block-matching algorithm. Rather than a simple Gaussian prior to regularize the super-resolution image recovery, they use a Huber Markov Random Field (HMRF). The first or second spatial derivative of the super-resolution image is found, and the Huber potential function is used to penalize discontinuities. The Huber function is quadratic for small values of input, but linear for larger values (see Section 3.4.2 and Figure 3.12), so it penalizes edges less severely than a Gaussian prior, whose potential function is purely quadratic. This Huber function models the statistics of real images more closely than a purely quadratic function, because real images contain edges, and so have much heavier-tailed first-derivative distributions than can be modelled by a Gaussian.

Farsiu *et al.* [35, 36, 37, 38] move to the other extreme, and base their image prior on the L_1 norm, referred to as *Total Variation* (TV) priors. This leads to a function that can be computed very rapidly because there is no multiplication involved, unlike the Gaussian and Huber-MRF priors, which involve squared terms. The authors'

Figure 2.5: **Farsiu *et al.*'s fast and robust SR method**, using a TV prior, taken from [37]. Left and Centre: input images 1 and 50 from a 55-image sequence. Right: output of their robust method with a TV prior. Notice that while the method is highly over-constrained, their approach is constructed in such a way as to create a consistent super-resolution image even thought the camel in the image moves independently of the rest of the scene in a few of the frames.

main stated aim is speed and computational efficiency, and they explore several ways of formulating quick solutions by working with L_1 norms, rather than the more common L_2 norms, to solve the super-resolution problem. One set of their results with their "fast and robust" super-resolution method is given in Figure 2.5. Ng *et al.* also use the TV prior [78] for super-resolving digital video sequences, though they also assume Gaussian noise (*i.e.* L_2 errors on the term involving the low-resolution images). They compare their TV regularization to Laplacian regularization (a form of GMRF), and show that the TV regularization gives more favourable signal-to-noise ratios on the reconstructed super-resolution images.

Capel and Zisserman [20] compare the back-projection method of Irani and Peleg to simple spatial-domain ML approaches, and show that these perform much less well on a text image sequence than the HMRF method, and also to a regularized *Total Variation* (TV) estimator. Later they also show that the simple constraint of bounding the image pixel values in the ML solution improves the solution significantly [18] (see also Section 3.3.1 and Figure 3.6 of this thesis).

Other work by Capel and Zisserman [17, 18] considers super-resolution as a second step after image mosaicing, where the image registration (using a full 8-

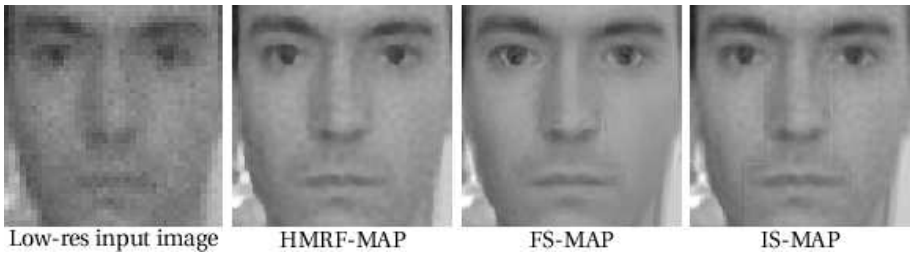


Figure 2.6: **Capel's face super-resolution**, taken from [21]. Left: one of thirty synthetic low-resolution input images at a zoom factor of three. Centre-left: the Huber-MRF reconstruction. Centre-right: MAP reconstruction using PCA faces and a face-space prior. Right: MAP reconstruction using PCA faces and an image-space prior.

degrees-of-freedom projective homography) is carried out in the mosaicing step. Also proposed [18, 21] are high-resolution image priors for dealing specifically with faces by building up a *Principal Component Analysis* (PCA) representation from training data. The face is divided into 6 regions, and *eigenimage* elements learnt for each. This information is then included as priors either by constraining the high resolution image to lie near the PCA subspace, using an optimization algorithm to optimize over all possible super-resolution images, or by constraining the solution to lie within the subspace but making use of the variance estimates for each principal component. In the latter case, the optimization need only be over the component weightings, so the dimension of the problem is considerably smaller than if each high-resolution pixel were considered individually. Some of Capel's results are shown in Figure 2.6. This method is extended by Gunturk *et al.* [55] to carry out face recognition using a face feature vector obtained directly from the low-resolution images using a similar super-resolution process.

As well as faces, a number of authors have concentrated on improving functions to encode prior knowledge for particular super-resolution applications, such as text-reading. A bimodal prior was suggested by Donaldson and Myers [27], because text

images tend to contain mainly black or white pixels. Fletcher *et al.* create a super-resolution system particularly for reading road signs [42], and also use text-specific prior knowledge in their algorithm.

The work of Baker and Kanade [4, 5, 6] offers a careful analysis of the sources of noise and poor reconstruction in the ML case, by considering various forms of the point spread function and their limitations. The MAP formulation of super-resolution is introduced with the suggestion of recognition-based super-resolution. This method works by partitioning the low-resolution space into a set of classes, each of which has a separate prior model. For instance, if a face is detected in the low resolution image set, a face-specific prior over the super-resolution image will be used. The low-resolution classification is made by using a pyramid of multi-scale Gaussian and Laplacian images built up from training data, and the specific priors are imposed by constraining the gradient in the super-resolution image to be as close as possible to that of the closest-matching pyramid feature vector. Example results show that even with inputs of constant intensity, given a face-specific prior, their algorithm will "hallucinate" a face into the super-resolution image.

Various attempts have been made to put bounds on the utility of super-resolution techniques; Baker and Kanade discussed the limits in [4, 6], and more recently Lin and Shum formalized limits for super-resolution under local translation [70], though in this case only for a "box" point-spread function. Most cases for which theoretical limits can be derived are considerably less complex than super-resolution algorithms used in practice.

2.4.3 Full posterior distributions

A more recent trend in MAP super-resolution has been to take the full posterior distribution of the super-resolution image into account, rather than just taking the maximal posterior value as the single image estimate. This is only possible for a subset of the useful priors on the image, and generally Gaussian priors are used in spite of the fact they tend to produce over-smooth results. Finding the covariance of the high-resolution intensity values is useful in making further estimates, especially in estimating imaging parameters.

Woods *et al.* [121] present an iterative EM (*Expectation Maximization*) algorithm for the joint registration, blind deconvolution, and interpolation steps which also estimates the blur, noise, and motion parameters. All the distributions in the problem setup are assumed Gaussian, so the posterior is also Gaussian. The covariance matrix is very large, and the EM update equations require the evaluation of the inverse of this matrix. By using Fourier domain methods (as described in 2.1.1) along with some fairly severe restrictions on their forward model (integer high-resolution-pixel shifts only, with all images sharing the same zoom, blur and decimation), they are able to derive direct updates for both the image noise parameter and the parameter governing the Gaussian prior on the super-resolution image.

Finally, Tipping and Bishop [112] take a different view of super-resolution and marginalize out the unknown super-resolution image to carry out the image registration using exactly the same model structure as the MAP super-resolution problem. Again, a Gaussian form of the prior over the super-resolution image is selected in order to make this possible; a more sophisticated prior like the Huber-MRF would lead to an intractable integral. Unlike [121], Tipping and Bishop do evaluate the spatial-domain matrices, though because of the computational restrictions, they use

very small low-resolution images of 9×9 low-resolution pixels in the part of their algorithm where they perform parameter estimation.

2.5 Image registration in super-resolution

Almost all multi-frame image super-resolution algorithms need to have some estimate of the motion relating the low-resolution input frames. There are a very small number of reconstruction-based super-resolution approaches which do not use motion, but instead employ changes in zoom (Joshi *et al.* [64, 65]) or defocus (Rajagopalan and Chaudhuri [95, 96]) in order to build up the constraints on the high-resolution image pixel intensities necessary to estimate the super-resolution image.

Within the vast majority of methods which do use sub-pixel image motion as the main cue for super-resolution, many authors assume that the image registration problem is solved to a sufficient degree that a satisfactory image registration is available *a priori*, so the registration process requires little to no direct discussion. Other authors describe the registration schemes they use, and of particular interest are the methods which take account of the super-resolution model in order to improve the registration estimates.

A typical method for registering images is to look for interest points in the low-resolution image set, then use robust methods to estimate the point correspondences [41, 116] and compute homographies between images. In an early super-resolution paper, Irani and Peleg [58] used an iterative technique to warp the low-resolution images into each others frames to estimate registration parameters (in this case two shifts and a rotation per image) prior to the super-resolution reconstruction phase. Other authors use hierarchical block-matching algorithms to register their

Figure 2.7: **Hardie *et al.*'s joint registration and super-resolution**, taken from [56]. Left: one of the sixteen infra-red low-resolution image inputs. Right: the high-resolution MAP image estimate.

input images, again as an independent step prior to the actual super-resolution reconstruction [102].

Hardie *et al.* [56] made use of the super-resolution image estimate in order to refine the registration of the low-resolution images. They restricted the motion model to be shifts of integer numbers of high-resolution image pixels, and searched over the grid of possible alignments for each low-resolution image between iterations of their MAP reconstruction algorithm. Example images taken from [56] are shown in Figure 2.7.

Much more recently, Chung *et al.* [25] revisited this idea, using an affine registration. The imaging model they select falls into the trap of resampling the high-resolution image using bilinear interpolation in the high-resolution frame before applying a second operation to average over high-resolution pixel values in order to produce their low-resolution frames. As Capel [18] explains, this can give a poor approximation to the generative model. This is likely to occur when the image registration causes a change in zoom factor, or perspective foreshortening (though Chung *et al.*'s model is only affine). However, even with the limitations of their model, the au-

thors show that considering the registration and super-resolution problems together, a more accurate super-resolution image estimate can be obtained.

Tipping and Bishop [112] searched a continuous space of shift and rotation parameters to find the parameter settings which maximized the *marginal data likelihood*, which takes into account the super-resolution image's interaction with the low-resolution images, even though the exact values of the super-resolution image intensities have themselves been integrated out of the problem in their formulation. Once Tipping and Bishop have estimated the registration, a more conventional MAP reconstruction algorithm is used to estimate the high-resolution image.

All these cases use parametric estimates for the image registration, generally with low numbers of registration parameters per image. At the other end of the spectrum, a number of super-resolution algorithms use optic flow to estimate the motion of each pixel using dense image-to-image correspondences [3, 44, 50, 78, 122].

It is not just the geometric registration which can change between the low-resolution images. Changes in scene illumination or in camera white balance can lead to photometric changes as well. Capel [18] makes a linear photometric registration estimate per image, based on fitting a linear function to pixel correspondences obtained after a geometric registration and warping into the same frame. More recently, Gevrekci and Gunturk [51] have also used a linear photometric model fitted after geometric registration in order to improve the dynamic range of input images acquired over highly variable lighting conditions.

2.6 Determination of the point-spread function

Diffraction-limited optical systems coupled with sensors of various geometries lead to a variety of possible point-spread function shapes [11, 15, 50]. Unfortunately, except for rare super-resolution applications where the imaging hardware can be entirely calibrated ahead of time and is then used to image a static scene, most super-resolution applications face some uncertainty about the exact shape of the point-spread function. In a few cases, the blur can be estimated by measuring the blur on step edges or point light sources in the image, or by taking additional calibration shots [76], but for most applications, determination of the point-spread function is a hard problem.

Blind image deconvolution is a closely-related problem, in which the goal is to take an image blurred according to some unknown kernel, and separate out the underlying source image and the kernel image, using no further inputs. A review of Blind Image Deconvolution is given in [68], though many of the methods are not entirely suitable for real super-resolution applications; for example, *zero sheet separation* is highly sensitive to noise and prone to numerical inaccuracy for large datasets, and the performance of frequency domain methods also suffers when image noise masks the nulls in the image Fourier transform which would otherwise contain information about the blur function. The two most appropriate methods for super-resolution PSF determination are those with the best ability to cope with image noise, and these are Reeves and Mercereau's "*Blur identification by generalized cross-validation*" [99], and Lagendijk *et al.*'s "*Maximum Likelihood image and blur identification*" [69].

Both these approaches have more recent counterparts in image super-resolution.

Cross-validation approaches are widely used to tune model parameters by examining how well a model learned on one subset of the data performs on another disjoint subset, and *Generalized Cross-Validation* (GCV) extends the leave-one-out variant of this (where the data used in evaluation consist of each element alone in turn). The GCV blur identification method was taken up in multi-frame image super-resolution by Nguyen *et al.* [79, 80, 81, 82, 83], who consider the blur identification and image restoration problem as a special case of image super-resolution where the low-resolution pixels all lie on a fixed grid. They also extend the blur-learning method to handle the learning of a prior strength parameter for image super-resolution.

The maximum likelihood approach for blur identification is used both in Tipping and Bishop's "*Bayesian image super-resolution*" [112], and in Abad *et al.*'s "*Parameter estimation in super-resolution image reconstruction*" [1] (and similarly [75]). Both pieces of work begin with a Gaussian data error function and a Gaussian image prior over the super-resolution image, then integrate the super-resolution image out of the problem to leave an equation that can by maximized with respect to the variables of interest. In the case of [112], this is the PSF standard deviation and some geometric image registrations; for [1] this is a selection of PSF values and a parameter governing the strength of the prior on the high-resolution image.

Wang *et al.* [120] use a MAP expression for the PSF parameter, a strong patch-based high-resolution image prior, and importance re-sampling to obtain values for the PSF parameter from its approximate distribution.

Recently, Variational Bayes has been applied both in the field of blind image deconvolution and in super-resolution. Original work by Miskin and MacKay [73] deals with blind source separation and blur kernel estimate. This is extended by Fergus *et al.* [39] to cover a much wider class of photographic images. The de-

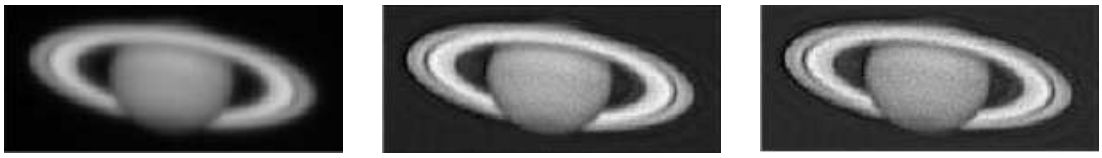


Figure 2.8: **Molina *et al.*'s blind deconvolution using a variational approach to recover the blur and the image.** Left: astronomical image displaying significant blur. Centre and Right: Two possible reconstructions; images taken from [74].

convolution approach of Molina *et al.* [74] is also similar to Miskin and MacKay's deblurring work, and builds on the Variational Bayes approach to blur determination in super-resolution. Figure 2.8 shows a blurred astronomical image, and two possible restored images from [74].

2.6.1 Extensions of the simple blur model

Most of the PSF determination techniques mentioned above relate to the estimation of parameters for blurs from specific families like Gaussians or circular blur functions, and only the variational work [39, 73, 74] builds up a non-parametric pixel-wise representation of the blur.

In between the simple one-parameter blur model and the full pixel-wise representation, several bodies of work model motion blur in addition to Gaussian or circular PSFs, making use of the motion of an input video sequence [7, 24, 50, 98].

The image blur plays a key role in depth-from-defocus, where several differently-defocused images are used to estimate a depth map. The extent of the spatially varying PSF for any point in the image is related to the depth of the object in the scene, and some camera parameters. Given that the depth stays the same while the camera parameters are varied, one can acquire several samples with different parameters and then eliminate the unknowns. The problem can be solved in several

ways, both in the image domain with MRFs and in the spatial frequency domain [95, 96]. The main drawback of this approach is that the available data must include several differently-focused frames, and suitably large changes of focus may not occur in a general low-resolution image sequence.

2.7 Single-image methods

Single-image super-resolution methods cannot hope to improve the resolution by overcoming the Nyquist limit, so any extra detail included in the high-resolution image by such algorithms must come from elsewhere, and prior information about the usual structure of a high-resolution image is consequently an important source.

The single-image super-resolution method proposed by Freeman *et al.* [45] is considered state-of-the-art. It learns the relationship between low- and high-resolution image patches from training data, and uses a Markov Random Field (MRF) to model these patches in an image, and to propagate information between patches using *Belief Propagation*, so as to minimize discontinuities at high-resolution patch boundaries. This work is based on the framework for low-level vision developed by Freeman *et al.* [46, 47, 48, 49].

Tappen *et al.* [109] also use Belief Propagation to produce impressive single-image super-resolution results, this time exploiting a high-resolution image prior based on natural image statistics to improve the image edge quality in the high-resolution outputs. The example in Figure 2.9 shows results from [109] for both this method and the original Freeman *et al.* [45] example-based super-resolution method.

An undisclosed algorithm in a commercial Photoshop plug-in, Genuine Fractals, by LizardTech [43], is also used as a benchmark for single-image super-resolution

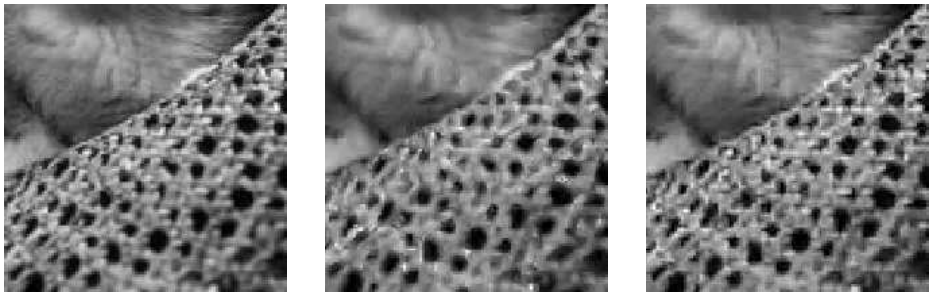


Figure 2.9: **Tappen *et al.*'s sparse derivative prior super-resolution** (taken from [109]). Left: Original image, before being decimated to have half the number of pixels in each dimension. Centre: The result of performing Freeman *et al.*'s single-image super-resolution method on the low-resolution image. Right: the result of using Tappen *et al.*'s sparse derivative prior super-resolution.

(see Figure 1.3). The method is not based on training data from other images, and it is marketed as being able to resize images to six hundred percent of the original without loss of image quality, though [45] states that it suffers from blur in textured regions and fine lines.

A method by Storkey [106] was inspired by work on structural image analysis [107]. Rather than using a model for the blur or image degradation during the imaging procedure, a latent image space is formed, and links are dynamically created between nodes in this space and the low- and high-resolution images. The method is restricted to a fixed zoom factor, and the resulting image has very sharp edges where there is a boundary between regions of the high-resolution image that are influenced by different latent nodes.

Sun *et al.* [108] perform what they term *Image Hallucination* on individual low-resolution images to obtain high-resolution output images in which plausible high-resolution details have been invented based on a "primal sketch" prior constructed from several unrelated high-resolution training images. The results display plausible levels of high-frequency information in the image edges at a zoom factor of three

in each spatial direction, though in some cases appear to suffer from an edge over-sharpening phenomenon similar to that described above.

Jiji *et al.* [62, 63] work with wavelet coefficients as an image representation, rather than using the pixel values in various frequency bands to estimage high-frequency image components as in Freeman *et al.*'s work. Wavelets allow for more localized frequency analysis than global image filtering or fourier transforms, though regularization is required to keep the outputs visually smooth and free from wavelet artifacts. Such single-frame wavelet methods are also improved upon by Temizal and Vlachos [111], who use local corellations in wavelet coefficients to improve their performance.

Finally, Begin and Ferrie [8] suggest an extension to the Freeman *et al.* [45] which deals with the estimation of the PSF *and* tries to take uncertainty in the parameter value into account. The blur estimate itself is made using the Lucy-Richardson algorithm, but the dictionary of patches used in the super-resolution phase is constructed from image pairs where the low-resolution image has been subjected to a range of blurs concentrated around the kernel size found using Lucy-Richardson.

It is important to note that while none of these single-image methods needs to perform registration/motion estimation between multiple inputs, these methods still highlight the great importance of having a good model and of exploiting prior knowledge about working in the image domain.

2.8 Super-resolution of video sequences

A common source of input data for super-resolution algorithms is digital video data, where the goal is either to capture one high-resolution still frame from a low-quality video sequence, or to produce an entire high-definition version of the video itself.

Several authors who tackle this problem simply employ POCS-based super-resolution image reconstruction, or techniques based on iterated back-projection [61], but there are additional constraints which can generally be exploited in video super-resolution. In [14], Borman and Stevenson use a Huber-MRF prior not only in the spatial domain, as described in 2.4.2, but also in the temporal domain at the same time by using a finite difference approximation to the second temporal derivative.

The ability to deal with the full spatio-temporal structure is also exploited by Schechtman *et al.* [103, 104] in considering *Space-Time Resolution*; they consider a space-time volume swept out by a video sequence and seek to increase the resolution along the temporal axis as well as the spatial axes by taking multiple videos as sources. An example of the temporal aspect of their super-resolution is shown in Figure 2.10.

Figure 2.10: **Examples of Schechtman *et al.*'s space-time super-resolution**, which appears in [103] and [104]. Left and centre: two of 18 input images with low frame-rates. Right: an output image which has been super-resolved in time, so that the path of the bouncing basket ball can be seen even more clearly.

Another space-time application is described by Pelletier *et al.* [88], who use a super-resolution model to combine two video streams. High-resolution frames are available at a very low frame rate, low-resolution frames are available at a higher frame rate, and the goal is to synthesize a high-resolution, high-frame-rate output video.

An interesting challenge in video super-resolution is introduced when video input sequences are compressed, *e.g.* by being saved in the MPEG video format. Some methods designed especially to work with digital video take advantage of this by modelling the discrete cosine transform component of the video compression algorithms directly, and by taking into account the quantization that the compression involves [2, 54].

Finally, Bishop *et al.* [10] propose a video super-resolution method for improving on the spatial resolution of a video sequence using a hallucination method which is most closely related to the single-image methods of the previous section. High-resolution information for each patch in the low-resolution frames is suggested by matching medium-resolution candidates from a dictionary, and consistency is enforced both spatially, to make each frame a plausible image, and temporally, to minimize flicker in the resulting video sequence.

2.9 Colour in super-resolution

Early approaches to colour in super-resolution tended to concentrate on how to extend the grey-scale algorithm to handle colour images where each pixel had a three-value colour. Irani *et al.* [59] transformed the colours into a YIQ (luminance and chrominance) representation, and carried out most of the work only on the

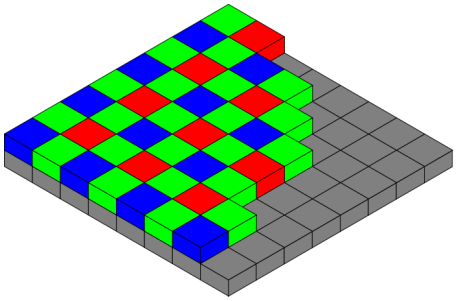


Figure 2.11: **The Bayer pattern on a sensor** (image obtained from Wikipedia under the GNU Free Documentation License).

luminance component. Later, Tom and Katsaggelos [114] used the additional colour channels to improve their image pre-registration step, noting that registering each colour channel separately can give rise to three different registrations between a single pair of images.

More recently, attention has been paid to the way in which pictures are created. When a digital camera images a scene, it typically senses red, green and blue channels spatially separately, giving rise to a colour-interleaved *Bayer* pattern, as shown in Figure 2.11. Most cameras compensate for this as an internal processing step, but algorithms have been developed to treat this *demosaicing* problem in a similar way to the inference of high-resolution pixel values in super-resolution; the offset grids of red, green and blue measurements bear certain similarities to offset low-resolution images, and in the case of Bayer patterns in cameras, the offsets involved are known to a high degree of accuracy because of the geometry of the CCD array.

In [109], Tappen *et al.* use the same natural image prior and belief propagation approach they use for super-resolution on the colour demosaicing problem and show convincing results. Vega *et al.* [119] also follow a probabilistic super-resolution model to improve interpolation from a single Bayer-pattern low-resolution image. For the multi-frame super-resolution setting, a colour model accounting for the Bayer pat-

tern is used by Farsiu *et al.* [37, 35], and again significant improvements are shown when the model takes the colour pattern into account. Note however that the model is only applicable when digital inputs are captured using a Bayer-pattern camera, and subsequently saved in an uncompressed format; general low-resolution colour datasets will not necessarily meet these criteria, *e.g.* if the low-resolution images have been saved as lossy JPEG images, or captured using traditional film cameras (*e.g.* scanned movie film), or with other digital imaging devices which employ different colour filtering.

2.10 Image sampling and texture synthesis

Methods such as Freeman *et al.*'s example-based super-resolution [45], discussed above, work well because the high-resolution information missing from the low-resolution inputs is recovered using data sampled from other images. This patch-sampling method had not previously been used in multiple-image super-resolution because most schemes rely on parametric priors which lead to simple and quick optimization schemes.

One of the contributions of the work presented in this thesis, published initially in [92], is to apply sample-based priors to multiple-image super-resolution. The prior we use is based on the texture synthesis scheme of Efros and Leung [28]. While classical texture synthesis methods work by characterizing statistics, filter responses, or wavelet descriptions of sample textures, then using these to generate new texture samples, the approach of [28] recognizes that the initial texture sample itself can be used to represent a probability model for the texture distribution directly by sampling.

First, the square neighbourhood around the pixel to be synthesized (excluding any regions where there texture has not yet been generated) is compared with similar regions in the sample image, and the closest few (all those within a certain distance) are used to build a histogram of central pixel values, from which a value for the newly-synthesized pixel is drawn. The synthesis proceeds one pixel at a time until all the required pixels in the new image have been assigned a value.

This algorithm has one main variable factor which governs its overall behaviour, and that is its definition of the neighbourhood, which is determined both by the size of the square window used, and by a Gaussian weighting factor applied to that window, so that pixels near to the one being synthesized carry more weight in the matching than those further away.

Even more recent work on patch-based super-resolution has been carried out by Wang *et al.* [120], who use strong sample-based priors to estimate even the image point-spread function in a MAP setting. Multi-image versions of a system similar to that used by Freeman *et al.* in [45] are applied by Rajaram, Das Gupta and colleagues [26, 97], though in these cases the authors constrain the problem by using low-resolution images with a precise regular spacing on a grid, reducing the motion model to be the same constrained translation-only motion model often used in frequency-domain super-resolution approaches.

Chapter 3

The anatomy of super-resolution

This chapter introduces the notation used for the remainder of the thesis. We use a generative model to describe the image formation process in terms of a motion model and camera optics, and we cover some of the standard image priors used in super-resolution. The second purpose of this chapter is to bring out more of the structure of the general multiframe super-resolution problem, in order to motivate the approaches followed subsequently.

The data available to us are low-resolution images of a single scene. In order to work with these, we need to know about their alignment to one another, the extent of any lighting variations between images in the set, and the kind of blur, discretization and sampling that have been introduced during the imaging process. Thus we have a single scene (high-resolution image), linked to several low-resolution images through several geometric and photometric transforms.

3.1 The generative model

A generative model is a parameterized, probabilistic model of data generation, which attempts to capture the forward process by which observed data (in this case low resolution images) is generated by an underlying system (the scene and imaging parameters), and corrupted by various noise processes. This translates to a top-down view of the super-resolution problem, starting with the scene or high-resolution image, and resulting in the low-resolution images, via the physical imaging and noise processes.

For super-resolution, the generative model approach is intuitive, since the goal is to recover the initial scene, and an understanding of the way it has influenced the observed low-resolution images is crucial. The generative model's advantage over classical descriptive models is that it allows us to express a probability distribution directly over the "hidden" high-resolution image *given* the low-resolution inputs, while handling the uncertainty introduced by the noise.

A high-resolution scene $\mathbf{x}$, with N pixels (represented as an $N \times 1$ vector), is assumed to have generated a set of K low-resolution images, where the k^{th} such image is $\mathbf{y}^{(k)}$, and has M pixels. The warping, blurring and subsampling of the scene is modelled by an $M \times N$ sparse matrix $\mathbf{W}^{(k)}$ [18, 112], and a global affine photometric correction results from multiplication and addition across all pixels by scalars $\lambda_1^{(k)}$ and $\lambda_2^{(k)}$ respectively [18]. Thus the generative model for one of the low-resolution images is

$$\mathbf{y}^{(k)} = \lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} + \lambda_2^{(k)}\mathbf{1} + \mathcal{N}\left(\mathbf{0}, \beta^{-1}\mathbf{I}\right), \tag{3.1}$$

where **1** is a vector of ones, and the final term on the right is a noise term consisting of *i.i.d.* samples from a zero-mean Gaussian with precision β, or alternatively with standard deviation σ_N, where $\beta^{-1} = \sigma_N^2$. This generative model is illustrated in Figure 3.1 for two images with differences in the geometric and photometric parameter values, leading to two noticeably different low-resolution images from the same scene.

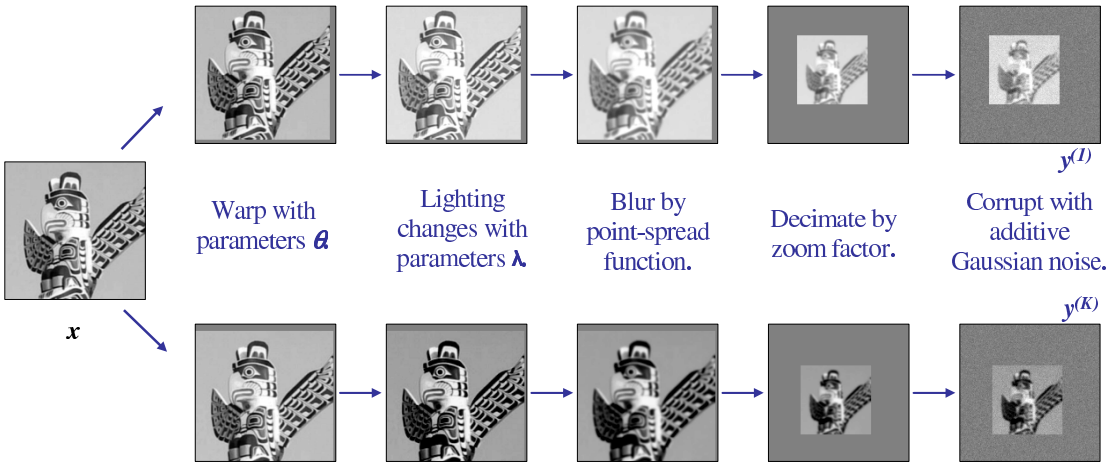


Figure 3.1: **The generative model for two typical low-resolution images.** On the left is the single ground truth scene, and on the extreme right are the two images (in this case $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(K)}$, which might represent the first and last images in a K-image sequence) as they are observed by the camera sensors.

The generative model for a set of low-resolution images was shown in Figure 1.5. Given a set of low resolution images like this, $\{\mathbf{y}^{(k)}\}$, the goal is to recover $\mathbf{x}$, without knowing the values associated with $\{\mathbf{W}^{(k)}, \boldsymbol{\lambda}^{(k)}, \sigma_N\}$.

3.2 Considerations in the forward model

While specific elements of $\mathbf{W}$ are unknown, it is still highly structured, and generally can be parameterized by relatively few values compared to its overall number of

non-zero elements, though this depends upon the type of motion assumed to exist between the input images, and on the form of the point-spread function.

3.2.1 Motion models

Early super-resolution research was predominantly concerned with simple motion models where the registration typically had only two or three degrees of freedom per image, *e.g.* from datasets acquired using a flatbed scanner and an image target. Some models are even more restrictive, and in addition to the 2DoF shift-only registration, the low-resolution image pixel centres are assumed to lie on a fixed integer grid on the super-resolution image plane [56, 83].

Affine (6DoF) and planar projective (8DoF) motion models are generally applicable to a much wider range of common scenes. The 8DoF case is most suitable for modelling planar or approximately planar objects captured from a variety of angles, or for cases where the camera centre rotates about its optical centre, *e.g.* during a panning shot in a movie.

However, some real-world examples, like 3D scenes with translating cameras, will still be a problem for registrations based on homographies, because of multiple motions and occlusion. In these cases, optic flow methods are used to estimate flow of image intensity from one shot to the next in the form of a vector field. Such models use visibility masks to handle occlusions, which are then dealt with as a special case in the super-resolution algorithm.

Most of the work in this thesis is based upon planar projective homographies, since the model applies to a wide variety of interesting problems. Additionally, small image regions and short time sequences can often be adequately approximated using

the 8DoF transform even when the true underlying motion of the overall scene is not completely described.

3.2.2 The point-spread function

To go from a high-resolution image (or a continuous scene), to a low-resolution image, the function representing the light levels reaching the image plane of the low-resolution image is convolved with a *point spread function* (PSF) and sampled at discrete intervals to represent the low-resolution image pixels. This point spread function can be decomposed into factors representing the blurring caused by camera optics and the spatial integration performed by a CCD sensor [6].

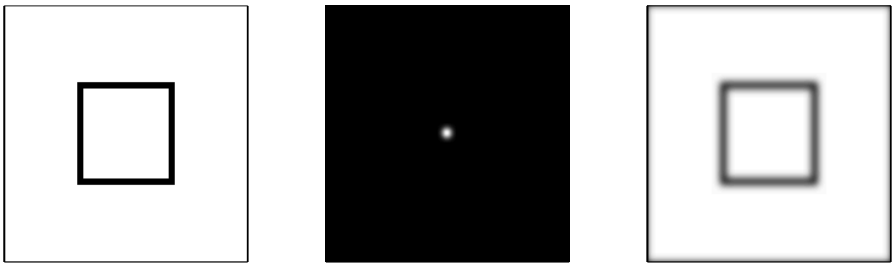


Figure 3.2: **Point-spread function example.** Left: an example image as it reaches the camera. Centre: A 2D Gaussian modelling the camera's point-spread function. Right: The resulting observed image.

Generally, the PSF is approximated by a simple parametric function centred on each low-resolution pixel: the two most common are an isotropic 2D Gaussian with a covariance $\sigma^2_{PSF}\mathbf{I}$, or a circular disk (top-hat function) with radius r_{PSF}. Figure 3.2 illustrates the blurring effect of a Gaussian PSF in the plane of the low-resolution image.

3.2.3 Constructing $\mathbf{W}^{(k)}$

In Chapter 1 we saw that each low-resolution image pixel can be created by dropping a blur kernel into the high-resolution scene and taking the corresponding weighted sum of the pixel intensity values. The centre of the blur kernel is given by the location of the centre of low-resolution pixel when its location is mapped into the frame of the high-resolution image. This means that the i^{th} row in $\mathbf{W}^{(k)}$ represents the kernel for the i^{th} low-resolution image pixel over the whole of the high-resolution image, and $\mathbf{W}^{(k)}$ is therefore sparse, because pixels far from the kernel centre should not have significantly non-zero weights.

Dropping the superscript (k) part of the notation for clarity, we can formalize the construction of each row of $\mathbf{W}$ in the case of a simple similarity transform with an isotropic Gaussian PSF as

$$W_{ij} = \frac{\widetilde{W_{ij}}}{\sum_{j'} \widetilde{W_{ij'}}} \tag{3.2}$$

$$\widetilde{W_{ij}} = \exp\left\{-\frac{\|\mathbf{v}_j - \mathbf{u}'_i\|_2^2}{2\sigma^2}\right\} \tag{3.3}$$

where $\mathbf{u}'_i$ is the 2D location of the i^{th} pixel's centre when projected into the frame of the high-resolution image, and $\mathbf{v}_j$ is the 2D location of the j^{th} high-resolution pixel in its own frame. In (3.3), the point-spread standard deviation σ is expressed in terms of the size of a high-resolution pixel as in [112], but from now on, we will measure the PSF covariance in terms of low-resolution image pixels, because using high-resolution pixels gives a measurement which is dependent on zoom factor, and which can vary and deviate from its isotropy under various types of warping (*e.g.* perspective), as we shall see in a moment.

For a homography $\mathbf{H}$ parameterized by vector $\boldsymbol{\theta}$, the motion model gives us

$$\mathbf{u}'_i = \mathbf{H}(\boldsymbol{\theta})\,\mathbf{u}_i \tag{3.4}$$

where $\mathbf{u}_i$ is the pixel centre in the low-resolution image, and both $\mathbf{u}'_i$ and $\mathbf{u}_i$ are in *homogeneous coordinates* (see *e.g.* [57]). $\mathbf{H}$ is some 3×3 projective matrix which is a function of the vector $\boldsymbol{\theta}$; note that we adopt the convention that the registrations are always represented as a mapping *from* the frame of the low-resolution image *into* the frame of the super-resolution image.

For motion models more complicated than a similarity transform (horizontal and vertical shift, rotation and zoom), the simple derivation for a row of $\mathbf{W}^{(k)}$ is inadequate, because the shape of the PSF when projected from the low- to high-resolution frames undergoes the same warping function specified in the motion model, as illustrated for projective transforms in Figure 3.3.

We assume that the local deformation of any given blur kernel in the high-resolution image is approximately affine, even for the projective case, so that an isotropic Gaussian PSF becomes a general 2D Gaussian kernel with covariance $\mathbf{C}_{\mathrm{PSF}}$ according to

$$\mathbf{C}_{\mathrm{PSF}} \;=\; \sigma_{\mathrm{PSF}}^{-2}\,(\nabla\mathbf{H})\,\mathbf{I}_{2\times 2}\,(\nabla\mathbf{H})^T\,, \tag{3.5}$$

where $\nabla\mathbf{H}$ is the Jacobian of the transform evaluated at the image of the low-resolution pixel centre in the high-resolution frame. This means that $\mathbf{C}_{\mathrm{PSF}}$ is effectively a function of the registration parameters $\boldsymbol{\theta}$ and of the low-resolution pixel

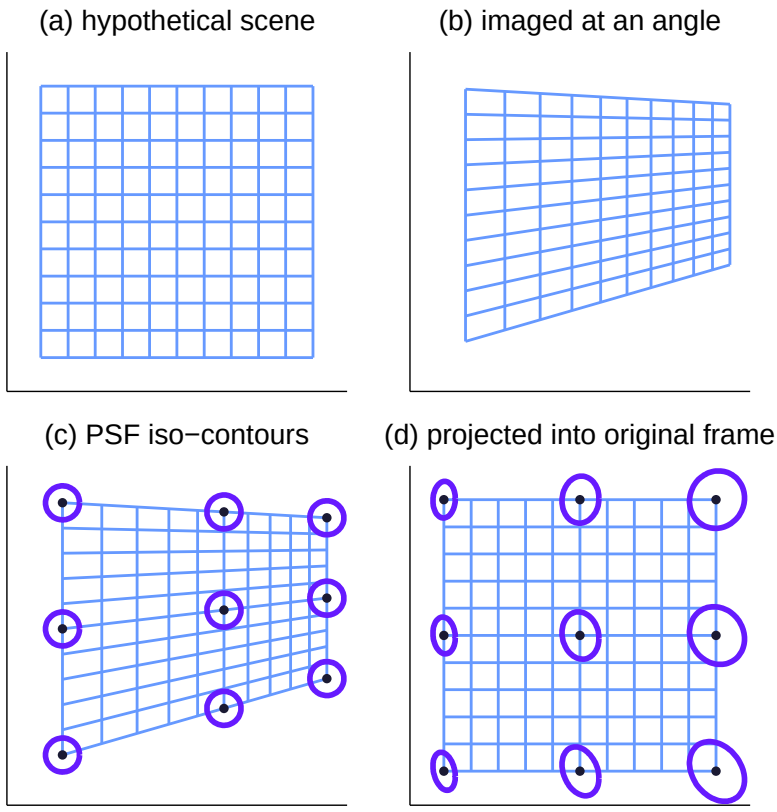


Figure 3.3: **The PSF kernels under projective transformation.** (a) A grid representing a hypothetical scene. (b) The scene is imaged at an angle, so perspective effects are observed, and grid cells to the right appear smaller than those to the left of the image. (c) The scene is imaged with a constant isotropic PSF, so each PSF kernel in the *image* is the same – note that all circles plotted onto this image are exactly the same size. (d) When projected back into the original frame, not only do the kernels change shape, but the change in size and area varies spatially as well.

location, $\mathbf{u}_i$. Each row of $\mathbf{W}$ is now

$$\widetilde{W_{ij}} \;=\; \exp\left\{-\frac{1}{2}\left(\mathbf{v}_j - \mathbf{u}'_i\right)^T \mathbf{C}_{\mathrm{PSF}}\left(\mathbf{v}_j - \mathbf{u}'_i\right)\right\}. \tag{3.6}$$

Again, anything over around three standard deviations is assumed to be zero, creating a sparse set of pixel weights. These are normalized to sum to unity and vectorized to form the i^{th} row of $\mathbf{W}^{(k)}$ exactly as in (3.2) for the simple case above.

3.2.4 Related approaches

In Capel's work [18], a bilinear interpolation step is also incorporated in order to approximate a continuous smooth PSF from the discretized values. However, for anything up to a similarity transform, this is equivalent to convolving the PSF kernel with a truncated Gaussian kernel, and in other cases, we would expect the effect to be so barely different that it justifies removing the bilinear step to simplify the algorithm.

A more detailed and computationally costly algorithm for finding a matrix $\mathbf{W}^{(k)}$ for an arbitrary motion model and PSF is given in [11]. In addition to the possibility of a PSF which is not radially symmetric, the algorithm does not make any local affine assumption. Since the support for a single blur mask projected from one point in the low-resolution image may undergo perspective warping, the area associated with any pair of neighbouring low-resolution pixels will not necessarily be the same (as they would in the affine approximation), so the Jacobian of the transform must be considered *at each PSF kernel pixel individually*, and the relative weights adjusted accordingly. Again, this adds greatly to the computational cost of

finding a particular $\mathbf{W}^{(k)}$ matrix without improving the accuracy of the model very significantly.

3.3 A probabilistic setting

From the generative model given in (3.1) and the assumption that the noise model is Gaussian, the likelihood of a low-resolution image $\mathbf{y}^{(k)}$, given the high-resolution image $\mathbf{x}$, geometric registration $\boldsymbol{\theta}^{(k)}$ and photometric registration $\boldsymbol{\lambda}^{(k)}$, may be expressed

$$p\left(\mathbf{y}^{(k)} \left| \mathbf{x}, \boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)} \right.\right) = \left(\frac{\beta}{2\pi}\right)^{\frac{M}{2}} \exp\left\{-\frac{\beta}{2}\left\|\mathbf{y}^{(k)} - \lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} - \boldsymbol{\lambda}_2^{(k)}\right\|_2^2\right\}, \quad (3.7)$$

where $\mathbf{W}^{(k)}$ is a function of the PSF and of $\boldsymbol{\theta}^{(k)}$.

It can be helpful to think in terms of the *residual* errors, where the residual refers to the parts of the data (in this case our low-resolution images) which are not explained by the model (*i.e.* the high-resolution estimate), given values for all the imaging parameters. We define the k^{th} residual, $\mathbf{r}^{(k)}$, to be

$$\mathbf{r}^{(k)} = \mathbf{y}^{(k)} - \lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} - \boldsymbol{\lambda}_2^{(k)}. \quad (3.8)$$

Using this notation, the compact form of the data likelihood for the whole low-resolution dataset may be written

$$p\left(\left\{\mathbf{y}^{(k)}\right\} | \mathbf{x}, \left\{\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}\right\}\right) = \left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}} \exp\left\{-\frac{\beta}{2}\sum_{k=1}^{K}\left\|\mathbf{r}^{(k)}\right\|_2^2\right\}. \quad (3.9)$$

3.3.1 The *Maximum Likelihood* solution

The *Maximum Likelihood* (ML) solution to the super-resolution problem is simply the super-resolution image which maximizes the probability of having observed the dataset,

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \underset{\mathbf{x}}{\operatorname{argmax}} \left(p \left(\{\mathbf{y}^{(k)}\} | \mathbf{x}, \{\boldsymbol{\theta}^{(k)} \boldsymbol{\lambda}^{(k)}\} \right) \right). \tag{3.10}$$

If all other parameters are known, $\hat{\mathbf{x}}_{\mathrm{ML}}$ can be computed directly as the *pseudoinverse* of the problem. Neglecting the photometric parameters for the moment, if $\mathbf{y}^{(k)} = \mathbf{W}^{(k)}\mathbf{x} + \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I})$, then the pseudoinverse would be

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \left(\mathbf{W}^T\mathbf{W}\right)^{-1}\mathbf{W}^T\mathbf{y}, \tag{3.11}$$

where $\mathbf{W}$ is the $KM \times N$ stack of all K of the $\mathbf{W}^{(k)}$ matrices, and $\mathbf{y}$ is the $KM \times 1$ stack of all the vectorized low-resolution images. Re-introducing the photometric components gives

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \left(\sum_{k=1}^{K} \lambda_1^{(k)2} \mathbf{W}^{(k)T} \mathbf{W}^{(k)} \right)^{-1} \left[\sum_{k=1}^{K} \lambda_1^{(k)} \mathbf{W}^{(k)T} \left(\mathbf{y}^{(k)} - \boldsymbol{\lambda}_2^{(k)} \right) \right]. \tag{3.12}$$

Thus we can solve for $\hat{\mathbf{x}}_{\mathrm{ML}}$ directly if we know $\left\{\mathbf{W}^{(k)}, \boldsymbol{\lambda}^{(k)}\right\}$ and the PSF. This can be a time-consuming process if the $\mathbf{W}$ matrices are large or have many non-zero elements, and if the matrix $\mathbf{W}^T\mathbf{W}$ is singular (*e.g.* when $KM < N$) the direct inversion is problematic. Instead, the ML solution can be found efficiently using a gradient descent algorithm like Scaled Conjugate Gradients (SCG) [77], by

minimizing the relevant terms in the negative log likelihood:

$$\mathcal{L} = \frac{1}{2}\sum_{k=1}^{K}\left\|\mathbf{y}^{(k)} - \lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} - \boldsymbol{\lambda}_2^{(k)}\right\|_2^2 \tag{3.13}$$

$$= \frac{1}{2}\sum_{k=1}^{K}\left\|\mathbf{r}^{(k)}\right\|_2^2 \tag{3.14}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{x}} = \sum_{k=1}^{K} -\lambda_1^{(k)}\mathbf{W}^{(k)T}\mathbf{r}^{(k)}. \tag{3.15}$$

When this is initialised with a reasonable estimate of the super-resolution image, this scheme can be used to improve it iteratively, even when $KM < N$.

Note that $\mathcal{L}$ is essentially a quadratic function of $\mathbf{x}$, so this problem is *convex*. A unique global minimum exists, and gradient-descent methods (which include SCG) can find it given enough steps. Typically we might need up to N steps (where there are N pixels in the super-resolution image) to solve exactly for $\mathbf{x}$, but generally far fewer iterations are needed to obtain a good image. Using SCG, small super-resolution images (under 200×200 pixels) tend to require fewer than 50 iterations before the super-resolution image intensity values change by less than a grey level per iteration.

The ML solution in practice

Unfortunately, ML super-resolution is an ill-conditioned problem whose solution is prone to corruption by very strong high-frequency oscillations. A set of synthetic *Keble* datasets[1] is introduced in Figure 3.4, where the number of images and the amplitude of the additive Gaussian noise is varied. The noise amplitude is measured

[1]The ground truth image shows a section of brick wall and part of a stained-glass window from Keble College, Oxford.

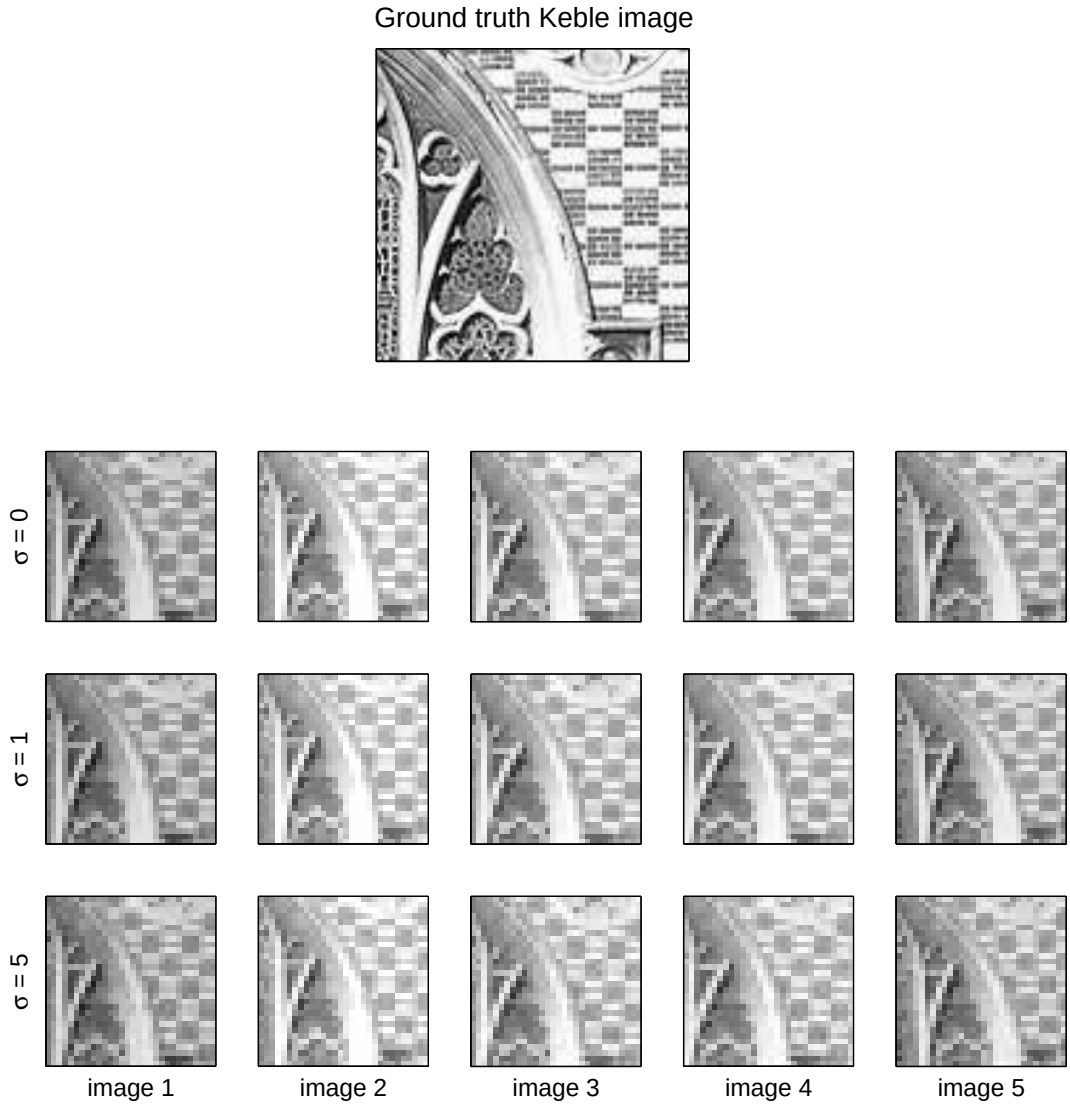


Figure 3.4: **The synthetic Keble dataset.** Top: ground truth Keble image. Below: low-resolution images generated according to the forward model, with a Gaussian PSF of *std* 0.4 low-resolution pixels, and a zoom factor of 4. The ten registration parameters per image (8DoF projective transform plus linear photometric model) were chosen randomly from a suitable uniform distribution, and 64 different registrations were used. Each column represents a different registration vector, and each row represents a different level of the Gaussian *i.i.d.* noise added to the low-resolution images, measured in grey levels, assuming an 8-bit image (*i.e.* 256 grey levels).

here in *grey levels*, with one grey level being one 255^{th} of the intensity range (*e.g.* assuming 8-bit images, which have 256 possible intensity values per pixel).

The ML super-resolutions for some of these are shown in Figure 3.5, and as the noise level increases, the scene information in many super-resolution images is quickly swamped by these oscillation patterns. Here pixels are represented by values in the range $\left[-\frac{1}{2}, \frac{1}{2}\right]$, but the ML noise pattern has values many orders of magnitude higher than this (though rounded to $\pm\frac{1}{2}$ for viewing purposes). For a very low number of images, the problem is underconstrained, and little of the noise pattern is seen. For 4 to 16 images (at this zoom factor of 4), the noise pattern dominates the output severely at the higher input image noise levels. As the number of images increases further, the noise patterns begin to disappear again because the problem becomes much better constrained.

In Figure 3.6, the same datasets are super-resolved, again using the ML cost function, but with values bounded to be in the range $\left[-\frac{1}{2}, \frac{1}{2}\right]$, so that the intensities cannot grow as they do in the standard ML case. We can see that the image results are significantly better, but that the images found at the higher noise levels still appear grainy.

3.3.2 The *Maximum a Posteriori* solution

A prior over $\mathbf{x}$ is usually introduced into the super-resolution model to avoid solutions which are subjectively very implausible to the human viewer. Section 2.4 introduced the *Maximum a Posteriori* (MAP) approach, which we explain here in terms of the generative model and its probabilistic interpretation. We then go on to cover a few of the general image priors commonly selected for image super-resolution

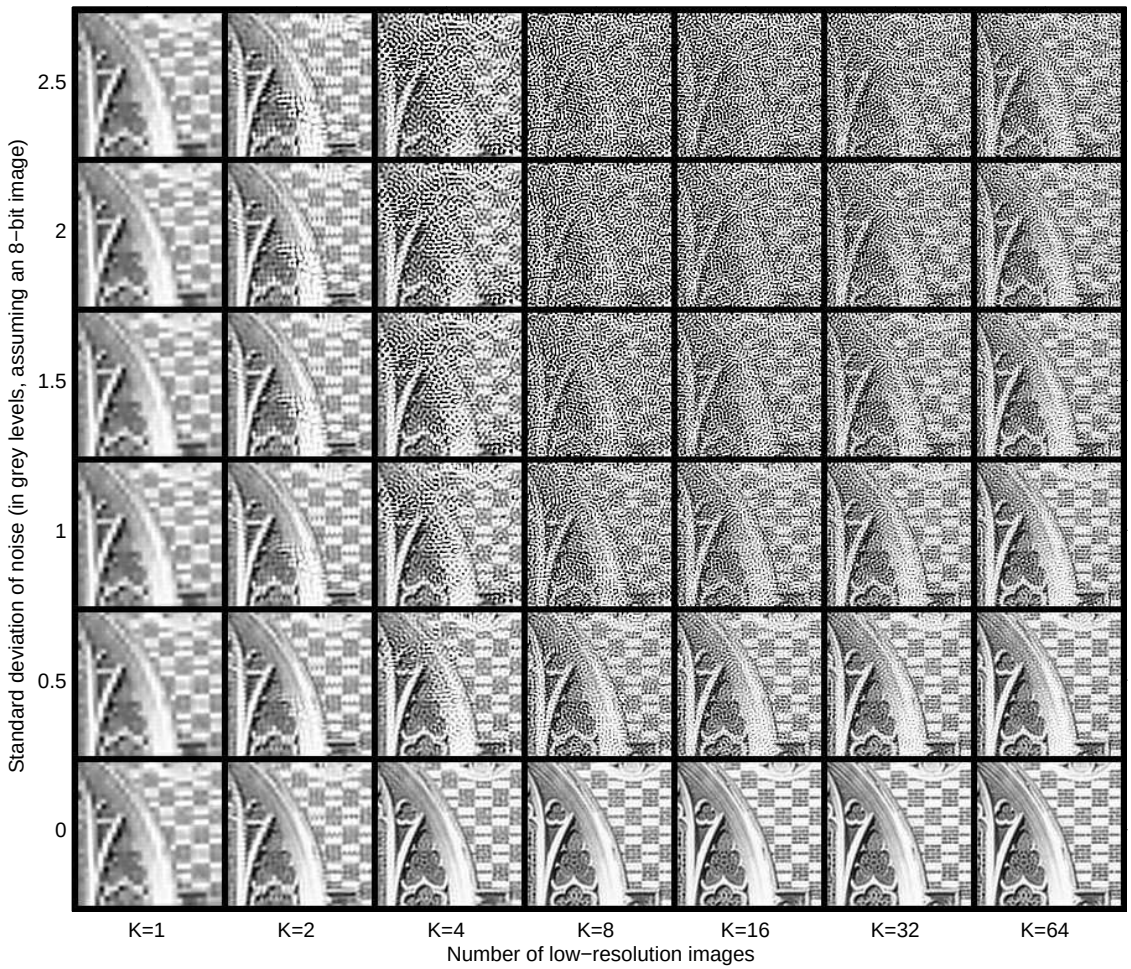


Figure 3.5: **The ML super-resolution estimate.** Synthetic datasets with varying numbers of images and varying levels of additive Gaussian noise were super-resolved using the ML algorithm. If the number of input images is small, then the system of equations is underconstrained, and the ML solution tends to be reasonably free from the characteristic "chequer-board" pattern, but also lacks much of the rest of the high-frequency information (see the first two columns). When the input images have not been corrupted by noise, the outputs are generally of the same high quality (see bottom row). The problem of ill-conditioning is most obvious when there are intermediate numbers of images, and increased noise, as seen in the top-middle and top-middle-right regions of the set of super-resolution images.

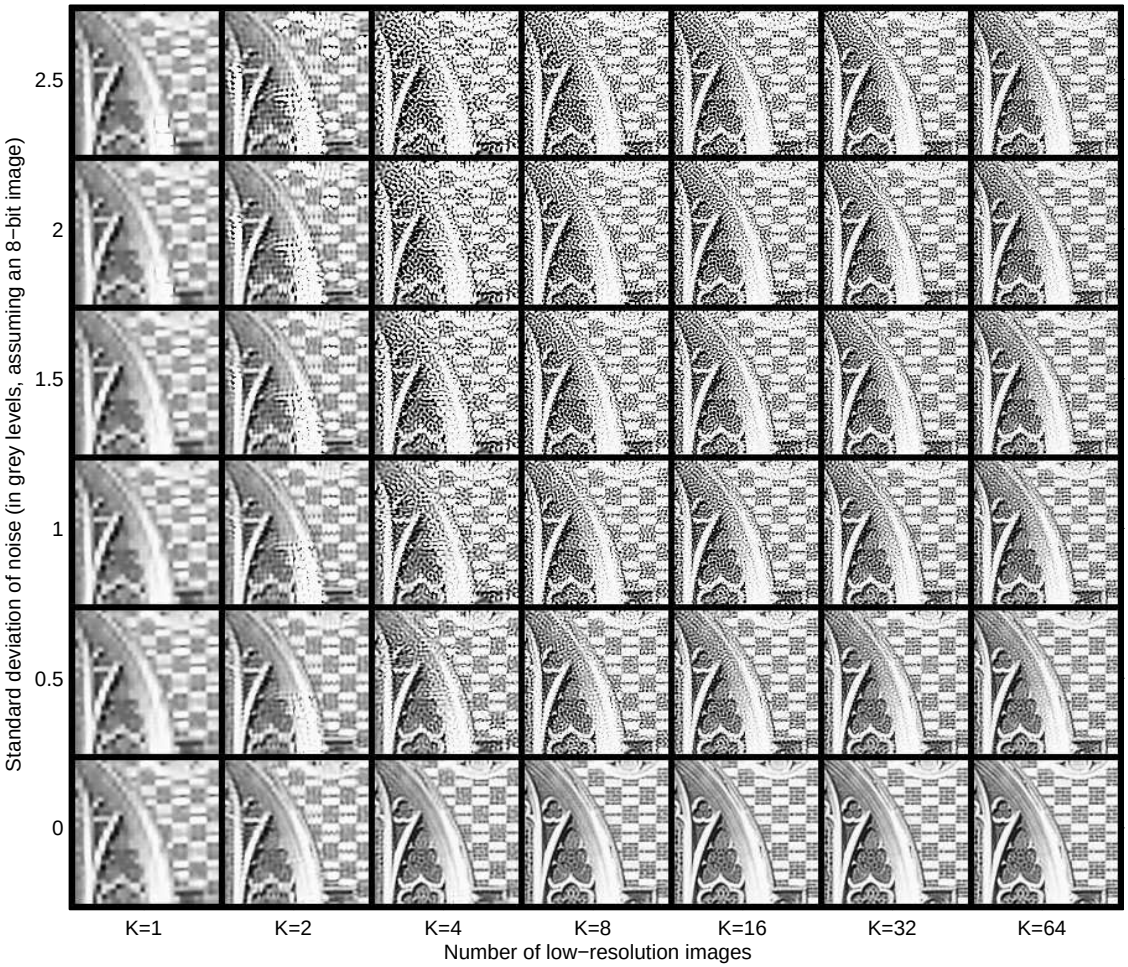


Figure 3.6: **The bounded ML super-resolution estimate.** These images were were created in exactly the same way as those in Figure 3.5, except that here the pixel values were constrained to lie in the interval $[-1/2, 1/2]$. More of the outline of the window in the scene is visible in the top-middle section of these results than in the set of results without any pixel value constraints. However, the resulting super-resolution images in this problem region are still noisy and unnatural-looking.

in the next section.

The MAP estimate of the super-resolution image comes about by an application of Bayes' theorem,

$$p(x|d) \;=\; \frac{p(d|x)p(x)}{p(d)}. \tag{3.16}$$

The left hand side is known as the *posterior* distribution over x, and if d (which in this case might represent our observed data) is held constant, then $p(d)$ may be considered as a normalization constant. It is, however worth noting that it is also the case that

$$p(d) \;=\; \int p(d|x)p(x)\mathrm{dx}. \tag{3.17}$$

Applying these identities to the super-resolution model, we have

$$p\left(\mathbf{x}|\left\{\mathbf{y}^{(k)},\boldsymbol{\theta}^{(k)},\boldsymbol{\lambda}^{(k)}\right\}\right) \;=\; \frac{p\left(\left\{\mathbf{y}^{(k)}\right\}|\mathbf{x},\left\{\boldsymbol{\theta}^{(k)},\boldsymbol{\lambda}^{(k)}\right\}\right)p(\mathbf{x})}{p\left(\{\mathbf{y}^{(k)}\}\,|\,\{\boldsymbol{\theta}^{(k)},\boldsymbol{\lambda}^{(k)}\}\right)} \tag{3.18}$$

If we again assume that the denominator is a normalization constant in this case — it is not a function of $\mathbf{x}$ — then the MAP solution, $\mathbf{x}_{\mathrm{MAP}}$, can be found by maximizing the numerator with respect to $\mathbf{x}$, giving

$$\hat{\mathbf{x}}_{\mathrm{MAP}} = \underset{\mathbf{x}}{\mathrm{argmax}}\, p(\{\mathbf{y}^{(k)}\}|\mathbf{x},\{\boldsymbol{\theta}^{(k)}\})p(\mathbf{x}). \tag{3.19}$$

We take the objective function $\mathcal{L}$ to be the negative log of the numerator of (3.18),

and minimize $\mathcal{L}$ with respect to $\mathbf{x}$. The objective function and its gradient are

$$\mathcal{L} = -\log\left(p(\mathbf{x})\right) + \frac{\beta}{2}\sum_{k=1}^{K}\left\|\mathbf{r}^{(k)}\right\|_2^2 \tag{3.20}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{x}} = \frac{\partial}{\partial\mathbf{x}}\left[-\log\left(p(\mathbf{x})\right)\right] - \sum_{k=1}^{K}\lambda_1^{(k)}\mathbf{W}^{(k)T}\mathbf{r}^{(k)}. \tag{3.21}$$

Of course, in order to solve this, we still require a form for the image prior $p(\mathbf{x})$.

In general we want the prior to favour smoother solutions than the ML approach typically yields, so it is usual for the prior to promote smoothness by penalizing excessive gradients or higher derivatives. Log priors that are *convex* and *continuous* are desirable, so that gradient-descent methods like SCG [77] can be used along with (3.20) and (3.21) to solve for $\mathbf{x}$ efficiently. A least-squares-style penalty term for image gradient values leads to a Gaussian image prior which gives a closed-form solution for the super-resolution image. However, natural images *do* contain edges where there are locally high image gradients which it is undesirable to smooth out.

Figure 3.7 shows the improvement in super-resolution image estimates that can be achieved using a very simple prior on the super-resolution image, $\mathbf{x}$. The super-resolution images were reconstructed using exactly the same input datasets as Figures 3.5 and 3.6, but this time a Huber prior was used on image gradients, and all of the noise present in the ML solutions is gone. We will consider the forms and relative benefits of a small selection of popular MAP super-resolution priors in the next section.

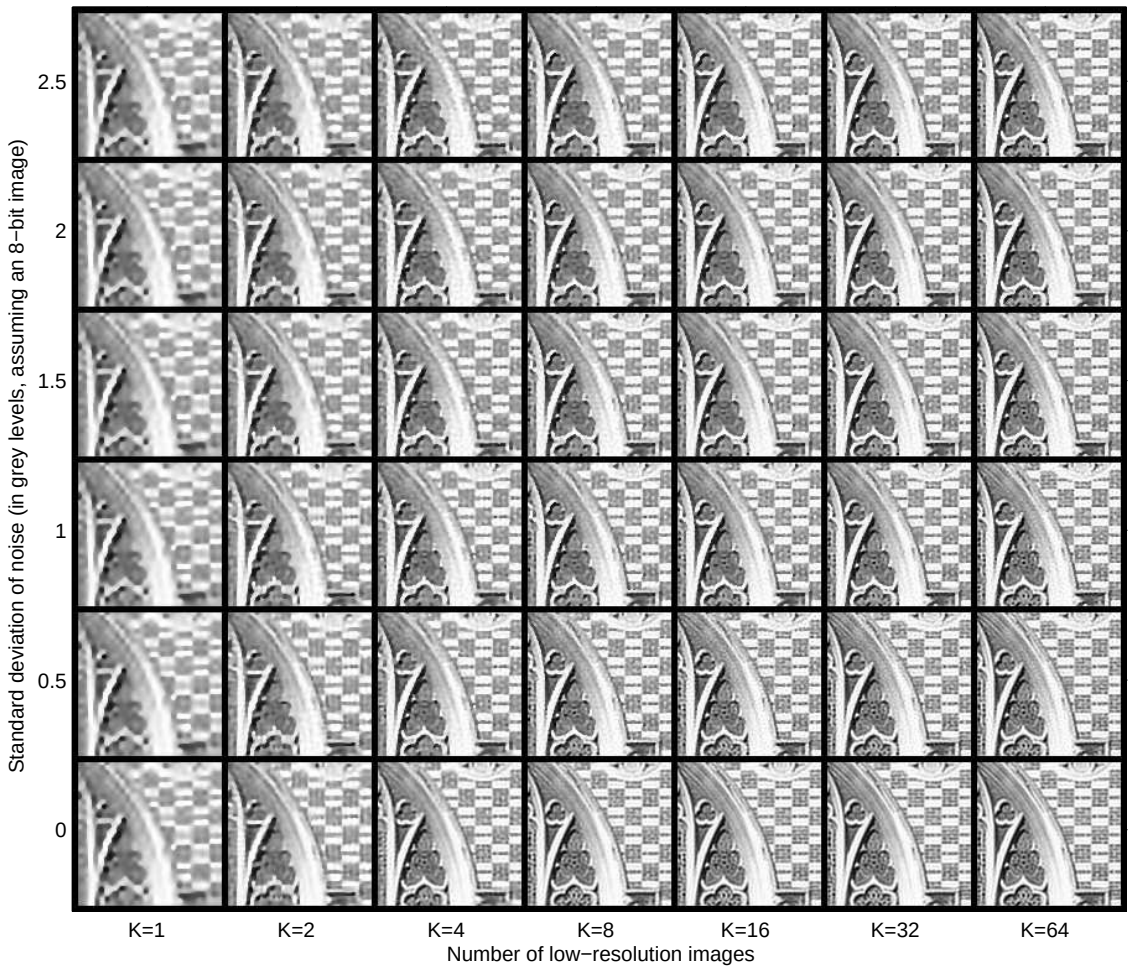


Figure 3.7: ***Maximum a Posteriori* super-resolution images.** This figure uses exactly the same input data as 3.5, but uses the *Maximum a Posteriori* method to infer the super-resolution images, which introduces a prior over the super-resolution image pixels. To construct the results above, the Huber prior was used (see Section 3.4), with parameter $\alpha = 0.05$ and various settings of parameter β, in order to obtain the sharpest possible results. Notice that in *every* case, the scene is clearly visible, and the super-resolution images in the upper middle and upper right of the grid no longer exhibit the noise present in the ML super-resolution images. However, the lack of detail on the two left-columns is still present, since this arises from a lack of input images, rather than a conditioning problem.

3.4 Selected priors used in MAP super-resolution

While ostensibly the prior is merely required to steer the objective function away from the "bad" solutions, in practice the exact selection of image prior does have an impact on the image reconstruction accuracy and on the computational cost of the algorithm, since some priors are much more expensive to evaluate than others.

This section introduces a few families of image priors commonly used in super-resolution, examines their structure, and derives the relevant objective functions to be optimized in order to make a MAP super-resolution image estimate in each case.

3.4.1 GMRF image priors

Gaussian Markov Random Field (GMRF) priors arise from a formulation where the gradient of the super-resolution solution is penalized, and correspond to specifying a Gaussian distribution over $\mathbf{x}$:

$$p(\mathbf{x}) = (2\pi)^{-\frac{N}{2}} |\mathbf{Z}_x|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mathbf{x}^T \mathbf{Z}_x^{-1}\mathbf{x}\right\}, \tag{3.22}$$

where N is the size of the vector $\mathbf{x}$, and $\mathbf{Z}_x$ is the covariance of a zero-mean Gaussian distribution:

$$p(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{Z}_x). \tag{3.23}$$

For super-resolution using any zero-mean GMRF prior, we have:

$$\mathcal{L} = \beta \|\mathbf{r}\|^2 + \mathbf{x}^T \mathbf{Z}_x^{-1} \mathbf{x} \tag{3.24}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = -2\beta\lambda_1 \mathbf{W}^T \mathbf{r} + 2\mathbf{Z}_x^{-1}\mathbf{x}, \tag{3.25}$$

where $\mathcal{L}$ and its derivative can be used in a gradient-descent scheme to find the MAP estimate for $\mathbf{x}$.

Because the data error term and this prior are both Gaussian, it follows that the posterior distribution over $\mathbf{x}$ will also be Gaussian. It is possible to derive a closed-form solution in this case:

$$\mathbf{x}_{\text{GMRF}} = \beta\boldsymbol{\Sigma}\left(\sum_{k=1}^{K} \lambda_1^{(k)} \mathbf{W}^{(k)T}\left(\mathbf{y}^{(k)} - \boldsymbol{\lambda}_2^{(k)}\right)\right) \tag{3.26}$$

$$\boldsymbol{\Sigma} = \left[\mathbf{Z}_x^{-1} + \beta\left(\sum_{k=1}^{K} \lambda_1^{(k)2} \mathbf{W}^{(k)T}\mathbf{W}^{(k)}\right)\right]^{-1}, \tag{3.27}$$

where $\boldsymbol{\Sigma}$ here is the covariance of the posterior distribution. However, the size of the matrices involved means that the iterative approach using SCG is far more practical for all but the very smallest of super-resolution problems.

Depending on the construction of the matrix $\mathbf{Z}_x$, the GMRF may have several different interpretations.

Capel's GMRF

The general gradient-suppressing GMRF employed by Capel in [18] has the form

$$p(\mathbf{x}) = \frac{1}{Z}\exp\left\{-\gamma||\mathbf{D}\mathbf{x}||_2^2\right\}, \tag{3.28}$$

where $\mathbf{D}$ is a matrix which pre-multiplies $\mathbf{x}$ to give a vector of first-order approximations to the magnitude of the image gradient in horizontal, vertical and two perpendicular diagonal directions. This gives us

$$\mathbf{Z}_x^{-1} = \frac{\gamma}{2}\mathbf{D}^T\mathbf{D}, \tag{3.29}$$

though in reality, we don't have to deal with $\mathbf{Z}_x$ or $\mathbf{Z}_x^{-1}$ explicitly. Figure 3.8 shows the four gradient pairs on an example 10×10-pixel super-resolution image. Each of these images is vectorized and transposed to form one row of the $4N \times N$ gradient matrix $\mathbf{D}$.

Figure 3.8: **The four gradient pair cliques used in Capel's GMRF**, shown for a 10×10 toy super-resolution image.

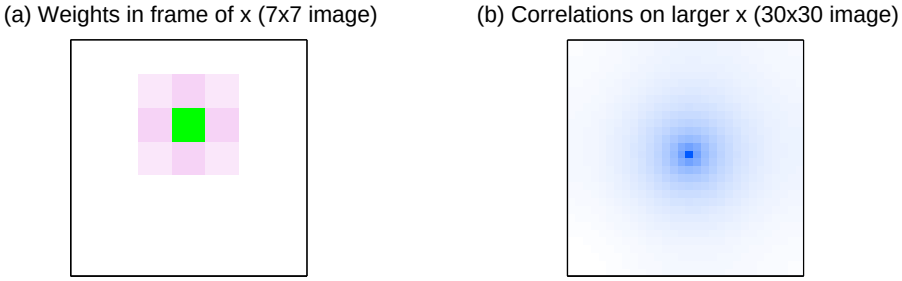


Figure 3.9: **Capel's GMRF prior.** (a) the weights found using the $\mathbf{D}$ matrix of (3.29), for a 7×7-pixel image. Green represents positive weights, with purple for negative ones. (b) The correlations induced over the image as a whole, this time for a 30×30-pixel image, highlighting the large-scale correlations.

The left part of Figure 3.9 shows the weights used in a typical row of $\mathbf{Z}_x^{-1}$ for a 7×7-pixel image. In order to look at the correlations $\mathbf{Z}_x^{-1}$ implies, we would like

to invert it, but the $\mathbf{Z}_x^{-1}$ matrix is singular – this can be understood because the value $\mathbf{Dx}$ is unchanged if the same constant is added to all pixels in $\mathbf{x}$, since $\mathbf{D}$ is constructed only from pixel differences. Adding a small multiple of the identity (*e.g.* a factor of 10^{-3}) leads to a poorly-conditioned but invertible matrix, allowing us to examine the correlations on the high-resolution image frame by assuming $\mathbf{Z}_x \approx (\mathbf{Z}_x^{-1} + 10^{-3}\mathbf{I})^{-1}$. We see that even though the weights used in the prior relate only to four immediately-neighbouring pixels, large-scale correlations are seen across most of the image.

Hardie *et al.*'s GMRF

In [56], Hardie *et al.* also use a prior of the form

$$p(\mathbf{x}) = \frac{1}{Z} \exp\left\{-\gamma ||\mathbf{D}'\mathbf{x}||^2\right\}, \tag{3.30}$$

but the form of $\mathbf{D}'$ is entirely different from Capel's $\mathbf{D}$ matrix. Rather than modelling each directional gradient individually, this prior uses a small discrete approximation to the Laplacian-of-Gaussian filter, so the result for each pixel is the difference between its own value and the average of its four cardinal neighbours. Thus $\mathbf{D}'$ is an $N \times N$ sparse matrix where the i^{th} row has an entry of 1 at position i, and four entries of $-\frac{1}{4}$ corresponding to the four cardinal neighbours of pixel i, with the remainder of the values being zero. As with Capel's GMRF, we can use (3.29) to find the corresponding covariance matrix $\mathbf{Z}_x$ for a GMRF.

Like Capel's GMRF, Hardie's GMRF is composed only from pixel differences, and is therefore singular. The correlations in the high-resolution frame that are implied by $\mathbf{Z}_x^{-1}$ are examined in the same way as those in the previous section,

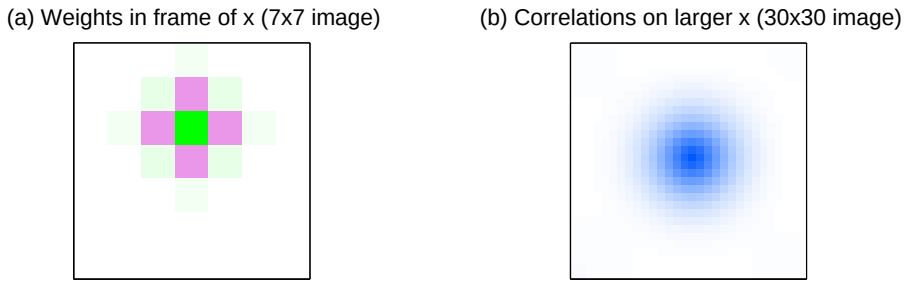


Figure 3.10: **Hardie *et al.*'s GMRF prior.** (a) the weights found using the $\mathbf{D}'$ matrix of (3.30), representing an approximation to the Laplacian, shown for a 7×7-pixel image. Green represents positive weights, with purple for negative ones. (b) The correlations induced over the image as a whole, this time for a 30×30-pixel image, highlighting the large-scale correlations.

where a small multiple of the identity matrix is added to $\mathbf{Z}_x^{-1}$ before inverting. Figure 3.10 shows the weights and correlations over the image space that we get from Hardie *et al.*'s prior. Notice that because the Laplacian estimate favours constant image regions, the correlation extends further in the image space than the equivalent correlations in Capel's GMRF.

Tipping and Bishop's GMRF

In their *Bayesian Image Super-resolution* paper [112], Tipping and Bishop treat the high-resolution image as a Gaussian Process, and suggest a form of Gaussian image prior where $\mathbf{Z}_x$ is calculated directly according to

$$Z_x(i,j) = A \exp\left\{-\frac{||\mathbf{v}_i - \mathbf{v}_j||^2}{r^2}\right\}, \tag{3.31}$$

where $\mathbf{v}_i$ is the two-dimensional position of the pixel that is lexicographically i^{th} in super-resolution image $\mathbf{x}$, r defines the distance scale for the correlations on the MRF, and A determines their strength.

This differs from the other two GMRF priors described above because here the long-range correlations in the high-resolution space are described explicitly, rather than resulting from the short-range weights prescribed in a difference matrix. One reshaped row of the resulting $\mathbf{Z}_x$ is shown in Figure 3.11. The matrix is considerably better-conditioned than either of the other GMRFs above, so we can view both the covariance matrix (left) and its inverse (right) easily. The image on the right of the figure can be viewed as if $\mathbf{Z}_x^{-1}$ were created by taking $\|\mathbf{Dx}\|$ for some other operator $\mathbf{D}$, but in this case it represents a more complex filter than the sum-of-gradients or Laplacian-of-Gaussian forms seen above.

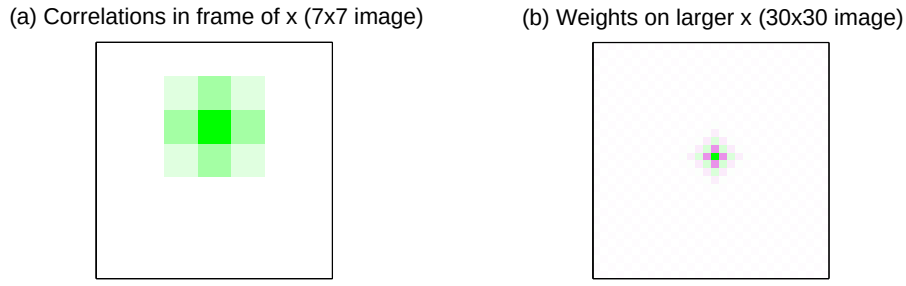


Figure 3.11: **Tipping and Bishop's GMRF prior.** (a) The correlations in the image frame, as described by (3.31), for one pixel in a 7×7-pixel image. (b) The corresponding image weightings these correlations would have resulted from, had this GMRF covariance matrix been formed in the same way as those in Figures 3.9 or 3.10. In both cases, green represents positive values, and purple represents negative values.

3.4.2 Image priors with heavier tails

BTV prior

The *Bilinear Total Variation* (BTV) prior is used by Farsiu *et al.* [35]. It compares the high-resolution image to versions of itself shifted by an integer number of pixels

in various directions, and weights the resulting absolute image differences to form a penalty function. This leads again to a prior that penalizes high spatial frequency signals, but is less harsh than a Gaussian because the norm chosen is L_1 rather than L_2.

The BTV image prior may be written

$$p(\mathbf{x}) = \frac{1}{Z}\exp\left\{\nu \sum_{l,m=-n}^{n} \alpha^{(|l|+|m|)} \left\|\mathbf{x} - \mathbf{S}_i^l \mathbf{S}_j^m \mathbf{x}\right\|_1\right\}, \tag{3.32}$$

where ν, α and n are parameters of the prior, Z is the normalization constant, and $\mathbf{S}_i^l$ and $\mathbf{S}_j^m$ are matrices realising the shift operations in the horizontal and vertical directions by l and m pixels respectively.

For super-resolution with the BTV prior, we have:

$$\mathcal{L} = \sum_{k=1}^{K} \beta \|\mathbf{r}^{(k)}\|_2^2 + \nu \sum_{l,m=-n}^{n} \alpha^{(|l|+|m|)} \left\|\mathbf{x} - \mathbf{S}_i^l \mathbf{S}_j^m \mathbf{x}\right\| \tag{3.33}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{x}} &= -2\beta \sum_{k=1}^{K} \lambda_1^{(k)} \mathbf{W}^{(k)T} \mathbf{r}^{(k)} \\ &\quad + \nu \sum_{l,m=-m}^{n} \alpha^{(|l|+|m|)} \left[\mathbf{I} - \mathbf{S}_j^{-m} \mathbf{S}_i^{-l}\right] \operatorname{sign}\left(\mathbf{x} - \mathbf{S}_i^l \mathbf{S}_j^m \mathbf{x}\right).\end{aligned} \tag{3.34}$$

Note that Farsiu *et al.* use a Laplacian noise model, rather than our Gaussian model, so for their own optimizations, the data error term above would have an L_1 norm, rather than this L_2 norm.

Huber prior

The Huber function is used as a simple prior for image super-resolution which benefits from penalizing edges less severely than any of the Gaussian image priors. The form of the prior is

$$p(\mathbf{x}) = \frac{1}{Z} \exp\left\{ -\nu \sum_{g \in \mathcal{D}(\mathbf{x})} \rho(g, \alpha) \right\}, \tag{3.35}$$

where $\mathcal{D}$ is the same set of gradient estimates as (3.28), given by $\mathbf{Dx}$. The parameter ν is a prior strength somewhat similar to a variance term, Z is the normalization constant, and α is a parameter of the Huber function specifying the gradient value at which the penalty switches from being quadratic to being linear:

$$\rho(x, \alpha) = \begin{cases} x^2, & \text{if } |x| \leq \alpha \\ 2\alpha|x| - \alpha^2, & \text{otherwise.} \end{cases} \tag{3.36}$$

For a very simple 1D case, some Huber functions and their corresponding probability functions

$$p(x) \;=\; \frac{1}{Z} \exp\left\{ -\nu \rho(x, \alpha) \right\} \tag{3.37}$$

are shown in Figure 3.12. For this 1D case, we can show by integration that

$$Z \;=\; \frac{1}{\nu\alpha} \exp\left\{ -2\nu\alpha^2 \right\} + \left(\frac{\pi}{\nu} \right)^{\frac{1}{2}} \operatorname{erf}\left\{ \alpha\nu \right\}. \tag{3.38}$$

Unfortunately the prior over the image is much harder to normalize, as the structure

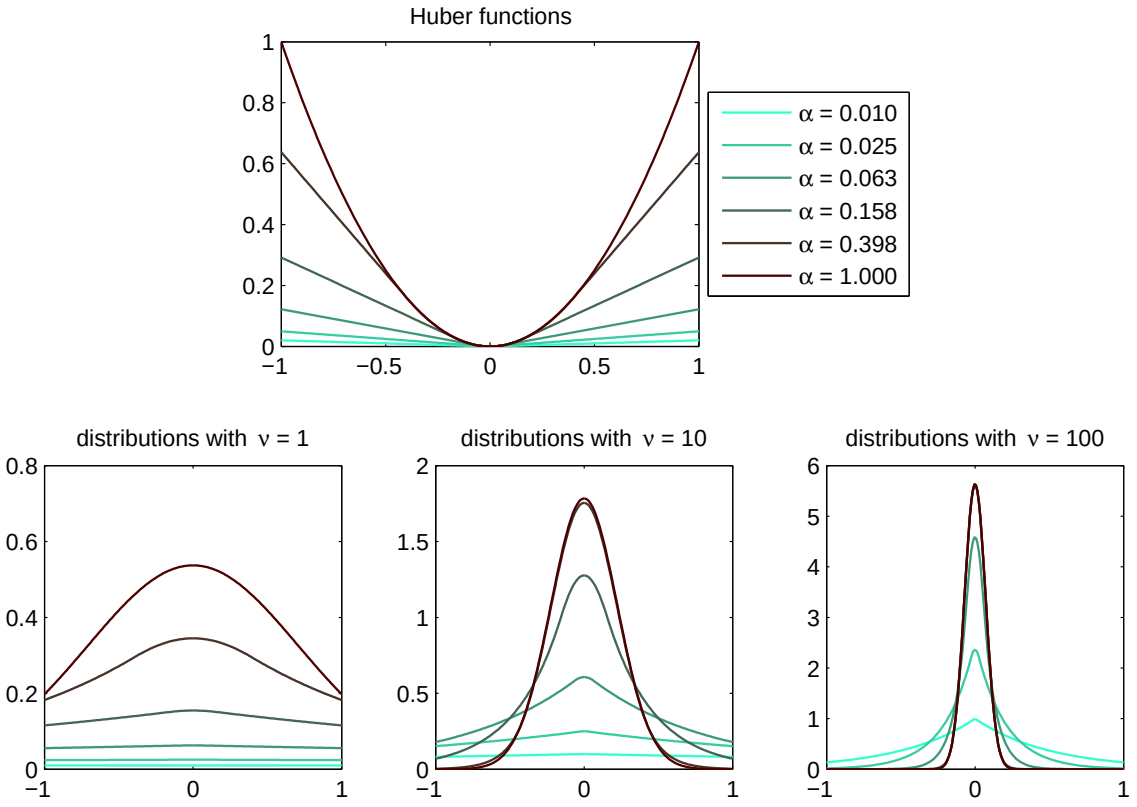


Figure 3.12: **Huber functions and corresponding distributions.** Top: Several Huber functions corresponding to a set of logarithmically-spaced α values. Bottom: three sets of distributions, each using the set of Huber functions and a different value of the prior strength parameter, ν. From left to right, the ν values are 1, 10, 100.

of the image means that each pair of pixel differences cannot be assumed independent, and so no closed-form solution exists. This makes it hard to learn Huber-MRF priors directly from data, because it rules out optimizing $p(\mathbf{x})$ with respect to α and ν, or writing down an explicit form of $p(\alpha, \nu | \mathbf{x})$. However, there are still methods available that allow us to learn values for α and ν, which we will return to in Chapter 4.

Regardless of the difficulty in normalization, super-resolving with the Huber-MRF prior it is very straight-forward. The objective function and its gradient with respect to the high-resolution image pixels are

$$\mathcal{L} = \sum_{k=1}^{K} \beta \|\mathbf{r}^{(k)}\|_2^2 + \nu \sum_{g \in \mathcal{D}\mathbf{x}} \rho(g, \alpha) \tag{3.39}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = -2\beta \sum_{k=1}^{K} \lambda_1^{(k)} \mathbf{W}^{(k)T} \mathbf{r}^{(k)} + \nu \mathbf{D}^T \rho'(\mathbf{D}\mathbf{x}, \alpha) \tag{3.40}$$

where

$$\rho'(x) = \begin{cases} 2x, & \text{if } |x| \leq \alpha \\ 2\alpha \text{ sign}(x), & \text{otherwise.} \end{cases} \tag{3.41}$$

and $\mathbf{D}$ is again Capel's version of the gradient operator (Section 3.4.1). This has the advantage over the TV prior that unless $\alpha \to 0$, the function and its gradient with respect to $\mathbf{x}$ are continuous as well as convex, and can be solved easily using gradient-descent methods like SCG.

3.5 Where super-resolution algorithms go wrong

There are many reasons why simply applying the MAP super-resolution algorithm to a collection of low-resolution images may not yield a perfect result immediately. This section consists of a few brief examples to highlight the causes of poor super-resolution results from what should be good and well-behaved low-resolution image sets. At the end of the section is a note on a few situations in which the data violates the generative model assumptions of Section 3.1.

The super-resolution problem involves several closely-interrelated components: geometric registration, photometric registration, parameters for the prior, noise estimates and the point-spread function, not to mention the estimates of the values of the high-resolution image pixels themselves. If one component is estimated badly, there can be a knock-on effect on the values of the other parameters needed in order to produce the best super-resolution estimate for a given low-resolution dataset. A few examples are given in the following subsections.

3.5.1 Point-spread function example

A bad estimate of the size and shape of the point-spread function kernel leads to a poor super-resolution image, because the weights in the system matrix do not accurately reflect the responsibility each high-resolution pixel (or scene element) takes for the measurement at any given low-resolution image pixel. The solution which minimizes the error in the objective function does not necessarily represent a good super-resolution image in this case, and it is common to see "ringing" around edges in the scene. These ringing artifacts can be attenuated by the prior on $\mathbf{x}$, so in general, a dataset with an incorrectly-estimated PSF parameter will require a

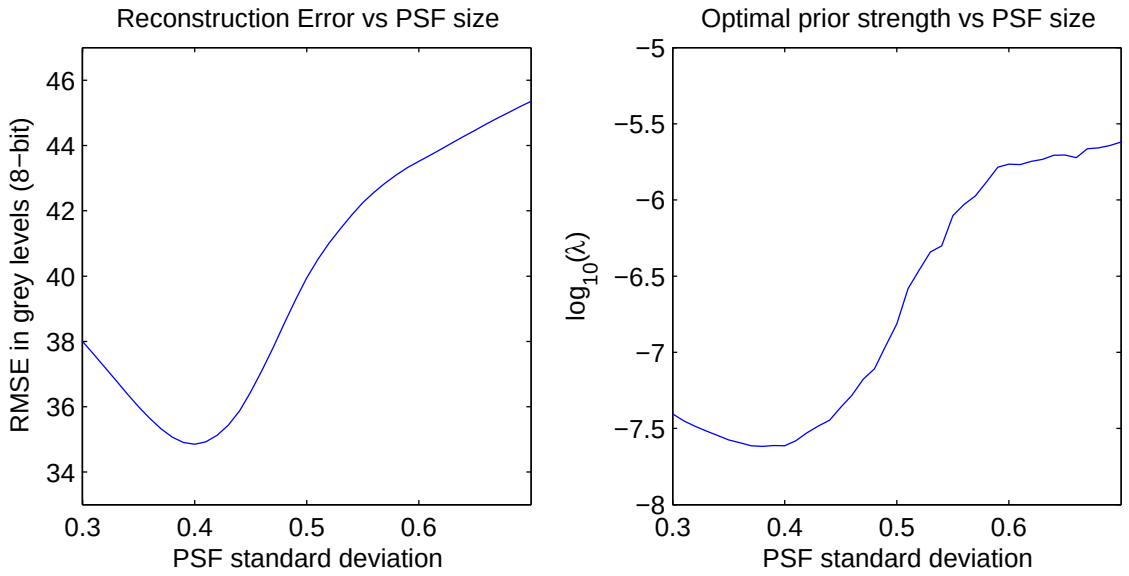


Figure 3.13: **Reconstructing the Keble dataset with various PSF estimates: errors.** The left-hand plot shows the RMS error, with respect to the ground truth Keble image, of super-resolution images recovered as the PSF standard deviation, γ, was varied from 0.3 low-resolution pixels up to 0.7 low-resolution pixels; the ground truth value is 0.4, and as expected, the curve has its minimum at this value. The right-hand plot shows the value of the prior strength at which the minimum error is achieved for each setting of γ. When γ is overestimated by a factor of 50%, the prior needs to be almost two orders of magnitude larger to reconstruct the image as well as possible.

stronger image prior to create a reasonable-looking super-resolution image than a dataset where the PSF size and shape are known accurately.

To illustrate this, we take 16 images from the synthetic Keble dataset (see Figure 3.4) with a noise standard deviation of 2.5 grey levels, and we super-resolve this at a zoom factor of 4 using the known geometric and photometric registration values, and a Huber prior with $\alpha = 0.05$. We vary the PSF standard deviation, γ, and find the optimal value for the prior strength ratio (the value ν/β in (3.39)) for each γ setting.

The results are plotted in Figure 3.13, which clearly shows that as γ moves away

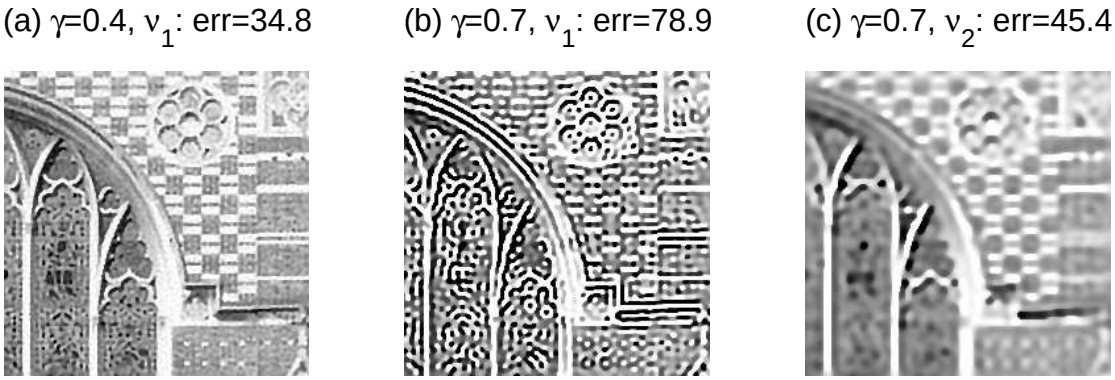


Figure 3.14: **Reconstructing the Keble dataset with various PSF estimates: images.** (a) The best reconstruction achieved at the correct value, $\gamma = 0.4$, for which the optimal prior strength ratio is $10^{-7.61} = 2.43e{-}8$. (b) The super-resolution image obtained with the same prior as the left-hand image, but with $\gamma = 0.7$. Heavy ringing is induced by the bad PSF estimate. (c) The best possible reconstruction using $\gamma = 0.7$. This time the prior strength ratio is $10^{-5.62} = 2.40e - 6$, almost 100 times stronger than for the optimal image, even though the input images themselves are identical.

from its true value of 0.4 low-resolution image pixels, the error increases, and the prior strength ratio needed to achieve the minimal error also increases. Figure 3.14 shows three images from the results. The first is the image reconstructed with the true γ, showing a very good super-resolution result. The next is the image reconstructed with $\gamma = 0.7$, and is a very poor super-resolution result. The final image of the three shows the super-resolution image reconstructed with the same poor value of γ, but with a much stronger prior; while the result is smoother than our ideal result, the quality is definitely superior to the middle case.

The important point to note is that all these images are constructed using *exactly the same* input data, and only γ and ν were varied. The consequence of this kind of relationship is that even if we are able to make a *reasonably* accurate estimate of each of the hyperparameters we need for super-resolution, the values themselves must be selected together in order to guarantee us a good-looking super-resolution

image.

3.5.2 Photometric registration example

The photometric part of the model accounts for relative changes in illumination between images, either due to changes in the incident lighting in the scene, or due to camera settings such as automatic white balance and exposure time. When the photometric registration has been calculated using pixel correspondences resulting from the geometric registration step and bilinear interpolation onto a common frame, we expect some error because both sets of images are noisy, and because such interpolation does not agree with the generative model of how the low-resolution images relate to the original scene.

To understand the effect of errors in the photometric estimates, we run several super-resolution reconstructions of the Keble dataset of Figure 3.4 with a noise standard deviation of 2.5 grey levels, using the ground truth point-spread function (a Gaussian with *std* 0.4 low-resolution pixels), the true geometric registration, and a set of photometric shift parameters which are gradually perturbed by random amounts, meaning that each image is assumed to be globally very slightly brighter or darker than it really is relative to the ground truth image.

For each setting of the photometric parameters, a set of super-resolution images was recovered using the Huber-MAP algorithm with different strengths of Huber prior. The plots in Figure 3.15 show the lowest error (left) and prior strength ratio, $\log_{10}(\nu/\beta)$ (right) for each case. Figure 3.16 shows the deterioration of the quality of the super-resolution image for the cases where the sixteen photometric shift parameters (one per image) were perturbed by an amount whose standard

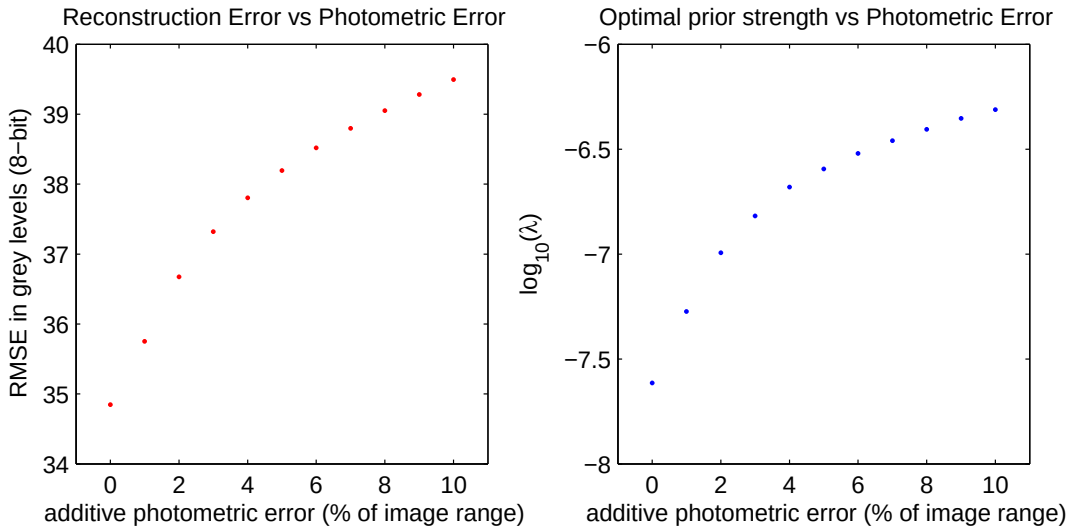


Figure 3.15: **Reconstructing the Keble dataset with photometric error: errors**. The left-hand plots shows how the RMSE of the reconstruction increases with the uncertainty in the photometric shift parameter: the standard deviation of the error in photometric shift was varied from 0 to 10% of the total intensity range of the image. The right-hand plot shows the prior strength setting necessary to achieve the best reconstruction for each setting of the photometric parameters. In this case, the prior strength increases by well over an order of magnitude between the no-error case and the 10% case.

deviation was equal to 2% and 10% of the image range respectively.

The edges are still very well localized even in the 10% case, because the geometric parameters are perfect. However, the ill conditioning caused in the linear system of equations solved in the Huber-MAP algorithm means that the optimal solutions require stronger and stronger image priors as the photometric error increases, and this results in the loss of some of the high frequency detail, like the brick pattern and stained-glass window leading, which are visible in the error-free solution.

3.5.3 Geometric registration example

Errors from two different sources can also be very closely-coupled in the super-resolution problem. In this example, we show that errors in some of the geometric parameters $\boldsymbol{\theta}^{(k)}$ can to a small extent be mitigated by a small increase in the size of the blur kernel.

We take images from the synthetic Frog dataset of Figure 3.17, which is constructed in exactly the same way as the Keble dataset, but using an entirely different scene as the ground truth high-resolution image. Again, we take 16 low-resolution images at a zoom factor of 4, a PSF width of 0.4 low-resolution pixels, and various levels of *i.i.d.* Gaussian noise. Because the ground-truth image is much smoother than the Keble image, with fewer high-frequency details, it is in general easier to super-resolve, and leads to reconstructions with much lower RMS error than the Keble image dataset.

Errors are applied to the $\boldsymbol{\theta}$ parameters governing the horizontal and vertical shifts of each image, with standard deviations of 0, 0.01, 0.02, 0.03, 0.04 and 0.05 low-resolution pixels. For each of these six registrations of the input data, a set of

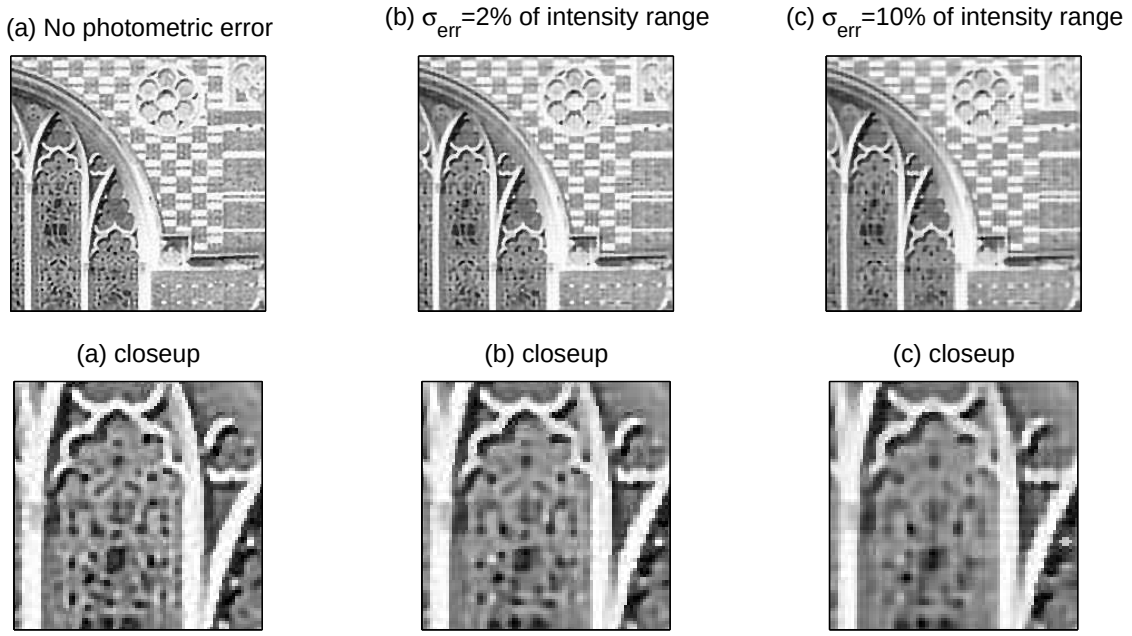


Figure 3.16: **Reconstructing the Keble dataset with photometric error: images**. The upper row shows the full super-resolution image, and the lower row shows a close-up of the main window, where the different levels of detail are very noticeable. (a) Reconstruction with ground truth photometric parameters: the sources of error here are merely the additive noise on the low-resolution images and the loss of high frequency detail due to subsampling. (b) The best reconstruction possible with a 2% error on the photometric shift parameters, $\lambda_2^{(k)}$. (c) The best reconstruction possible with a 10% error. Notice that the edges are still well-localized, but finer details are smoothed out due to the increase in prior strength.

Figure 3.17: **The synthetic Frog dataset.** At the top is the ground truth Frog image. Below is an array of low-resolution images generated from this ground truth image. Like the Keble dataset in Figure 3.4, all these low-resolution images have a Gaussian point-spread function of *std* 0.4 low-resolution pixels, a zoom factor of 4, an 8DoF projective transform, and a linear photometric model. Each column represents a different registration vector, and each row represents a different level of the Gaussian *i.i.d.* noise added to the low-resolution images, measured in grey levels, assuming an 8-bit image (*i.e.* 256 grey levels). Because large areas of the frog image are smooth, this dataset represents less of a challenge to reconstruct than the Keble image, since smoothness-based image priors describe the distribution of its pixels more effectively than they describe the Keble image.

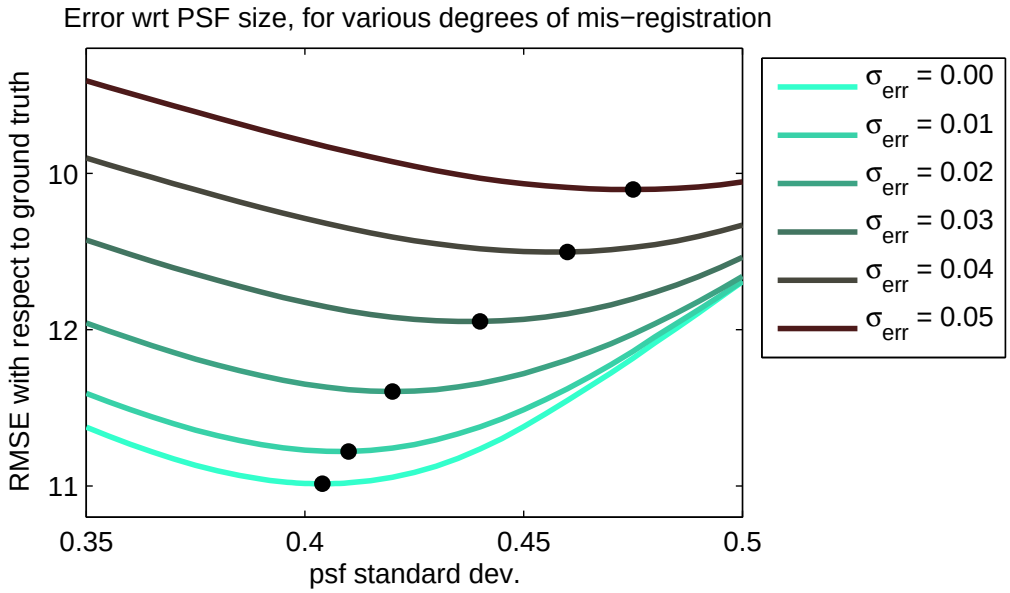


Figure 3.18: **Reconstructing the Frog image with small errors in geometric registration and point-spread function size.** The six colours represent six levels of additive random noise added to the shift parameters in the geometric registration. The curves represent the optimal error as the PSF parameter γ was varied about its ground truth value of 0.4. The larger the registration error, the bigger the error in γ is in order to optimize the result.

super-resolution images is recovered as the PSF standard deviation, γ is varied, and for each setting of γ, the prior strength ratio giving the best reconstruction is found.

Figure 3.18 shows how the best error for each of the six registration cases varies with the point-spread function size. When the geometric registration is known accurately, the minimum falls at the true value of γ. However, as the geometric registration parameters drift more, the point at which the lowest error is found for any given geometric registration increases. This can be explained intuitively because the uncertainty in the registration tends to spread the assumed influence of each high-resolution pixel over a larger area of the collection of low-resolution images.

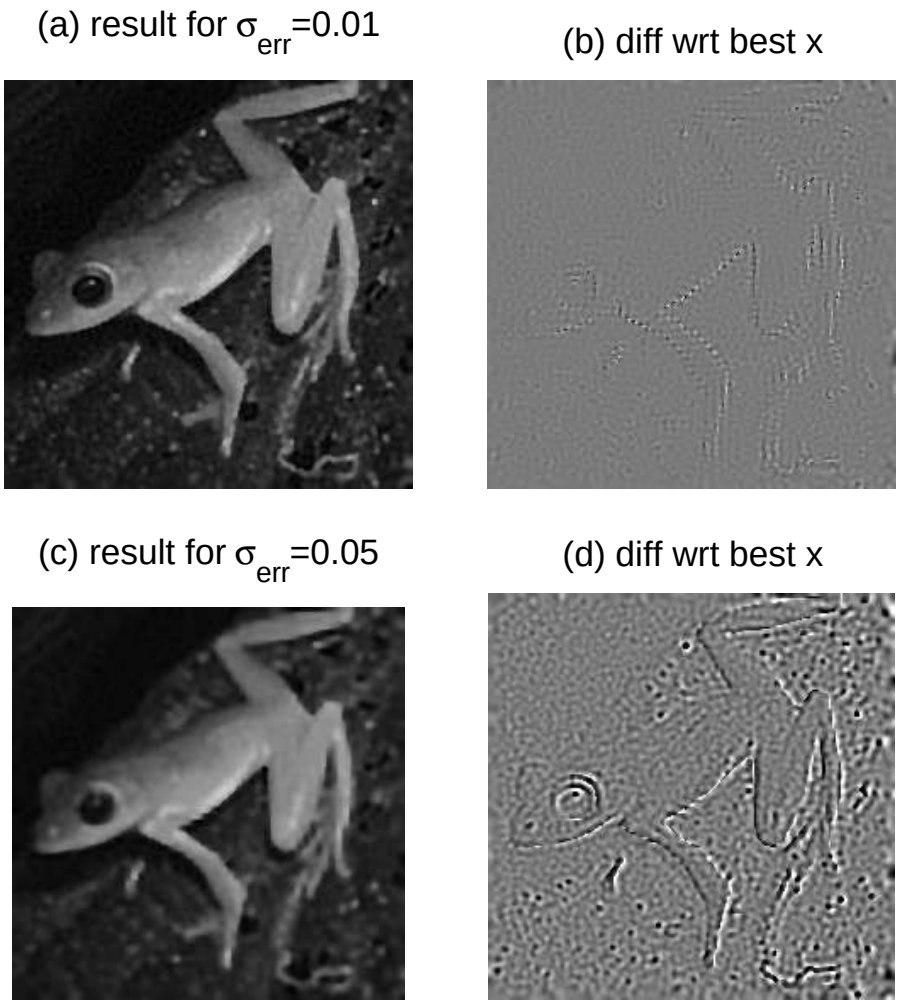


Figure 3.19: **Reconstructing the Frog dataset with smell levels of geometric error.** (a) The best result with $\sigma_{err} = 0.01$, corresponding to the second-lowest black dot in Figure 3.18. (b) The difference between this reconstruction and the best reconstruction given no geometric error (corresponding to the lowest black dot in Figure 3.18), amplified by a factor of four. (c) The result with $\sigma_{err} = 0.05$ (uppermost black dot of Figure 3.18). (d) The difference between this reconstruction and the best reconstruction, amplified by a factor of four.

Figure 3.19 shows two high-resolution images reconstructed with geometric registration error standard deviations of 0.01 and 0.05 low-resolution pixels, and also shows the difference between these and the best reconstruction for this dataset using the true registration. Localization problems and the effects of a stronger image prior are clearly visible along the strong image edges, particularly around the frog's eye, especially where the standard deviation is as high as one twentieth of a low-resolution pixel.

3.5.4 Shortcomings of the model

A few situations that might arise in the imaging process are not covered by the generative model, and we conclude this chapter by considering these.

The biggest shortfall of the imaging model used so far is that many imaging conditions cannot be described adequately using the planar projective transform. However, many other transforms can be described as locally affine or locally projective, so this is not as restrictive as it might appear. For very non-rigid scenes, and those with multiple motions, it is possible to use optic flow, but the size of the latent registration space increases considerably, because motion and occlusion information for each pixel needs to be stored [44]. However, the registration itself can be accounted for in the construction of the $\mathbf{W}^{(k)}$ matrices if desired, and while the estimation of the registration space becomes correspondingly harder, the treatment of the other parts of super-resolution, such as the photometric registration and the high-resolution image priors discussed above need not change much.

Image datasets captured using modern digital cameras are likely to have had a significant amount of pre-processing applied before being saved out to image files.

These include white balance/gamma correction, adjustment for optimal dynamic range, and lossy compression in order for the images to fit more efficiently into the memory media.

Lossy compression is a problem because methods like the JPEG algorithm destroy exactly the high-frequency information in the input images that we want to use in super-resolution image reconstruction. The compression algorithm works by taking 8×8-pixel blocks in the image, finding the *Discrete Cosine Transform* (DCT) of each block, and quantizing the 64 coefficients. The granularity of the quantization depends on how sensitive we as humans are to each spatial frequency, so that frequencies we are less aware of are quantized more coarsely, therefore requiring fewer bits to store.

Figure 3.20 shows a selection of images from the Frog and Keble synthetic datasets which have been saved as lossy JPEG images with various quality levels using Matlab's `imwrite` routine.

For each JPEG quality level (from 5% to 100% in increments of 5%, giving 20 datasets per ground truth image), the standard deviation of the pixel-wise additive error induced by the compression artifacts was calculated. Control image sets were created using *i.i.d.* Gaussian image noise at each of these standard deviations. For each dataset, the best reconstruction was found (across all settings of the Huber prior strength, with $\alpha = 0.05$), and these are plotted in Figure 3.21. While the errors are comparable for the high quality (less lossy) JPEG images, as the quality decreases, the JPEG image reconstructions are significantly worse than the noise model would predict based on *i.i.d.* Gaussian noise of the same standard deviation.

It is interesting to note that the optimal prior ratio for each of the four cases follow approximately the same curve, even though the minimal errors for each of

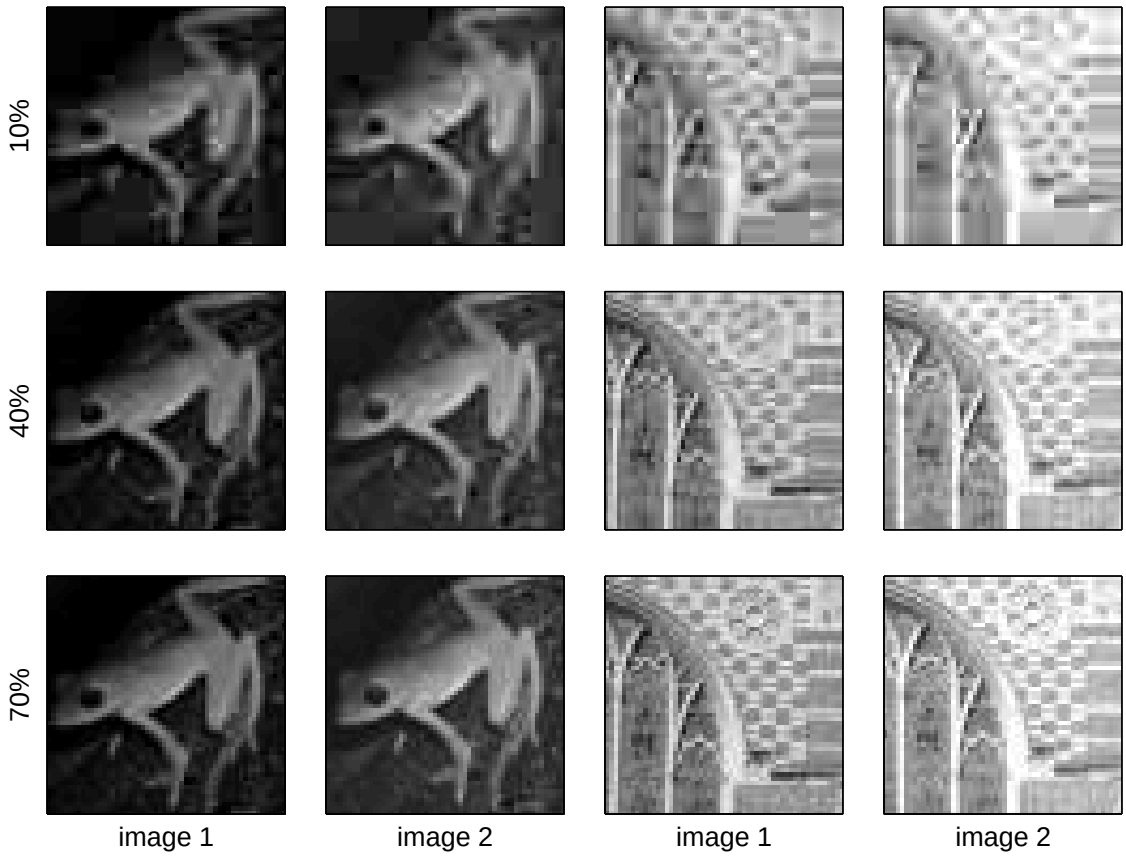


Figure 3.20: **The "Frog" and "Keble" datasets.** The first two images of the Frog and Keble datasets, saved as lossy JPEG images with quality levels of 10% (poorest quality of those shown), 40% and 70% (best quality of those shown) respectively. For ground truth images, see Figures 3.4 and 3.17.

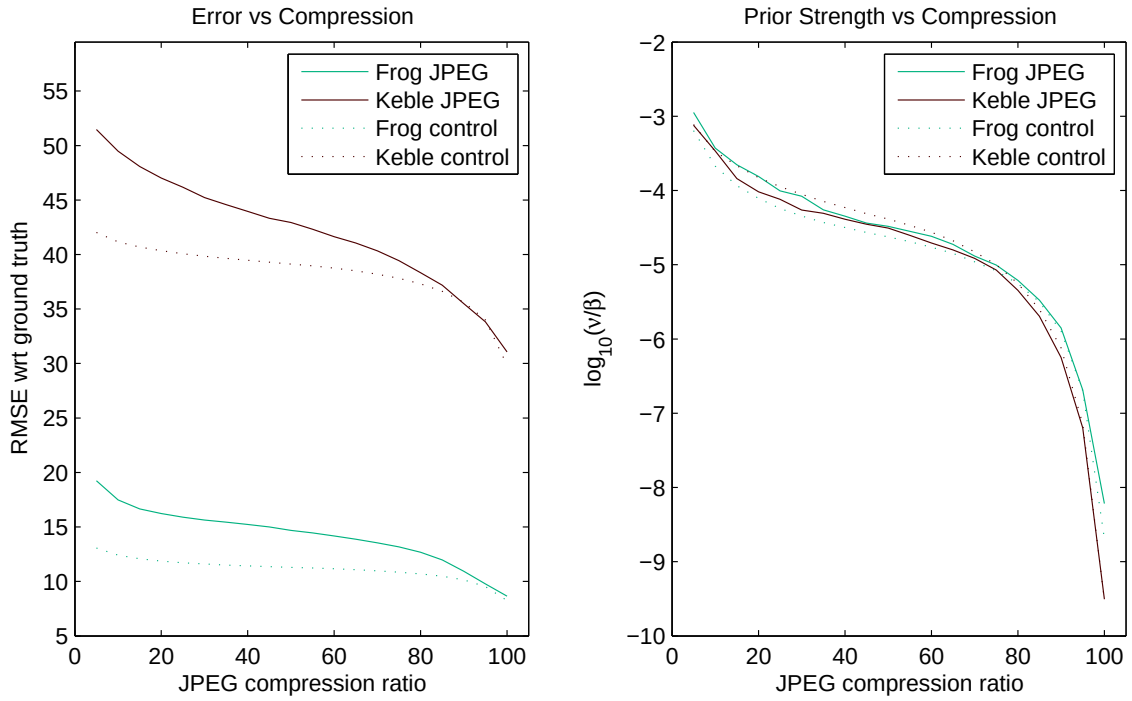


Figure 3.21: **The underperformance of super-resolution images on JPEG-compressed image data.** The solid curves on the left-hand plot show the errors achieved using JPEG input images as the compression ratio is changed. The dotted curves indicate the errors achieved using datasets with equivalent *i.i.d.* Gaussian noise (and hence obey the forward model), which in all cases are lower. The right-hand plot shows the strength ratio $\log_{10}(\nu/\beta)$ for each of the four groups of datasets. While the quality of the image results vary considerably, the prior strengths all match each other remarkably well.

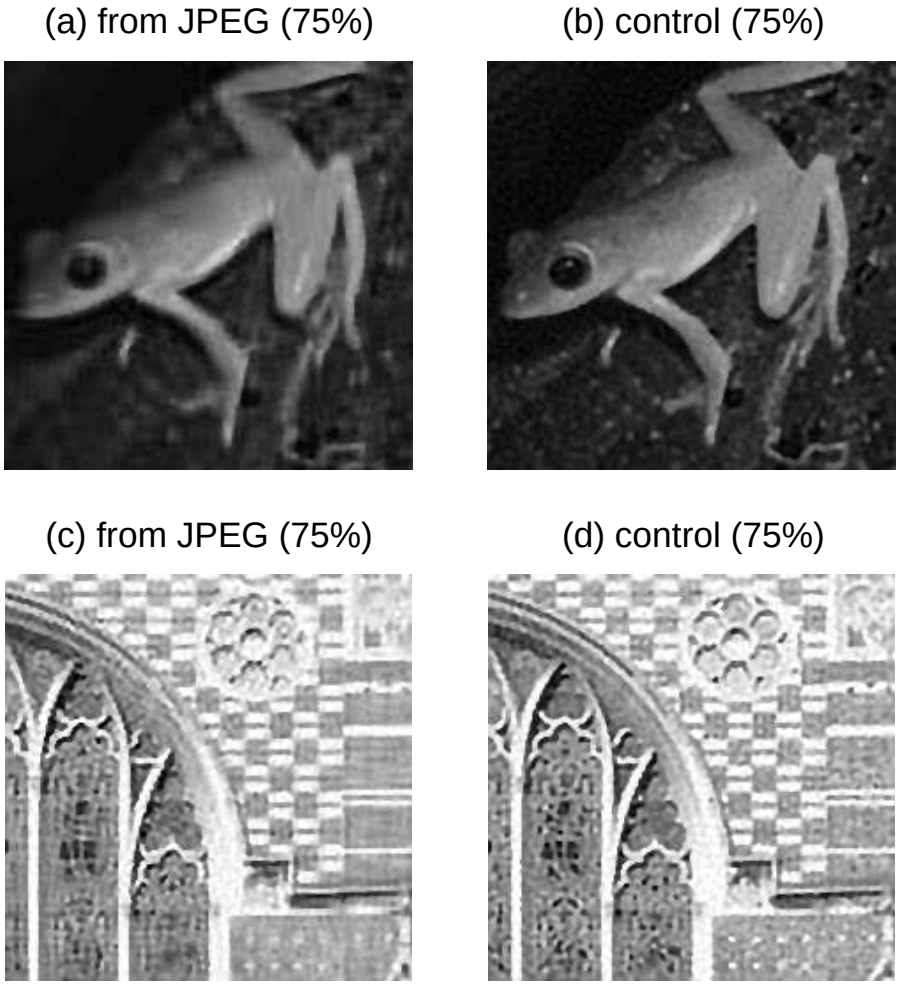


Figure 3.22: **Reconstructing images corrupted with non-Gaussian noise.** (a) the Frog image, reconstructed using 16 images from the Frog dataset which had been saved as JPEG images at a quality setting of 75%. (b) The corresponding Frog image reconstruction where Gaussian noise of the same standard deviation is added instead; the image quality is significantly better, especially on the tree surface and the specularity in the frog's eye. (c) Keble image, reconstructed using 16 images from the Keble dataset which had been saved as JPEG images at a quality setting of 75%. (d) Corresponding Keble image reconstruction, using Gaussian *i.i.d.* noise of the same standard deviation. Again, this is significantly better than the JPEG image case, in particular in capturing the window leading and the brick texture.

the four groups of datasets are very different. Finally, Figure 3.22 shows the super-resolution images from the JPEG and control sets at the 75% quality level, in order for the image quality to be inspected visually. In both cases, even at this relatively high quality level, the details missing from the JPEG case but preserved in the Gaussian nose case are clearly visible.

These observations are important because the same type of image compression used in JPEG images also forms part of the MPEG video compression algorithm, and many video datasets are compressed this way; if the model were not capable of handling such an input source, this would indeed be a problem. It is also possible to extend the model to include the degradation due to the JPEG compression in the forward model, but the quantization is a nonlinear operation, so this prevents us representing the low-resolution inputs as linear functions of the super-resolution image we want to recover, and makes the system less easy to solve. In the plots of Figure 3.21, we have seen that the changed noise type does not greatly change the behaviour of the rest of the model with respect to the corresponding best prior strength setting, so even when using JPEG-compressed images, we chose to accept the limitations of the generative model as described in Section 3.1.

3.6 Super-resolution so far

This chapter has laid out the foundations of the probabilistic approach to image super-resolution, as well as introducing notation for the various parts of the problem we need to deal with, along with the objective functions and function gradients necessary to find super-resolution image results in several different ways and using several different image priors.

We have highlighted several important ways in which a super-resolution algorithm can under-perform when it is presented with erroneous latent parameters (geometric and photometric registrations and point-spread function), and have shown that these parameters themselves are highly coupled through the generative model, so that errors in one may effect the error surface with respect to the others as well.

Finally, we have summarized the main ways in which the model presented does not conform to all of the image input data sources we might wish to super-resolve, and have shown that for the common case of JPEG image errors, the model and the MAP solution outlined above still provide a successful scheme upon which to base further investigations into improving and automating this multi-frame image super-resolution process as far as possible.

Chapter 4

The simultaneous approach

In this chapter we introduce an algorithm to estimate a super-resolution image at the same time as finding the low-resolution image registrations (both geometric and photometric), and selecting the strength and shape of image prior necessary for a good super-resolution result. We show that this *simultaneous* approach offers visible benefits on results obtained from real data sequences. This work was originally published in [93], and developed in [91].

The problem of multi-frame image super-resolution is often broken down into several distinct stages: image registration or motion estimation, low-resolution image blur estimation, selection of a suitable prior, and super-resolution image estimation. However, these stages are seldom truly independent, and this is too often ignored in current super-resolution techniques. In Chapter 2 we mentioned the few exceptions which do learn a registration model, and saw that often they are either tied to very restrictive motion models, or use unrealistic GMRF-style priors which blur out edges, or have extremely high computational costs.

We also expect that each scene will have different underlying image statistics, and a super-resolution user generally has to hand-tune these in order to obtain an output which preserves as much richness and detail as possible from the original scene without encountering problems with conditioning, even when using a GMRF.

Taken together, these observations motivate the development of an approach capable of registering the images at the same time as super-resolving and tuning a prior, in order to get the best possible result from any given dataset.

4.1 Super-resolution with registration

Even very small errors in the registration of the low-resolution dataset can have consequences for the estimation of other super-resolution components. The examples in Section 3.5 showed some of the ways in which one super-resolution component affects the estimation of another. In particular, the example in Section 3.5.2 showed that error in the photometric registration requires an increase in prior strength, and the example of Section 3.5.3 showed that slight perturbations in the geometric registration can give the appearance of a larger point-spread function, so the importance of acquiring as accurate a registration estimate as possible is clear.

Standard approaches to super-resolution *first* determine the registration, *then* fix it and optimize a function like the MAP objective function of (3.20) with respect only to $\mathbf{x}$ to obtain the final super-resolution estimate. However, if the set of input images is assumed to be noisy, it is reasonable to expect the registration to be adversely affected by the noise.

In contrast, we make use of the high-resolution image estimate common to all the low-resolution images, and aim to find a solution in terms of the high-resolution

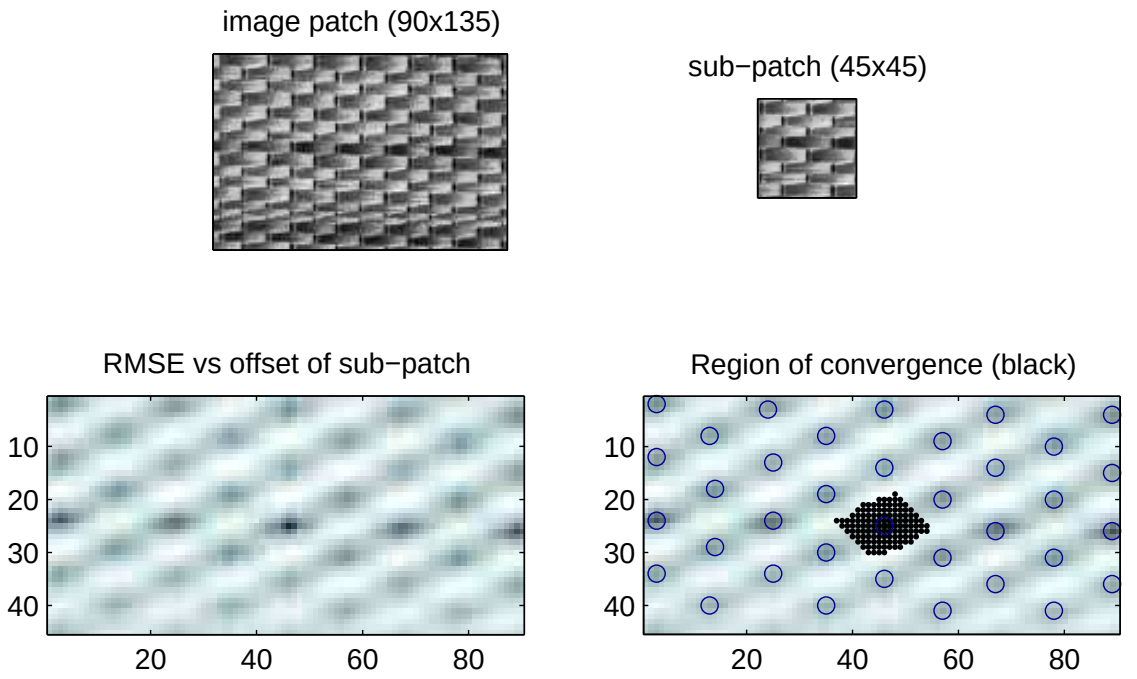


Figure 4.1: **Example of a non-convex registration problem.** Top: the 45×45-pixel sub-patch (right) is taken from a 90×135-pixel image (left). Now each of the possible 45×45-pixel sub-windows of the image is compared to this sub-patch, and the error at each offset from the top left corner is recorded, giving the plot on the bottom left. Darker regions show lower error, and we can see that there are multiple minima other than the single global minimum near the centre of the plot. The black region in the bottom-right plot shows all the offsets from which the global minimum can be reached by gradient descent; additionally, all the local minima are circled.

image $\mathbf{x}$, the set of geometric registration parameters, $\boldsymbol{\theta}$ (which parameterize $\mathbf{W}$), and the photometric parameters $\boldsymbol{\lambda}$ (composed of the λ_1 and λ_2 values), at the same time, i.e. we determine the point at which

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = 0. \tag{4.1}$$

The registration problem itself is not convex, and repeating textures can cause naïve intensity-based registration algorithms to fall into local minima, though when

initialized sensibly, very accurate results are obtained. Figure 4.1 shows the local optimum problem on repeating texture, along with the region of convergence for which gradient descent optimization will reach the global minimum. The pathological case where the footprints of the low-resolution images fail to overlap in the high-resolution frame can be avoided by adding an extra term to $\mathcal{L}$ to penalize large deviations in the registration parameters from the initial registration estimate (see below).

4.1.1 The optimization

The simultaneous super-resolution and image registration problem closely resembles the well-studied problem of Bundle Adjustment [116], in that the camera parameters and image features (which are 3D points in Bundle Adjustment) are found simultaneously. Because most high-resolution pixels are observed in most frames, the super-resolution problem is closest to the "strongly convergent camera geometry" setup, and conjugate gradient methods are expected to converge rapidly [116].

The objective function for simultaneous registration and super-resolution is very similar to the regular MAP negative log likelihood, except that it is optimized with respect to the registration parameters as well as the super-resolution image estimate, *e.g.*

$$\mathcal{L} = \sum_{k=1}^{K} \beta \|\mathbf{r}^{(k)}\|_2^2 + \nu \sum_{g \in \mathcal{D}\mathbf{x}} \rho(g, \alpha) \tag{4.2}$$

$$[\mathbf{x}_{\text{MAP}}, \boldsymbol{\theta}_{\text{MAP}}, \boldsymbol{\lambda}_{\text{MAP}}] = \underset{\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}}{\operatorname{argmax}} \mathcal{L}, \tag{4.3}$$

where (4.2) is the same as (3.39), repeated here for convenience, though any reason-

able image prior can be used in place of the Huber-MRF here. The residuals are of course functions of the registration parameters: $\mathbf{r}^{(k)} = \mathbf{y}^{(k)} - \lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} - \boldsymbol{\lambda}_2^{(k)}$.

Using the Scaled Conjugate Gradients (SCG) implementation from Netlab [77], rapid convergence is observed up to a point, beyond which a slow steady decrease in the negative log likelihood gives no subjective improvement in the solution, but this extra computation can be avoided by specifying sensible convergence criteria. The gradient with respect to $\mathbf{x}$ is given by (3.40), and the gradients with respect to $\mathbf{W}^{(k)}$ and the photometric registration parameters are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(k)}} = -2\beta\lambda_1^{(k)}\mathbf{r}^{(k)}\mathbf{x}^T \tag{4.4}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_1^{(k)}} = -2\beta\mathbf{x}^T\mathbf{W}^{(k)T}\mathbf{r}^{(k)} \tag{4.5}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_2^{(k)}} = -2\beta\sum_{i=1}^{M} r_i^{(k)}. \tag{4.6}$$

The gradient of the elements of $\mathbf{W}^{(k)}$ with respect to $\boldsymbol{\theta}^{(k)}$ could be found directly, but for projective homographies it is simpler to use a finite difference approximation, because as well as the location, shape and size of the PSF kernel's footprint in the high-resolution image frame, each parameter also affect the entire normalisation of $\mathbf{W}^{(k)}$, as in (3.2), requiring a great deal of computation to find the exact derivatives of each individual matrix element.

4.2 Learning prior strength parameters from data

For most forms of MAP super-resolution, we need to determine values for free parameters like the prior strength, and prior-specific additional parameters like the

Huber-MRF's α value. In order to learn parameter values in a usual ML or MAP framework, it would be necessary to be able to evaluate the partition function (normalization constant), which is a function of ν and α. For example, for the Huber-MRF, we have

$$p(\mathbf{x}) = \frac{1}{Z(\nu,\alpha)} \exp\left\{-\nu \sum_{g\in\mathcal{D}_\mathbf{x}} \rho(g,\alpha)\right\}, \tag{4.7}$$

and so the full negative log likelihood function is

$$\mathcal{L} = \frac{1}{2}\sum_{k=1}^{K} \beta\|\mathbf{r}^{(k)}\|_2^2 - \log p(\mathbf{x}) \tag{4.8}$$

$$= \frac{1}{2}\sum_{k=1}^{K} \beta\|\mathbf{r}^{(k)}\|_2^2 + \nu \sum_{g\in\mathcal{D}_\mathbf{x}} \rho(g,\alpha) - \log Z(\nu,\alpha). \tag{4.9}$$

For a ML solution to ν and α, we would optimize this with respect to those two variables, and could in fact neglect the entire data-error term, since it does not depend on these variables. However, in Section 3.4.2 we showed that the partition function $Z(\nu,\alpha)$ for these sorts of edge-based priors is not easy to compute.

Rather than setting the prior parameters using an ML or MAP technique, therefore, we turn to cross-validation for parameter-fitting. However, it is necessary to determine these parameters while still in the process of converging on the estimates of $\mathbf{x}$, $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$. This is done by removing some *individual low-resolution pixels* from the problem, solving for $\mathbf{x}$ using the remaining pixels, then projecting this solution back into the original low-resolution image frames. The quality is determined by comparing these values with the validation pixels using the L_1 norm, though the L_2 norm or the Huber potential are also suitable and give comparable results in

practice. The selected α and ν should minimize this cross-validation error.

This defines a subtly different cross-validation approach to those used previously for image super-resolution, because validation pixels are selected at random from the collection of KM *individual low-resolution pixels* comprising the overall problem, rather than selecting whole *images* from the K inputs. This distinction is very important when uncertainty in the registrations is assumed, since validation *images* can be misregistered in their entirety. Assuming independence of the registration error on each frame given $\mathbf{x}$, the pixel-wise validation approach has a clear advantage, because pixels from each differently-registered image will be chosen.

4.2.1 Gradient descent on α and ν

In determining a search direction in (ν, α)-space, we make a small change in the parameters, then optimize $\mathcal{L}$ *w.r.t.* $\mathbf{x}$, starting with the current $\mathbf{x}$ estimate, for *just a few steps* to determine whether the parameter combination improves the estimate of $\mathbf{x}$, as determined by cross-validation. The intermediate optimization to re-estimate $\mathbf{x}$ does not need to run to convergence in order to determine whether the new (ν, α)-direction makes an improvement and is therefore worthy of consideration for gradient descent.

This scheme is much faster than the usual approach of running a complete optimization for a number of parameter combinations, especially useful if the initial estimate is poor. An arbitrary 5% of pixels are used for validation, ignoring regions within a few pixels of edges, to avoid boundary complications.

1. **Initialize PSF, image registrations, super-resolution image and prior parameters** according to the initializations given in Section 4.3.4.
2. (a) **(Re)-sample the set of validation pixels**, selecting them as described in section 4.2.
   (b) **Update α and ν (prior parameters)**. Perform gradient descent on the cross-validation error to improve the values of α and ν as in section 4.2.1.
   (c) **Update the super-resolution image and the registration parameters.** Optimize $\mathcal{L}$ (equation 4.2) jointly with respect to $\mathbf{x}$ (super-resolution image), $\boldsymbol{\lambda}$ (photometric transform) and $\boldsymbol{\theta}$ (geometric transform).
3. If the maximum absolute change in α, ν, or any element of $\mathbf{x}$, $\boldsymbol{\lambda}$ or $\boldsymbol{\phi}$ is above preset convergence thresholds, return to 2.

Figure 4.2: **Basic structure of the simultaneous algorithm.** Convergence criteria and thresholds are detailed in section 4.3.1.

4.3 Considerations and algorithm details

In this section, we fill out the remaining details of the simultaneous super-resolution approach, which consists of three distinct components.

The first component incorporates a set of convenient initialization methods for the registrations, prior parameters, and the estimate of $\mathbf{x}$, which even by themselves give a quick and often-reasonable super-resolution estimate. These are structured so as to find as good an estimate as possible in spite of the uncertainty associated with all the variables at this stage.

The second and third algorithm components are the MAP estimation and the the cross-validation regularizer update. These two steps form the body of an iterative loop which continually re-estimates each of the component parts in the super-resolution problem. The algorithm is summarized in Figure 4.2, and the remainder of this section fills out the rest of the details of how each step is implemented.

4.3.1 Convergence criteria and thresholds

Convergence for the *simultaneous* algorithm is defined to be the point at which all parameters change by less than a preset threshold in successive iterations. The outer loop is repeated till this point, typically taking 3-10 iterations.

The threshold is defined differently depending on the nature of the parameter. For $\mathbf{x}$, we look for the point at which the iteration has failed to change any pixel value in $\mathbf{x}$ by more than 0.3 grey levels (*e.g.* 1/850 of the image range). The same threshold number (*i.e.* 0.3) is used for the registration parameter values, after they have been shifted and scaled as described in section 4.3.2. For ν and α, we work with the log values of the parameters (since neither should take a negative value), and look for a change of less than 10^{-4} between subsequent iterations.

In the inner loop iterations use the same convergence criteria as the outer loop, but additionally the number of steps for the update of $\mathbf{x}$ and $\boldsymbol{\theta}$ (algorithm part 2c) is limited to 20, and the number if steps for the prior update (algorithm part 2b) is limited to ten, so that the optimization is divided more effectively between the two groups or parameters.

4.3.2 Scaling the parameters

The elements of $\mathbf{x}$ are scaled to lie in the range $[-\frac{1}{2}, \frac{1}{2}]$, and the geometric registration is decomposed into a "fixed" component, which is the initial mapping from $\mathbf{y}^{(k)}$ to $\mathbf{x}$, and a projective correction term, which is itself decomposed into constituent shifts, rotations, axis scalings and projective parameters, which are the geometric registration parameters, $\boldsymbol{\theta}$. The registration vector for a low-resolution image k, $\boldsymbol{\theta}^{(k)}$, is concatenated with the photometric registration parameter vector, $\boldsymbol{\lambda}^{(k)}$, to

give one ten-element parameter vector per low-resolution image.

A typical image estimate $\mathbf{x}$ might be expected to have a standard deviation of about 0.35 units on its pixel intensity values, where this value is found empirically by averaging over pixel intensities from several high-resolution images. In order to make the vector over which we are optimizing uniform in characteristics, the registration parameter values are shifted and scaled so that they are also zero mean with a *std* of 0.35 units. This is done by considering the distribution of registration vectors over the range of motions anticipated, and assuming each parameter type (*e.g.* shifts or shearing parameters) can be treated independently.

The prior over registration values suggested in Section 4.1 simply needs to penalize large values in the registration vector. This can be accomplished by adding a multiple of the square of each registration parameter value onto the objective function, leading to a Gaussian form, *i.e.*

$$-\log p\left(\boldsymbol{\theta}^{(k)}\right) \;=\; \text{const.} + \frac{1}{2\sigma_\theta^2}\boldsymbol{\theta}^{(k)T}\boldsymbol{\theta}^{(k)}. \tag{4.10}$$

4.3.3 Boundary handling in the super-resolution image

The way in which the boundaries of the super-resolution image are handled is not always described in detail, though it forms an important part of most super-resolution schemes.

The point-spread function produces a "spread" of pixel influences in both directions between the high-resolution image and the low-resolution dataset: each low-resolution pixel depends on a linear combination of high-resolution pixels according to the PSF, and at the same time, each high-resolution pixels is in the

receptive field of several low-resolution pixels.

In order for the generative model to be accurate, the receptive field for the PSF associated with every low-resolution pixel must be fully supported within the high-resolution image. This must be the case because the i^{th} row of the $\mathbf{W}^{(k)}$ matrix describes how the whole super-resolution image contributes to the i^{th} pixel of $\mathbf{y}^{(k)}$, and, other than photometric effects, the equation leaves no room for the pixel to depend on anything *other than* $\mathbf{x}$. Therefore the naïve assumption that it is best only to consider the high-resolution pixels for which we have full coverage in the low-resolution image set is incorrect.

Capel's method expands the support for pixels near the borders of the low-resolution images by introducing a "fixed zone" around the border of the high-resolution image. Intensity values for pixels in this region are roughly approximated in advance using the *average image* (see Section 4.3.4), before the rest of the super-resolution image is estimated by an iterative MAP process. This removes most of the bias which would otherwise be introduced by having low-resolution images with unsupported PSFs, though it still does not guarantee that the supporting "fixed" pixel estimates will be accurate.

A further issue with the fixed-boundary approach is that it assumes a fixed registration, and in a simultaneous setting, we would have to re-estimate the "fixed" boundary at each iteration anyway. A preferable approach is to select a larger region of the super-resolution image to consider in the reconstruction, so that it can support the PSFs of all of the low-resolution images' pixels with extra space to accommodate small changes in the registration. As computing time is not the primary concern for our accurate super-resolution algorithms, the additional processing time required to deal with the larger number of high-resolution pixels is acceptable.

This boundary handling is similar to [112], though we use a different approach to the registration, and so our method can rely simply on the weak prior over $\boldsymbol{\theta}$ (Section 4.3.2) to prevent the registration from drifting far enough off-centre that the PSFs are no longer fully-supported.

4.3.4 Initialization

In our experiments, input images are assumed to be pre-registered by a standard algorithm [57] (*e.g.* RANSAC on features detected in the images) such that points at the image centres correspond to within a small number of low-resolution pixels. This takes us comfortably into the region of convergence illustrated in Figure 4.1 for the global optimum even in cases with considerable repeating texture like the weave pattern shown there.

The Huber image prior parameters are initialized to around $\alpha = 0.01$ and $\nu = \beta/10$; as these are both strictly positive quantities, logs of the values are used. A candidate PSF is selected in order to compute the *average image*, $\mathbf{a}$, which is a stable though excessively smooth approximation to $\mathbf{x}$, as shown in Figure 4.3. Each pixel in $\mathbf{a}$ is a weighted combination of pixels in $\mathbf{y}$, such that a_i depends strongly on y_j if y_j depends strongly on x_i according to the weights in $\mathbf{W}$. Lighting changes must also be taken into consideration, so

$$\mathbf{a} = \mathbf{S}^{-1}\mathbf{W}^T\boldsymbol{\Lambda}_1^{-1}(\mathbf{y} - \boldsymbol{\lambda}_2), \tag{4.11}$$

where $\mathbf{W}$, $\mathbf{y}$, $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\lambda}_2$ are the stacks of the K groups of $\mathbf{W}^{(k)}$, $\mathbf{y}^{(k)}$, $\lambda_1^{(k)}\mathbf{I}$, and $\lambda_2^{(k)}\mathbf{1}$ respectively, and $\mathbf{S}$ is a diagonal matrix whose elements are the column sums of $\mathbf{W}$.

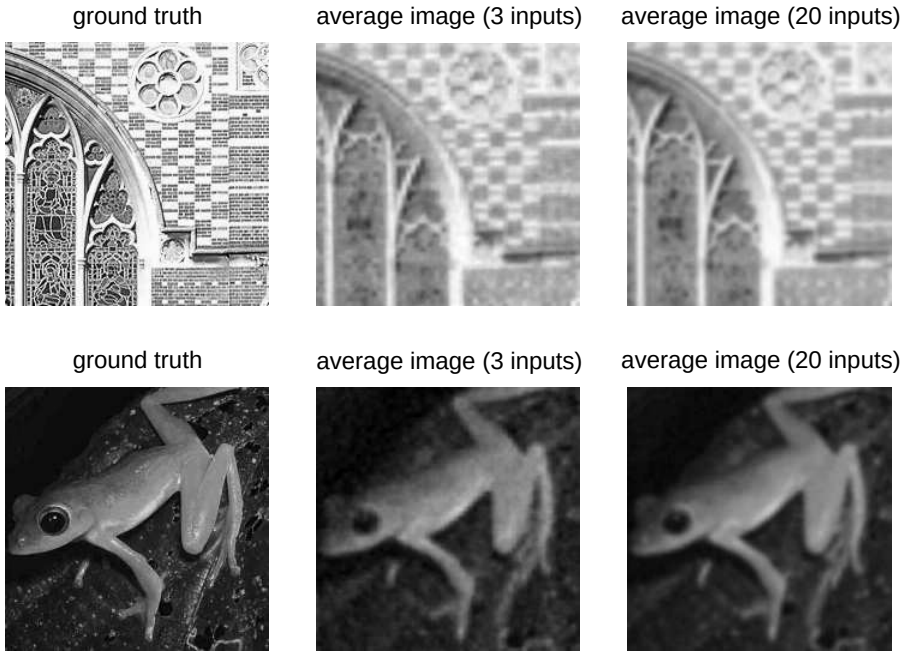


Figure 4.3: **Average Image Examples.** Left: Ground truth image; Centre: Average image using 3 low-resolution input images; Right: Average image using 20 input images. The top row uses the Keble dataset (see Figure 3.4), and the bottom row uses the Frog dataset (Figure 3.17). Both datasets included a noise standard deviation of 10 grey levels on the input images, but note that the average image is still smooth in spite of the relatively high noise level.

Notice that both inverted matrices are diagonal, so $\mathbf{a}$ is simple to compute. Note that the resulting images shown in Figure 4.3 are generally much smoother than the corresponding high-resolution frame calculated from the same input images will be, but that the average image itself is very robust to noise on the low-resolution images. In the work of Cheeseman *et al.* [22], this is referred to as the *composite image*.

In order to get a registration estimate, it is possible to optimize $\mathcal{L}$ of 4.2 with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ only, by using $\mathbf{a}$ in place of $\mathbf{x}$ to estimate the high-resolution pixels. This provides a good estimate for the registration parameters, without requiring $\mathbf{x}$ or the prior parameters, and we refer to the output from this step as the *average image registration*. We find empirically that this out-performs popular alternatives such as mapping images to a common reference frame by bilinear interpolation and setting the registration parameters to minimise the resulting pixel-wise error.

To initialize $\mathbf{x}$, we begin with $\mathbf{a}$, and use the SCG algorithm with the ML solution equations as in (3.15) and (3.15) to improve the result. The optimization is terminated after around $\frac{K}{4}$ steps (where K is the number of low-resolution images), before the instabilities dominate the solution. This gives a sharper starting point than initializing with $\mathbf{a}$ as in [18]. When only a few images are available, a more stable ML solution can be found by using a constrained optimization to bound the pixel values so they must lie in the permitted image intensity range.

Additionally, any pixels in the borders of $\mathbf{x}$ which are not considered in the average image (because they are in the receptive fields of none of the low-resolution pixels at the current PSF size and registration), are initialized by sampling. In the case of a Gaussian prior over $\mathbf{x}$, the entire vector of uninitialized pixels can be sampled using one draw from the marginalized prior distribution. In practice, for priors

such as the Huber-MRF, it is easier to sample pixels sequentially, conditioned only on their neighbours which have already been initialized and progressing outwards. The resulting values, while not equivalent to a single draw from the prior, are both quick to compute and entirely sufficient for the purposes of initialization.

4.4 Evaluation on synthetic data

Before applying the simultaneous algorithm to real low-resolution image data, an evaluation is performed on synthetic data, generated using the generative model (3.1) applied to ground truth images. The first experiment uses only the simultaneous registration and super-resolution part of the algorithm, and the second covers the cross-validation. Before presenting these experiments, it is worth considering the problem of making comparisons with ground-truth images when the registration itself is part of the algorithm.

4.4.1 Measuring image error with registration freedom

When we perform image super-resolution in the simultaneous setting, we learn the registration for each of K low-resolution images with respect to the super-resolution frame. Even when we deal with synthetic data, where there are *true* homographies used to generate the low-resolution images from the ground-truth high-resolution image, the algorithm has no information about the true values, so no *direct* constraints can be placed on the alignment between the ground truth image and the super-resolution image. This means that care has to be taken when comparing the result to ground truth in a quantitative sense.

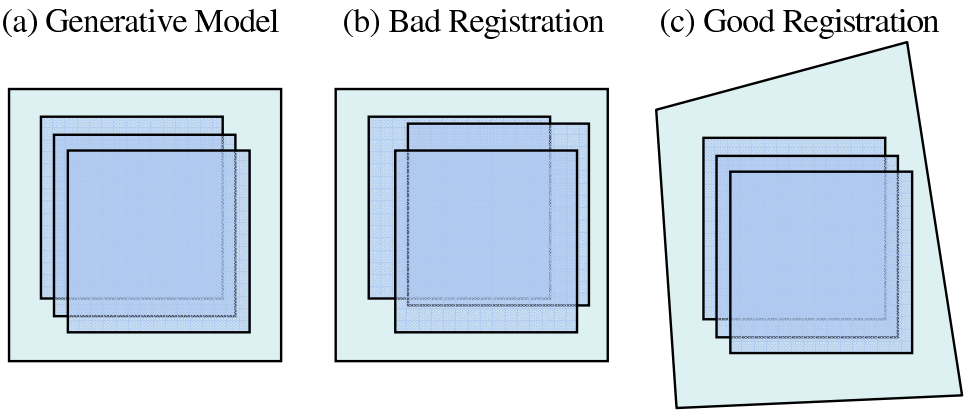


Figure 4.4: **An illustration of *gauge freedom* in the geometric registration.** (a) The generation of the three images from a ground truth scene, given three ground truth registration vectors. (b) The same three images are registered with respect to the frame of a super-resolution image. In this case, the middle frame has drifted with respect to the other two, so the registration as a whole is bad. (c) Another registration of the three images. In this case, the *relative* registrations of the three images are the same as the true registrations, even though their relationship to the "true" scene frame has been warped by a perspective transform. This is a good registration, because all the variation can be accounted for in a single homography between the frame of the ground truth and the frame of the super-resolution image.

When determining the quality of a registration estimate, only the relative registrations of the low-resolution image set are important. Notice that only $K-1$ registration vectors are needed for this; the "free" registration parameter vector, in the case of projective planar homographies, can be thought of as parameterizing the free space of homographies between the ground truth frame and the super-resolution image frame. This freedom is illustrated in Figure 4.4.

If all the low-resolution images are registered perfectly, then there is a single homography which relates the frame of the super-resolution image to that of the ground truth image, and this can be calculated from the pair of "true" and learnt registrations from any one of the low-resolution images, *i.e.* if $\mathbf{H}_{\text{true}}$ maps from $\mathbf{y}^{(k)}$ to the ground truth image, and $\mathbf{H}_{\text{learnt}}$ maps from $\mathbf{y}^{(k)}$ to the super-resolution image,

then the homography from the super-resolution image to the ground truth image is $\mathbf{H}_{\text{true}}\mathbf{H}_{\text{learnt}}^{-1}$. Indeed, if we assume the registration will be perfect, it is sufficient to *clamp* the registration of one of the low-resolution images to its ground truth value, and only optimize to find the other $K-1$ registrations.

However, because we acknowledge the existence of noise in our dataset, we expect a small degree of error in any registration outcome. This means that every real registration is a case like situation (b) in Figure 4.4. In this simple case, *most* registrations would lead to a common overall homography, but were we to select the poorly-registered image as the one whose registration parameters were clamped, then there would be a *global* mis-registration of the two high-resolution frames, and it is likely that image edges in the super-resolution image would not line up exactly with edges in the ground truth frame, for instance, meaning that a large pixel-wise image error is measured for this super-resolution frame.

For all these reasons, we maintain estimates of all K registrations as the K registration vectors, $\boldsymbol{\theta}^{(k)}$, rather than clamping any parameters. Where direct pixel-wise comparisons of super-resolution images are made, the registration is corrected for the global high-resolution mapping by selecting *e.g.* the mean shift across all the images, and the super-resolution estimate is corrected for the additional transform. A few additional iterations, optimizing over $\mathbf{x}$ only, are used in this case to ensure that the image does not deviate from the optimum point of (4.1) due to pixel boundary effects in the re-sampling.

4.4.2 Registration example

Low-resolution images were generated from the synthetic "Eyechart" image at a zoom factor of 4, with each pixel being corrupted by additive Gaussian noise to give a SNR of $30dB$. The image is of text and has 256 grey levels (scaled to lie in $\left[-\frac{1}{2}, \frac{1}{2}\right]$ for the experiments), though the majority of the pixels are black or white. The low-resolution images are 30×30 pixels in size.

Values for a shift-only geometric registration, $\boldsymbol{\theta}$, and a 2D photometric registration $\boldsymbol{\lambda}$ are sampled independently from uniform distributions. The ground truth image and two of the low-resolution images generated by the forward model are shown in Figure 4.5. The mean intensity is clearly different, and the vertical shift is easily observed by comparing the top and bottom edge pixels of each low-resolution image.

An initial registration was then carried out using the *average image registration* technique discussed in section 4.3.4. This is taken as the "fixed" registration for comparison with the joint MAP algorithm, and it differs from the ground truth by an average of 0.0142 pixels, and 1.00 grey levels for the photometric shift. Allowing the joint MAP super-resolution algorithm to update this registration while super-resolving the image resulted in registration errors of just 0.0024 pixels and 0.28 grey levels given the optimal prior settings (see below).

We sweep through values of the prior strength parameter ν, keeping the Huber parameter α set to 0.04. The noise precision parameter β is chosen so that the noise in every case is assumed to have a standard deviation of 5 grey levels, so a single "prior strength" quantity equal to $\log_{10}(\nu/\beta)$ is used. For each prior strength, both the fixed-registration and the joint MAP methods are applied to the data, and the *root mean square error* (RMSE) compared to the ground truth image is calculated.

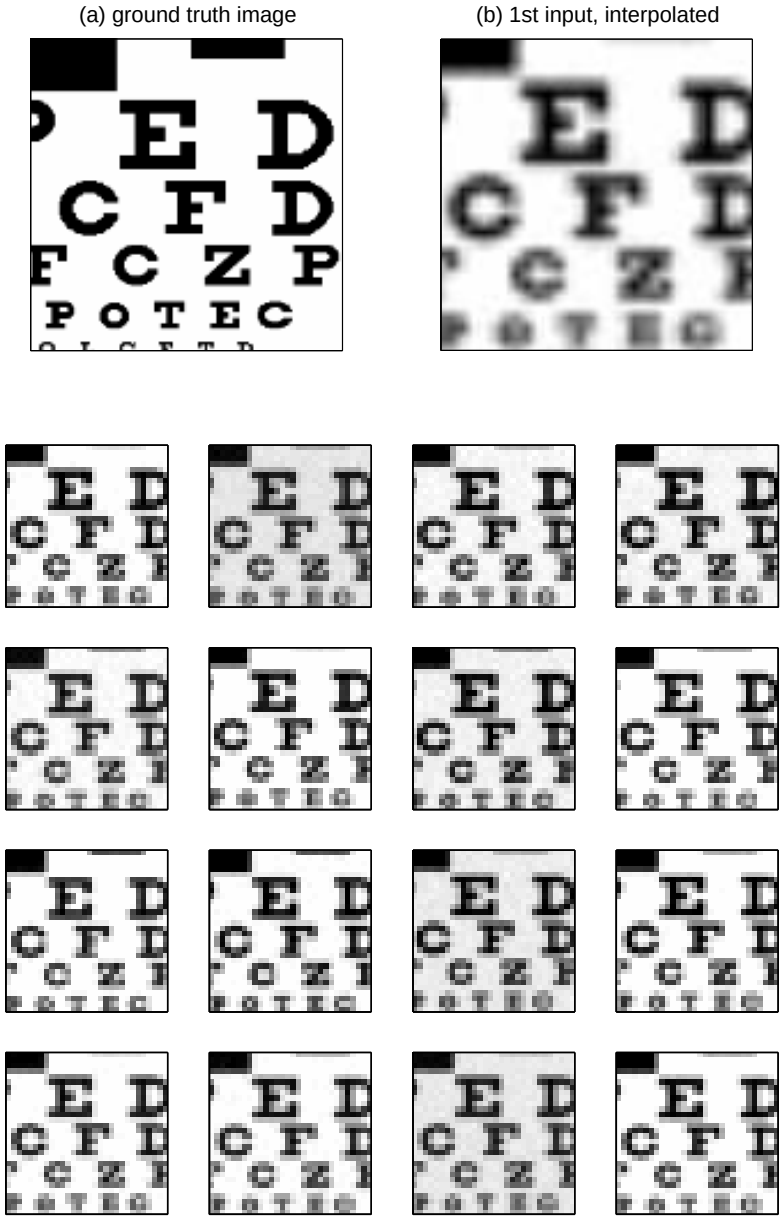


Figure 4.5: **Synthetic Eyechart Data.** (a) Ground truth image (150×150 pixels); (b) First of the 16 low-resolution input images, interpolated up by a factor of 4 in each direction; Bottom: the 16 low-resolution images, 30×30 pixels, generated from the ground truth image at a zoom factor of 4, PSF standard deviation of 0.42 low-resolution pixels, and additive Gaussian noise with an SNR of 30dB.

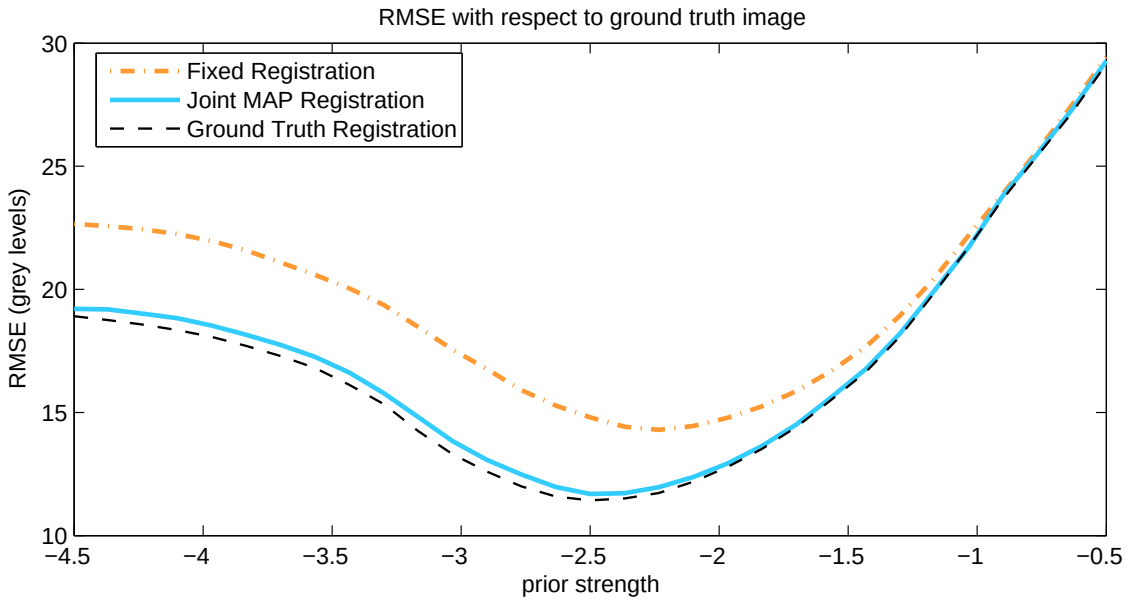


Figure 4.6: **Synthetic data results.** RMSE compared to ground truth, plotted for the fixed and joint MAP algorithms, and for the Huber super-resolution image found using the ground truth registration.

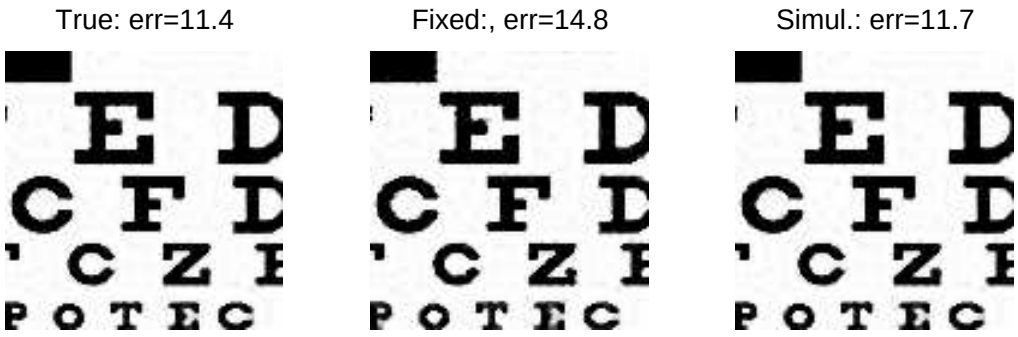


Figure 4.7: **Synthetic data image results.** These three images are recovered using a prior strength setting of -2.5. Left: using the true registration (error is 11.4 grey levels). Centre: using the fixed registration. The error is 14.8 grey levels, and small errors can be seen at the sharb black-white edges of some letters because of the small mis-registrations. Right: using the simultaneous approach; the error is 11.7 grey levels), which is almost as low as the case using ground-truth registration.

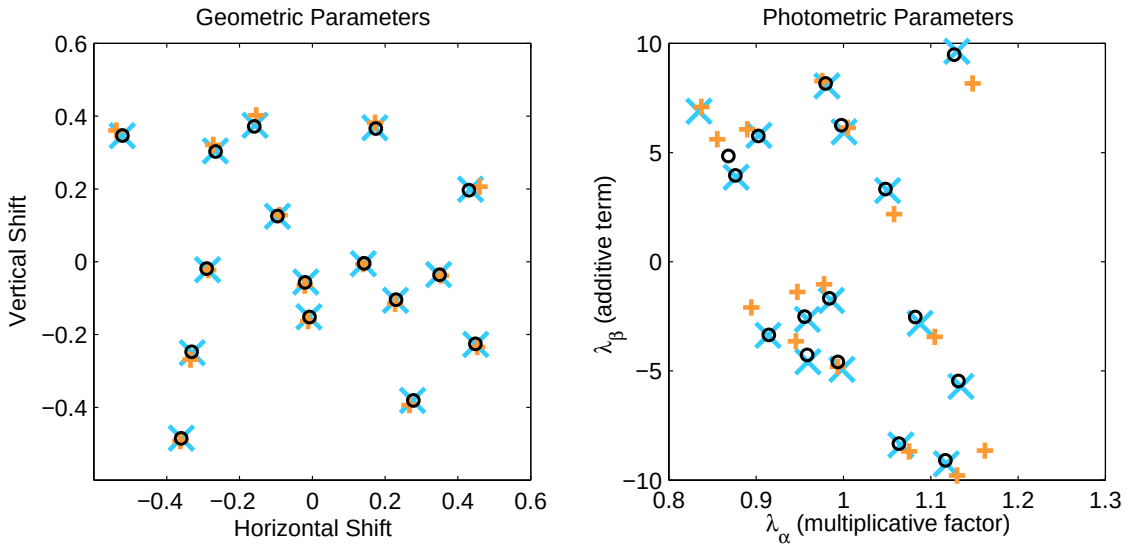


Figure 4.8: **Synthetic data registration results.** These two plots show the registration values for the initial average image method of section 4.3.4 (orange "+"), our joint MAP algorithm (blue "×") and ground truth (black "○") registrations. The left gives the geometric shift parameters, and the right gives the two photometric parameters. In most cases, the joint MAP registration value is considerably closer to the true value than the initial "fixed" value is.

The RMSE compared to the ground truth image for both the fixed registration and the joint MAP approach are plotted, in Figure 4.6, along with a curve representing the performance if the ground truth registration is known. The three images corresponding to the prior strength setting of -2.5 are shown in Figure 4.7. Examples of the improvement in geometric and photometric registration parameters are shown Figure 4.8. Note that we have not learned the prior values in this synthetic-data experiment, in order to plot how the value of ν affects the output.

4.4.3 Cross-validation example

A second experiment is used to verify that the cross-validation-based prior learning phase is viable. The cross-validation error is measured by holding a percentage of

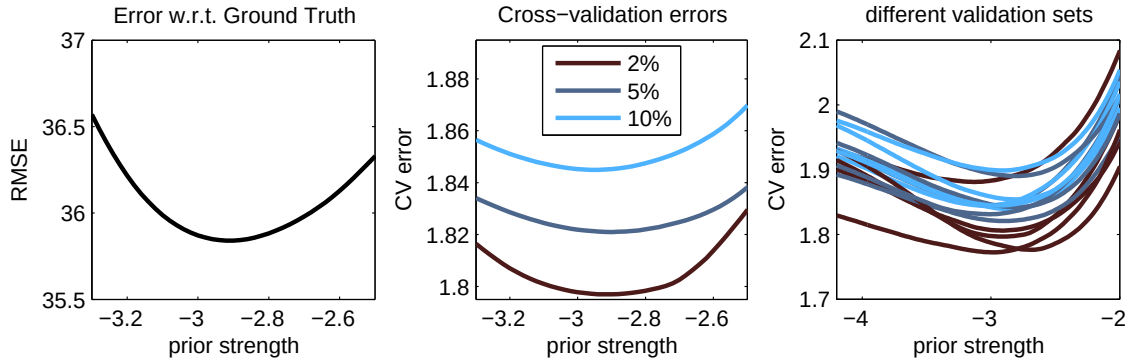


Figure 4.9: **Cross-validation errors on synthetic data.** Left: The error with respect to ground truth on the Keble dataset for this noise level. The minimum point is at a log strength ratio of -2.92. Centre: Three cross-validation curves, corresponding to 2%, 5% and 10% of pixels being selected for the validation sets. The smaller the number of pixels taken out of the reconstruction, the lower the error tends to be. The minima are at log strength ratios of -2.90, -2.90 and -2.96 respectively. Right: More cross-validation curves at each ratio, showing that all the minima lie very close to each other on the curve.

the low-resolution pixels in each image back, and performing Huber-MAP super-resolution using the remaining pixels. The super-resolution image is then projected back down into the withheld set, and the mean absolute error is recorded. This is done for three different validation set sizes (2%, 5% and 10%), and at a range of prior strengths, where all the prior strength values are given as $\log(\nu/\beta)$.

The low-resolution images were generated from the "Keble" image (see Figure 3.4), and the results are plotted in Figure 4.9, along with a plot of the error measured from reconstructing the image using all low-resolution pixels and comparing the results with the ground truth high-resolution image. The best ground-truth comparison occurs when the log prior strength ratio is -2.92. Reconstructed images for strengths of -0.5 and -3 are shown in Figure 4.10 along with the corresponding validation pixels and errors.

In the cross-validation plots shown in the centre of the figure, the curves' minima

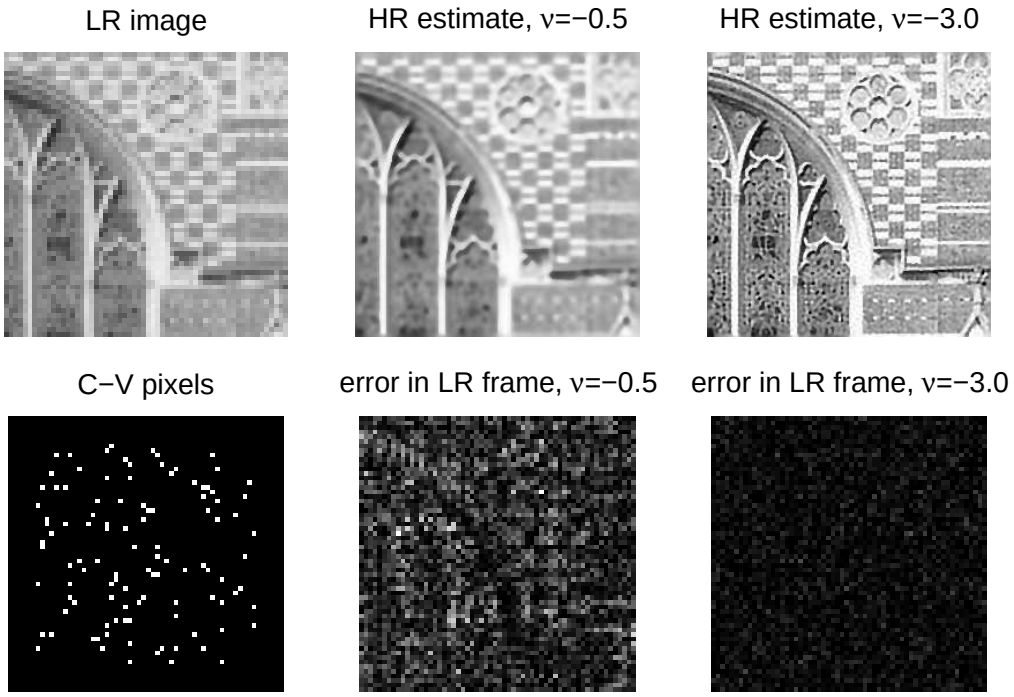


Figure 4.10: **Cross-validation on synthetic data: images.** Images showing cross-validation on the synthetic Keble dataset. Left: one low-resolution image and a set of cross-validation pixels (5%) selected from it. Centre: the high-resolution image reconstructed from the remaining 95% of low-resolution pixels with the prior strength equal to -0.5 (top) and the absolute error when this solution is projected down into the first low-resolution frame. Right: Another high-resolution image and absolute error, this time with the prior strength equal to -3. In both cases, the error values have been multiplied by 12 for ease of viewing.

are at -2.90, -2.90 and -2.96 respectively, which is very close indeed. The final plot shows that for a variety of different random choices of validation pixel sets, the minima are all very close. All the curves are also very smooth and continuous, meaning that we expect optimization using the cross-validation error to be straightforward to achieve.

4.5 Tests on real data

The performance of simultaneous registration, super-resolution and prior updating is evaluated using real data from a variety of sources. This is contrasted with a "fixed-registration" approach, whereby registrations between the inputs are found then fixed before the super-resolution process. This fixed registration is also initialized as in Section 4.3.4, then refined using an intensity-based scheme. Finally $\mathcal{L}$ is optimized *w.r.t.* $\mathbf{x}$ *only* to obtain a high-resolution estimate.

4.5.1 Surrey library sequence

An area of interest is highlighted in the 30-frame Surrey Library sequence from `http://www.robots.ox.ac.uk/~vgg/data/`. The camera motion is a slow pan through a small angle, and the sign on a wall is illegible given any one of the inputs alone. Gaussian PSFs with $std = 0.375, 0.45, 0.525$ are selected, and used in both algorithms. There are 77003 elements in $\mathbf{y}$, and $\mathbf{x}$ has 45936 elements with a zoom factor of 4. $\mathbf{W}$ has around 3.5×10^9 elements, of which around 0.26% are non-zero with the smallest of these PSF kernels, and 0.49% with the largest. Most instances of the simultaneous algorithm converge in 2 to 5 iterations. Results in Figure 4.12 show that while both algorithms perform well with the middle PSF size, the simultaneous-registration algorithm handles the worse PSF estimates more gracefully.

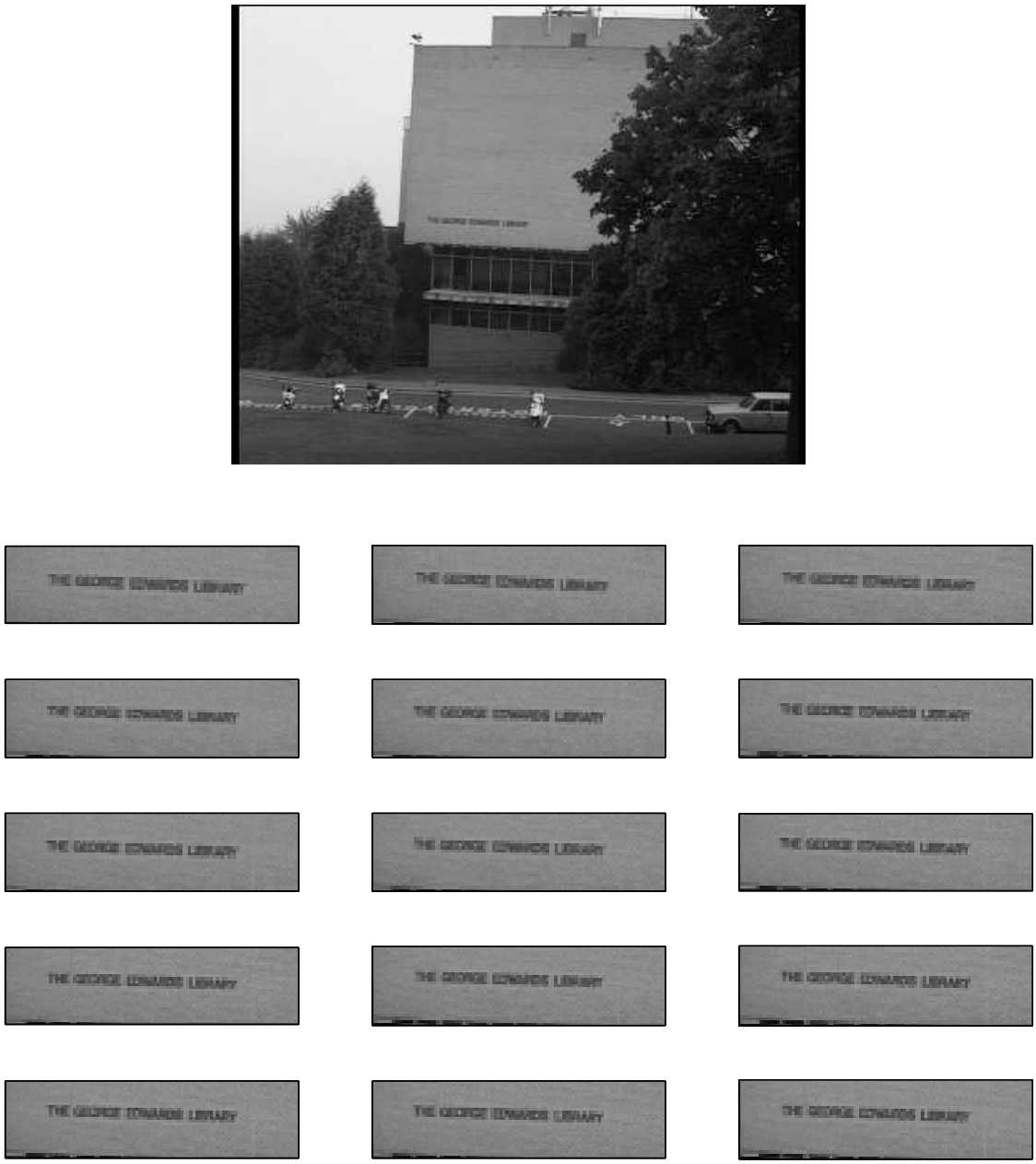


Figure 4.11: **The "Surrey library" dataset.** Top: one of the 30 original frames from the library sequence. Bottom: every second input from the sequence, showing the area cropped out around the region of interest.

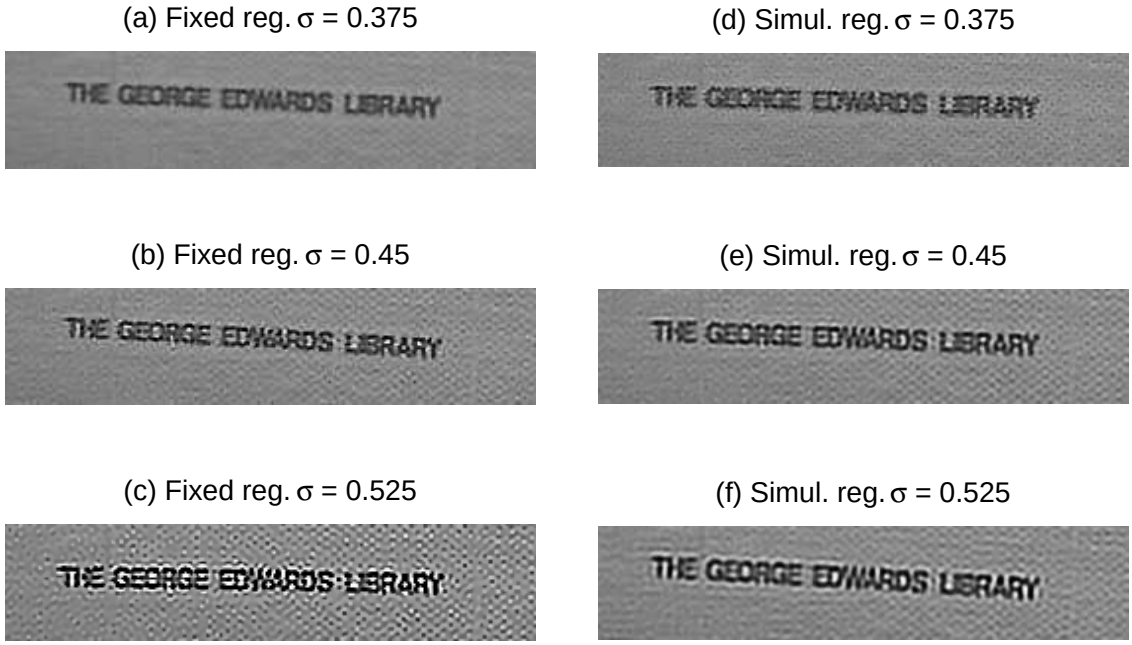


Figure 4.12: **Results on the Surrey Library sequence.** Left column (a,b,c) Super-resolution found using fixed registrations. Right column (d,e,f) Super-resolution images using our algorithm. while both algorithms perform well with the middle PSF size, the simultaneous-registration algorithm handles the worse PSF estimates more gracefully.

4.5.2 Eye-test card sequence

The second experiment uses just 10 images of an eye-test card, captured using a webcam. The card is tilted and rotated slightly, and image brightness varies as the lighting and camera angles change. The inputs are shown in Figure 4.13. Note that in all the low-resolution images, as well as the interpolated version of the first frame, the text on the bottom row of the card is illegible.

Gaussian PSFs with $std = 0.3, 0.375, 0.45$ are used for both the fixed-registration and simultaneous algorithms to super-resolve this dataset. The results are shown in Figure 4.14. Note that the bottom row of text can now be made out, and as before, the simultaneous version gives superior performance over the range of PSF sizes.

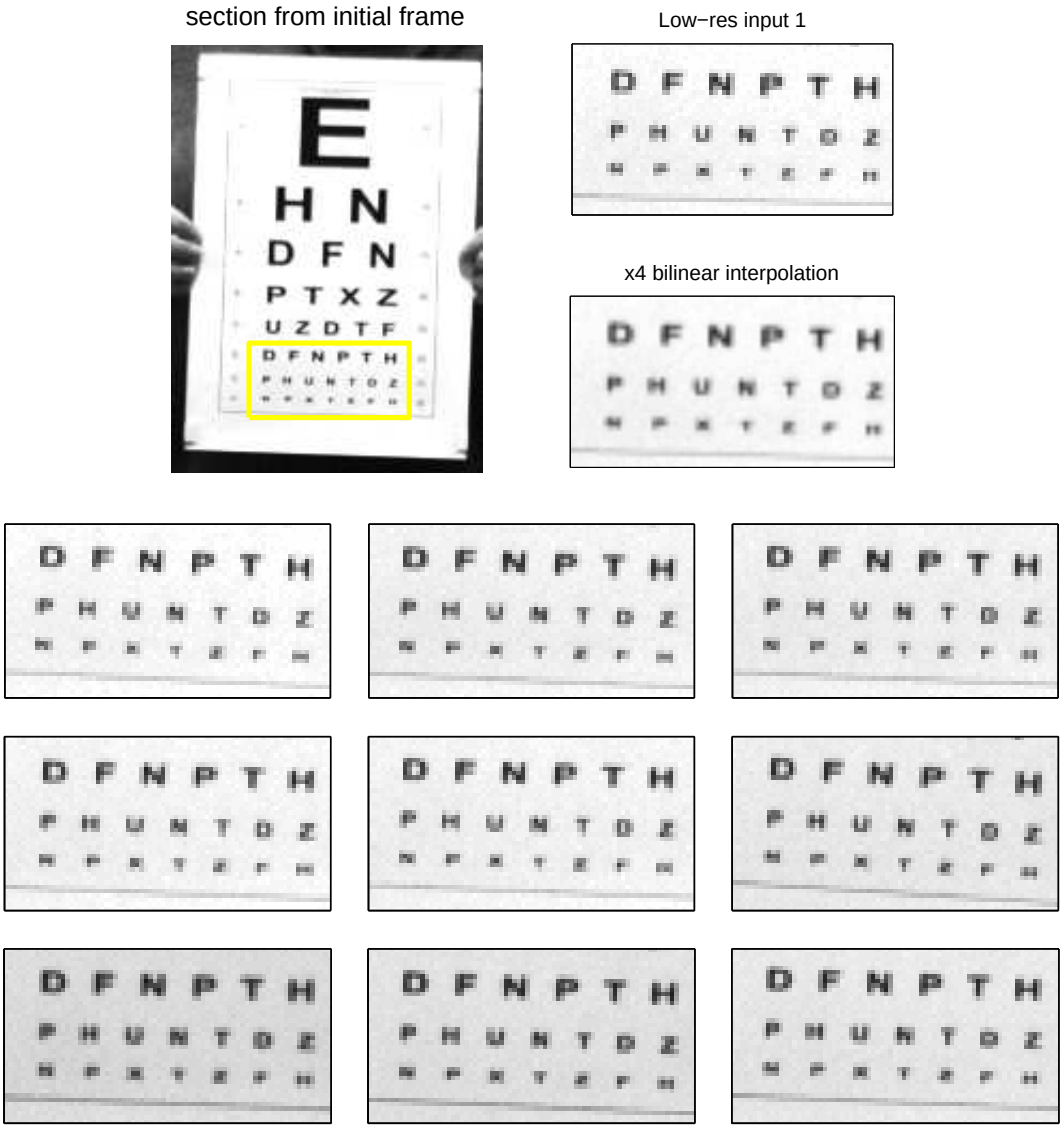


Figure 4.13: **The ten-frame eye-test-card sequence.** Top: one of the original frames, giving the image context and highlighting the region of interest. Top right: The first region of interest, accompanied by a second version whose pixel density has been increased by a factor of four in the horizontal and vertical directions using bilinear interpolation. Bottom: the remaining nine low-resolution input regions of interest.

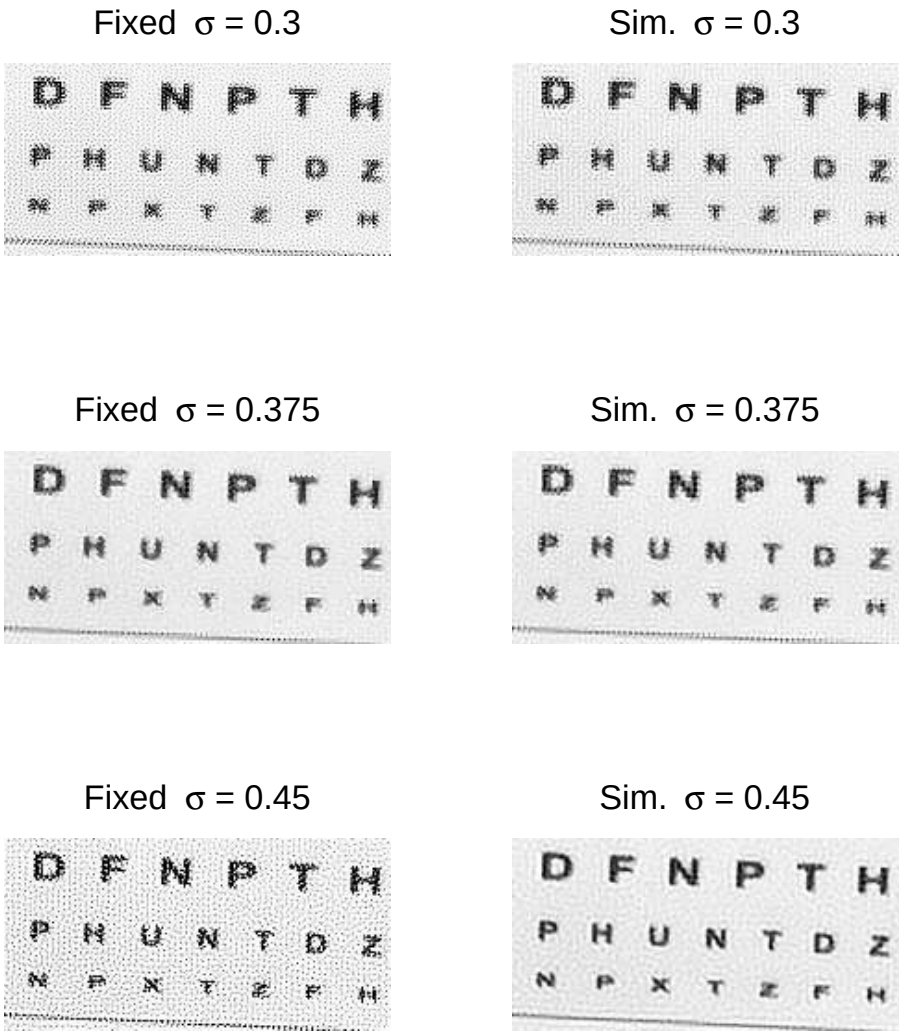


Figure 4.14: **Results on the Eye-chart dataset.** First column: results obtained by fixing registration prior to super-resolution; Second column: results obtained using the simultaneous approach.

4.5.3 Camera "9" sequence

The model is adapted to handle DVD input, where the aspect ratio of the input images is 1.25:1, but they represent 1.85:1 video. The correction in the horizontal scaling is incorporated into the "fixed" part of the homography representation, and the PSF is assumed to be anisotropic. This avoids an undesirable interpolation of

Figure 4.15: **The Camera "9" sequence.** Top Left: Raw DVD frame for Camera "9" sequence, at original aspect ratio. Top Right: the first low-resolution region of interest, along with a version bilinearly interpolated into a super-resolution frame, correcting for the DVD aspect ratio. Bottom: ten of the remaining 28 low-resolution frames.

the inputs prior to super-resolving, which would lose high-frequency information, or working with squashed images throughout the process, which would violate the assumption of an isotropic prior on $\mathbf{x}$. The sequence consists of 29 I-frames[1] from the movie *Groundhog Day*. An on-screen hand-held TV camera moves independently of the real camera, and the logo on the side is chosen as the interest region. Disk-shaped PSFs with radii of 1, 1.4, and 1.8 pixels are used. In both the eye-test card and camera "9" sequences, the simultaneously-optimized super-resolution images again appear subjectively better to the human viewer, and are more consistent across different PSFs.

4.5.4 "Lola" sequences

A selection of results obtained from difficult DVD input sequences (Figure 4.17) take from the movie *Lola Rennt* is shown in Figure 4.18. In the "cars" sequence, there are just 9 I-frames showing a pair of cars, and the areas of interest are the car number plates. The "badge" sequence shows the badge of a bank security officer. Seven I-frames are available, but all are dark, making the noise level proportionally very high. Significant improvements at a zoom factor of 4 (in each direction) can be seen.

4.6 Conclusions

A novel method for combining super-resolution with image registration and the tuning of image prior parameters has been presented. Results on real data from several

[1] I-Frames are encoded as complete images, rather than requiring nearby frames in order to render them.

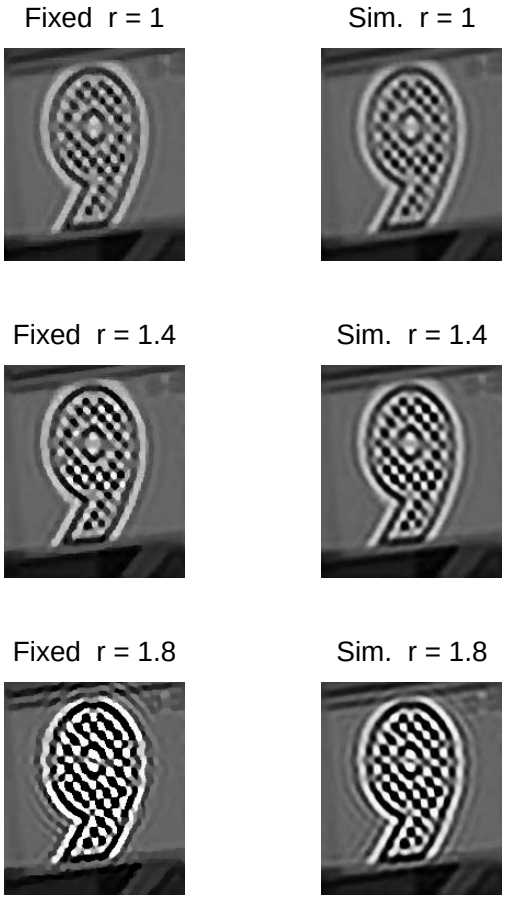
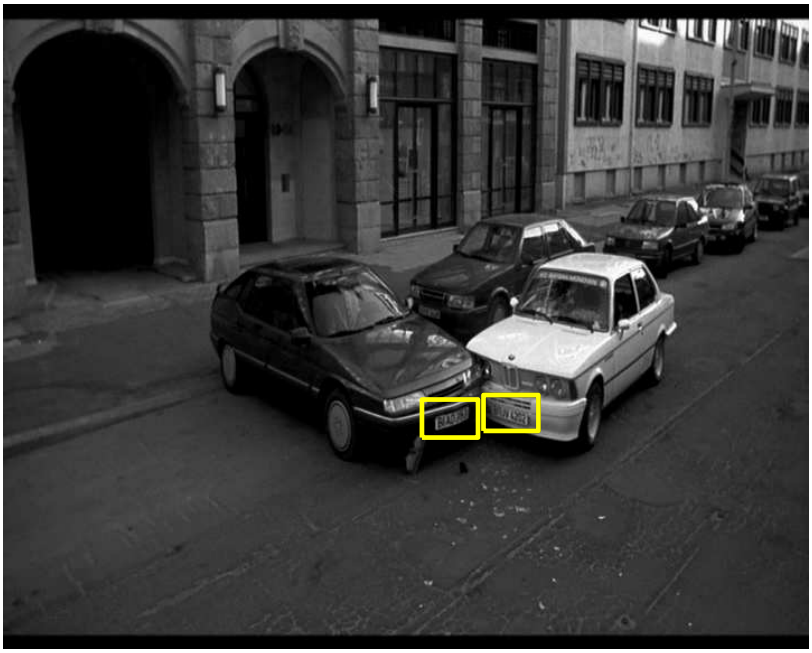


Figure 4.16: **Camera "9" super-resolved outputs.** First column: results obtained by fixing registration prior to super-resolution; Second column: results obtained using the simultaneous approach.

sources show this approach to be superior to the practice of fixing the registration prior to the super-resolution process in terms of eventual registration quality and fidelity to the ground truth high-resolution image in synthetic data experiments.

original DVD frame (cars sequence)



original DVD frame (badge sequence)

Figure 4.17: **Real data from the movie *Lola Rennt* on DVD: two raw DVD frames.**

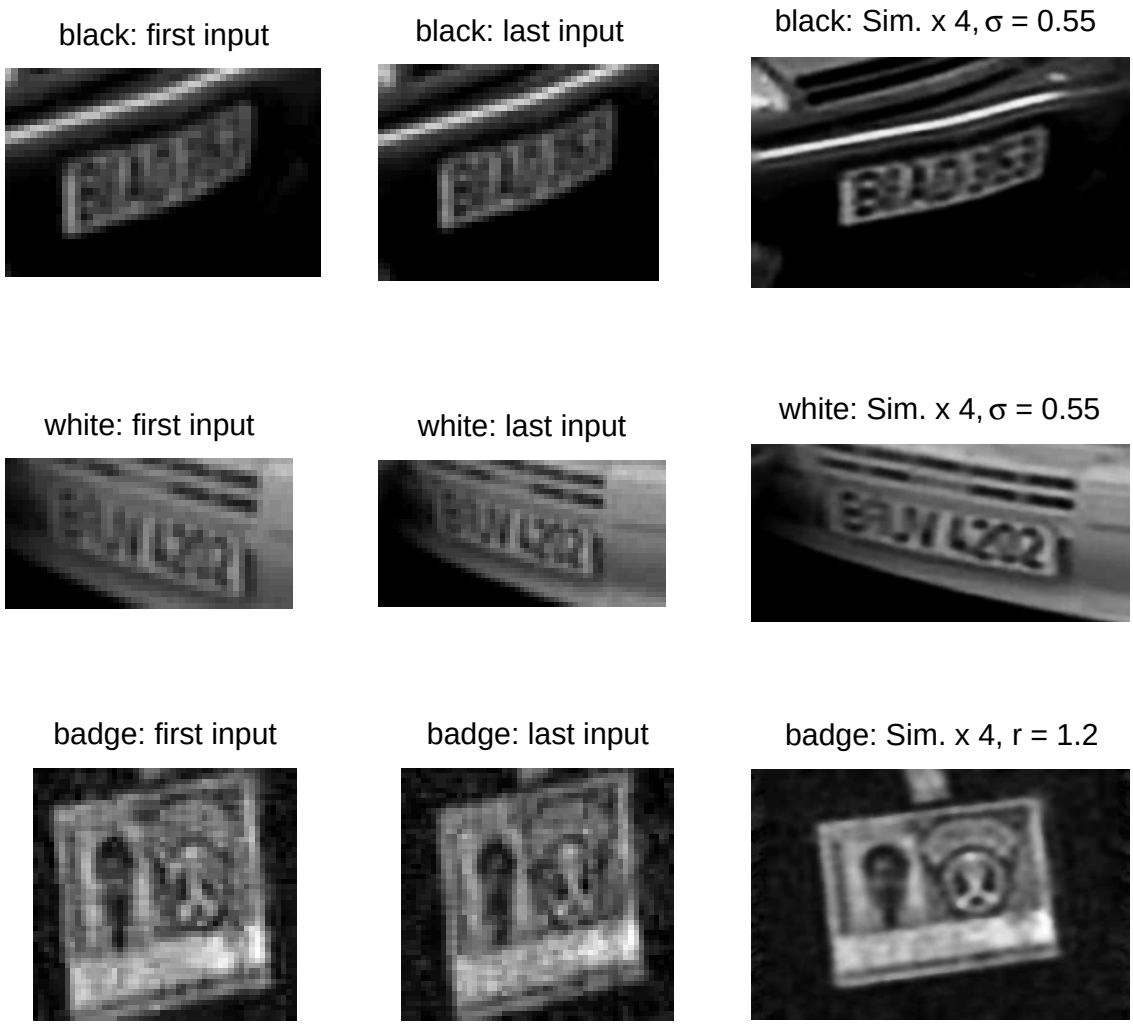


Figure 4.18: **Results from the *Lola Rennt* DVD sequences.** First two columns: the selected interest regions, shown at the DVD aspect ratio. Final (rightmost) column: The same interest regions super-resolved using the simultaneous method. Top: black car's number plate; Middle: white car's number plate; Bottom: security guard's ID badge (intensities have been scaled for ease of viewing). Examples of the full movie frames from which these datasets were taken are shown in Figure 4.17.

Chapter 5

Integrating over the imaging parameters

This chapter develops a method to handle uncertainty in the set of estimated imaging parameters (the geometric and photometric registrations, and the point-spread function) in a principled manner. These parameters are also known as *nuisance parameters* because they are not directly part of the desired output of the algorithm, which in this case is the high resolution image. We take a Gaussian prior over the errors in the preliminary estimate of the registration and marginalize over the possible values, leading to an iterative algorithm for estimating a high-resolution image, which we call the registration-marginalizing approach.

We also examine the alternative Bayesian approach of Tipping and Bishop [112], which marginalizes over the high-resolution image in order to make a *maximum marginal likelihood* point estimate of the imaging parameters. This gives an improvement in the accuracy of the recovered registration (measured against known

truth on synthetic data) compared to the *Maximum a Posteriori* (MAP) approach, but has two important limitations: (i) it is restricted to a Gaussian image prior in order for the marginalization to remain tractable, whereas others have shown improved image super-resolution results are produced using distributions with heavier tails (see the discussion in Section 3.4), and (ii) it is computationally expensive due to the very large matrices required by the algorithm, so the registration is only possible over very small image patches, which take a long time to register accurately. In contrast our approach allows for a more realistic image prior and operates with smaller matrix sizes.

This work was first published in [89], and later developed in [90] to include marginalization of a point-spread function parameter as well as the registration parameters.

5.1 Bayesian marginalization

Both the registration-marginalizing and image-marginalizing derivations proceed along similar lines mathematically, though the effect of choosing a different variable of integration makes a large difference to the way in which the super-resolution algorithms proceed.

5.1.1 Marginalizing over registration parameters

The goal of super-resolution is to obtain a high-resolution image, so our proposed approach treats the other parameters – geometric and photometric registrations and the point-spread function – as "nuisance variables" which might be marginalized out

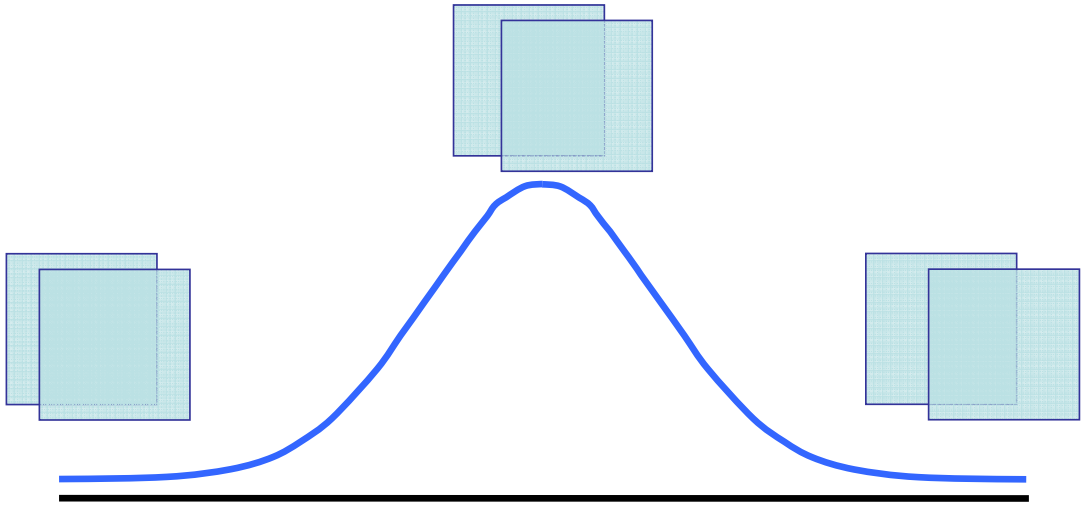


Figure 5.1: **Marginalizing over the registration parameters.** The blue curve shows a Gaussian *p.d.f.* over the possible values taken by a single registration parameter – in this case the horizontal offset of the front-most low-resolution frame. The most probable value, in the centre of the plot, represents the initial point estimate of the image registration. Rather than assigning all other possible registrations zero probability, which is the usual approach, this method assumes that there is a non-zero probability that nearby registrations are correct instead, and that the probability drops as the registration parameter values get further from the initial estimate.

of the problem. This marginalization is illustrated in Figure 5.1.

If we collect these parameters into a single vector, $\boldsymbol{\phi}$, we can write this marginalizations simply as

$$p\left(\mathbf{x}\left|\left\{\mathbf{y}^{(k)}\right\}\right.\right) = \int p\left(\mathbf{x},\boldsymbol{\phi}\left|\left\{\mathbf{y}^{(k)}\right\}\right.\right)\mathrm{d}\boldsymbol{\phi} \tag{5.1}$$

$$= \int \frac{p\left(\mathbf{x},\boldsymbol{\phi}\right)}{p\left(\left\{\mathbf{y}^{(k)}\right\}\right)} p\left(\left\{\mathbf{y}^{(k)}\right\}\left|\mathbf{x},\boldsymbol{\phi}\right.\right)\mathrm{d}\boldsymbol{\phi} \tag{5.2}$$

$$= \frac{p\left(\mathbf{x}\right)}{p\left(\left\{\mathbf{y}^{(k)}\right\}\right)} \int p\left(\left\{\mathbf{y}^{(k)}\right\}\left|\mathbf{x},\boldsymbol{\phi}\right.\right) p\left(\boldsymbol{\phi}\right)\mathrm{d}\boldsymbol{\phi}. \tag{5.3}$$

Notice that $p\left(\mathbf{x},\boldsymbol{\phi}\right) = p\left(\mathbf{x}\right)p\left(\boldsymbol{\phi}\right)$ (because the super-resolution image and registration parameters are independent) and that $p\left(\mathbf{x}\right)$ and $p\left(\left\{\mathbf{y}^{(k)}\right\}\right)$ can be taken

outside the integral. This leaves us free to choose any suitable super-resolution image prior $p(\mathbf{x})$, rather that being constrained to picking a Gaussian merely to make the integral tractable, as in the image-marginalizing case we will discuss in Section 5.1.2.

We require a prior distribution over the registration and PSF parameters, $\boldsymbol{\phi}$, which appear within the integral. We assume that a preliminary image registration (geometric and photometric) and an estimate of the PSF are available, and we also assume that these registration values are related to the ground truth registration values by unknown zero-mean Gaussian-distributed additive noise.

The registration estimate itself can be obtained using classical registration methods, either intensity-based [59] or estimation from image points [57]. There is a rich literature of *Blind Image Deconvolution* concerned with estimating an unknown blur on an image [68].

We introduce a vector $\boldsymbol{\delta}$ to represent the perturbations from ground truth in our initial parameter estimate. For the parameters of a single image, we have

$$\begin{bmatrix} \boldsymbol{\theta}^{(k)} \\ \lambda_1^{(k)} \\ \lambda_2^{(k)} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{\theta}}^{(k)} \\ \bar{\lambda_1}^{(k)} \\ \bar{\lambda_2}^{(k)} \end{bmatrix} + \boldsymbol{\delta}^{(k)} \tag{5.4}$$

where $\bar{\boldsymbol{\theta}}^{(k)}$ and $\bar{\boldsymbol{\lambda}}^{(k)}$ are the estimated registration, and $\boldsymbol{\theta}^{(k)}$ and $\boldsymbol{\lambda}^{(k)}$ are the true registration. The stacked vector $\boldsymbol{\delta}$ is then composed as

$$\boldsymbol{\delta}^T = \left[\boldsymbol{\delta}^{(1)T}, \boldsymbol{\delta}^{(2)T}, \ldots, \boldsymbol{\delta}^{(K)T}, \delta_\gamma\right] \tag{5.5}$$

where the final entry is for the PSF parameter so that $\gamma = \bar{\gamma} + \delta_\gamma$, and $\bar{\gamma}$ is the initial estimate.

The vector $\boldsymbol{\delta}$ is assumed to be distributed according to a zero-mean Gaussian and the diagonal matrix $\mathbf{V}$ is constructed to reflect the confidence in each parameter estimate. This might mean a standard deviation of a tenth of a low-resolution pixel on image translation parameters, or a few grey levels' shift on the illumination model, for instance, giving

$$\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \mathbf{V}). \tag{5.6}$$

The final step before we can perform the integral of (5.3) is to bring out the dependence on $\boldsymbol{\phi}$ in $p\left(\left\{\mathbf{y}^{(k)}\right\}|\mathbf{x}, \boldsymbol{\phi}\right)$. Starting with

$$p\left(\left\{\mathbf{y}^{(k)}\right\}|\mathbf{x}, \boldsymbol{\phi}\right) = \left(\frac{\beta}{2\pi}\right)^{-\frac{KM}{2}} \exp\left\{-\frac{\beta}{2}\mathbf{e}(\boldsymbol{\delta})\right\} \tag{5.7}$$

where

$$\mathbf{e}(\boldsymbol{\delta}) = \sum_{k=1}^{K} \left\| \mathbf{y}^{(k)} - \lambda_1^{(k)} \mathbf{W}\left(\boldsymbol{\theta}^{(k)}, \gamma\right)\mathbf{x} - \boldsymbol{\lambda}_2^{(k)} \right\|_2^2 \tag{5.8}$$

and where where $\boldsymbol{\theta}$, $\boldsymbol{\lambda}$ and γ are functions of $\boldsymbol{\delta}$ and the initial registration values, we can then expand the integral in (5.3) to an integral over $\boldsymbol{\delta}$,

$$\begin{aligned} p\left(\mathbf{x}|\left\{\mathbf{y}^{(k)}\right\}\right) &= \frac{p(\mathbf{x})\left|\mathbf{V}^{-1}\right|^{1/2}\beta^{KM/2}}{p\left(\left\{\mathbf{y}^{(k)}\right\}\right)(2\pi)^{(KM+Kn+1)/2}} \\ &\quad \times \int \exp\left\{-\frac{\beta}{2}\mathbf{e}(\boldsymbol{\delta}) - \frac{1}{2}\boldsymbol{\delta}^T\mathbf{V}^{-1}\boldsymbol{\delta}\right\} \mathrm{d}\boldsymbol{\delta}. \end{aligned} \tag{5.9}$$

We can expand $\mathbf{e}(\boldsymbol{\delta})$ as a second-order Taylor series about the parameter estimates $\left\{\bar{\boldsymbol{\theta}}^{(k)}, \bar{\boldsymbol{\lambda}}^{(k)}\right\}$ and $\bar{\gamma}$ in terms of the vector $\boldsymbol{\delta}$, so that

$$e(\boldsymbol{\delta}) = f + \mathbf{g}^T\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T\mathbf{H}\boldsymbol{\delta}. \tag{5.10}$$

Values for f, $\mathbf{g}$ and $\mathbf{H}$ can be found numerically (for geometric registration parameters) and analytically (for the photometric parameters) from $\mathbf{x}$ and $\left\{\mathbf{y}^{(k)}, \boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}\right\}$. Details of these equations and computations are given in Appendix A.2.

We are now in a position to evaluate the integral in (5.9) using the identity [9]

$$\int \exp\left\{-\mathbf{b}\mathbf{x} - \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}\right\} \mathrm{d}\mathbf{x} = 2\pi^{\frac{d}{2}} |\mathbf{A}|^{-\frac{1}{2}} \exp\left\{\mathbf{b}^T\mathbf{A}^{-1}\mathbf{b}\right\}, \tag{5.11}$$

where d is the dimension of the vector $\mathbf{b}$.

The exponent in the integral in (5.9), becomes

$$a = -\frac{\beta}{2}\mathbf{e}(\boldsymbol{\delta}) - \frac{1}{2}\boldsymbol{\delta}^T\mathbf{V}^{-1}\boldsymbol{\delta} \tag{5.12}$$

$$= -\frac{\beta}{2}f - \frac{\beta}{2}\mathbf{g}^T\boldsymbol{\delta} - \frac{1}{2}\boldsymbol{\delta}^T\left[\frac{\beta}{2}\mathbf{H} + \mathbf{V}^{-1}\right]\boldsymbol{\delta}. \tag{5.13}$$

so that

$$\int \exp\{a\} \mathrm{d}\boldsymbol{\delta} = \exp\left\{-\frac{\beta}{2}f\right\} \int \exp\left\{-\frac{\beta}{2}\mathbf{g}^T\boldsymbol{\delta} - \frac{1}{2}\boldsymbol{\delta}^T\mathbf{S}\boldsymbol{\delta}\right\} \mathrm{d}\boldsymbol{\delta} \tag{5.14}$$

$$= \exp\left\{-\frac{\beta}{2}f\right\} (2\pi)^{\frac{nK+1}{2}} |\mathbf{S}|^{-\frac{1}{2}} \exp\left\{\frac{\beta^2}{8}\mathbf{g}^T\mathbf{S}^{-1}\mathbf{g}\right\} \tag{5.15}$$

where $\mathbf{S} = \left[\frac{\beta}{2}\mathbf{H} + \mathbf{V}^{-1}\right]$ and n is the number of registration parameters (geometric

and photometric) per image. Using this integral result along with (5.9), the final expression for our conditional distribution of the super-resolution image is

$$p\left(\mathbf{x}\left|\left\{\mathbf{y}^{(k)}\right\}\right.\right) = \frac{p\left(\mathbf{x}\right)}{p\left(\left\{\mathbf{y}^{(k)}\right\}\right)}\left(\frac{\beta^{KM}\left|\mathbf{V}^{-1}\right|}{\left(2\pi\right)^{KM}\left|\mathbf{S}\right|}\right)^{\frac{1}{2}}\exp\left\{-\frac{\beta}{2}f+\frac{\beta^{2}}{8}\mathbf{g}^{T}\mathbf{S}^{-1}\mathbf{g}\right\}. \quad (5.16)$$

To arrive at an objective function that we can optimize using gradient descent methods, we take the negative log likelihood and neglect terms which are not functions of $\mathbf{x}$. Using the Huber image prior from Section (3.4.2), this gives

$$\mathcal{L} = \frac{\nu}{2}\rho\left(\mathbf{Dx},\alpha\right)+\frac{\beta}{2}f+\frac{1}{2}\log\left|\mathbf{S}\right|-\frac{\beta^{2}}{8}\mathbf{g}^{T}\mathbf{S}^{-1}\mathbf{g}. \quad (5.17)$$

This is the function we optimize with respect to $\mathbf{x}$ to compute the super-resolution image. The dependence of the various terms on $\mathbf{x}$ can be sumarised

$$\begin{aligned} f(\mathbf{x}) &= \sum_{k=1}^{K}\left\|\mathbf{y}^{(k)}-\lambda_{1}^{(k)}(\boldsymbol{\delta})\mathbf{W}^{(k)}(\boldsymbol{\delta})\mathbf{x}-\boldsymbol{\lambda}_{2}^{(k)}(\boldsymbol{\delta})\right\|_{2}^{2} \qquad \text{(scalar)} & (5.18)\\ \mathbf{g}(\mathbf{x}) &= \frac{\partial f(\mathbf{x})}{\partial\boldsymbol{\delta}} \qquad (p\times 1 \text{ gradient vector}) & (5.19)\\ \mathbf{H}(\mathbf{x}) &= \frac{\partial\mathbf{g}(\mathbf{x})}{\partial\boldsymbol{\delta}} \qquad (p\times p \text{ Hessian matrix}) & (5.20)\\ \mathbf{S}(\mathbf{x}) &= \frac{\beta}{2}\mathbf{H}(\mathbf{x})+\mathbf{V}^{-1} \qquad (p\times p \text{ matrix}), & (5.21) \end{aligned}$$

where $\boldsymbol{\delta}$ is the p-element vector of "nuisance variables" (*e.g.* registrations and PSF size), which is assumed to be Gaussian distributed with covariance $\mathbf{V}$.

5.1.2 Marginalizing over the super-resolution image

We will outline the marginalization method used in [112] here, since it is useful for comparison with our method, and also because the model used here extends theirs, by adding photometric parameters, which introduces extra terms to the equations.

The prior used in [112] takes the form of a zero-mean Gaussian over the pixels in $\mathbf{x}$ with covariance $\mathbf{Z}_x$. A simplified version has already been discussed in Section 3.4.1 (equations (3.22) and (3.31)), but if we consider the exact form of the probability and its normalizing constant, it is

$$p(\mathbf{x}) \;=\; (2\pi)^{-\frac{N}{2}} \, |\mathbf{Z}_x|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \mathbf{x}^T \mathbf{Z}_x^{-1} \mathbf{x} \right\}. \tag{5.22}$$

This is used to facilitate the marginalization over the super-resolution pixels in order to arrive at an expression for the marginal probability of the low-resolution image set conditioned only on the set of imaging parameters. Taking $\mathbf{y}$ to be a stacked vector of all the input images, and $\boldsymbol{\lambda}_2$ to be a stack of the $\boldsymbol{\lambda}_2^{(k)}$ vectors, this distribution is

$$\mathbf{y} \;=\; \mathcal{N}\left(\mathbf{y} \,|\boldsymbol{\lambda}_2, \mathbf{Z}_y\right), \tag{5.23}$$

where

$$\mathbf{Z}_y \;=\; \beta^{-1}\mathbf{I} + \boldsymbol{\Lambda}_1 \mathbf{W} \mathbf{Z}_x \mathbf{W}^T \boldsymbol{\Lambda}_1^T. \tag{5.24}$$

Here $\boldsymbol{\Lambda}_1$ is a matrix whose diagonals are given by the $\lambda_1^{(k)}$ values of the corresponding low-resolution images, and $\mathbf{W}$ is the stack of individual $\mathbf{W}^{(k)}$ matrices. This

expression is derived in Appendix B.1.2.

The objective function, which does not depend on $\mathbf{x}$, is optimized with respect to $\left\{\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}\right\}$ and γ, and is given by

$$\mathcal{L} = \frac{1}{2}\left[\beta \sum_{k=1}^{K} \left\| \mathbf{y}^{(k)} - \lambda_1^{(k)} \mathbf{W}^{(k)} \boldsymbol{\mu} - \boldsymbol{\lambda}_2^{(k)} \right\|_2^2 + \boldsymbol{\mu}^T \mathbf{Z}_x^{-1} \boldsymbol{\mu} - \log|\boldsymbol{\Sigma}|\right] \quad (5.25)$$

where

$$\boldsymbol{\Sigma} = \left[\mathbf{Z}_x^{-1} + \beta \sum_{k=1}^{K} \lambda_1^{(k)2} \mathbf{W}^{(k)T} \mathbf{W}^{(k)}\right]^{-1} \quad (5.26)$$

$$\boldsymbol{\mu} = \beta \boldsymbol{\Sigma} \left(\sum_{k=1}^{K} \lambda_1^{(k)} \mathbf{W}^{(k)T} \left(\mathbf{y}^{(k)} - \boldsymbol{\lambda}_2^{(k)} \right) \right), \quad (5.27)$$

which is derived in Appendix B.1.1. The gradient of this expression with respect to the registration parameters is derived in Appendix B.2.

The expression for the posterior mean $\boldsymbol{\mu}$ is the closed form of the overall MAP solution for the super-resolution image. However, in [112], the optimization over registration and blur parameters is carried out with low-resolution image patches of just 9×9 pixels, rather than the full low-resolution images, because of the computational cost involved in computing the terms in (5.25) — even for a tiny 50×50-pixel high-resolution image, the $\mathbf{Z}_x$ and $\boldsymbol{\Sigma}$ matrices are 2500×2500. The full-sized super-resolution image can then be computed by fixing the optimal registration and PSF values and finding $\boldsymbol{\mu}$ using the full-sized low-resolution images, $\mathbf{y}^{(k)}$, rather than the 9×9 patches. This is exactly equivalent to solving the usual MAP super-resolution approach of Section 3.3.2, with $p(\mathbf{x})$ defined as in (3.22), using the covariance of (3.31).

In comparison, the dimensionality of the matrices in the terms comprising the registration-marginalizing objective function (5.17) is in most cases much lower than those in (5.25). This means the terms arising from the marginalization are far less costly to compute, so our algorithm can be run on entire low-resolution images, rather than just patches.

5.1.3 Implementation notes

The objective function (5.17) can be optimized using *Scaled Conjugate Gradients* (SCG) [77], noting that the gradient can be expressed

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{x}} &= \frac{\nu}{2}\mathbf{D}^T\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\rho\left(\mathbf{D}\mathbf{x},\alpha\right)+\frac{\beta}{2}\frac{\mathrm{df}}{\mathrm{d}\mathbf{x}}-\frac{\beta^2}{4}\boldsymbol{\xi}^T\frac{\mathrm{d}\mathbf{g}}{\mathrm{d}\mathbf{x}} \\ &\quad+\left[\frac{\beta}{4}\mathrm{vec}\left(\mathbf{S}^{-1}+\frac{\beta^2}{8}\boldsymbol{\xi}\boldsymbol{\xi}^T\right)^T\right]\frac{\mathrm{dvec}(\mathbf{H})}{\mathrm{d}\mathbf{x}},\end{aligned} \tag{5.28}$$

where

$$\boldsymbol{\xi} = \mathbf{S}^{-1}\mathbf{g}, \tag{5.29}$$

and where *vec* is the matrix vectorization operator. Derivatives of f, $\mathbf{g}$ and $\mathbf{H}$ with respect to $\mathbf{x}$ can be found analytically for photometric parameters, and numerically (using the analytic gradient of $e^{(k)}\left(\boldsymbol{\delta}^{(k)}\right)$ with respect to $\mathbf{x}$) with respect to the geometric parameters. A derivation of the gradient expression is given in Appendix A.1.

The upper part of $\mathbf{H}$ is block-diagonal $nK \times nK$ sparse matrix, and the final $(nK+1)^{\mathrm{th}}$ row and column are non-sparse, assuming that the blur parameter is

shared between the images, as it might be in a short video sequence, for instance, and that the image registration errors for two different images are independent. Notice that the value f in (5.17) is simply the reprojection error of the current estimate of $\mathbf{x}$ at the mean registration parameter values, *i.e.* the value of (5.8) evaluated at $\bar{\boldsymbol{\theta}}^{(k)}$, $\bar{\boldsymbol{\lambda}}^{(k)}$ and $\bar{\gamma}$. Gradients of this expression with respect to the $\boldsymbol{\lambda}$ parameters, and with respect to $\mathbf{x}$ can both be found analytically. To find the gradient with respect to a geometric registration parameter $\theta_i^{(k)}$, and elements of the Hessian involving it, a central difference scheme involving only the k^{th} image is used. Details of the construction of the Taylor expansions and their gradients with respect to $\mathbf{x}$ are in Appendix A.2.

Initialization

Mean values for the registration are computed by standard registration techniques, and $\mathbf{x}$ is initialized at these parameters, first using the average image, then refining the estimate using a few (typically around $\frac{K}{4}$) iterations of the ML solution to give a sharp estimate.

For all our experiments, pixel values are scaled to lie between $-\frac{1}{2}$ and $\frac{1}{2}$, in accordance with [112], and the $\mathbf{W}$ matrices are computed assuming a Gaussian PSF, so γ represents the length scale parameter (standard deviation) of a Gaussian blur kernel.

In our implementation, the parameters representing the photometric registration values are scaled so that they share the same standard deviations as the $\boldsymbol{\theta}$ parameters, which represent the sub-pixel geometric registration shifts, which makes the matrix $\mathbf{V}$ a multiple of the identity. The scale factors are chosen so that one standard deviation in λ_2 gives a 10-grey-level shift, and one standard deviation in λ_1

varies pixel values by around 10 grey levels at mean image intensity.

5.2 Results

The first three experiments use synthetic datasets in order to allow a quantitative measure of performance to be taken with respect to known ground truth high-resolution images. In the first two experiments, only the marginalization over the geometric and photometric registrations is tested, then the PSF parameter is also introduced to the list over which the system integrates. The final experiment shows the full system working on real data, and compares the results to the standard Huber-MAP method, and to the approach of [112].

5.2.1 Butterfly sequence

The first experiment in this section demonstrates the improvement this registration marginalization can attain on a simple synthetic dataset. The generative model is used to generate 16-image datasets at a zoom factor of four, with a fixed PSF (standard deviation 0.2 low-resolution pixels), and low noise (30dB, corresponding to additive noise with standard deviations of approximately 1.55 grey levels for this image). Each 30×30-pixel low-resolution image is related to the ground truth image by a 2D translation-only motion model (in addition to the zoom factor), and the two-parameter global linear illumination model, giving a total of four registration parameters per low-resolution image. The sub-pixel perturbations are evenly spaced over a grid up to plus or minus one half of a low-resolution pixel, giving a similar setup to that described in [83], but with additional lighting variation.

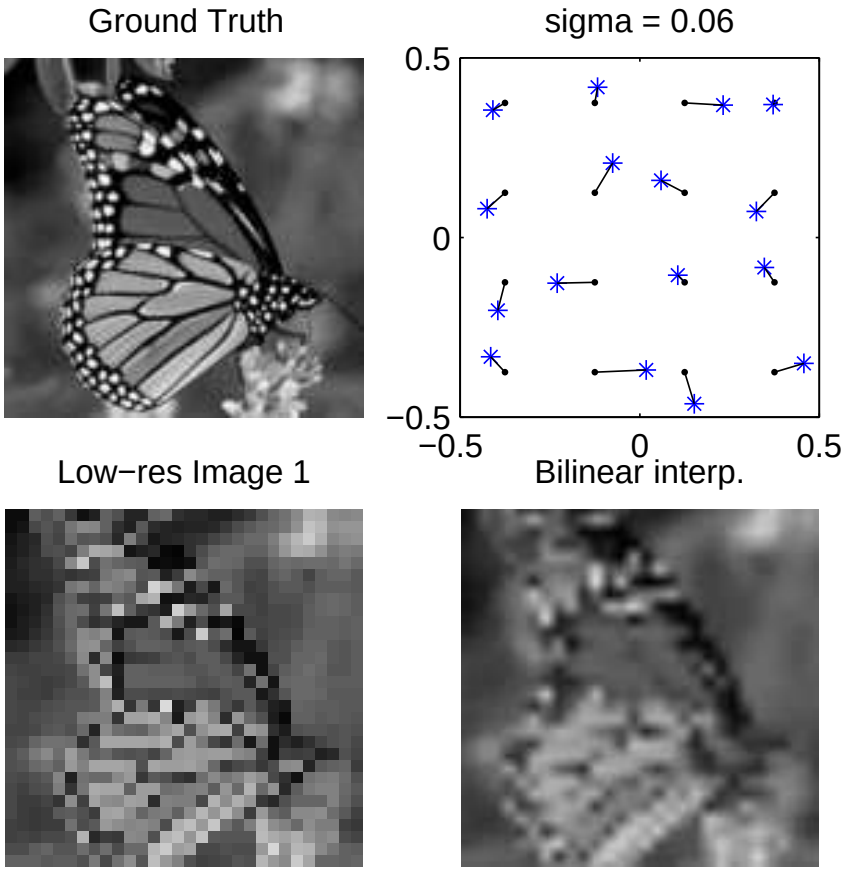


Figure 5.2: **The synthetic "butterfly" dataset.** Top left: ground truth high-resolution image. Top right: example of sub-pixel image registration and artificially-introduced geometric registration errors, plotted on a single low-resolution image pixel. Dots on the grid denote the true shift sizes (measured in low-resolution pixels), and blue stars mark the values as passed to the super-resolution reconstruction algorithms. Bottom left: an example low-resolution image. Bottom right: a second image warped into the high-resolution image frame using bilinear interpolation, increasing the number of pixels by a factor of 16. See Figure 5.3 and Table 5.1 for the super-resolution results.

Before solving to find super-resolution images, the registration data are corrupted by the addition of *i.i.d.* Gaussian noise. The noise samples corrupting the dataset have standard deviations of 0.02–0.14 low-resolution pixels, with corresponding intensity shift standard deviations of around 1.53–10.71 grey-levels.

Figure 5.2 shows the ground-truth butterfly image, a plot of the image registration errors within a single low-resolution pixel for the case when the standard deviation on the registration error is 0.06 low-resolution pixels, and two of the sixteen low-resolution images, one of which has been interpolated up to the original image resolution size using bilinear interpolation.

Three super-resolution algorithms are used to construct super-resolution images from these datasets: ML, plain Huber-MAP and registration-marginalizing (which also uses the same Huber prior). In this first experiment we will consider integrating only over the geometric and photometric registrations, so the true value of the PSF parameter is supplied to all three algorithms. In both cases where the Huber prior is involved, the Huber parameter is set to $\alpha = 0.04$. For the registration-marginalizing approach, the error distribution over the registration parameters is assumed to have a standard deviation of 0.005 pixels, which is found to perform well in spite of being smaller than the true registration perturbation distributions.

Figure 5.3 shows the results of the three super-resolution algorithms. The ML case quickly becomes unrecognizable as a butterfly as the registration error increases. The Huber-MAP approach (using prior strength $\log_{10}(\nu/\beta) = -1.6$) performs better, but errors and speckling in the vicinity of the black-white edges become apparent as the image registration noise level increases. The most plausible-looking high-resolution image for each of the four datasets comes from the registration-marginalizing approach, which succeeds in maintaining much cleaner transitions at

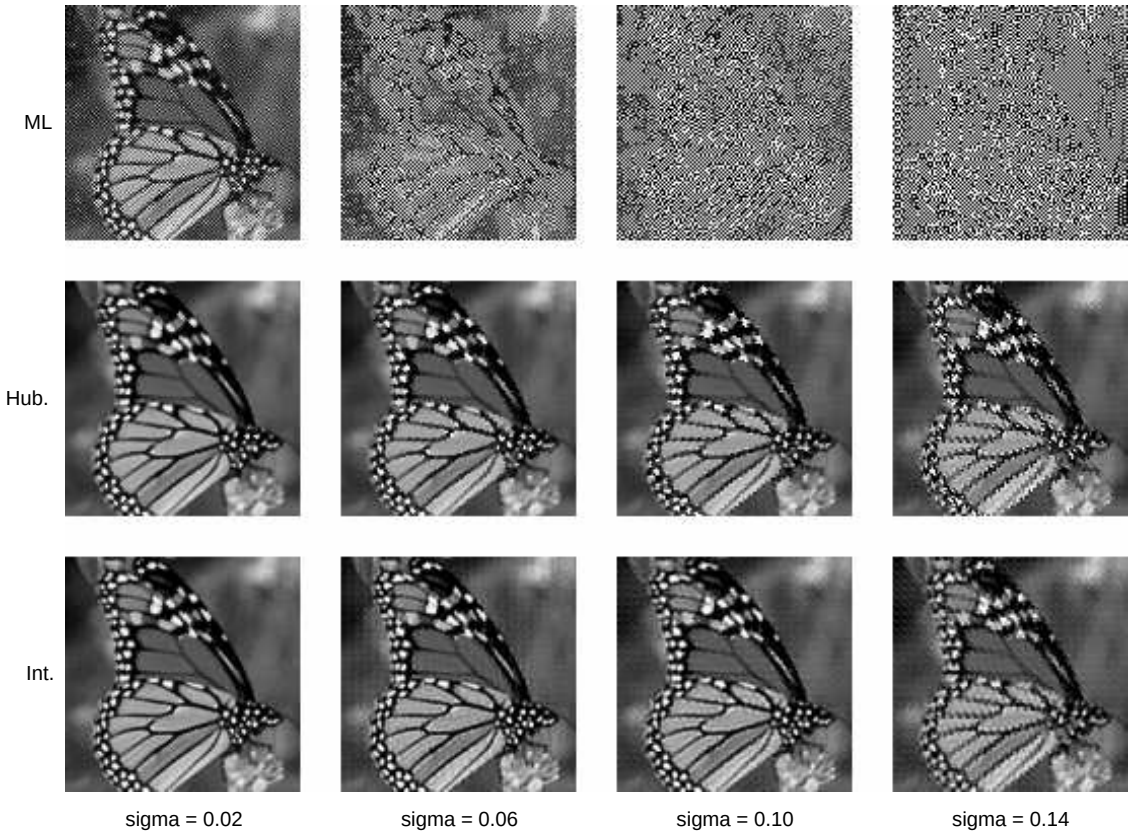


Figure 5.3: **Results on the synthetic "butterfly" dataset.** Super-resolution images found using ML (top), Huber-MAP (middle) and registration-marginalizing recovery methods. The mis-registration error gets progressively worse from left to right, with the standard deviations being 0.02, 0.06, 0.10, and finally 0.14 low-resolution pixels. The bottom right corner (registration-marginalizing) image is visibly better than the equivalent Huber image directly above. The RMS errors for each image compared to the ground truth high-resolution image are given in Table 5.1.

RMSE	*std* of registration error			
Method	0.02	0.06	0.10	0.14
ML	50.56	328.07	904.91	1771.06
Huber	10.61	15.59	20.12	31.53
Integr. A	7.33	11.89	13.45	21.32
Integr. B	9.27	12.96	13.10	22.62

Table 5.1: **RMS errors for the butterfly experiment.** The errors are measured with respect to the ground truth image, which is shown in Figure 5.3. The two rows for the registration-marginalizing approach indicate the error scores for different ν values in the Huber prior. The "A" row uses a slightly weaker prior (see text) and corresponds to the "Int" images of Figure 5.3, whereas the "B" row uses a stronger prior equal to that used in the Huber-MAP super-resolution algorithm, and is given here for comparison only.

the edge boundaries. The strength of the Huber prior used in the registration-marginalizing approach does not need to be as high as the regular Huber-MAP approach, since some of the regularization power comes from the new terms in the objective function (5.17), so in this case we choose $\log_{10}(\nu/\beta) = -2.2$. Considering only the images in the final column, the registration-marginalizing approach reduces the RMSE by over 32%.

Table 5.1 gives the RMSE for each of the images of Figure 5.3 (ML, Huber and Integr. "A"). Also given for comparison are the RMSE values for the registration-marginalizing algorithm when the stronger image prior ($\log_{10}(\nu/\beta) = -1.6$) is used (Integr. "B").

5.2.2 Synthetic eyechart sequence

The second experiment examines the behaviour of the registration-marginalizing method over a larger range or parameter values, again without marginalizing over the point-spread-function. To show that the approach can handle multiple classes

of image, the synthetic Eyechart sequence of Figure 4.5 is used.

Geometric and photometric registration parameters were initialized to the identity, and the images were registered using an iterative intensity-based scheme. The resulting parameter values were used to recover two sets of super-resolution images: one using the standard Huber-MAP algorithm, and the second using our extension integrating over the geometric and photometric registration uncertainties. The Huber parameter α was fixed at 0.01 for all runs, and ν was varied over a range of possible values representing ratios between ν and the image noise precision β.

The images giving lowest RMS error from each set are displayed Figure 5.4. Visually, the differences between the images are subtle, though the bottom row of letters is better defined in the output from our algorithm. Plotting the RMSE as a function of ν in Figure 5.5, we see that the registration-marginalizing approach achieves a lower error, compared to the ground truth high-resolution image, than the standard Huber-MAP algorithm for any choice of prior strength, $\log_{10}(\nu/\beta)$. Because ν and β are free parameters in the algorithm, it is an advantage that the marginalizing approach is less sensitive to variation in their values.

5.2.3 Face sequence

The face experiment uses a sixteen-image synthetic dataset generated from a face image in the same manner as the eyechart dataset above. The image size is smaller – 15×15 pixels – and the noise level is set to $25dB$ (approximately 3.94 grey levels). It is difficult to identify the individual in any one of the low-resolutions images; the full set of 16 is shown in Figure 5.6.

In addition to the error in registration parameters arising from registering these

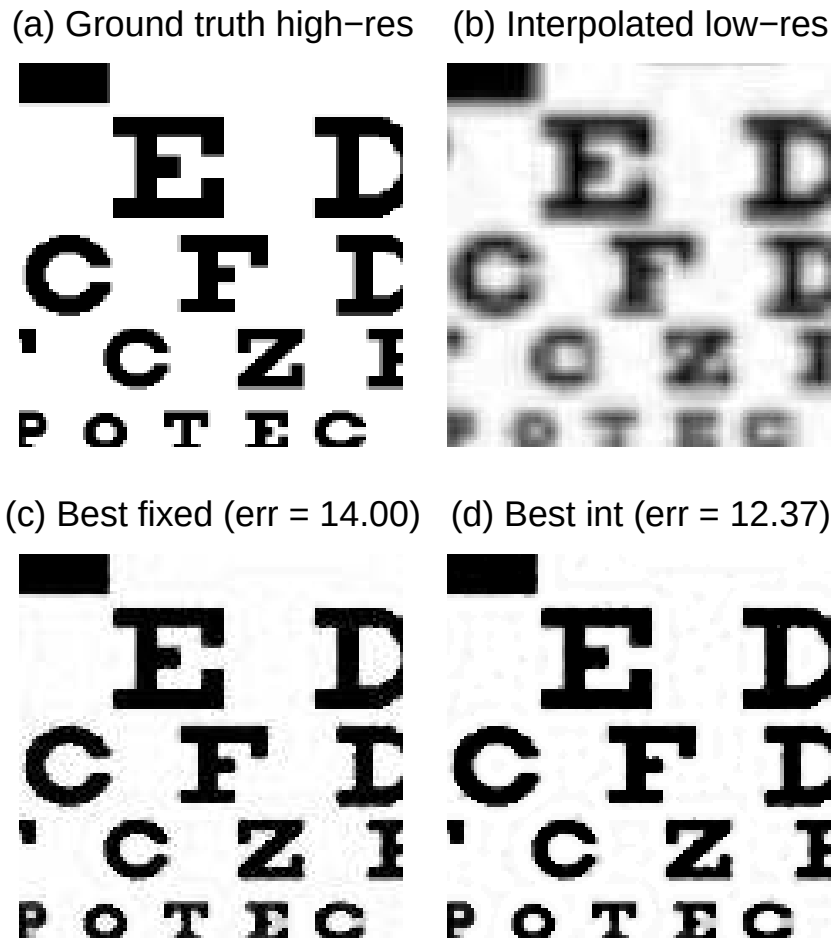


Figure 5.4: **Super-resolving the eyechart dataset.** (a) ground truth image (only the central recoverable part is shown); (b) interpolated low-resolution image; (c) best (minimum MSE) image from the regular Huber-MAP algorithm, having super-resolved the dataset multiple times with different prior strength settings; and (d) the best result using our approach of integrating over $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$. As well as having a lower RMSE, note the improvement in black-white edge detail on some of the letters on the bottom line.

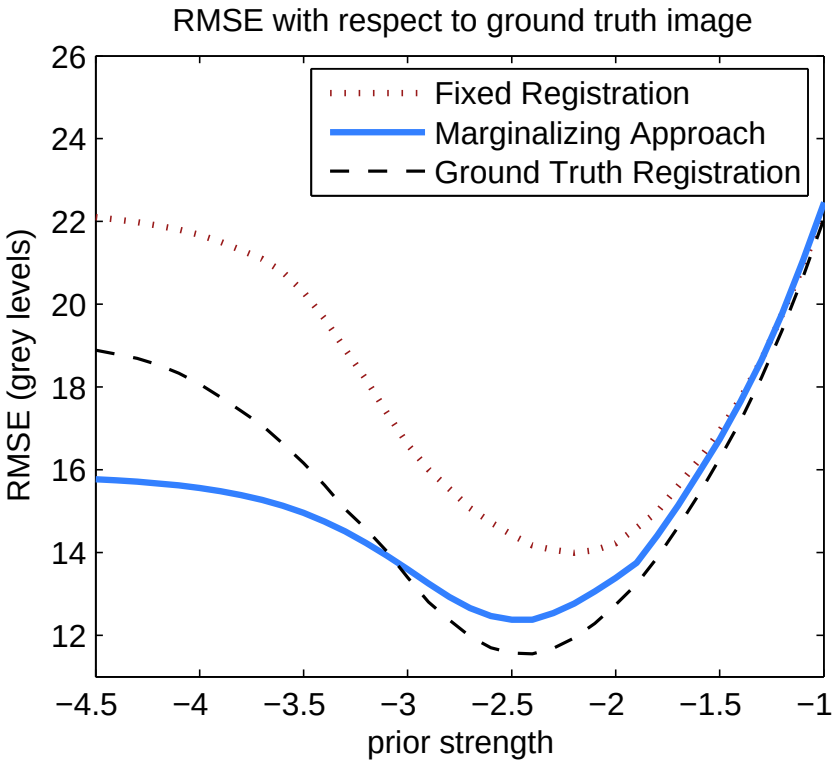


Figure 5.5: **Eyechart Dataset.** The plot shows the variation of RMSE with prior strength for the standard Huber-prior MAP super-resolution method and our approach integrating over $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$.

Figure 5.6: **The face datasets.** There are 16 15 × 15-pixel low-resolution images generated from a common high-resolution version at a zoom factor of 4. The variation in intensity is clearly visible, and the sub-pixel displacements necessary for multi-frame image super-resolution are also visible.

small input images, we also introduce uncertainty in the value of the point-spread function parameter, γ, by chosing a value that is 0.05 low-resolution pixels bigger than the true point-spread function standard deviation. PSF kernel estimation, particularly in such small images with such a high proportion of pixels lying near the image borers, is a hard problem, so this level of error is not unexpected in a general super-resolution problem with unknown parameters.

The full registration-marginalizing approach was applied (including marginalizing over the PSF, γ), so $\boldsymbol{\delta}$ was a 65-element vector comprised of four registration parameters per image, plus the single log-psf parameter. The result was compared to the plain Huber-MAP super-resolution algorithm with no integration, over a range of possible image prior strengths. The registration-marginalizing without the psf-integration was also used, providing a third image estimate for each prior strength value. For the results shown, we take the distribution over the registration to have standard deviation 0.01 low-resolution pixels, and use the same standard deviation for the PSF parameter.

The errors for each approach are plotted against prior strength in Figure 5.7. As anticipated, the plain Huber-MAP approach gives the greatest error overall, and the full registration-marginalizing approach which takes the point-spread function uncertainty into account gives the lowest error overall, though the version which handles only the 64 image registration parameters also handles the problem better than the plain Huber-MAP method.

The best output images from the plain Huber-MAP and the full registration-marginalizing approach are shown in Figure 5.8, along with one of the low-resolution images interpolated up in to the high-resolution frame, to give an idea of the improvement made by the super-resolution. As this was intentionally a challenging dataset

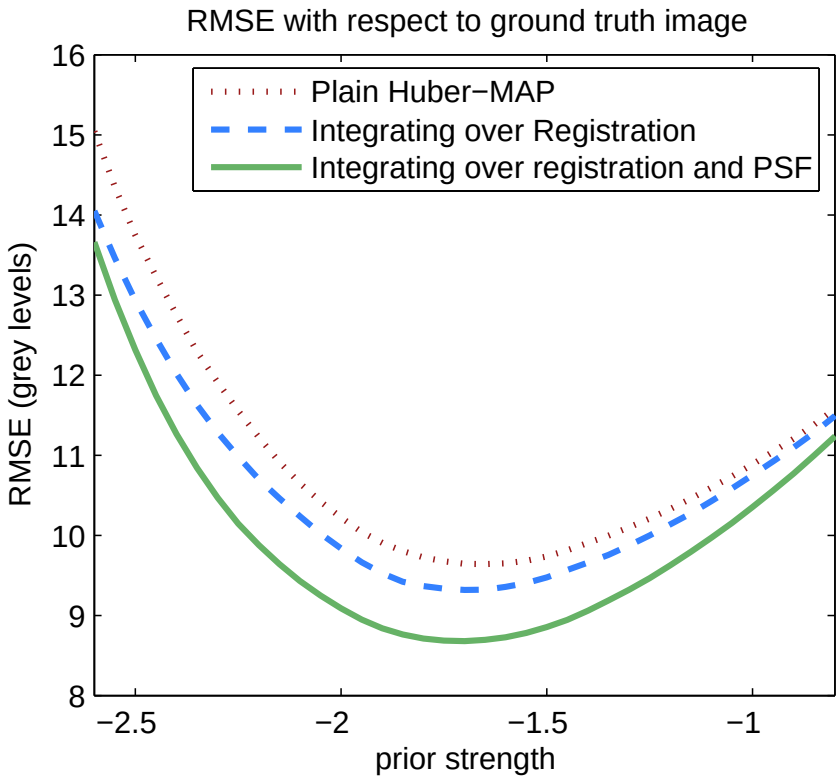


Figure 5.7: **Face Dataset.** Plot showing the variation of RMSE with prior strength for the standard Huber-prior MAP super-resolution method, the approach integrating over $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$, and the approach integrating over γ as well as the geometric and photometric registrations. As expected, the result which integrates over the uncertainty in all the estimates produces a lower-error reconstruction than the standard method.

with relatively high noise levels, both face reconstructions exhibit some noise around the cheek region. While the absolute differences between these two images are subtle, improvements in the reconstruction of the cheek to the left and around the eye to the right of the image may be discernible in the registration-marginalizing case, which leads to its lower error than the other methods when compared to ground truth.

Figure 5.8: **Super-resolving the face dataset.** (a) interpolated low-resolution image; (b) best (minimum MSE) image found using the Huber method (c) super-resolution result using our approach of integrating over $\boldsymbol{\theta}$, $\boldsymbol{\lambda}$ and γ. Again, the difference in results is subtle, but still perceptible. For the images shown, the Huber RMSE is 9.64, while the registration-marginalizing RMSE is 8.68.

5.2.4 Real data

The final experiment uses real data with a 2D translation motion model and a 2-parameter lighting model exactly as above; the low-resolution images appear in Figure 5.9. Homographies were provided with the data, but were not used. Instead, an iterative illumination-based registration was used on the sub-region of the images chosen for super-resolution, and this agreed with the provided homographies to within a few hundredths of a pixel.

Super-resolution images were created for a number of image prior strengths, and equivalent values to those quoted in [19] were selected for the Huber-MAP recovery, following a subjective evaluation of other possible parameter settings. For the registration-marginalizing approach, a similar parameter error distribution as that used in the synthetic experiments was assumed. Finally, Tipping and Bishop's method was extended to cover the illumination model and used to register and super-resolve the dataset, using the same PSF standard deviation (0.4 low-resolution pixels) as the other methods.

The three sets of results on the real data sequence are shown in Figure 5.10. To facilitate a better comparison, a sub-region of each is expanded to make the letter details clearer. The Huber prior tends to make the edges unnaturally sharp, though it is very successful at regularizing the solution elsewhere. Between the Tipping and Bishop image and the registration-marginalizing approach, the text appears more clear in our method, and the regularization in the constant background regions is slightly more successful. Also note that the Gaussian prior on the image-marginalizing method is zero-mean, so in this case having a strong enough prior to suppress the background noise has also biased the output image towards the mid-grey zero value, making the white regions appear darker than they do in the other methods.

5.3 Discussion

It is possible to interpret the extra terms introduced into the objective function in the derivation of the registration-marginalizing method as an extra regularizer term or image prior. Considering (5.17), the first two terms are identical to the standard MAP super-resolution problem using a Huber image prior. The two additional terms constitute an additional distribution over $\mathbf{x}$ in the cases where the parameter covariance $\mathbf{S}$ is not dominated by $\mathbf{V}$; as the distribution over $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ tightens to a single point, the terms tend to constant values.

The intuition behind the method's success is that this extra prior resulting from the final two terms of (5.17) will favour image solutions which are not acutely sensitive to minor adjustments in the image registration. The images of Figure 5.11 illustrate the type of solution which would score poorly. To create the figure, one

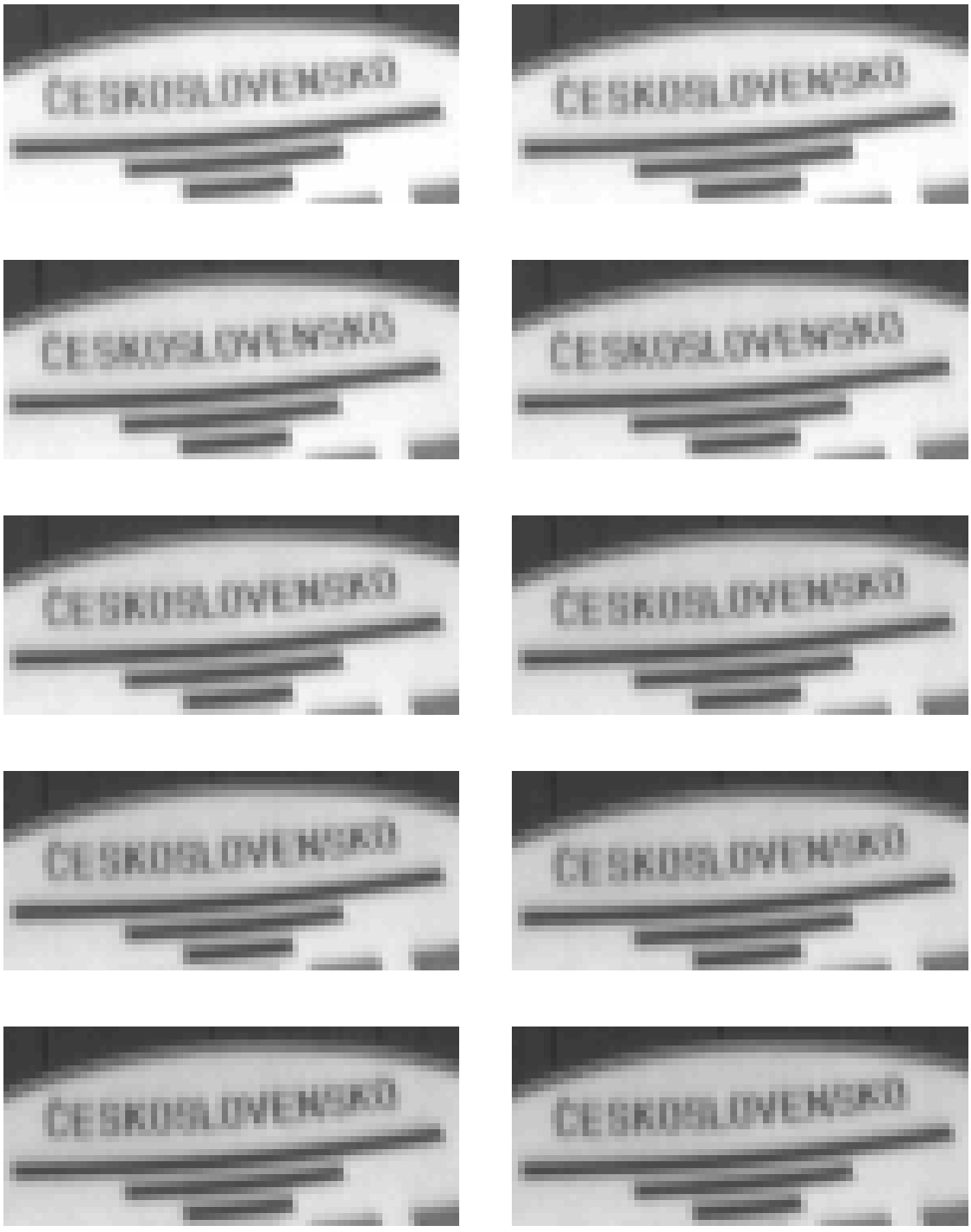


Figure 5.9: **The ten input images in the real dataset.** Photometric differences are clearly visible, with the later images being darker in general.

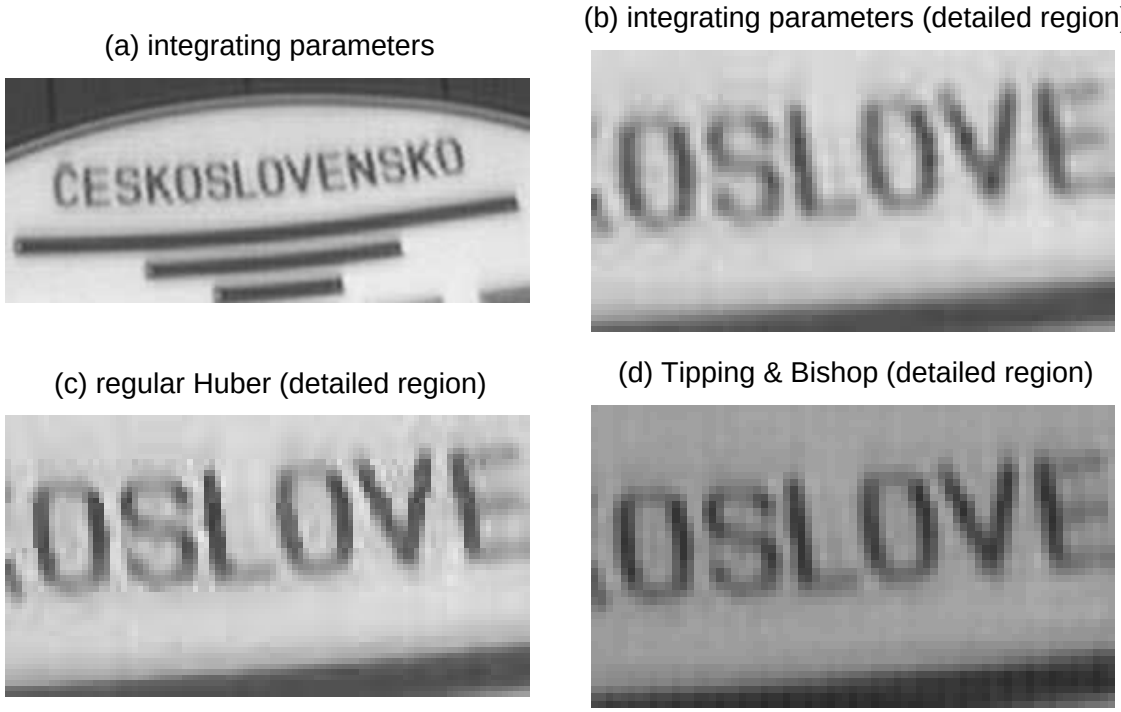


Figure 5.10: **Super-resolving the "Československo" sequence using three different methods.** (a) The full super-resolution output from our algorithm. (b) Detailed region of the central letters, again with our algorithm. (c) Detailed region of the regular Huber-MAP super-resolution image, using parameter values suggested in [19], which are also found to be subjectively good choices. The edges are slightly artificially crisp, but the large smooth regions are well regularized. (d) Close-up of letter detail for comparison with Tipping and Bishop's method of marginalization. The Gaussian form of their prior leads to a more blurred output, or one that over-fits to the image noise on the input data if the prior's influence is decreased.

dataset was used to produce two super-resolved images, using two independent sets of registration parameters which were randomly perturbed by an *i.i.d.* Gaussian vector with a standard deviation of only 0.04 low-resolution pixels. The chequer-board pattern typical of ML super-resolution images can be observed, and the difference image on the right shows the drastic contrast between the two image estimates.

The ability of the registration-marginalizing method to suppress the chequer-board pattern can also be observed in the results of the experiment with the butterfly dataset; in Figure 5.3, the chequer-board-like noise on the regular Huber-MAP super-resolution images is significantly diminished in the registration-marginalizing method's outputs.

5.4 Conclusion

In this chapter we have outlined two contrasting Bayesian approaches to image super-resolution, and presented the latest work in integrating over the imaging parameters as the "nuisance variables" in the super-resolution problem.

The registration-marginalizing approach to super-resolution shows several advantages over Tipping and Bishop's original image-integrating algorithm. These are a formal treatment of registration uncertainty, the use of a much more realistic image prior, and the computational speed and memory efficiency relating to the smaller dimension of the space over which we integrate. The results on real and synthetic images with this method show an advantage over the popular MAP approach, and over the result from Tipping and Bishop's method, largely owing to our more favourable prior over the super-resolution image.

Note that while the examples in this paper are confined to a translation-only

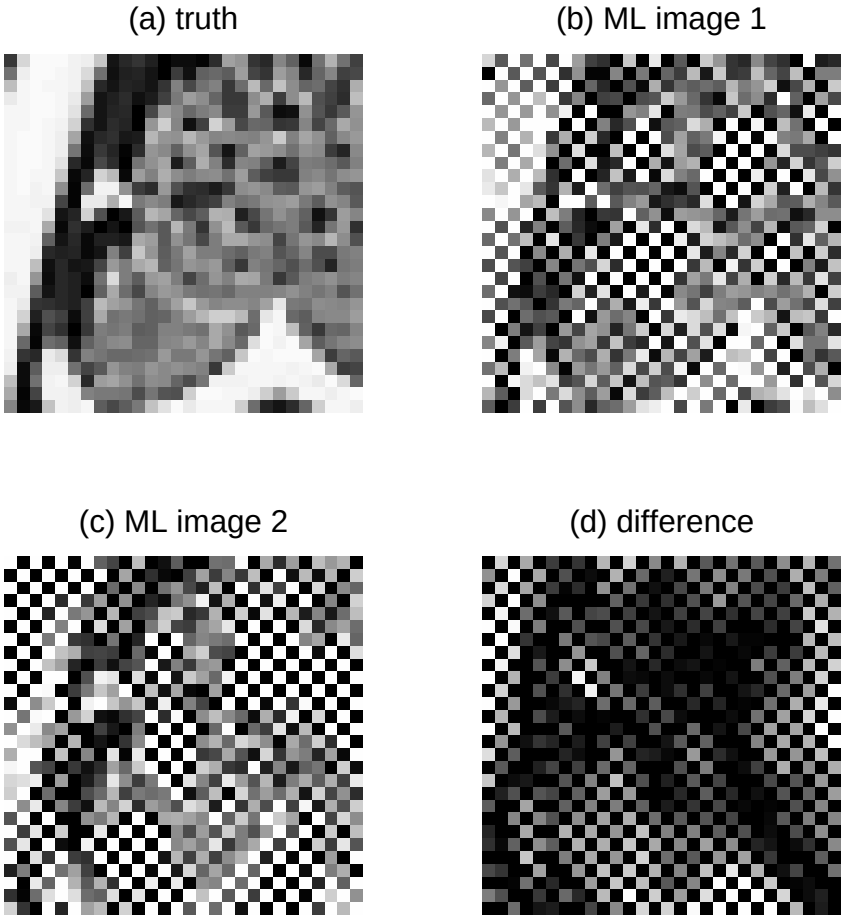


Figure 5.11: **An example of the effect of tiny changes in the registration parameters.** (a) Ground truth image from which a 16-image low-resolution dataset was generated. (b,c) Two ML super-resolution estimates. In both cases, the same dataset was used, but the registration parameters were perturbed by an *i.i.d.* vector with standard deviation of just 0.04 low-resolution pixels. (d) The difference between the two solutions. In all these images, values outside the valid image intensity range have been rounded to white or black values.

motion model, there is no constraint in the mathematical derivation which prevents it from being applied to more complex parametric motion models such as affine or planar projective homographies.

This chapter has demonstrated a quantitative improvement in super-resolution image quality can be achieved by the registration-marginalizing method, compared to the known ground truth image on synthetic datasets. The results on the real data "Československo" example show the method is capable of handling the errors and uncertainty introduced in real practical systems.

Chapter 6

Texture-based image priors for super-resolution

In this Chapter, we develop a new prior for use in MAP super-resolution, based on *sample image patches* extracted from example images with similar textures to the super-resolution image we seek to reconstruct, based on work first published in [92]. Up until now, we have concentrated on *parametric* priors over the super-resolution image as a way to produce plausible high-resolution images. They capture observations about the general statistics of high-resolution images: the Huber-MRF prior on image gradients approximates a true edge histogram in observed data, and Hardie *et al.*'s Laplacian approximation [56] embodies the belief that images are in general smooth. However, the actual image observations these priors try to approximate have been thrown away.

The complexity of most real-world scenes is hard to capture in a parametric way, so rather than discarding real-world observations, we propose a texture-based

image prior which works directly from patches in previously-observed images in order to evaluate how probable a candidate high-resolution image is. In this way we can approximate the *p.d.f.* of a target image class using a sample-based patch representation. The trade-off we make is that more of the problem has become "searching" – looking through previously-seen images – as opposed to "learning", where natural images are examined ahead of time to determine that a Huber-shaped prior might be appropriate.

In using specific textures, we also sacrifice the overall generality of parametric priors such as the Huber prior over image gradients, but in its place we get an incredibly powerful non-parametric description of a particular sub-class of images which we allow the prior to "observe" before we tackle a given super-resolution problem.

This chapter begins by formalizing the theory of using patches from example images to build a high-resolution image prior. We then present experiments to show how effective this idea is on several classes of image where the texture is very apparent. Comparisons are made between the performance of our texture-based prior, and that of the Huber prior over image gradients. Finally, we discuss the performance and possible limitations of a texture-based prior, and show an example of how it can behave when given an "incorrect" example image to work with.

6.1 Background and theory

The basic method of solution for a MAP multiframe image super-resolution problem was outlined in 3.3.2, and here we concentrate just on developing the form of the prior, $p(\mathbf{x})$.

The approach we take is motivated by the philosophy of Efros and Leung [28], who used non-parametric samples for texture synthesis. In their approach, a square neighbourhood around the pixel to be synthesized (excluding any regions where there texture has not yet been generated) is compared with similar regions in the sample image, and then the closest few (all those within a certain distance) are used to build a histogram of central pixel values from which a value for the newly-synthesized pixel is drawn. The synthesis proceeds one pixel at a time until all the required pixels in the new image have been assigned a value. The algorithm has one main variable factor which governs its overall behaviour, and that is its definition of the neighbourhood, which is determined both by the size of the square window used, and by a Gaussian weighting factor applied to that window, so that pixels near to the one being synthesized carry more weight in the matching than those further away.

Another patch-based approach (developed at the same time as the texture-based prior presented here) was used by Fitzgibbon *et al.* in [40] for image-based rendering. For their application the likelihood function was multi-modal, and a prior was needed to help ensure consistency across the output image. Their prior was posed as a product over probabilities of image patches, where each probability was a function of the distance from that patch to the closest patch found in a texture dictionary composed of images of the same object region from other viewpoints.

For our application, a small image patch around each high-resolution image pixel is selected. We can learn a distribution for this central pixel's intensity value by examining the values at the centres of similar patches from other images. Each pixel x_i has a neighbourhood region $\mathcal{R}(x_i)$ consisting of the pixels around it, but not including x_i itself. For each $\mathcal{R}(x_i)$, we find the closest neighbourhood patch in

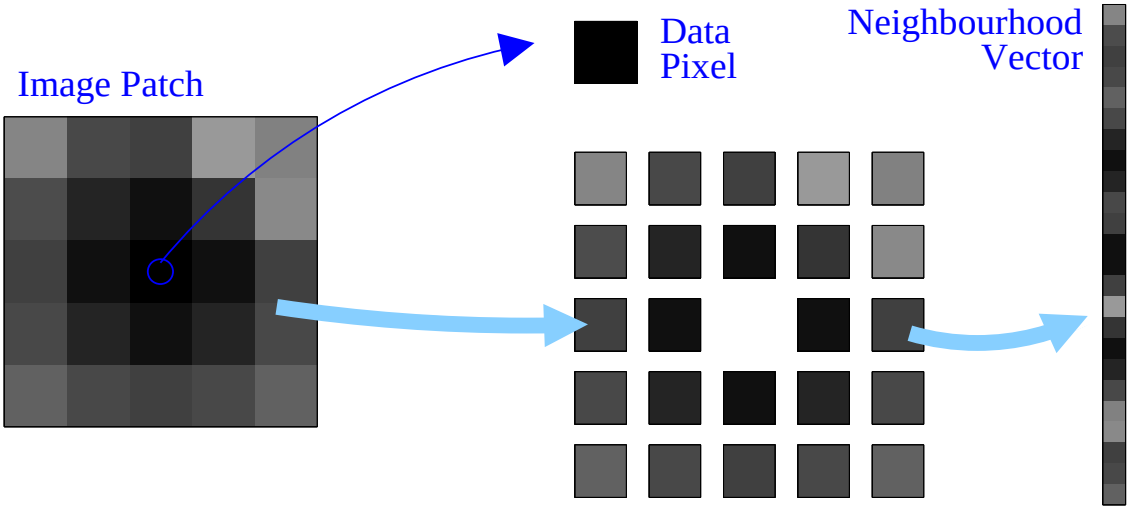


Figure 6.1: **Forming the neighbourhood vector from an image patch.** The central pixel is held back, so that when a match is made between this neighbourhood vector and a new trial patch, the central pixel value can be returned as a prototype for what value we might expect to find associated with the centre of the new patch.

the set of sampled patches, and thus the central pixel associated with this nearest neighbour, $L_{\mathcal{R}}(x_i)$. This decomposition of the image patch into a neighbourhood vector and a central "data" pixel is illustrated in Figure 6.1.

The intensity of the original pixel is assumed to be Gaussian-distributed with mean equal to the intensity of this central pixel, and with some precision ϕ,

$$x_i \sim \mathcal{N}(L_{\mathcal{R}}(x_i), \phi^{-1}) \tag{6.1}$$

leading to a prior of the form

$$p(\mathbf{x}) \propto \exp\left\{-\frac{\phi}{2}\left\|\mathbf{x} - L_{\mathcal{R}}(\mathbf{x})\right\|_2^2\right\}. \tag{6.2}$$

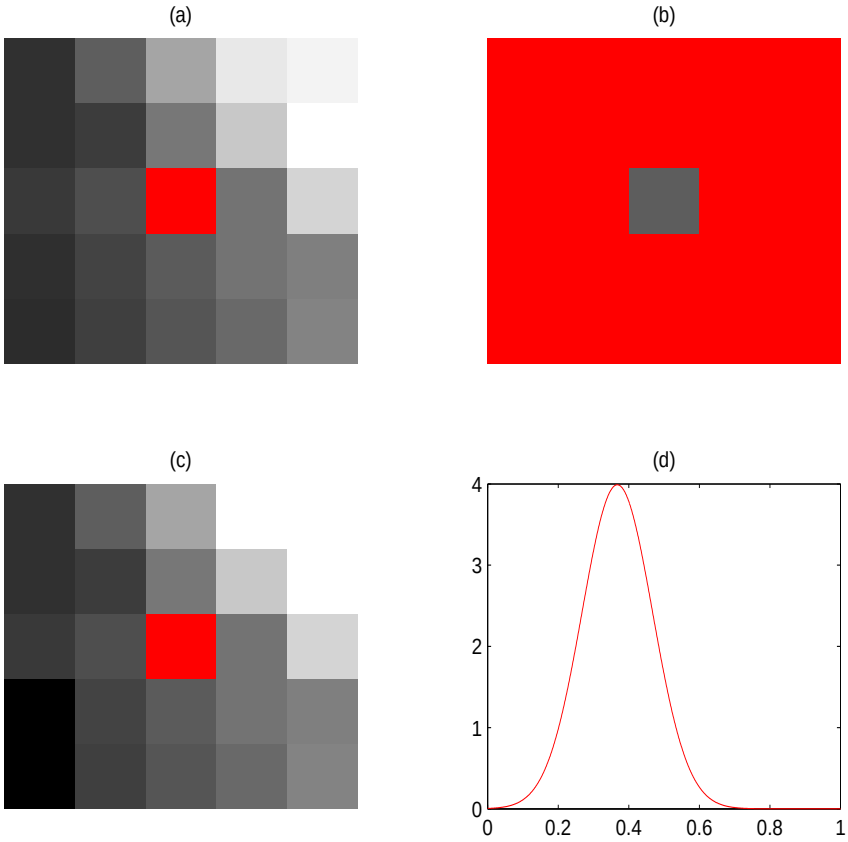


Figure 6.2: **Evaluating the prior for a particular pixel.** (a) the patch neighbourhood $\mathcal{R}(x_i)$, for a 5×5 patch size; (b) the central pixel under consideration; (c) the closest matching patch sampled from the training image, $L_{\mathcal{R}}(x_i)$; (d) the distribution we expect (b) to have been drawn from, which is centred around the value of the middle pixel of (c), and has precision given by ϕ.

This distribution is illustrated in Figure 6.2.

With reference to the basic MAP equations (3.19) and (3.20), we have

$$\mathbf{x}_{\text{MAP}} = \arg\max_{\mathbf{x}} \left(\frac{\phi}{2} \|\mathbf{x} - L_{\mathcal{R}}(\mathbf{x})\|_2^2 + \frac{\beta}{2} \sum_{k=1}^{K} \|\mathbf{r}^{(k)}\|_2^2 \right), \tag{6.3}$$

where $\mathbf{r}^{(k)}$ is the residual as defined in (3.8). Scaling the two terms on the right to

leave a single unknown ratio ϕ' instead of the two precision factors ϕ and β, the final objective function can be expressed

$$\mathcal{L} = \phi' ||\mathbf{x} - L_{\mathcal{R}}(\mathbf{x})||_2^2 + \sum_{k=1}^{K} \left\|\mathbf{r}^{(k)}\right\|_2^2 . \tag{6.4}$$

6.1.1 Image patch details

The image patch regions $\mathcal{R}(x_i)$ are square windows centred on x_i. Because we must be able to centre an image patch on any pixel we optimize over, those nearest to the image borders must be estimated using a different scheme. These border pixels are initialized using the same technique as Section 4.3.4, *i.e.* starting with the *average image*, and refining the estimate using a small number of ML iterations.

The patches in the texture dictionary are normalized to sum to unity, and centre weighted as in [28] by a 2-dimensional Gaussian kernel. For simplicity, the patches are always chosen to be squares with an odd number of pixels to a side, and the Gaussian kernel standard deviations are chosen to be equal to the floor of half the width of the square, *i.e.* a 7×7 patch would use a Gaussian with a standard deviation of 3 pixels. The width of the image patches used for a particular super-resolution problem depends very much upon the scales of the textures present in the scene. Figure 6.3 illustrates patches taken from a small example text image at two different sizes. The smaller size (5×5-pixel patches) captures the types of curves and edges typically seen in images of anti-aliased text, and the larger size (11×11-pixel patches) captures many of the forms of the individual letters. Examples are shown with and without the Gaussian weightings applied.

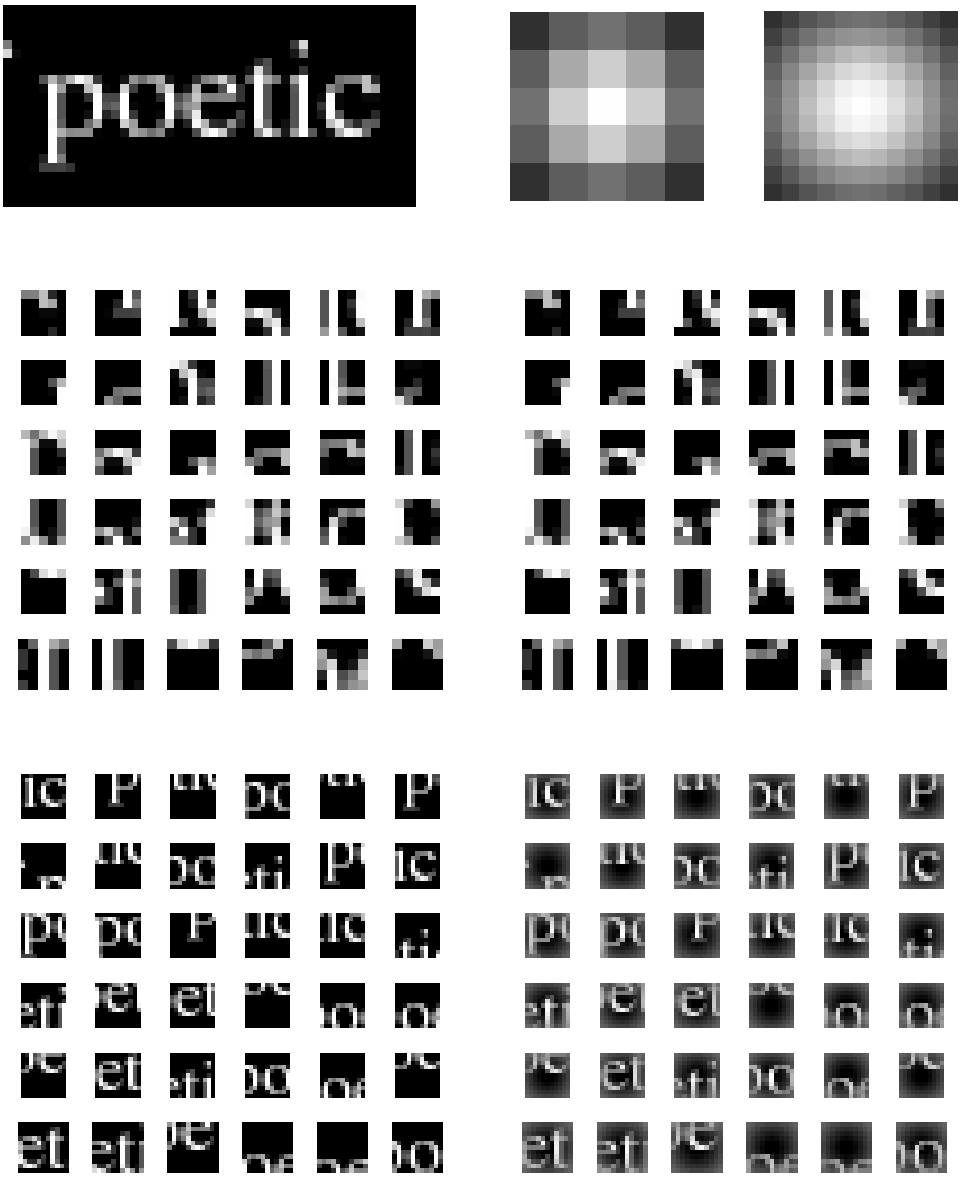


Figure 6.3: **Image patch example showing two sizes of patches.** Top row: a small image from which we extract patches, and two example Gaussian kernels of size 5×5 and 11×11 pixels. Middle left: a collection of 36 sample 5×5 patches, which capture the types of curves and edges typically seen in this type of text image. Middle right: The same collection of patches, but with the Gaussian kernel applied. Bottom left: a collection of 36 sample 11×11 patches, which bring out the forms of individual letters far more than in the smaller-patch case. Bottom right: the same 36 larger patches with a Gaussian kernel applied.

6.1.2 Optimization details

As with the MAP methods described previously, the objective function of equation 6.4 is optimized using SCG to obtain a super-resolution image estimate. This requires an expression for the gradient of the function with respect to $\mathbf{x}$. For speed, we approximate this by

$$\frac{d\mathcal{L}}{d\mathbf{x}} = 2\phi'\big(\mathbf{x} - L_{\mathcal{R}}(\mathbf{x})\big) - 2\sum_{k=1}^{K} \lambda_1^{(k)} \mathbf{W}^{(k)T} \mathbf{r}^{(k)}, \tag{6.5}$$

which assumes that small perturbations in the neighbours of $\mathbf{x}$ will not change the value returned by $L_{\mathcal{R}}(\mathbf{x})$. This is obviously not necessarily the case, but leads to a more efficient algorithm. The algorithm is no longer globally convex because of the nonlinearity introduced by the patch-matching step, though as long as the nearest neighbour of a patch in the super-resolution estimate remains constant, $\mathcal{L}$ is simply a quadratic in $\mathbf{x}$

Two methods for matching the patches were investigated. The first was an augmented KD-tree, which stored the centre pixel values of each patch as well as the patch data vector itself. The second approach employed a Delaunay triangulation method using on Matlab's `dsearchn` function, which was found to run more quickly in the Matlab implementation than the KD tree variation. Both methods returned the same patch for any given query. To speed up the search, duplicate patches, or those with many very close neighbours were removed from the patch dictionary.

6.2 Experiments

To test the performance of the texture-based prior, a collection of synthetic image datasets was generated from various textured images using the forward model described in Section 3.1. These datasets were then used to reconstruct the high-resolution image using a MAP approach, either with the texture-based prior or with a standard Huber-MAP prior. The RMS error of these reconstructions compared to the ground truth image gives a measure of how well the texture-based prior helps us in these richly-textured problems.

6.2.1 Synthetic data

The forward model for super-resolution imaging as described in Section 3.1 is used to generate the synthetic super-resolution datasets. Image intensities are all in the range $[-\frac{1}{2}, \frac{1}{2}]$, and the imaging model assumes ten degrees of freedom per low-resolution image: a planar projective transform (8 DoF) and a linear photometric transform (2 DoF). The PSF used is a Gaussian kernel with a standard deviation of 0.4 low-resolution pixels, and a zoom factor of two in each direction (*i.e.* a four times area decrease going from high to low resolution) is used.

Figures 6.4, 6.5 and 6.6 show three 100×100-pixel ground truth images[1], each accompanied by four corresponding 40×40 pixel low-resolution images generated from the ground truth images at half the resolution, and an additional image from which texture information is sampled for the image prior. In no cases do the areas of the images used for the prior overlap with the sections used in generating the datasets.

[1]Text grabbed from Greg Egan's novella *Oceanic*, published online at the author's website. Brick image from the Brodatz texture set. Beads image from *http://textures.forrest.cz/*.

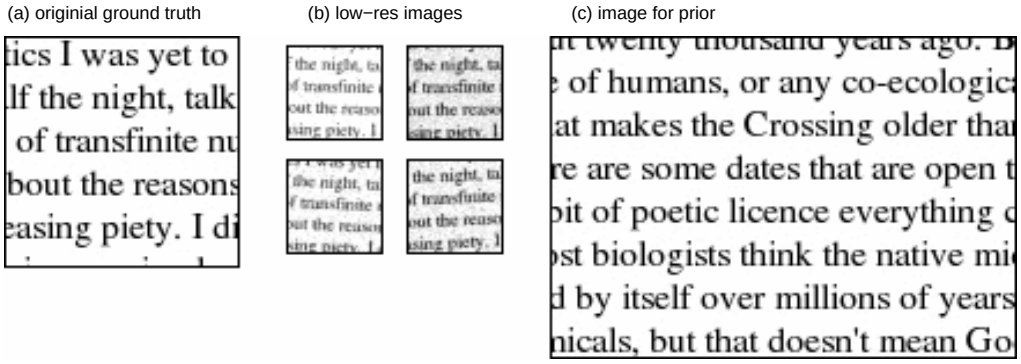


Figure 6.4: **Image data for the "text" example.** (a) The ground truth text image. (b) Four of the low-resolution images generated as synthetic inputs for super-resolution. (c) The sample of an image of different text using the same scale and font as the super-resolution image we want to reconstruct from the low-resolution images.

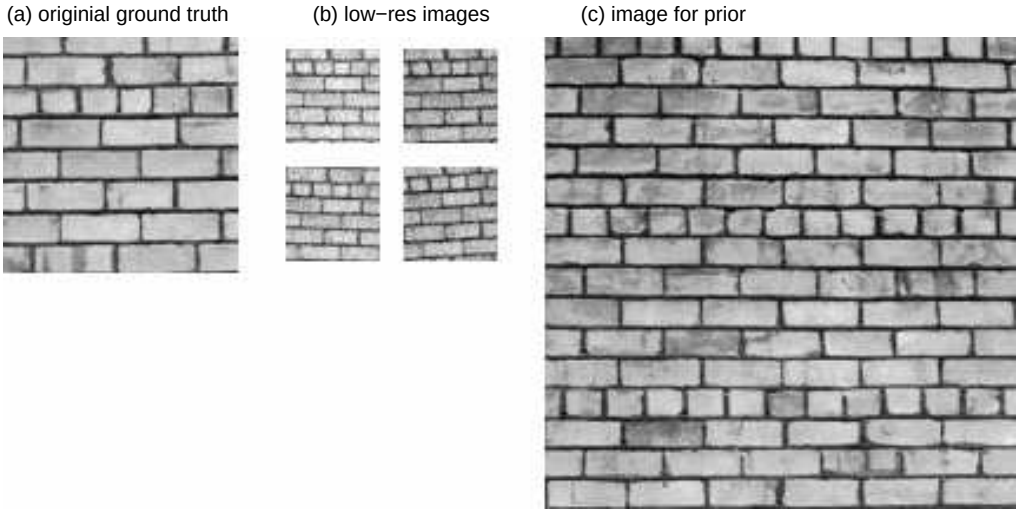


Figure 6.5: **Image data for the "brick" example.** (a) The ground truth brick image. (b) Four of the low-resolution images generated as synthetic inputs for super-resolution. (c) A large texture sample is used for the prior because the texture actually contains a lot of variation.

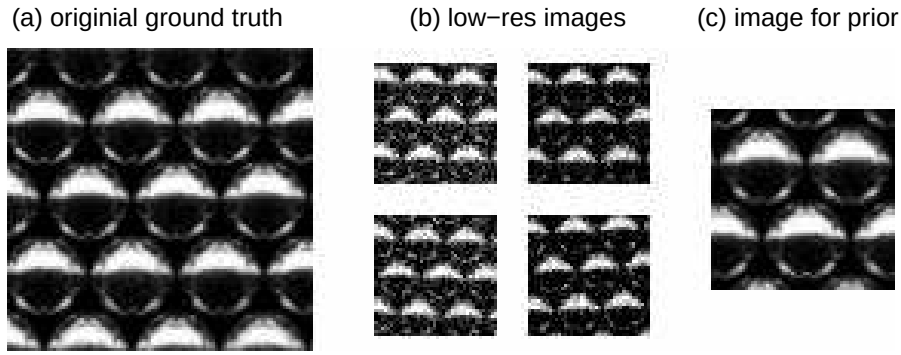


Figure 6.6: **Image data for the "beads" example.** (a) The ground truth beads image. (b) Four of the low-resolution images generated as synthetic inputs for super-resolution. (c) The sample used to make the image prior. In this case only a very small section is required because the texture is very regular.

6.2.2 Results

Our aim was to reconstruct the central 50×50 pixel section of the original ground truth image. Figures 6.7, 6.8 and 6.9 show the difference in super-resolution image quality that can be obtained using the sample-based prior over the Huber prior using identical input sets as described above.

For each Huber super-resolution image, we ran a set of reconstructions, varying the Huber parameter α and the prior strength parameter ν. The image shown for each input number/noise level pair is the one which gave the minimum RMS error when compared to the ground-truth image; these are very close to the "best" images chosen from the same sets by a human subject.

The images shown for the sample-based prior are again the best (in the sense of having minimal RMS error) of several runs per image. We varied the size of the sample patches from 5 to 13 pixels in edge length – computational cost meant that larger patches were not considered. Compared to the Huber images, we tried

relatively few different patch size and ϕ-value combinations for our sample-based prior; again, this was due to our method taking longer to execute than the Huber method. Consequently, the Huber parameters are more likely to lie close to their own optimal values than our sample-based prior parameters are.

We also present images recovered using a "wrong" texture. We generated ten low-resolution images from a picture of a leaf, and used patches from a small black-and-white spiral to build the texture dictionary (Figure 6.10). A selection of results are shown in Figure 6.11, where we varied the ϕ parameter governing the prior's contribution to the output image. Using a low value gives an image not dissimilar to the ML solution; using a significantly higher value makes the output follow the form of the prior much more closely, and here this means that the grey values get lost as the evidence for them from the data term is swamped by the black-and-white pattern of the prior.

6.3 Discussion and further considerations

The images of Figures 6.7 to 6.9 show that our prior offers a qualitative improvement over the generic prior, especially when few input images are available (see the top row in each set of image results). In general, larger patch sizes (11×11 pixels) give smaller errors for the noisy inputs, while small patches (5×5) are better for the less noisy images.

Quantitatively, our method gives an RMS error of approximately 25 grey levels from only 2 input images with 2 grey levels of additive Gaussian noise on the "text" input images, whereas the best Huber-MAP prior super-resolution image for that image set and noise level uses all 10 available input images, and still has an RMS

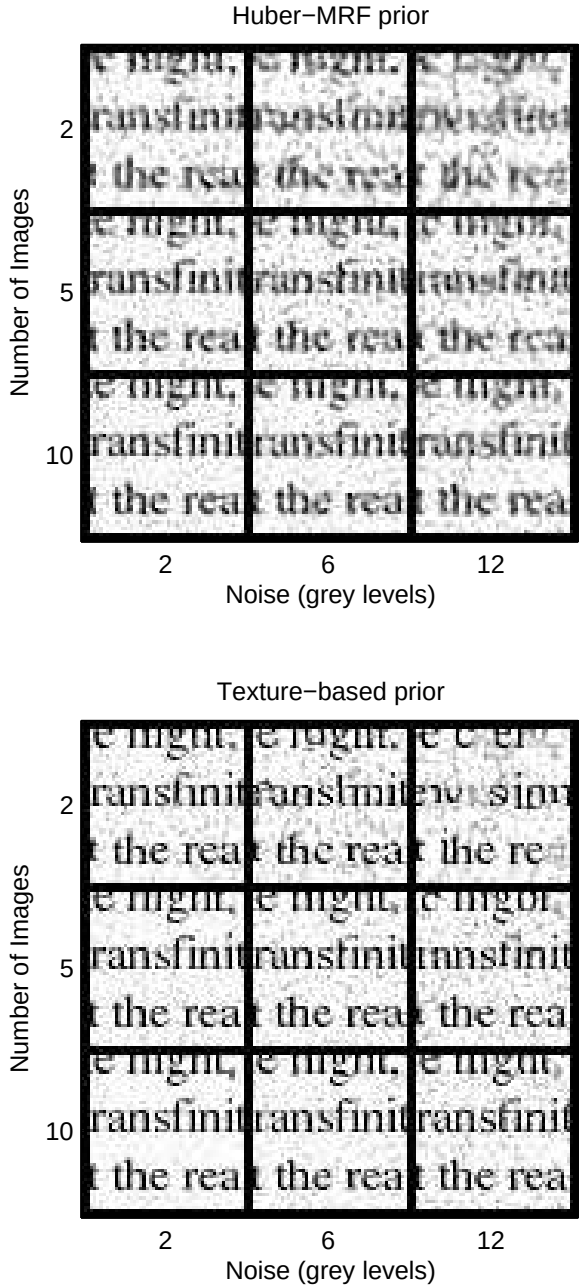


Figure 6.7: **"Text" results: Huber-MRF and texture-based.** Top: super-resolution results using the Huber-MRF prior on datasets with 2, 5, or 10 images and a noise standard deviation of 2, 6 or 12 grey levels. Bottom: The texture-based prior on the same nine datasets.

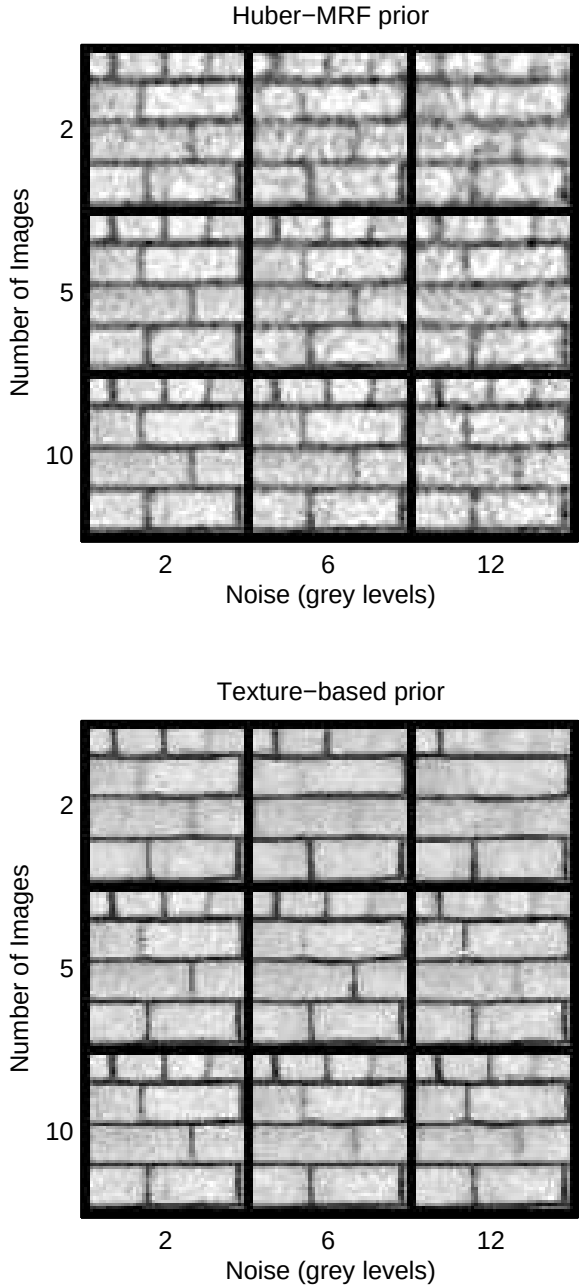


Figure 6.8: **"Brick" results: Huber-MRF and texture-based.** Top: super-resolution results using the Huber-MRF prior on datasets with 2, 5, or 10 images and a noise standard deviation of 2, 6 or 12 grey levels. Bottom: The texture-based prior on the same nine datasets.

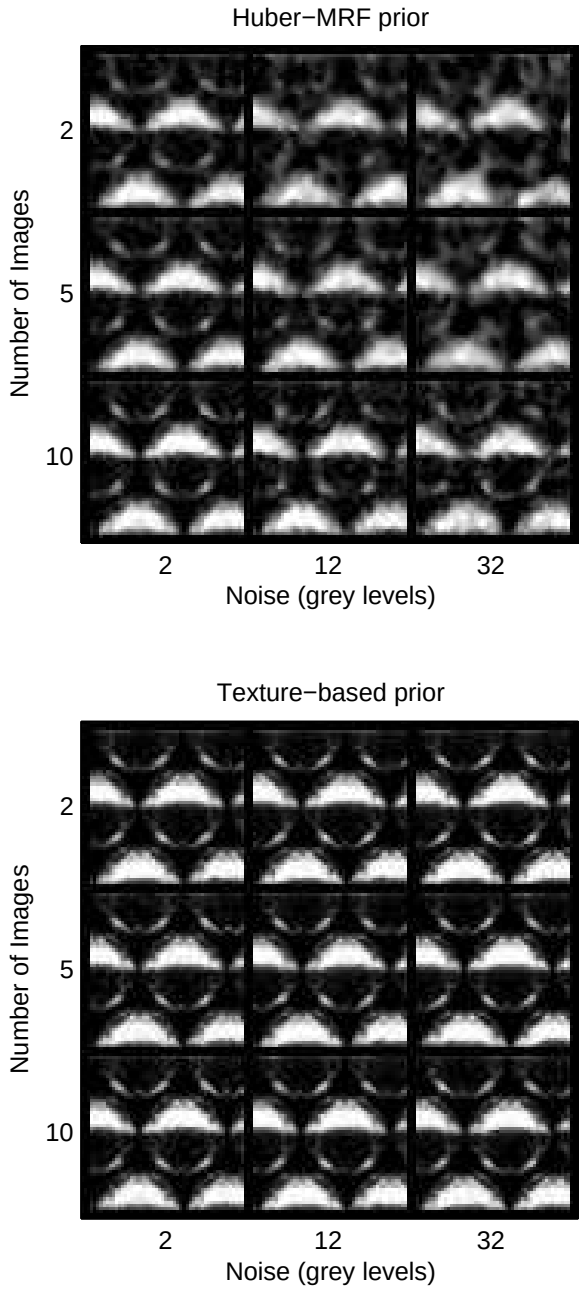


Figure 6.9: **"Beads" results: Huber-MRF and texture-based.** Top: super-resolution results using the Huber-MRF prior on datasets with 2, 5, or 10 images and a noise standard deviation of 2, 12 or 32 grey levels. Bottom: The texture-based prior on the same nine datasets. Note how well the texture-based version performs even on the extremely noisy case (right-hand column of both sets).

Figure 6.10: **Ground truth leaf image and a "bad" texture match.** The ground truth "leaf" image (120×120 pixels, left) is used to generate the low-resolution image data. The "spiral" image (80×80 pixels, right) is a poor choice of image from which to build a patch dictionary for super-resolution reconstruction.

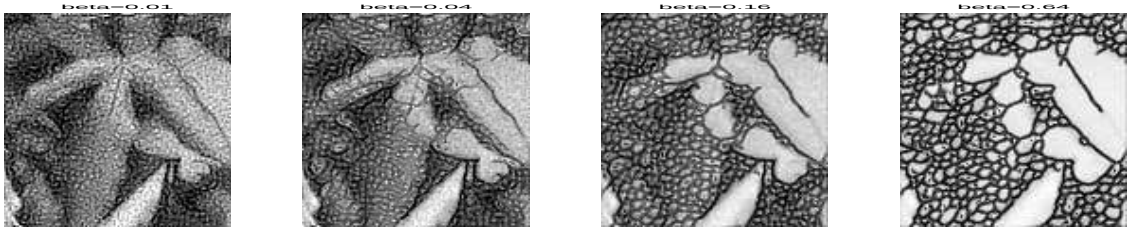


Figure 6.11: **Reconstruction using a "bad" texture.** These four 120×120 super-resolution images of the "leaf" image are reconstructed using different values of the prior strength parameter ϕ: 0.01, 0.04, 0.16, 0.64, from left to right, using a patch dictionary formed from the "spiral" image.

error score of almost 30 grey levels.

Figure 6.12 plots the RMS errors from the Huber-MAP and sample-based priors against each other. In all cases, the sample-based method fares better, with the difference most notable in the text example.

Further work on improving the computational complexity of the texture-based prior algorithm still needs to be carried out. For finding the gradient with respect to the high-resolution image pixels, the same k-nearest-neighbour variation introduced

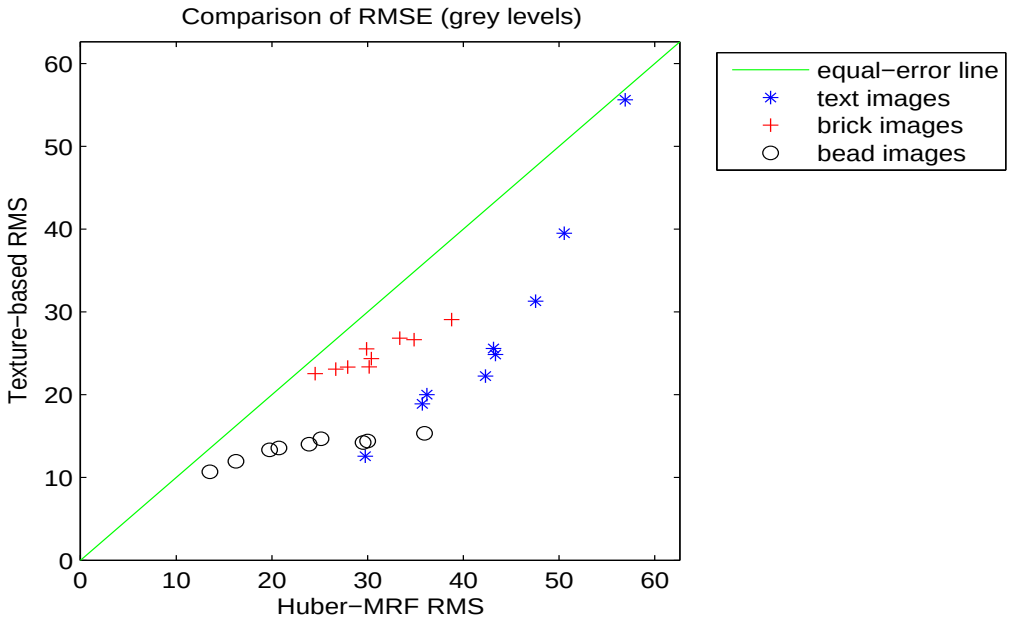


Figure 6.12: **Comparison of RMS errors.** The error with respect to ground truth is measured for each image in Figures 6.7, 6.8 and 6.9 (18 images per figure), and the result from each texture-based prior image is plotted against the Huber-MAP prior error for the corresponding dataset (*i.e.* identical noise and number of images). This gives nine datapoints per texture type, as shown. In every case, the error associated with the texture-based prior is lower than the Huber-MRF error.

in [28], where multiple neighbourhoods are found for each pixel under consideration, could be adopted to smooth this response, which may also lead to better super-resolution image outputs.

Since in general the textures for the prior will not be invariant to rotation and scaling, consideration should be given to the orientation of the super-resolution image frame, *i.e.* so that the scene shares its horizontal and vertical directions with the image set from which the prior's patch dictionary was sampled. The optimal patch size will be a function of the image textures as well as noise levels in the input dataset, so learning it as a parameter of an extended model is another direction of

interest.

Finally, handling multiple image textures from different dictionaries in the same optimization would allow the texture-based prior to be applied in more situations. Existing specialist super-resolution techniques for faces rely on registering the low-resolution images precisely so that different face regions (eyes, nose *etc.*) occupy specific pixels in the super-resolution image [18]. A texture-based prior could make use of this by learning a dictionary from many registered faces, and only checking the patches from face regions within a few pixels of the current candidate patch. Similarly, any application that incorporates both recognition and super-resolution could draw patches from more specialized dictionaries as regions in an image are recognized (*e.g.* field, road or water textures for satellite images).

Chapter 7

Conclusions and future research directions

7.1 Research contributions

The two main themes around which the research presented in this thesis have been based are firstly the benefits of using sensible and applicable image priors to help the super-resolution reconstruction process, and secondly the ways in which a super-resolution image estimate can be improved by considering uncertainties in the underlying image registrations and other parameters as part of the image estimation process.

Chapter 3 discusses the interdependent structure of multi-frame super-resolution, and explores the way errors induced in one part of the problem have coupled effects in the estimation of other components when the registrations and imaging parameters are estimated in a sequential manner before the main image reconstruction is carried out. No other surveys of super-resolution have touched on the way in which different sub-components of the problem interrelate in this way.

In Chapter 4 the question of how to handle the uncertainty in the parameter estimates in order to avoid the errors presented in the previous chapter is answered

with an algorithm for making a simultaneous point estimate all the values concerned: the super-resolution image, geometric and photometric registration parameters, and parameters for a non-Gaussian image prior. The work presented shows that image registration accuracy can be improved by taking the high-resolution image estimate into account, and these improvements are significant enough to be visible easily even on noisy input data from a DVD source.

A second key point in Chapter 4 is that the selection of parameter values for a parametric image prior, such as the Huber-MRF, can be made with reference to the input data. As well as taking another degree of trial-and-error out of the super-resolution problem, the scheme we present avoids the flaw in the cross-validation suggestion of previous authors, because it is handled in the same iterative framework as the image registration, so problems with mis-registered input images, which would cause serious problems for such methods previously, are avoided here.

An alternative answer to the question of handling uncertainty in the parameter estimates is presented in Chapter 5, which takes the Bayesian approach of integrating uncertainty out of the problem. We propose a scheme for marginalizing over residual uncertainty in the registrations, leaving an objective function that can be optimized directly with respect to the variables of interest, which are the intensity values of the super-resolution image pixels. As in the previous chapter, the approach is successful because it considers the super-resolution problem as a whole. In addition to the registration parameters estimated in the previous chapter, some slight improvement is also shown here in marginalizing over the parameter governing the point-spread function size. On the theme of correct choices of image priors, this method also shows an additional advantage over previous Bayesian super-resolution techniques because where they are limited to Gaussian priors for reasons of tractability, this

marginalization is viable for a wide range of image priors, including, but not limited to, Huber-MAP and texture-based priors.

Finally, Chapter 6 explores the possibilities of using image patches to improve the prior term in the MAP approach. Patches have been popular in single-image super-resolution methods, and the approach here shows how the patch-based methods can be brought into multi-frame super-resolution problems. In this work, we take the particular case of highly-textured scenes and show how a prior based on samples from images known to have similar textures to the target high-resolution image can be used to achieve a very significant improvement in output image quality over standard "smoothness" priors.

7.2 Where super-resolution algorithms go next

The following subsections outline some major directions in which the model and approaches for multi-frame image super-resolution presented in this thesis can be extended to cover more data sources and create high-quality super-resolution images in a more fully automatic way requiring even less user guidance.

7.2.1 More on non-parametric image priors

One technique which is successful in single-image super-resolution is the practice of working with high-frequency information represented as the difference between the interpolated low-resolution image and the super-resolved image itself, as in [45, 108]. In the multi-frame case, we have the *average image* to act as a blurred estimate of the high-resolution image, and there is scope for learning priors over the high-resolution image based on the difference between the average image and the super-resolution

estimate. The particular advantage in this approach is that the high-frequency-only patches are likely to generalize across scene types more effectively than the normalized image patches presented in Chapter 6 of this thesis, which need to be chosen specifically for the type of texture present in the scene.

The techniques used for these single-image super-resolution algorithms, particularly the "primal sketch" prior of Sun *et al.* [108] are especially effective for scenes with low texture content but strong edges, where the edge details and contours can be modelled by transfer of local edge information from other images. The patch-based prior cases examined in this thesis were most effective where the high-resolution image was rich in high-frequency information which varied considerably over the scale of only a few pixels, so a sketch-style prior would be useful for a different range of scenes which would complement the work covered so far in this thesis.

7.2.2 Registration models and occlusion

Affine and planar-projective mappings between the low-resolution images and the super-resolution image are sufficient to model a wide range of scenes, especially because small image regions often behave in a locally-affine way even when the global motion is non-affine. Authors whose motion models go beyond what can be handled by these simple parameterized homographies usually opt for optic flow as a motion model, and have to cope with all the common drawbacks of optic flow, such as poor localization in smooth areas and poor performance in the presence of significant image noise, and also have a far larger latent space to contend with.

An intermediate approach capturing the strengths of both styles of motion model for super-resolution would be to assume the low-resolution images are related to the super-resolution frame by a *Radial Basis Map* (RBM), which represents the motion

of a set of "centres" in each image, deforming the rest of the image to accommodate the motion while minimizing curvature. The geometric image registrations can therefore be encoded as point-to-point mappings, giving a representation that is much lower in dimension than a corresponding optic flow map.

In the same way as local affine approximations are used to map the point-spread-function into the high-resolution frame in Section 3.2.3, a locally-affine approximation can also be made for the RBM-mapped low-resolution pixel and the PSF kernel centred at its location. An example of a synthetic dataset generated and super-resolved using RBM transformations is shown in Figure 7.1. However, the best method for incorporating the registration of these images into the super-resolution framework in a way that is effective for simultaneous optimization (as in Chapter 4) remains an area for active research.

7.2.3 Learning the blur

The simultaneous approach of Chapter 4 in this thesis was found to be better at handling slight errors in the blur than an algorithm which fixed all the registration parameter estimates ahead of time, but the algorithm itself did not include any blur estimation step as part of its iterations. The geometric-error example of Section 3.5.3 showed that when there is registration error, the PSF giving the best super-resolution reconstruction does not necessarily represent the true blur. Even extending the cross-validation part of the simultaneous algorithm shows little success in accurately recovering a point-spread kernel.

While Chapter 5 showed that small PSF estimate errors can be marginalized out of the problem in the same way as the registration parameter errors, the algorithm again did not directly estimate parameters. However, recent advances in blind image

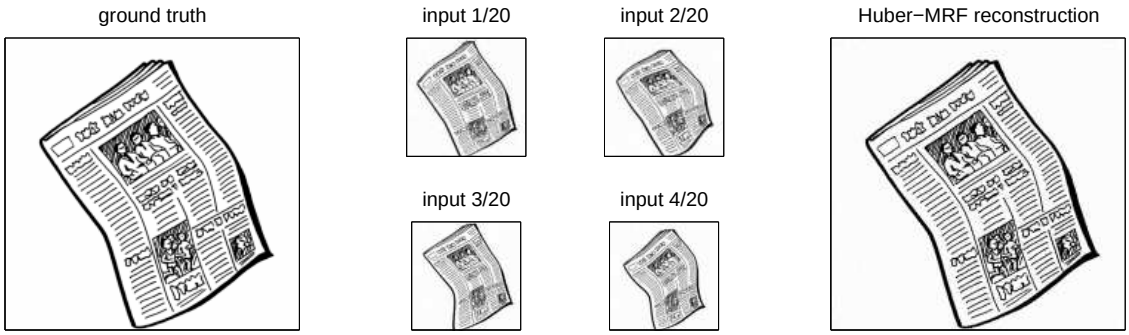


Figure 7.1: **An example of super-resolution using inputs related by a radial basis registration.** Left: The ground truth image (a cartoon newspaper). Centre: four of the 20 low-resolution images generated under the forward model. The PSF blur is Gaussian with standard deviation 0.4 low-resolution pixels, and the zoom factor is 2.4. Note the very different shapes taken on by the newspaper in the different low-resolution views because of the RBM model. Right: the Huber-MAP super-resolution result when the registration is known with very high accuracy. Incorporating the registration step for RBM models into the super-resolution framework described in this thesis as successfully as the planar-projective registration model used in Chapter 4 is a desirable extension to this work.

deconvolution suggest that Variational methods will enable the learning of a PFS kernel.

The form of the problem used in Fergus *et al.* [39] is particularly attractive because of the carefully-considered non-Gaussian prior, though the image representation — working with image derivatives — is not so immediately applicable to image super-resolution work. The model used by Molina *et al.* [74] imposes a typical Gaussian prior in a framework, and is much closer to typical multi-frame super-resolution models. In both cases, a blur kernel for the input image is learnt, and a deblurred scene is recovered.

Several extensions to the approach should be possible, and indeed fruitful, for image super-resolution. Firstly, the additional image information available from having multiple views of the scene should improve noise robustness when the super-resolution model is applied. Secondly, if the input images are blurred using different PSF kernels, *e.g.* because they come from hand-held camera photographs under conditions in which hand-shake becomes noticeable, then the inputs provide much stronger constraints about which image effects the scene is responsible for, and which are due to the blur.

Finally, inhomogeneous, space-varying blurs are something which very few super-resolution algorithms are able to cope with at present, but which frequently occur in real data sequences, *e.g.* when a subject is much closer to the camera than other background objects, or when camera shake causes rotation about the camera's principal axis, and that in turn causes motion blur. Multi-frame approaches in which the point-spread for each low-resolution pixel is considered separately – possibly with neighbourhood consistency constraints encoded as prior information over the blur field – should be able to perform better than standard single-image deblurring

methods, and in the process, the super-resolution image found as a result of a system which learns the blurs like this while taking into account the super-resolution generative model should produce even more accurate results.

7.2.4 Further possibilities with the model

The super-resolution model we use has several more interesting possibilities which remain to be explored. The first of these is the idea of drawing samples from the posterior over the super-resolution image, rather than simply selecting the mode as the MAP image estimate. By looking at the covariance of the estimate, rather than simply the estimate itself, we can visualize which structures in the image we are most uncertain about, which could offer valuable information in situations where the image estimates are used for recognition, *e.g.* face recognition from security footage.

Another extension to the model we have presented is to reconstruct a high-resolution video sequence rather than a single super-resolution image of the scene. With planar-projective homographies and no occlusion, this is a straight-forward problem because the high-resolution frames are also all related by homographies. However, considering more flexible motion models and occlusions, as discussed in Section 7.2.2 will be more of a challenge. The natural extension for our model is to consider all super-resolution images jointly (augmenting the image prior to include temporal information), allowing either the *simultaneous* or *parameter-integrating* approaches of Chapters 4 and 5 to be applied.

At present, the model we use assumes that the constraints from every pixel are valid, which would not be the case if an unwanted object appears in one or more frames of a sequence *e.g.* if a DVD sequence to super-resolve accidentally contains the first frame of the subsequent shot as well as the "good" frames, or more commonly

if the input sequence contains specularities, which move depending on the viewing angle. In these cases, an explicit method to flag "suspect" low-resolution pixels (even all the pixels from one of the images) and handle the outliers properly could be added to the model.

If information about the specific imaging conditions is available, then a number of further refinements can be made. If colour images come from a digital camera with a Bayer pattern (see Section 2.9 and Figure 2.11), then we can account for each different colour in the generative model. If there is significant white-balance or gamma-correction on the input data, these nonlinearities should be taken into account in the model, though in general these are camera-specific and setting-specific distortions that are difficult to work with unless some image meta-data is also available. Finally, there are factors like lens distortion and sensor saturation, which are less specific effects, and which could be handled by the model, though solving a model with saturation is less straight-forward because it introduces a significant nonlinearity to the model.

7.3 Closing remarks

This thesis has focused on the problem of multi-frame image super-resolution, and how to improve upon the quality or accuracy of estimated output images over the state of the art algorithms. However, the main themes of this work – finding more plausible image priors to encode our human intuition about "good" images, treating the whole of a problem together to improve the accuracy of each estimate that needs to be made, and dealing in a sensible way with the uncertainties inevitable such a problem – are all applicable to a range of other interesting problems, and the

approaches illustrated here will hopefully be of interest and of help to future work in other aspects of computer vision and problems further afield.

Appendix A

Marginalizing over the imaging parameters

In Chapter 5 we introduced a scheme for integrating over the imaging parameter uncertainty in the super-resolution model to leave an objective function which can be optimized directly with respect to the super-resolution image pixel intensities. This objective function is

$$\mathcal{L} = \frac{\nu}{2}\rho\left(\mathbf{Dx}, \alpha\right) + \frac{\beta}{2}f + \frac{1}{2}\log|\mathbf{S}| - \frac{\beta^2}{8}\mathbf{g}^T\mathbf{S}^{-1}\mathbf{g}, \tag{A.1}$$

where

$$\mathbf{S} = \left[\frac{\beta}{2}\mathbf{H} + \mathbf{V}^{-1}\right], \tag{A.2}$$

and where $\mathbf{V}$ is the covariance of the zero-mean Gaussian prior over the imaging parameter perturbation vector $\boldsymbol{\delta}$. In the first section of this appendix, we derive the gradient of this objective function with respect to $\mathbf{x}$, which we need in order to use gradient-descent methods to find the optimal super-resolution image estimate. In the second section of this appendix, we include a note on the numerical methods used to evaluate $\mathbf{g}$ and $\mathbf{H}$.

A.1 Differentiating the objective function *w.r.t.* x

Since $\mathbf{V}$ and $\mathbf{H}$ are both symmetric, $\mathbf{S}$ is also symmetric, so $\mathbf{S} = \mathbf{S}^T$, which will simplify some of the following expressions. The gradient of (A.1) with respect to $\mathbf{x}$ is

$$\begin{aligned}\frac{d\mathcal{L}}{d\mathbf{x}^T} &= \frac{\nu}{2}\mathbf{D}^T\rho'\left(\mathbf{D}\mathbf{x},\alpha\right)+\frac{\beta}{2}\frac{df}{d\mathbf{x}^T}-\frac{\beta^2}{4}\boldsymbol{\xi}^T\frac{d\mathbf{g}}{d\mathbf{x}^T}\\ &\quad+\frac{\beta}{4}\mathrm{vec}\left(\mathbf{S}^{-1}+\frac{\beta^2}{8}\boldsymbol{\xi}\boldsymbol{\xi}^T\right)^T\frac{d\mathrm{vec}(\mathbf{H})}{d\mathbf{x}^T},\end{aligned} \tag{A.3}$$

where

$$\boldsymbol{\xi} = \mathbf{S}^{-1}\mathbf{g}, \tag{A.4}$$

which we will now derive.

First, the objective function (A.1) is split into its four individual terms

$$\mathcal{L}_1 = \frac{\nu}{2}\rho\left(\mathbf{Dx}, \alpha\right) \tag{A.5}$$

$$\mathcal{L}_2 = \frac{\beta}{2} f \tag{A.6}$$

$$\mathcal{L}_3 = \frac{1}{2}\log|\mathbf{S}| \tag{A.7}$$

$$\mathcal{L}_4 = -\frac{\beta^2}{8}\mathbf{g}^T\mathbf{S}^{-1}\mathbf{g}. \tag{A.8}$$

Considering $\mathcal{L}_1$ and applying the chain rule, we obtain

$$\frac{\partial \mathcal{L}_1}{\partial \mathbf{x}} = \frac{\nu}{2}\mathbf{D}^T\frac{\partial \rho\left(\mathbf{Dx}, \alpha\right)}{\partial(\mathbf{Dx})} \tag{A.9}$$

$$= \frac{\nu}{2}\mathbf{D}^T\rho'\left(\mathbf{Dx}, \alpha\right), \tag{A.10}$$

where we define

$$\rho'(\mathbf{z}, \alpha) = [\rho'(z_1, \alpha), \ldots, \rho'(z_n, \alpha)] \tag{A.11}$$

$$\rho'(z, \alpha) = \begin{cases} 2z & \text{if } |z| < \alpha \\ 2\alpha \text{ sign}(z) & \text{otherwise.} \end{cases} \tag{A.12}$$

The gradient of $\mathcal{L}_2$ is simply

$$\frac{\partial \mathcal{L}_2}{\partial \mathbf{x}^T} = \frac{\beta}{2}\frac{\partial f}{\partial \mathbf{x}^T}, \tag{A.13}$$

which we leave in this form.

Using the chain rule for $\mathcal{L}_3$, we get

$$\frac{\partial \mathcal{L}_3}{\partial \mathbf{x}^T} = \frac{1}{2}\frac{\partial \log|\mathbf{S}|}{\partial \mathrm{vec}(\mathbf{S})^T}.\frac{\partial \mathrm{vec}(\mathbf{S})}{\partial \mathrm{vec}(\mathbf{H})^T}.\frac{\partial \mathrm{vec}(\mathbf{H})}{\partial \mathbf{x}^T}. \tag{A.14}$$

Then using the identity

$$\frac{\partial \log|\mathbf{X}|}{\partial \mathbf{X}} = \mathbf{X}^{-T} \tag{A.15}$$

we have

$$\frac{\partial \log|\mathbf{S}|}{\partial \mathrm{vec}(\mathbf{S})} = \mathrm{vec}\left(\mathbf{S}^{-T}\right)^T = \mathrm{vec}\left(\mathbf{S}^{-1}\right)^T, \tag{A.16}$$

and with the definition of $\mathbf{S}$ in (A.2), we have

$$\frac{\partial \mathrm{vec}(\mathbf{S})}{\partial \mathrm{vec}(\mathbf{H})^T} = \frac{\beta}{2}, \tag{A.17}$$

so

$$\frac{\partial \mathcal{L}_3}{\partial \mathbf{x}^T} = \frac{\beta}{4}\mathrm{vec}\left(\mathbf{S}^{-T}\right)^T \frac{\partial \mathrm{vec}(\mathbf{H})}{\partial \mathbf{x}^T}. \tag{A.18}$$

The final term $\mathcal{L}_4$ is more complicated because both $\mathbf{g}$ and $\mathbf{H}$ (hence $\mathbf{S}$) are functions of $\mathbf{x}$. First holding $\mathbf{S}$ constant and using the identity

$$\frac{\partial \mathbf{z}^T\mathbf{A}\mathbf{z}}{\partial \mathbf{z}} = \left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{z} \tag{A.19}$$

then

$$\frac{\partial \mathcal{L}_4}{\partial \mathbf{g}^T} = -\frac{\beta^2}{8}\left[\left(\mathbf{S}^{-1}+\mathbf{S}^{-T}\right)\mathbf{g}\right]^T \tag{A.20}$$

$$= -\frac{\beta^2}{4}\mathbf{g}^T\mathbf{S}^{-1}. \tag{A.21}$$

Second, holding $\mathbf{g}$ constant and using the chain rule, we obtain

$$\frac{\partial \mathcal{L}_4}{\partial \mathrm{vec}\left(\mathbf{H}\right)^T} = \frac{\partial \mathcal{L}_4}{\partial \mathrm{vec}\left(\mathbf{S}^{-1}\right)^T}.\frac{\partial \mathrm{vec}\left(\mathbf{S}^{-1}\right)}{\partial \mathrm{vec}\left(\mathbf{S}\right)^T}.\frac{\partial \mathrm{vec}\left(\mathbf{S}\right)}{\partial \mathrm{vec}\left(\mathbf{H}\right)^T}, \tag{A.22}$$

and as above, the final factor evaluates to $\frac{\beta}{2}$.

Using the identities

$$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{a}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{a}^T, \tag{A.23}$$

$$\mathrm{vec}\left(\mathbf{b}\mathbf{c}^T\right) = \left[\mathbf{c}^T \otimes \mathbf{b}^T\right]^T \tag{A.24}$$

where $\otimes$ is the Kronecker product (see *e.g.* [71, 72]), we can see that

$$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{a}}{\partial \mathrm{vec}\left(\mathbf{X}\right)^T} = \mathrm{vec}\left(\mathbf{a}\mathbf{a}^T\right)^T \tag{A.25}$$

$$= \mathbf{a}^T \otimes \mathbf{a}^T \tag{A.26}$$

so

$$\frac{\partial \mathcal{L}_4}{\partial \mathrm{vec}\left(\mathbf{H}\right)^T} = \frac{\beta}{2}\frac{\partial \mathcal{L}_4}{\partial \mathrm{vec}\left(\mathbf{S}^{-1}\right)^T}.\frac{\partial \mathrm{vec}\left(\mathbf{S}^{-1}\right)}{\partial \mathrm{vec}\left(\mathbf{S}\right)^T} \tag{A.27}$$

$$= -\frac{\beta^3}{16}\frac{\partial \mathbf{g}^T\mathbf{S}^{-1}\mathbf{g}}{\partial \mathrm{vec}\left(\mathbf{S}^{-1}\right)^T}.\frac{\partial \mathrm{vec}\left(\mathbf{S}^{-1}\right)}{\partial \mathrm{vec}\left(\mathbf{S}\right)^T} \tag{A.28}$$

$$= -\frac{\beta^3}{16}\left[\mathbf{g}^T \otimes \mathbf{g}^T\right]\frac{\partial \mathrm{vec}\left(\mathbf{S}^{-1}\right)}{\partial \mathrm{vec}\left(\mathbf{S}\right)^T} \tag{A.29}$$

Now using the identities

$$\frac{\partial \mathrm{vec}\left(\mathbf{X}^{-1}\right)}{\partial \mathrm{vec}\left(\mathbf{X}\right)^T} = -\mathbf{X}^{-T} \otimes \mathbf{X}^{-1} \tag{A.30}$$

$$\left[\mathbf{A} \otimes \mathbf{B}\right]\left[\mathbf{C} \otimes \mathbf{D}\right] = \mathbf{AC} \otimes \mathbf{BD}, \tag{A.31}$$

we get

$$\frac{\partial \mathcal{L}_4}{\partial \mathrm{vec}\left(\mathbf{H}\right)^T} = -\frac{\beta^3}{16}\left[\mathbf{g}^T \otimes \mathbf{g}^T\right]\left[-\mathbf{S}^{-T} \otimes \mathbf{S}^{-1}\right] \tag{A.32}$$

$$= \frac{\beta^3}{16}\left[\mathbf{g}^T \otimes \mathbf{g}^T\right]\left[\mathbf{S}^{-1} \otimes \mathbf{S}^{-1}\right] \tag{A.33}$$

$$= \frac{\beta^3}{16}\left[\mathbf{g}^T\mathbf{S}^{-1} \otimes \mathbf{g}^T\mathbf{S}^{-1}\right] \tag{A.34}$$

Collecting up all the individual terms, we have

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}^T} = \frac{\partial \mathcal{L}_1}{\partial \mathbf{x}^T} + \frac{\partial \mathcal{L}_2}{\partial \mathbf{x}^T} + \frac{\partial \mathcal{L}_3}{\partial \mathbf{x}^T} + \frac{\partial \mathcal{L}_4}{\partial \mathbf{g}^T}.\frac{\partial \mathbf{g}}{\partial \mathbf{x}^T} + \frac{\partial \mathcal{L}_4}{\partial \text{vec}\,(\mathbf{H})^T}.\frac{\partial \text{vec}\,(\mathbf{H})}{\partial \mathbf{x}^T} \quad \text{(A.35)}$$

$$= \frac{\nu}{2}\mathbf{D}^T\rho'\,(\mathbf{D}\mathbf{x},\alpha) + \frac{\beta}{2}\frac{\partial f}{\partial \mathbf{x}^T} + \frac{\beta}{4}\text{vec}\left(\mathbf{S}^{-T}\right)^T\frac{\partial \text{vec}\,(\mathbf{H})}{\partial \mathbf{x}^T}$$
$$-\frac{\beta^2}{4}\mathbf{g}^T\mathbf{S}^{-1}.\frac{\partial \mathbf{g}}{\partial \mathbf{x}^T} + \frac{\beta^3}{16}\left[\mathbf{g}^T\mathbf{S}^{-1}\otimes\mathbf{g}^T\mathbf{S}^{-1}\right]\frac{\partial \text{vec}\,(\mathbf{H})}{\partial \mathbf{x}^T} \quad \text{(A.36)}$$

$$= \frac{\nu}{2}\mathbf{D}^T\rho'\,(\mathbf{D}\mathbf{x},\alpha) + \frac{\beta}{2}\frac{\partial f}{\partial \mathbf{x}^T} - \frac{\beta^2}{4}\mathbf{g}^T\mathbf{S}^{-1}\frac{\partial \mathbf{g}}{\partial \mathbf{x}^T}$$
$$+\frac{\beta}{4}\left[\text{vec}\left(\mathbf{S}^{-T}\right)^T + \frac{\beta^2}{4}\left[\mathbf{g}^T\mathbf{S}^{-1}\otimes\mathbf{g}^T\mathbf{S}^{-1}\right]\right]\frac{\partial \text{vec}\,(\mathbf{H})}{\partial \mathbf{x}^T} \quad \text{(A.37)}$$

Now letting $\xi = \mathbf{S}^{-1}\mathbf{g}$,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}^T} = \frac{\nu}{2}\mathbf{D}^T\rho'\,(\mathbf{D}\mathbf{x},\alpha) + \frac{\beta}{2}\frac{\partial f}{\partial \mathbf{x}^T} - \frac{\beta^2}{4}\boldsymbol{\xi}^T\frac{\partial \mathbf{g}}{\partial \mathbf{x}^T}$$
$$+\frac{\beta}{4}\left[\text{vec}\left(\mathbf{S}^{-T}\right)^T + \frac{\beta^2}{4}\left[\boldsymbol{\xi}^T\otimes\boldsymbol{\xi}^T\right]\right]\frac{\partial \text{vec}\,(\mathbf{H})}{\partial \mathbf{x}^T} \quad \text{(A.38)}$$

$$= \frac{\nu}{2}\mathbf{D}^T\rho'\,(\mathbf{D}\mathbf{x},\alpha) + \frac{\beta}{2}\frac{\partial f}{\partial \mathbf{x}^T} - \frac{\beta^2}{4}\boldsymbol{\xi}^T\frac{\partial \mathbf{g}}{\partial \mathbf{x}^T}$$
$$+\frac{\beta}{4}\left[\text{vec}\left(\mathbf{S}^{-T}\right)^T + \frac{\beta^2}{4}\text{vec}\left(\boldsymbol{\xi}\boldsymbol{\xi}^T\right)^T\right]\frac{\partial \text{vec}\,(\mathbf{H})}{\partial \mathbf{x}^T} \quad \text{(A.39)}$$

$$= \frac{\nu}{2}\mathbf{D}^T\rho'\,(\mathbf{D}\mathbf{x},\alpha) + \frac{\beta}{2}\frac{\partial f}{\partial \mathbf{x}^T} - \frac{\beta^2}{4}\boldsymbol{\xi}^T\frac{\partial \mathbf{g}}{\partial \mathbf{x}^T}$$
$$+\frac{\beta}{4}\text{vec}\left(\mathbf{S}^{-T} + \frac{\beta^2}{4}\boldsymbol{\xi}\boldsymbol{\xi}^T\right)^T\frac{\partial \text{vec}\,(\mathbf{H})}{\partial \mathbf{x}^T}. \quad \text{(A.40)}$$

The identity (A.24) was applied in reverse to get from (A.38) to (A.39), and the result in (A.40) is now identical to (A.3) earlier.

A.2 Numerical approximations for g and H

Considering a single image with an n-parameter motion model, so that $\boldsymbol{\delta}_{1:n}$ relate to the geometric registration, δ_{n+1} relates to λ_1, δ_{n+2} relates to λ_2. Forgetting about the dependence on γ for the moment, let

$$\begin{aligned} e(\boldsymbol{\delta}) &= \left|\left|\mathbf{y} - \lambda_1 \mathbf{W}\left(\bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{1:n}\right)\mathbf{x} - \lambda_2 \mathbf{1}\right|\right|_2^2 && \text{(A.41)} \\ e'(\boldsymbol{\delta}) &= \frac{d}{d\mathbf{x}} e(\boldsymbol{\delta}) && \text{(A.42)} \\ &= -2\lambda_1 \mathbf{W}\left(\bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{1:n}\right)^T \left(\mathbf{y} - \lambda_1 \mathbf{W}\left(\bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{1:n}\right)\mathbf{x} - \lambda_2 \mathbf{1}\right) && \text{(A.43)} \end{aligned}$$

where the superscript k is omitted for notational clarity. Remember that

$$\begin{aligned} \lambda_1 &= \bar{\lambda_1} + \delta_{n+1} && \text{(A.44)} \\ \lambda_2 &= \bar{\lambda_2} + \delta_{n+2}, && \text{(A.45)} \end{aligned}$$

so these $\boldsymbol{\lambda}$ values also depend on the vector $\boldsymbol{\delta}$. Immediately, it can be observed that

$$\begin{aligned} f &= e(\mathbf{0}) && \text{(A.46)} \\ \frac{df}{d\mathbf{x}} &= e'(\mathbf{0}). && \text{(A.47)} \end{aligned}$$

We now begin by finding the elements of $\mathbf{g}$ and $\mathbf{H}$ which depend only on the

photometric parameters. Differentiating *w.r.t.* λ_1 and λ_2 gives

$$\frac{\partial e(\boldsymbol{\delta})}{\partial \lambda_1} = -2\mathbf{x}^T\mathbf{W}^T(\mathbf{y} - \lambda_1\mathbf{W}\mathbf{x} - \lambda_2\mathbf{1}) \quad (A.48)$$

$$\frac{\partial e(\boldsymbol{\delta})}{\partial \lambda_2} = -2\sum(\mathbf{y} - \lambda_1\mathbf{W}\mathbf{x} - \lambda_2\mathbf{1}) \quad (A.49)$$

$$\frac{\partial^2 e(\boldsymbol{\delta})}{\partial \lambda_1^2} = 2\mathbf{x}^T\mathbf{W}^T\mathbf{W}\mathbf{x} \quad (A.50)$$

$$\frac{\partial^2 e(\boldsymbol{\delta})}{\partial \lambda_2^2} = 2M \quad (A.51)$$

$$\frac{\partial^2 e(\boldsymbol{\delta})}{\partial \lambda_1 \partial \lambda_2} = 2\sum \mathbf{W}\mathbf{x}, \quad (A.52)$$

where M is the number of pixels in vectorized image $\mathbf{y}$. The two gradient expressions (A.48) and (A.49) give values for the final two elements of $\mathbf{g}$, and the remaining three terms form the bottom left 2×2 block of $\mathbf{H}$.

Finding the gradients of these with respect to $\mathbf{x}$ gives

$$\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial e(\boldsymbol{\delta})}{\partial \lambda_1}\right) = -2\mathbf{W}^T(\mathbf{y} - 2\lambda_1\mathbf{W}\mathbf{x} - \lambda_2\mathbf{1}) \quad (A.53)$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial e(\boldsymbol{\delta})}{\partial \lambda_2}\right) = 2\lambda_1\mathbf{W}^T\mathbf{1} \quad (A.54)$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial^2 e(\boldsymbol{\delta})}{\partial \lambda_1^2}\right) = 4\mathbf{x}^T\mathbf{W}^T\mathbf{W} \quad (A.55)$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial^2 e(\boldsymbol{\delta})}{\partial \lambda_2^2}\right) = \mathbf{0} \quad (A.56)$$

$$\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial^2 e(\boldsymbol{\delta})}{\partial \lambda_1 \partial \lambda_2}\right) = 2\mathbf{W}^T\mathbf{1}. \quad (A.57)$$

The upper $n \times n$ block of $\mathbf{H}$, and the first n elements of $\mathbf{g}$, are found numerically. For simple motion models, the full analytic solution could be found, but the renormalization of $\mathbf{W}$ as the geometric registration changes can make the solution for

more general motion models increasingly expensive. Let

$$e_i(\boldsymbol{\delta}) = e(\boldsymbol{\delta} + \epsilon \mathbf{d}_i) \tag{A.58}$$

$$e'_i(\boldsymbol{\delta}) = e'(\boldsymbol{\delta} + \epsilon \mathbf{d}_i) \tag{A.59}$$

where $\mathbf{d}_i$ is a vector consisting of all zeros, except for a 1 at position i. Now we can see that

$$\mathbf{g}_{1:n} = \frac{d}{d\boldsymbol{\delta}_{1:n}} e(\boldsymbol{\delta}) \tag{A.60}$$

$$g_i = \frac{1}{2\epsilon}(e_i - e_{-i}) \tag{A.61}$$

$$\frac{dg_i}{d\mathbf{x}} = \frac{1}{2\epsilon}\left(e'_i - e'_{-i}\right). \tag{A.62}$$

Note that for each element g_i, only one of the $\mathbf{W}^{(k)}$ matrices from the set $\left\{\mathbf{W}^{(k)}\right\}$ needs to be considered, since it is assumed that registration parameters from different images are independent. Also note that $g_{n+1} = \frac{\partial e}{\partial \lambda_1}$ and $g_{n+2} = \frac{\partial e}{\partial \lambda_2}$ (defined in (A.48) and (A.49) above), completing the Jacobian vector.

The Hessian, $\mathbf{H}$, will be a symmetric block-diagonal matrix, because only parameters from the same input low-resolution image have any combined effect. For the upper square submatrix of the Hessian relating only to geometric parameters of

one image only,

$$\mathbf{H} = \frac{d^2}{d\boldsymbol{\delta}^2} e(\boldsymbol{\delta}) \tag{A.63}$$

$$H_{ij} = \begin{cases} \frac{1}{\epsilon^2}\left(e_i + e_{-i} - 2e_0\right) & i = j \\ \frac{1}{2\epsilon^2}\left(e_{i,j} + e_{-i,-j} - e_{i,-j} - e_{-i,j}\right) & \text{otherwise} \end{cases} \tag{A.64}$$

$$\frac{dH_{ij}}{d\mathbf{x}} = \begin{cases} \frac{1}{\epsilon^2}\left(e'_i + e'_{-i} - 2e'_0\right) & i = j \\ \frac{1}{2\epsilon^2}\left(e'_{i,j} + e'_{-i,-j} - e'_{i,-j} - e'_{-i,j}\right) & \text{otherwise.} \end{cases} \tag{A.65}$$

For the cross-terms involving both geometric and photometric registration parameters, where i indexes the geometric parameter

$$H_{i,(n+1)} = \frac{1}{2\epsilon}\left[\left(\frac{\partial e_{+i}(\boldsymbol{\delta})}{\partial \lambda_1}\right) - \left(\frac{\partial e_{-i}(\boldsymbol{\delta})}{\partial \lambda_1}\right)\right] \tag{A.66}$$

$$H_{i,(n+2)} = \frac{1}{2\epsilon}\left[\left(\frac{\partial e_{+i}(\boldsymbol{\delta})}{\partial \lambda_2}\right) - \left(\frac{\partial e_{-i}(\boldsymbol{\delta})}{\partial \lambda_2}\right)\right] \tag{A.67}$$

$$\frac{\partial H_{i,(n+1)}}{\partial \mathbf{x}} = \frac{1}{2\epsilon}\left[\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial e_{+i}(\boldsymbol{\delta})}{\partial \lambda_1}\right) - \frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial e_{-i}(\boldsymbol{\delta})}{\partial \lambda_1}\right)\right] \tag{A.68}$$

$$\frac{\partial H_{i,(n+2)}}{\partial \mathbf{x}} = \frac{1}{2\epsilon}\left[\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial e_{+i}(\boldsymbol{\delta})}{\partial \lambda_2}\right) - \frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial e_{-i}(\boldsymbol{\delta})}{\partial \lambda_2}\right)\right] \tag{A.69}$$

Finally, when the shared PSF parameter γ is also considered, it can be treated exactly as the terms for the geometric registration (in terms of finite difference approximation techniques), except that its effect on all images simultaneously must be taken into account, so the final row and column of $\mathbf{H}$ can be completely non-zero.

Appendix B

Marginalization over the super-resolution image

In this section we derive the equations for marginalization over the super-resolution image estimate, leaving a marginal probability of the input dataset given the imaging parameters. This is the approach taken by Tipping and Bishop [112], though here we use the super-resolution model described in Chapter 3, which includes photometric components not present in [112].

We also derive the gradient expression for the objective function in terms of the system matrices, $\mathbf{W}^{(k)}$, which can be used in gradient-descent-based algorithms for finding the ML estimate of the registration parameters.

B.1 Basic derivation

This section covers the derivation of the marginal probability of the low-resolution image set conditioned on the registration values. The objective function which we optimize with respect to the registration parameters is equal to the negative log of the marginal probability of the input images, neglecting any additive constants which are not functions of the imaging parameters. The log marginal is derived in Section B.1.1, and in Section B.1.2 this is shown to have an equivalent form as a straightforward Gaussian distribution over the low-resolution images.

B.1.1 The objective function

This derivation begins by assuming a Gaussian distribution over the high-resolution image and hence evaluating the integral over the image exactly, to obtain a closed-form expression for $p(\mathbf{y}|\boldsymbol{\phi})$. As in Chapter 5, $\boldsymbol{\phi}$ is used here to represent the set of γ, $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ parameters.

As a starting point, we take the model as described in Chapter 3. In particular, equations (3.22) and (3.9), which are

$$\begin{aligned} p(\mathbf{x}) &= (2\pi)^{-\frac{N}{2}} |\mathbf{Z}_x|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mathbf{x}^T\mathbf{Z}_x^{-1}\mathbf{x}\right\} \\ p\left(\left\{\mathbf{y}^{(k)}\right\}|\mathbf{x},\boldsymbol{\phi}\right) &= \left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}} \exp\left\{-\frac{\beta}{2}\sum_{k=1}^{K}\left\|\mathbf{r}^{(k)}\right\|_2^2\right\}, \end{aligned}$$

where

$$\mathbf{r}^{(k)} = \mathbf{y}^{(k)} - \lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} - \boldsymbol{\lambda}_2^{(k)}.$$

We also have Bayes' Theorem applied to the super-resolution model to obtain the posterior probability of the high-resolution image as in equation (3.18):

$$p\left(\mathbf{x}\left|\left\{\mathbf{y}^{(k)}\right\},\boldsymbol{\phi}\right.\right) = \frac{p\left(\left\{\mathbf{y}^{(k)}\right\}\left|\,\mathbf{x},\boldsymbol{\phi}\right.\right)p(\mathbf{x})}{p\left(\left\{\mathbf{y}^{(k)}\right\}\left|\,\boldsymbol{\phi}\right.\right)}.$$

Here, the normalizing constant in the denominator gives us the marginal probability over $\mathbf{y}$, so the integral we require is

$$p\left(\left\{\mathbf{y}^{(k)}\right\}\left|\,\boldsymbol{\phi}\right.\right) = \int p\left(\left\{\mathbf{y}^{(k)}\right\}\left|\,\mathbf{x},\boldsymbol{\phi}\right.\right)p(\mathbf{x})\mathrm{d}\mathbf{x}. \tag{B.1}$$

In order to keep the derivation as uncluttered as possible, we will switch to stacked notation so that $\mathbf{y} = \left[\mathbf{y}^{(1)T},\ldots,\mathbf{y}^{(K)T}\right]^T$, with $\mathbf{W}$ and $\boldsymbol{\lambda}_2$ being similarly stacked, and $\boldsymbol{\Lambda}_1$ being a square matrix whose diagonal entries contain the corresponding λ_1 values for each low-resolution image. We now have

$$\begin{aligned}
p\left(\mathbf{y}\left|\,\boldsymbol{\phi}\right.\right) &= \int p\left(\mathbf{y}\left|\,\mathbf{x},\boldsymbol{\phi}\right.\right)p(\mathbf{x})\mathrm{d}\mathbf{x} && \text{(B.2)}\\
&= \int (2\pi)^{-\frac{N}{2}}\left|\mathbf{Z}_x\right|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\mathbf{x}^T\mathbf{Z}_x^{-1}\mathbf{x}\right\}\\
&\quad\times\left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}}\exp\left\{-\frac{\beta}{2}\left\|\mathbf{y}-\boldsymbol{\Lambda}_1\mathbf{W}\mathbf{x}-\boldsymbol{\lambda}_2\right\|_2^2\right\}\mathrm{d}\mathbf{x} && \text{(B.3)}\\
&= (2\pi)^{-\frac{N}{2}}\left|\mathbf{Z}_x\right|^{-\frac{1}{2}}\left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}}\exp\left\{-\frac{\beta}{2}\left\|\mathbf{y}-\boldsymbol{\lambda}_2\right\|_2^2\right\}\\
&\quad\times\int\exp\left\{-\frac{1}{2}\mathbf{x}^T\left[\mathbf{Z}_x^{-1}\mathbf{x}+\beta\mathbf{W}^T\boldsymbol{\Lambda}_1^2\mathbf{W}\right]\mathbf{x}+\beta\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\boldsymbol{\Lambda}_1\mathbf{W}\mathbf{x}\right\}\mathrm{d}\mathbf{x} && \text{(B.4)}\\
&= (2\pi)^{-\frac{N}{2}}\left|\mathbf{Z}_x\right|^{-\frac{1}{2}}\left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}}\exp\left\{-\frac{\beta}{2}\left\|\mathbf{y}-\boldsymbol{\lambda}_2\right\|_2^2\right\}\\
&\quad\times\int\exp\left\{-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}+\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}\right\}\mathrm{d}\mathbf{x}, && \text{(B.5)}
\end{aligned}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are defined

$$\boldsymbol{\Sigma}^{-1} = \mathbf{Z}_x^{-1} + \beta \mathbf{W}^T \boldsymbol{\Lambda}_1^2 \mathbf{W} \tag{B.6}$$

$$\boldsymbol{\mu} = \beta \boldsymbol{\Sigma} \mathbf{W}^T \boldsymbol{\Lambda}_1 (\mathbf{y} - \boldsymbol{\lambda}_2) . \tag{B.7}$$

These are equivalent to the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ derivations in [112], and to equations (3.27) and (3.27), which use the usual non-stacked notation.

Completing the square in the exponent and noticing that the value of the integral is the inverse of the normalizing constant on the equivalent Gaussian distribution, we have

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\theta}) &= (2\pi)^{-\frac{N}{2}} |\mathbf{Z}_x|^{-\frac{1}{2}} \left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}} \exp\left\{-\frac{\beta}{2}\|\mathbf{y} - \boldsymbol{\lambda}_2\|_2^2 + \frac{1}{2}\boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}\right\} \\ &\quad \times \int \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right\} \mathrm{d}\mathbf{x} \qquad \text{(B.8)} \\ &= (2\pi)^{-\frac{N}{2}} |\mathbf{Z}_x|^{-\frac{1}{2}} \left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}} \exp\left\{-\frac{\beta}{2}\|\mathbf{y} - \boldsymbol{\lambda}_2\|_2^2 + \frac{1}{2}\boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}\right\} \\ &\quad \times (2\pi)^{\frac{N}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}} \qquad \text{(B.9)} \\ &= \left(\frac{|\boldsymbol{\Sigma}|}{|\mathbf{Z}_x|}\right)^{\frac{1}{2}} \left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}} \exp\left\{-\frac{\beta}{2}\|\mathbf{y} - \boldsymbol{\lambda}_2\|_2^2 + \frac{1}{2}\boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}\right\} . \qquad \text{(B.10)} \end{aligned}$$

Taking the first part of the exponent, $\|\mathbf{y} - \boldsymbol{\lambda}_2\|_2^2$, and completing the square, we

have

$$
\begin{aligned}
-\frac{\beta}{2}\left\|\mathbf{y}-\boldsymbol{\lambda}_2\right\|_2^2 &= -\frac{\beta}{2}\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\left(\mathbf{y}-\boldsymbol{\lambda}_2\right) && \text{(B.11)}\\
&= -\frac{\beta}{2}\left(\mathbf{y}-\boldsymbol{\lambda}_2-\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)^T\left(\mathbf{y}-\boldsymbol{\lambda}_2-\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)\\
&\quad -\beta\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}+\frac{\beta}{2}\boldsymbol{\mu}^T\mathbf{W}^T\boldsymbol{\Lambda}_1^2\mathbf{W}\boldsymbol{\mu} && \text{(B.12)}\\
&= -\frac{\beta}{2}\left\|r\right\|_2^2-\beta\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\frac{\beta}{2}\boldsymbol{\mu}^T\mathbf{W}^T\boldsymbol{\Lambda}_1^2\mathbf{W}\boldsymbol{\mu} && \text{(B.13)}\\
&= -\frac{\beta}{2}\left\|r\right\|_2^2-\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\frac{\beta}{2}\boldsymbol{\mu}^T\mathbf{W}^T\boldsymbol{\Lambda}_1^2\mathbf{W}\boldsymbol{\mu}, && \text{(B.14)}
\end{aligned}
$$

where

$$
\mathbf{r}_\mu = \mathbf{y}-\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}-\boldsymbol{\lambda}_2 \qquad \text{(B.15)}
$$

is the residual error found by projecting the MAP super-resolution image estimate $\boldsymbol{\mu}$ back into each of the low-resolution input images and taking the difference between the resulting projection and the observed image, also accounting for the photometric effects.

Now including the $+\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ term, the full exponent of (B.10) is

$$
\begin{aligned}
-\frac{\beta}{2}\left\|\mathbf{y}-\boldsymbol{\lambda}_2\right\|_2^2+\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} &= \frac{\beta}{2}\left\|r\right\|_2^2-\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\frac{\beta}{2}\boldsymbol{\mu}^T\mathbf{W}^T\boldsymbol{\Lambda}_1^2\mathbf{W}\boldsymbol{\mu} && \text{(B.16)}\\
&= \frac{\beta}{2}\left\|\mathbf{r}_\mu\right\|_2^2-\frac{1}{2}\boldsymbol{\mu}^T\left[\boldsymbol{\Sigma}^{-1}-\beta\mathbf{W}^T\boldsymbol{\Lambda}_1^2\mathbf{W}\right]\boldsymbol{\mu} && \text{(B.17)}\\
&= \frac{\beta}{2}\left\|\mathbf{r}_\mu\right\|_2^2-\frac{1}{2}\boldsymbol{\mu}^T\mathbf{Z}_x\boldsymbol{\mu} && \text{(B.18)}
\end{aligned}
$$

This is the form of the marginal probability of the input dataset that leads to the form of the objective function of [112], though ours also includes the photometric

part of the model. The log of the marginal $\mathbf{y}$ distribution is

$$\begin{aligned}\log p(\mathbf{y}|\boldsymbol{\theta}) &= -\frac{1}{2}\Big[\beta\|\mathbf{y}-\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}-\boldsymbol{\lambda}_2\|_2^2+\boldsymbol{\mu}^T\mathbf{Z}_x\boldsymbol{\mu}+\log|\mathbf{Z}_x| \\ &\quad -\log|\boldsymbol{\Sigma}|-KM\log\beta+KM\log(2\pi)\Big], \end{aligned} \tag{B.19}$$

which is a convenient when $KM > N$, because the log-determinant steps take matrices of size $N \times N$ rather than of size $KM \times KM$, even though the distribution is over the vector $\mathbf{y}$ which is itself of size KM.

B.1.2 The Gaussian distribution over y

To express the marginal probability as a straightforward Gaussian distribution over $\mathbf{y}$, we return to the exponent of (B.10), and instead rearrange it

$$\begin{aligned}&-\frac{\beta}{2}\|\mathbf{y}-\boldsymbol{\lambda}_2\|_2^2+\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ &= -\frac{\beta}{2}\|\mathbf{y}-\boldsymbol{\lambda}_2\|_2^2+\frac{\beta^2}{2}(\mathbf{y}-\boldsymbol{\lambda}_2)^T\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\Sigma}\mathbf{W}^T\boldsymbol{\Lambda}_1^T(\mathbf{y}-\boldsymbol{\lambda}_2) &\text{(B.20)}\\ &= -\frac{1}{2}(\mathbf{y}-\boldsymbol{\lambda}_2)^T\left[\beta\mathbf{I}-\beta^2\boldsymbol{\Lambda}_1\mathbf{W}\left[\mathbf{Z}_x^{-1}+\beta\mathbf{W}^T\boldsymbol{\Lambda}_1^2\mathbf{W}\right]^{-1}\mathbf{W}^T\boldsymbol{\Lambda}_1^T\right](\mathbf{y}-\boldsymbol{\Lambda}_1). &\text{(B.21)}\end{aligned}$$

Using the Woodbury Matrix Identity [52],

$$(\mathbf{A}+\mathbf{UCV})^{-1} = \mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{C}^{-1}\mathbf{V}\mathbf{A}^{-1}\mathbf{U}\right)\mathbf{V}\mathbf{A}^{-1}, \tag{B.22}$$

this becomes

$$\begin{aligned} & -\frac{\beta}{2}\left\|\mathbf{y}-\boldsymbol{\lambda}_2\right\|_2^2+\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ &= -\frac{1}{2}\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\left[\beta^{-1}\mathbf{I}+\boldsymbol{\Lambda}_1\mathbf{W}\mathbf{Z}_x\mathbf{W}^T\boldsymbol{\Lambda}_1^T\right]^{-1}\left(\mathbf{y}-\boldsymbol{\lambda}_2\right) \qquad \text{(B.23)} \\ &= -\frac{1}{2}\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\mathbf{Z}_y\left(\mathbf{y}-\boldsymbol{\lambda}_2\right). \qquad \text{(B.24)} \end{aligned}$$

Returning to the distribution of (B.10), we have

$$\begin{aligned} p\left(\mathbf{y}\left|\boldsymbol{\theta}\right.\right) &= \left(\frac{\left|\boldsymbol{\Sigma}\right|}{\left|\mathbf{Z}_x\right|}\right)^{\frac{1}{2}}\left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}}\exp\left\{-\frac{1}{2}\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\mathbf{Z}_y\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)\right\} \qquad \text{(B.25)} \\ &= (2\pi)^{-\frac{KM}{2}}\left|\mathbf{Z}_y\right|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\mathbf{Z}_y\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)\right\}, \qquad \text{(B.26)} \end{aligned}$$

where the change in normalizing constant can be verified using the definitions of $\mathbf{Z}_y$ and $\boldsymbol{\Sigma}$ and the Matrix Inversion Lemma [94],

$$\left|\mathbf{A}+\mathbf{U}\mathbf{C}\mathbf{V}^T\right| = \left|\mathbf{C}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right|\left|\mathbf{C}\right|\left|\mathbf{A}\right| \qquad \text{(B.27)}$$

to show that

$$\left|\mathbf{Z}_y\right| = \left|\boldsymbol{\Sigma}^{-1}\right|\left|\mathbf{Z}_x\right|\beta^{-\frac{KM}{2}}. \qquad \text{(B.28)}$$

Alternatively, the normalization constant for (B.26) can simply be inferred from the exponent, because we expect a normalized Gaussian distribution over $\mathbf{y}$.

B.2 Gradient *w.r.t.* the registration parameters

In order to optimize the log marginal of (B.19) with respect to the registration parameters using gradient-descent methods like SCG, we need to find the gradient of the function with respect to the $\mathbf{W}$, which is parameterized by the geometric registration $\boldsymbol{\phi}$ and the PSF parameter γ.

We begin in Section B.2.1 by finding the derivatives of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ with respect to $\mathbf{W}$, and put them together in Section B.2.2 to produce the final derivative.

B.2.1 Gradients *w.r.t.* μ and Σ

Definitions of $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ are given in (B.6) and (B.7). Both are functions of $\mathbf{W}$, and μ is also a function of $\boldsymbol{\Sigma}$ as well, so we begin with $\boldsymbol{\Sigma}$. We have

$$\frac{\partial \mathrm{vec}\left(\boldsymbol{\Sigma}\right)}{\partial \mathrm{vec}\left(\mathbf{W}\right)^T} = \frac{\partial \mathrm{vec}\left(\boldsymbol{\Sigma}\right)}{\partial \mathrm{vec}\left(\boldsymbol{\Sigma}^{-1}\right)^T} \cdot \frac{\partial}{\partial \mathrm{vec}\left(\mathbf{W}\right)^T} \left[\mathbf{Z}_x^{-1} + \beta \mathbf{W}^T \boldsymbol{\Lambda}_1^2 \mathbf{W}\right]. \tag{B.29}$$

We make use of the fact that

$$\begin{aligned} \mathrm{vec}\left(\beta \mathbf{W} \boldsymbol{\Lambda}_1^2 \mathbf{W}\right) &= \beta \left(\mathbf{I}_{KM} \otimes \mathbf{W}^T \boldsymbol{\Lambda}_1^2\right) \mathrm{vec}\left(\mathbf{W}\right) && \text{(B.30)} \\ &= \beta \mathbf{K}_{KM,N} \left(\mathbf{I}_{KM} \otimes \mathbf{W}^T \boldsymbol{\Lambda}_1^2\right) \mathrm{vec}\left(\mathbf{W}\right), && \text{(B.31)} \end{aligned}$$

and

$$\frac{\partial \mathrm{vec}\left(\boldsymbol{\Sigma}\right)}{\partial \mathrm{vec}\left(\boldsymbol{\Sigma}^{-1}\right)^T} = \boldsymbol{\Sigma}^T \otimes -\boldsymbol{\Sigma}, \tag{B.32}$$

to obtain

$$\frac{\partial \mathrm{vec}\left(\boldsymbol{\Sigma}^{-1}\right)}{\partial \mathrm{vec}\left(\mathbf{W}\right)^T} = \left(\mathbf{I}_{KM} + \mathbf{K}_{KM,N}\right)\left(\mathbf{I}_{KM} \otimes \mathbf{W}^T\boldsymbol{\Lambda}_1^2\right). \tag{B.33}$$

so that

$$\begin{aligned}\frac{\partial \mathrm{vec}\left(\boldsymbol{\Sigma}\right)}{\partial \mathrm{vec}\left(\mathbf{W}\right)^T} &= -\beta\left(\boldsymbol{\Sigma}\otimes\boldsymbol{\Sigma}\right)\left(\mathbf{I}_{KM} + \mathbf{K}_{KM,N}\right)\left(\mathbf{I}_{KM}\otimes\mathbf{W}^T\boldsymbol{\Lambda}_1^2\right) &\text{(B.34)}\\ &= -2\beta\left(\boldsymbol{\Sigma}\otimes\boldsymbol{\Sigma}\right)\left(\mathbf{I}_{KM}\otimes\mathbf{W}^T\boldsymbol{\Lambda}_1^2\right) &\text{(B.35)}\\ &= -2\beta\left(\boldsymbol{\Sigma}\otimes\boldsymbol{\Sigma}\mathbf{W}^T\boldsymbol{\Lambda}_1^2\right), &\text{(B.36)}\end{aligned}$$

where the expression is simplified significantly because $\boldsymbol{\Sigma}$ is a symmetric matrix.

Now considering $\boldsymbol{\mu}$, we have

$$\frac{\partial \mathrm{vec}\left(\boldsymbol{\mu}\right)}{\partial \mathrm{vec}\left(\boldsymbol{\Sigma}\right)^T} = \left[\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\boldsymbol{\Lambda}_1\mathbf{W}\right]\otimes\beta\mathbf{I}_N \tag{B.37}$$

and also

$$\mathrm{vec}\left(\boldsymbol{\mu}\right) = \left[\left(\mathbf{y}-\boldsymbol{\Lambda}_1\right)\boldsymbol{\Lambda}_1^T\otimes\beta\boldsymbol{\Sigma}\right]\mathbf{K}_{N,KM}\mathrm{vec}\left(\mathbf{W}\right) \tag{B.38}$$

where the latter comes from the identities

$$\begin{aligned}\mathrm{vec}\left(\mathbf{A}\mathbf{W}^T\mathbf{B}\right) &= \left(\mathbf{C}^T\otimes\mathbf{A}\right)\mathrm{vec}\left(\mathbf{W}^T\right) &\text{(B.39)}\\ \mathrm{vec}\left(\mathbf{W}^T\right) &= \mathbf{K}_{N,KM}\mathrm{vec}\left(\mathbf{W}\right) &\text{(B.40)}\end{aligned}$$

where $\mathbf{A}$ and $\mathbf{B}$ are arbitrary matrices of appropriate sizes, and $\mathbf{W}$ is our usual

stacked system matrix of size $KM \times N$.

Including the dependence of $\boldsymbol{\Sigma}$ on $\mathbf{W}$, the partial derivative of $\boldsymbol{\mu}$ with respect to $\mathbf{W}$ is

$$\begin{aligned}
\frac{\partial \mathrm{vec}\left(\boldsymbol{\mu}\right)}{\partial \mathrm{vec}\left(\mathbf{W}\right)^T} &= \left[\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T \boldsymbol{\Lambda}_1^T \otimes \beta\boldsymbol{\Sigma}\right] \mathbf{K}_{N,KM} \\
&\quad -2\beta\left(\left[\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T \boldsymbol{\Lambda}_1\mathbf{W}\right] \otimes \beta\mathbf{I}_N\right)\left(\boldsymbol{\Sigma} \otimes \boldsymbol{\Sigma}\mathbf{W}^T\boldsymbol{\Lambda}_1^2\right) \qquad \text{(B.41)} \\
&= \left[\beta\boldsymbol{\Sigma} \otimes \left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T \boldsymbol{\Lambda}_1\right] \\
&\quad -2\beta\left(\left[\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T \boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\Sigma}\right] \otimes \beta\boldsymbol{\Sigma}\mathbf{W}^T\boldsymbol{\Lambda}_1^2\right) \qquad \text{(B.42)} \\
&= \left[\beta\boldsymbol{\Sigma} \otimes \left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T \boldsymbol{\Lambda}_1\right] - 2\left(\boldsymbol{\mu} \otimes \beta\boldsymbol{\Sigma}\mathbf{W}^T\boldsymbol{\Lambda}_1^2\right). \qquad \text{(B.43)}
\end{aligned}$$

B.2.2 Gradient of the objective function

Using the form of $\log p\left(\mathbf{y}\,|\boldsymbol{\theta}\right)$ given in (B.19), there are three terms which will have non-zero gradient with respect to $\boldsymbol{\theta}$:

$$\begin{aligned}
\mathcal{L}_1 &= -\frac{\beta}{2}\left\|\mathbf{y}-\boldsymbol{\lambda}_2-\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right\|_2^2 \qquad \text{(B.44)} \\
\mathcal{L}_2 &= -\frac{1}{2}\boldsymbol{\mu}^T\mathbf{Z}_x^{-1}\boldsymbol{\mu} \qquad \text{(B.45)} \\
\mathcal{L}_3 &= \frac{1}{2}\log\left|\boldsymbol{\Sigma}\right|. \qquad \text{(B.46)}
\end{aligned}$$

Starting with $\mathcal{L}_1$, we can see that

$$\frac{\partial \mathcal{L}_1}{\partial \mathrm{vec}\left(\mathbf{W}\right)^T} = -\beta\left(\mathbf{y}-\boldsymbol{\lambda}_2-\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)^T \frac{\partial\left(-\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)}{\partial \mathrm{vec}\left(\mathbf{W}\right)^T} \qquad \text{(B.47)}$$

and that the factor $\mathbf{\Lambda}_1\mathbf{W}\boldsymbol{\mu}$ depends on $\mathbf{W}$ both directly and through $\boldsymbol{\mu}$, so that

$$\frac{\partial\left(-\mathbf{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)}{\partial\mathrm{vec}\left(\mathbf{W}\right)^T} = \left(\boldsymbol{\mu}^T \otimes -\mathbf{\Lambda}_1\right) - \mathbf{\Lambda}_1\mathbf{W}\frac{\partial\mathrm{vec}\left(\boldsymbol{\mu}\right)}{\partial\left(\mathbf{W}\right)^T}, \tag{B.48}$$

and therefore

$$\begin{aligned}\frac{\partial\mathcal{L}_1}{\partial\mathrm{vec}\left(\mathbf{W}\right)^T} &= -\beta\left(\mathbf{y}-\boldsymbol{\lambda}_2-\mathbf{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)^T\left[\left(\boldsymbol{\mu}^T\otimes-\mathbf{\Lambda}_1\right)-\mathbf{\Lambda}_1\mathbf{W}\frac{\partial\mathrm{vec}\left(\boldsymbol{\mu}\right)}{\partial\left(\mathbf{W}\right)^T}\right] &\text{(B.49)}\\ &= -\beta\mathrm{vec}\left(\mathbf{\Lambda}_1\left(\mathbf{y}-\boldsymbol{\lambda}_2-\mathbf{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)\boldsymbol{\mu}^T\right)\\ &\quad +\beta\left(\mathbf{y}-\boldsymbol{\lambda}_2-\mathbf{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)^T\mathbf{\Lambda}_1\mathbf{W}\frac{\partial\mathrm{vec}\left(\boldsymbol{\mu}\right)}{\partial\left(\mathbf{W}\right)^T}. &\text{(B.50)}\end{aligned}$$

Next we can see that

$$\frac{\partial\mathcal{L}_2}{\partial\mathrm{vec}\left(\mathbf{W}\right)^T} = -\boldsymbol{\mu}^T\mathbf{Z}_x^{-1}\frac{\partial\mathrm{vec}\left(\boldsymbol{\mu}\right)}{\partial\left(\mathbf{W}\right)^T} \tag{B.51}$$

so

$$\begin{aligned}\frac{\partial\left(\mathcal{L}_1+\mathcal{L}_2\right)}{\partial\mathrm{vec}\left(\mathbf{W}\right)^T} &= -\beta\mathrm{vec}\left(\mathbf{\Lambda}_1\left(\mathbf{y}-\boldsymbol{\lambda}_2-\mathbf{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)\boldsymbol{\mu}^T\right)\\ &\quad +\left[\beta\left(\mathbf{y}-\boldsymbol{\lambda}_2-\mathbf{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)^T\mathbf{\Lambda}_1\mathbf{W}-\boldsymbol{\mu}^T\mathbf{Z}_x^{-1}\right]\frac{\partial\mathrm{vec}\left(\boldsymbol{\mu}\right)}{\partial\left(\mathbf{W}\right)^T}. &\text{(B.52)}\end{aligned}$$

The bracketed expression pre-multiplying the partial derivative can be rearranged

to show that it cancels itself out, leaving only the first term of (B.52):

$$\begin{aligned}
&\beta\left(\mathbf{y}-\boldsymbol{\lambda}_2-\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)^T\boldsymbol{\Lambda}_1\mathbf{W}-\boldsymbol{\mu}^T\mathbf{Z}_x^{-1}\\
&\quad=\quad \beta\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\boldsymbol{\Lambda}_1\mathbf{W}-\beta\boldsymbol{\mu}^T\mathbf{W}^T\boldsymbol{\Lambda}_1^2\mathbf{W}-\boldsymbol{\mu}^T\mathbf{Z}_x^{-1} && \text{(B.53)}\\
&\quad=\quad \beta\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\boldsymbol{\Lambda}_1\mathbf{W}-\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1} && \text{(B.54)}\\
&\quad=\quad \left(\beta\left(\mathbf{y}-\boldsymbol{\lambda}_2\right)^T\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\Sigma}\right)\boldsymbol{\Sigma}^{-1}-\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1} && \text{(B.55)}\\
&\quad=\quad \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}-\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1} && \text{(B.56)}\\
&\quad=\quad \mathbf{0}. && \text{(B.57)}
\end{aligned}$$

Finally, considering $\mathcal{L}_3$, taking the identity given in (A.30), and the partial derivative from (B.36), we get

$$\begin{aligned}
\frac{\partial\mathcal{L}_3}{\partial\mathrm{vec}(\mathbf{W})^T} &= \frac{1}{2}\mathrm{vec}\left[\boldsymbol{\Sigma}^{-T}\right]^T\left(-2\beta\left(\boldsymbol{\Sigma}\otimes\boldsymbol{\Sigma}\mathbf{W}^T\boldsymbol{\Lambda}_1^2\right)\right) && \text{(B.58)}\\
&= \mathrm{vec}\left[-\beta\boldsymbol{\Lambda}_1^2\mathbf{W}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-T}\boldsymbol{\Sigma}\right]^T && \text{(B.59)}\\
&= \mathrm{vec}\left[-\beta\boldsymbol{\Lambda}_1^2\mathbf{W}\boldsymbol{\Sigma}\right]^T. && \text{(B.60)}
\end{aligned}$$

So the gradient of $\mathcal{L}$ with respect to the system matrix $\mathbf{W}$ is

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}} = -\beta\boldsymbol{\Lambda}_1\left(\mathbf{y}-\boldsymbol{\lambda}_2-\boldsymbol{\Lambda}_1\mathbf{W}\boldsymbol{\mu}\right)\boldsymbol{\mu}^T-\beta\lambda_1^2\mathbf{W}\boldsymbol{\Sigma}. \qquad \text{(B.61)}$$

Bibliography

[1] J. Abad, M. Vega, R. Molina, and A. K. Katsaggelos. Parameter estimation in super-resolution image reconstruction problems. In *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP-2003)*, volume III, pages 709–712. Hong Kong, April 2003.

[2] Y. Altunbasak, A. Patti, and R. Mersereau. Super-resolution still and video reconstruction from MPEG-coded video. *IEEE Trans. Circuits And Syst. Video Technol.*, 12:217—226, 2002.

[3] S. Baker and T. Kanade. Super resolution optical flow. Technical Report CMU-RI-TR-99-36, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, October 1999.

[4] S. Baker and T. Kanade. Limits on super-resolution and how to break them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[5] S. Baker and T. Kanade. Super-resolution: Reconstruction or recognition? In *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Baltimore, Maryland, June 2001. IEEE.

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

[7] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, UK*, pages 312–320. Springer-Verlag, 1996.

[8] I. Begin and F. P. Ferrie. Blind super-resolution using a learning-based approach. In *17th International Conference on Pattern Recognition*, volume 2, pages 85–89, 2004.

[9] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[10] C. M. Bishop, A. Blake, and B. Marthi. Super-resolution enhancement of video. In C. M. Bishop and B. Frey, editors, *Proceedings Artificial Intelligence and Statistics*, Key West, Florida, 2003.

[11] S. Borman. *Topics in Multiframe Superresolution Restoration.* PhD thesis, University of Notre Dame, Notre Dame, Indiana, May 2004.

[12] S. Borman and R. L. Stevenson. Spatial resolution enhancement of low-resolution image sequences: A comprehensive review with directions for future research. Technical report, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, USA, July 1998.

[13] S. Borman and R. L. Stevenson. Super-Resolution from Image Sequences - A Review. In *Proceedings of the 1998 Midwest Symposium on Circuits and Systems*, Notre Dame, IN, 1998.

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

[15] S. Borman and R. L. Stevenson. Image resampling and constraint formulation for multi-frame super-resolution restoration. In C. A. Bouman and R. L. Stevenson, editors, *Computational Imaging*, volume 5016 of *Proceedings of the SPIE*, pages 208–219, San Jose, CA, USA, January 2003.

[16] N. K. Bose, S. Lertrattanapanich, and M. B. Chappalli. Superresolution with second generation wavelets. *Signal Processing: Image Communication*, 19(5):387–391, May 2004.

[17] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara*, pages 885–891, June 1998.

[18] D. P. Capel. *Image Mosaicing and Super-resolution.* PhD thesis, University of Oxford, 2001.

[19] D. P. Capel. *Image Mosaicing and Super-resolution (Distinguished Dissertations).* Springer, ISBN: 1852337710, 2004.

[20] D. P. Capel and A. Zisserman. Super-resolution enhancement of text image sequences. In *Proceedings of the International Conference on Pattern Recognition*, pages 600–605, 2000.

[21] D. P. Capel and A. Zisserman. Super-resolution from multiple views using learnt image models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 627–634, 2001.

[22] P. Cheeseman, B. Kanefsky, R. Kraft, and J. Stutz. Super-resolved surface reconstruction from multiple images. Technical report, NASA, 1994.

[23] P. Cheeseman, R. Kanefsky, R. Kraft, J. Stutz, and R. Hanson. Super-resolved surface reconstruction from multiple images. In Glenn R. Heidbreder, editor, *Maximum Entropy and Bayesian Methods*, pages 293–308. Kluwer Academic Publishers, Dordrecht, the Netherlands, 1996.

[24] M. C. Chiang and T. E. Boult. Local blur estimation and super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–832, 1997.

[25] J. Chung, E. Haber, and J. Nagy. Numerical methods for coupled super-resolution. *Inverse Problems*, 22:1261–1272, June 2006.

[26] M. Das Gupta, S. Rajaram, N. Petrovic, and T. S. Huang. Non-parametric image super-resolution using multiple images. In *ICIP*, Genova, Italy, September 2005.

[27] K. Donaldson and D. K. Myers. Bayesian super-resolution of text in video with a text-specific bimodal prior. *International Journal on Document Analysis and Recognition*, 7(2–3):159–167, July 2005.

[28] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 1039–1046, September 1999.

[29] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *Image Processing*, 6(12):1646–1658, December 1997.

[30] M. Elad and A. Feuer. Super-resolution reconstruction of continuous image sequences. In *ICIP*, pages 817–834, Kobe, Japan, October 1999.

[31] M. Elad and A. Feuer. Super-resolution reconstruction of image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):817–834, 1999.

[32] M. Elad and A. Feuer. Super-resolution restoration of continuous image sequence - adaptive filtering approach. *IEEE Tranactions on Image Processing*, 8(3):387–395, March 1999.

[33] M. Elad and Y. Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space invariant blur. *IEEE Transactions on Image Processing*, 10(8):1187–93, August 2001.

[34] P. E. Eren, M.I. Sezan, and A. M. Tekalp. Robust, object-based high- resolution image reconstruction from low-resolution video. *IEEE Transactions on Image Processing*, 6(10):1446–1451, 1997.

[35] S. Farsiu, M. Elad, and P. Milanfar. A practical approach to super-resolution. In *Proc. of the SPIE: Visual Communications and Image Processing*, San-Jose, 2006.

[36] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Robust shift and add approach to super-resolution. *In Proc. of the 2003 SPIE Conf. on Applications of Digital Signal and Image Processing*, pages 121–130, August 2003.

[37] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology, Special Issue on High Resolution Image Reconstruction*, 14(2):47–57, August 2004.

[38] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *Image Processing*, 13(10):1327–1344, October 2004.

[39] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics, SIGGRAPH 2006 Conference Proceedings, Boston, MA*, 25:787–794, 2006.

[40] A. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1176–1183, October 2003.

[41] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. In *Proceedings of the British Machine Vision Conference*, pages 662–670, 2001.

[42] L. Fletcher, L. Petersson, N. Barnes, D. Austin, and A. Zelinsky. A sign reading driver assistance system using eye gaze. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Automation (ICRA2005)*, Barcelona, Spain, April 2005.

[43] LizardTech's Genuine Fractals. http://www.altamira-group.com/, 2000.

[44] R. Fransens, C. Strecha, and L Van Gool. A probabilistic approach to optical flow based super-resolution. In *Proc. Workshop on Generative Model Based Vision*, 2004.

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

[46] W. Freeman and E. Pasztor. Learning low-level vision. In *Proceedings of the International Conference on Computer Vision*, pages 1182–1189, 1999.

[47] W. Freeman and E. Pasztor. Learning to estimate scenes from images. In *Advances in Neural Information Processing Systems*, volume 11, 1999.

[48] W. Freeman and E. Pasztor. Markov networks for low-level vision. Technical Report MERL-TR99-08, MERL, 1999.

[49] W. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. In *International Journal of Computer Vision*, July 2000.

[50] T. F. Gee, T. P. Karnowski, and K. W. Tobin. Multiframe combination and blur deconvolution of video data. In *Proc. SPIE Vol. 3974, p. 788-795, Image and Video Communications and Processing 2000, Bhaskaran Vasudev; T. Russell Hsing; Andrew G. Tescher; Robert L. Stevenson; Eds.*, pages 788–795, April 2000.

[51] M. Gevrekci and B. K. Gunturk. Super resolution under photometric diversity of images. *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 36076, 12 pages, 2007.

[52] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, 3 edition, 1996.

[53] F. Guichard and L. Rudin. Image frame fusion by velocity estimation using region merging. *US Patent 5,909,251*, 1997.

[54] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Multiframe resolution-enhancement methods for compressed video. *IEEE Signal Processing Letters*, 9(6):170–174, June 2002.

[55] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. Hayes III, and R. M. Mersereau. Eigenface-domain super-resolution for face recognition. *IEEE Transactions on Image Processing, volume 12 (2003), number 5*, 12(5):597–606, 2003.

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

[57] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[58] M. Irani and S. Peleg. Super resolution from image sequences. *Proceedings of the International Conference on Pattern Recognition*, 2:115–120, June 1990.

[59] M. Irani and S. Peleg. Improving resolution by image registration. *Graphical Models and Image Processing*, 53:231–239, 1991.

[60] M. Irani and S. Peleg. Motion analysis for image enhancement:resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4:324–335, 1993.

[61] Z. Jiang, T.T. Wong, and H. Bao. Practical super-resolution from dynamic video sequences. In *CVPR03*, pages II: 549–554, 2003.

[62] C. V. Jiji and S. Chaudhuri. Single-frame image super-resolution through contourlet learning. *EURASIP J. Appl. Signal Process.*, 2006(1):235–235.

[63] C. V. Jiji and S. Chaudhuri. Single-frame image super-resolution using learned wavelet coefficients. *International Journal of Imaging Systems and Technology*, 14(3):105=–112, 2004.

[64] M. V. Joshi, S. Chaudhuri, and R. Panuganti. A learning-based method for image super-resolution from zoomed observations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(3):527–537, 2005.

[65] M.V. Joshi, S. Chaudhuri, and R. Panuganti. Super-resolution imaging: use of zoom as a cue. *Image and Vision Computing*, 22(14):1185–1196, December 2004.

[66] D. Keren, S. Peleg, and R. Brada. Image sequence enhancement using subpixel displacements. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–746, Ann Arbor, MI, June 1988.

[67] S. P. Kim, H. K. Bose, and H. M. Valenzuela. Recursive reconstruction of high-resolution image from noisy undersampled frames. *IEEE Trans. Acoust., Speech, Signal Processing*, 38:1013–1027, June 1990.

[68] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–46, May 1996.

[69] R. L. Lagendijk, A. M. Tekalp, and J. Biemond. Maximum likelihood image and blur identification: A unifying approach. *Optical Engineering*, 29:422–435, 1990.

[70] Z. Lin and H. Y. Shum. Fundamental limits of reconstruction-based super-resolution algorithms under local translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):83–97, 2004.

[71] H. Lutkepohl. *Handbook of Matrices*. Wiley, UK, 1996.

[72] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, UK, revised edition, 1999.

[73] J. Miskin and D. MacKay. *Ensemble learning for blind image separation and deconvolution*, chapter 7. Springer-Verlag Scientific Publishers, 2000.

[74] R. Molina, J. Mateos, and A. K. Katsaggelos. Blind deconvolution using a variational approach to parameter, image, and blur estimation. *IEEE Trans. on Image Processing*, 15(12):3715–3727, December 2006.

[75] R. Molina, M. Vega, J. Abad, and A. K. Katsaggelos. Parameter estimation in bayesian high-resolution image reconstruction with multisensors. *IEEE Transactions on Image Processing*, 12:1655–1667, December 2003.

[76] U. Mudenagudi, R. Singla, P. Kalra, and S. Banerjee. Super-resolution using graph cut. In P. J. Narayanan, Shree K. Nayar, and Heung-Yeung Shum, editors, *7th Asian Conference on Computer Vision*, pages 385–394, January 2006.

[77] I. Nabney. *Netlab algorithms for pattern recognition*. Springer, 2002.

[78] M. K. Ng, H. Shen, E. Y. Lam, and L. Zhang. A total variation regularization based super-resolution reconstruction algorithm for digital video. *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 74585, 16 pages, 2007.

[79] N. Nguyen. *Numerical Algorithms for Image Superresolution*. PhD thesis, Stanford University, July 2000.

[80] N. Nguyen, P. Milanfar, and G. Golub. Blind superresolution with generalized cross-validation using gauss-type quadrature rules. In *Proceedings of the 33rd Asilomar Conference on Signals, Systems, and Computers*, October 1999.

[81] N. Nguyen, P. Milanfar, and G. Golub. A computationally efficient super-resolution image reconstruction algorithm. Technical Report SCCM-99-04, Stanford University, 1999.

[82] N. Nguyen, P. Milanfar, and G. Golub. A computationally efficient superresolution image reconstruction algorithm. *IEEE Transactions on Image Processing*, 10(4):573–583, April 2001.

[83] N. Nguyen, P. Milanfar, and G. Golub. Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement. *IEEE Transactions on Image Processing*, 10(9):1299–1308, September 2001.

[84] A. J. Patti and Y Altunbasak. Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants. *IEEE Transactions on Image Processing*, 10(1):179–186, 2001.

[85] A. J. Patti, M. I. Sezan, and A. M. Tekalp. Robust methods for high quality stills from interlaced video in the presence of dominant motion. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2):328–342, April 1997.

[86] A. J. Patti, M. I. Sezan, and A. M. Tekalp. Super resolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Transactions on Image Processing*, pages 1064–1078, August 1997.

[87] S. Peleg, D. Keren, and L. Schweitzer. Improving image resolution using subpixel motion. *Pattern Recognition Letters*, 5(3):223–226, 1987.

[88] S. Pelletier, S. P. Spackman, and J. R. Cooperstock. High-resolution video synthesis from mixed-resolution video based on the estimate-and-correct method. In *WACV-MOTION '05: Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Volume 1*, pages 172–177, Washington, DC, USA, 2005. IEEE Computer Society.

[89] L. C. Pickup, D. P. Capel, S. J. Roberts, and A. Zisserman. Bayesian image super-resolution, continued. In *Advances in Neural Information Processing Systems 19*, pages 1089–1096. MIT Press, 2007.

[90] L. C. Pickup, D. P. Capel, S. J. Roberts, and A. Zisserman. Bayesian methods for image super-resolution. *The Computer Journal*, page bxm091, 2007.

[91] L. C. Pickup, D. P. Capel, S. J. Roberts, and A. Zisserman. Overcoming registration uncertainty in image super-resolution: Maximize or marginalize? *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 23565, 14 pages, 2007.

[92] L. C. Pickup, S. J. Roberts, and A. Zisserman. A sampled texture prior for image super-resolution. In *Advances in Neural Information Processing Systems*, pages 1587–1594, 2003.

[93] L. C. Pickup, S. J. Roberts, and A. Zisserman. Optimizing and learning for super-resolution. In *Proceedings of the British Machine Vision Conference*, 2006.

[94] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C (2nd Ed.)*. Cambridge University Press, 1992.

[95] A. N. Rajagopalan and S. Chaudhuri. Space-variant approaches to recovery of depth from defocused images. *Computer Vision and Image Understanding*, 68:309–329, 1997.

[96] A. N. Rajagopalan and S. Chaudhuri. A variational approach to recovering depth from defocused images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1158–1164, 1997.

[97] S. Rajaram, M. Das Gupta, N. Petrovic, and T. S. Huang. Learning-based non-parametric image superresolution. *EURASIP Journal on Applied Signal Processing*, 2006(6), 2006.

[98] A. Rav-Acha and S. Peleg. Two motion blurred images are better than one. *Pattern Recognition Letters*, 26:311–317, 2005.

[99] S. J. Reeves and R. M. Mersereau. Blur identification by the method of generalized cross-validation. *IEEE Transactions on Image Processing*, 1(3):301–311, 1992.

[100] L. Rudin and F. Guichard. Velocity estimation from images sequence and application to super-resolution. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 527–531, 1999.

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

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

[103] E. Shechtman, Y. Caspi, and M. Irani. Increasing space-time resolution in video. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, pages 753–768. Springer-Verlag, 2002.

[104] E. Shechtman, Y. Caspi, and M. Irani. Space-time super-resolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):531–545, 2005.

[105] H. Stark and P. Oskoui. High resolution image recovery from image-plane arrays, using convex projections. *J. Opt. Soc. Am. A*, 6(11):1715–1726, 1989.

[106] A. Storkey. Dynamic structure super-resolution. In *Advances in Neural Information Processing Systems 16*, 2003.

[107] A. Storkey and C. Williams. Dynamic positional trees for structural image analysis. In T. Jaakkola and T. Richardson, editors, *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, pages 298–304. Morgan Kaufmann, 2001.

[108] J. Sun, N. Zhang, H. Tao, and H. Shum. Image hallucination with primal sketch priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 729–736, Madison, USA, 2003.

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

[110] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan. High resolution image reconstruction from lower-resolution image sequences and spave-varying image restoration. In *Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing*, volume III, pages 169–172, San Francisco, CA, 1992.

[111] A. Temizel and T. Vlachos. Wavelet domain image resolution enhancement. *Vision, Image and Signal Processing*, 153(1):25–30, February 2006.

[112] M. E. Tipping and C. M. Bishop. Bayesian imge super-resolution. In S. Thrun, S. Becker, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 1279–1286, Cambridge, MA, 2003. MIT Press.

[113] B. C. Tom and A. K. Katsaggelos. Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 2539–2542, 1995.

[114] B. C. Tom and A. K. Katsaggelos. Resolution enhancement of video sequences using motion compensation. In *Proceedings of the IEEE International Conference on Image Processing*, volume I, pages 713–716, Lausanne, Switzerland, September 1996.

[115] B. C. Tom, A. K. Katsaggelos, and N. P. Galatsanos. Reconstruction of a high resolution image from registration and restoration of low resolution images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 553–557, 1994.

[116] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment: A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[117] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:13–27, 1984.

[118] H. Ur and D. Gross. Improved resolution from sub-pixel shifted pictures. *Graphical Models and Image Processing*, 52:181–186, March 1992.

[119] M. Vega, R. Molina, and A. K. Katsaggelos. A bayesian superresolution approach to demosaicing of blurred images. *EURASIP Journal on Applied Signal Processing*, 2006:Article ID 25072, 12 pages, 2006.

[120] Q. Wang, X. Tang, and H. Shum. Patch based blind image super resolution. In *Proceedings of the 10th International Conference on Computer Vision, Beijing, China*, volume 1, pages 709–716, 2005.

[121] N. A. Woods, N. P. Galatsanos, and A. K. Katsaggelos. Em-based simultaneous registration, restoration, and interpolation of super-resolved images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, September 2003.

[122] W. Zhao and H. S. Sawhney. Is super-resolution with optical flow feasible? In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, pages 599–613, London, UK, 2002. Springer-Verlag.

[123] A. Zomet, A. Rav-Acha, and S. Peleg. Robust super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii*, pages 645–650, December 2001.