

---

## *Foreward*

I am delighted to introduce the first book on Multimedia Data Mining. When I came to know about this book project undertaken by two of the most active young researchers in the field, I was pleased that this book is coming in early stage of a field that will need it more than most fields do. In most emerging research fields, a book can play a significant role in bringing some maturity to the field. Research fields advance through research papers. In research papers, however, only a limited perspective could be provided about the field, its application potential, and the techniques required and already developed in the field. A book gives such a chance. I liked the idea that there will be a book that will try to unify the field by bringing in disparate topics already available in several papers that are not easy to find and understand. I was supportive of this book project even before I had seen any material on it. The project was a brilliant and a bold idea by two active researchers. Now that I have it on my screen, it appears to be even a better idea.

Multimedia started gaining recognition in 1990s as a field. Processing, storage, communication, and capture and display technologies had advanced enough that researchers and technologists started building approaches to combine information in multiple types of signals such as audio, images, video, and text. Multimedia computing and communication techniques recognize correlated information in multiple sources as well as insufficiency of information in any individual source. By properly selecting sources to provide complementary information, such systems aspire, much like human perception system, to create a holistic picture of a situation using only partial information from separate sources.

Data mining is a direct outgrowth of progress in data storage and processing speeds. When it became possible to store large volume of data and run different statistical computations to explore all possible and even unlikely correlations among data, the field of data mining was born. Data mining allowed people to hypothesize relationships among data entities and explore support for those. This field has been put to applications in many diverse domains and keeps getting more applications. In fact many new fields are direct outgrowth of data mining and it is likely to become a powerful computational tool.





## Part I

# This is a Part





# 1

## *Multi-frame super-resolution from a Bayesian perspective*

Lyndsey Pickup, Stephen Roberts, Andrew Zisserman

*University of Oxford*

David Capel

*2d3 Inc.*

### CONTENTS

1.1	The generative model .....	4
1.1.1	Considerations in the forward model .....	5
1.1.2	A probabilistic setting .....	6
1.1.2.1	The <i>Maximum Likelihood</i> solution .....	7
1.1.2.2	The ML solution in practice .....	8
1.1.2.3	The <i>Maximum a Posteriori</i> solution .....	9
1.1.3	Selected priors used in MAP super-resolution .....	11
1.2	Where super-resolution algorithms go wrong .....	14
1.2.1	Point-spread function example .....	15
1.2.2	Photometric registration example .....	16
1.2.3	Geometric registration example .....	18
1.3	Simultaneous super-resolution .....	20
1.3.1	Super-resolution with registration .....	20
1.3.2	Learning prior strength parameters from data .....	22
1.3.3	Scaling and convergence .....	23
1.3.4	Initialization .....	24
1.3.5	Evaluation on synthetic data .....	25
1.3.6	Tests on real data .....	26
1.4	Bayesian marginalization .....	29
1.4.1	Marginalizing over registration parameters .....	30
1.4.2	Marginalizing over the super-resolution image .....	32
1.4.3	Implementation notes .....	34
1.4.4	Experimental Evaluation .....	34
1.4.5	Discussion .....	37
1.5	Concluding remarks .....	37

This chapter examines multi-frame image super-resolution in a probabilistic framework. Many multi-frame super-resolution algorithms begin by a point estimate of the unknown *latent* parameters like those describing the camera/scene motion model or the camera optics. The focus of this chapter is on alternatives to this practice that can yield superior super-resolution results.

We begin with the generative model and simple *Maximum Likelihood* (ML)

and *Maximum a Priori* (MAP) solutions for the super-resolution image. In the second section, the simple MAP algorithm, some consequences of inaccurate point estimates of some of the latent parameters are illustrated.

The third section introduces the simultaneous Maximum a Posteriori algorithm, which estimates the super-resolution image along with parameters such as the image registration, allowing the high-resolution information to influence and improve the estimates of the latent parameters. In the fourth section, we show how Bayesian marginalization can integrate the latent parameters out of the problem, leading to a cost function in terms of the low-resolution images which can be optimized with respect to the high-resolution pixels directly. We conclude with a brief discussion of the benefits of these two Bayesian approaches to super-resolution.

---

## 1.1 The generative model

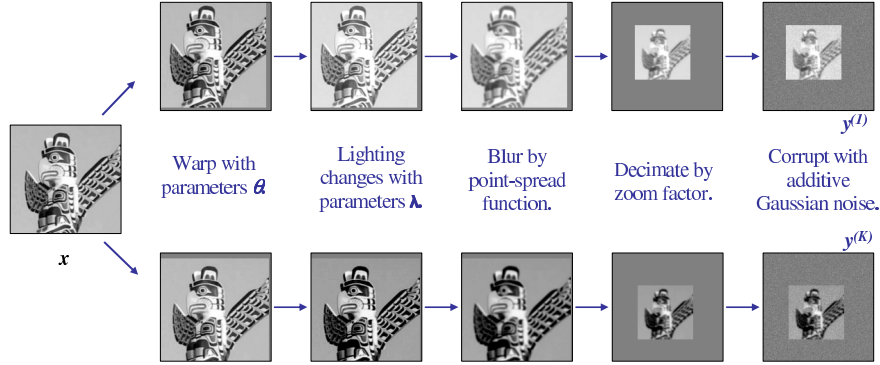
A generative model is a parameterized, probabilistic model of data generation, which attempts to capture the forward process by which observed data (in this case low resolution images) is generated by an underlying system (the scene and imaging parameters), and corrupted by various noise processes. This 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  $k^{\text{th}}$  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  $\mathbf{W}^{(k)}$  [3, 13], and a global affine photometric correction results from multiplication and addition across all pixels by scalars  $\lambda_1^{(k)}$  and  $\lambda_2^{(k)}$  respectively [3]. Thus the generative model for one of the low-resolution images is

$$\mathbf{y}^{(k)} = \lambda_1^{(k)} \mathbf{W}^{(k)} \mathbf{x} + \lambda_2^{(k)} \mathbf{1} + \mathcal{N}(\mathbf{0}, \beta^{-1} \mathbf{I}), \quad (1.1)$$

where  $\mathbf{1}$  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  $\beta$ , or alternatively with standard deviation  $\sigma_N$ , where  $\beta^{-1} = \sigma_N^2$ .



**FIGURE 1.1**  
The generative model for two typical low-resolution images.

Figure 1.1 shows the generative model for two typical greyscale low-resolution images in terms of the images at each step in the procedure. On the left is the single ground truth scene, and on the extreme right are the two images (in this case  $y^{(1)}$  and  $y^{(K)}$ , which might represent the first and last images in a  $K$ -image sequence) as they are observed by the camera sensors. Given a set of low resolution images like this,  $\{y^{(k)}\}$ , the goal is to recover  $x$ , without knowing the values associated with  $\{\mathbf{W}^{(k)}, \lambda^{(k)}, \sigma_N\}$ .

### 1.1.1 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.

**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 [6, 11].

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. Though it is equally possible to create  $\mathbf{W}$  matrices from more complex motion models such as optic

flow, the methods described in this chapter are based upon planar projective homographies.

**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 [1].

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  $\sigma_{PSF}^2 \mathbf{I}$ , or a circular disk (top-hat function) with radius  $r_{PSF}$ .

**The noise model:** The image noise is assumed to be *i.i.d.* Gaussian, which leads to an  $L_2$  data error measure. Several common types of image noise contain more structure than this model would predict, *e.g.* noise levels which depend on observed pixel intensities, or errors introduced by quantization artifacts in the JPEG or MPEG compression process. However, it can be shown that even for nonlinear noise such as JPEG quantization, the Gaussianity assumption performs well [12], and has the benefit that the problem remains convex. Some authors assume other families of noise, *e.g.* Laplacian or salt-and-pepper noise, for which the  $L_1$  norm is more appropriate [5]. It is worth noting that most of the algorithms presented in this chapter the  $L_1$  norm can be substituted into the derived objective functions if appropriate, though gradient expressions must then be adjusted accordingly.

**Constructing  $\mathbf{W}^{(k)}$ :** 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  $i^{\text{th}}$  row in  $\mathbf{W}^{(k)}$  represents the kernel for the  $i^{\text{th}}$  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.

### 1.1.2 A probabilistic setting

From the generative model given in (1.1) 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

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

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  $k^{\text{th}}$  residual,  $\mathbf{r}^{(k)}$ , to be

$$\mathbf{r}^{(k)} = \mathbf{y}^{(k)} - \lambda_1^{(k)} \mathbf{W}^{(k)} \mathbf{x} - \boldsymbol{\lambda}_2^{(k)}. \quad (1.2)$$

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

$$p\left(\{\mathbf{y}^{(k)}\} | \mathbf{x}, \{\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}\}\right) = \left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}} \exp\left\{-\frac{\beta}{2} \sum_{k=1}^K \|\mathbf{r}^{(k)}\|_2^2\right\}. \quad (1.3)$$

### 1.1.2.1 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,

$$\hat{\mathbf{x}}_{\text{ML}} = \underset{\mathbf{x}}{\operatorname{argmax}} \left( p\left(\{\mathbf{y}^{(k)}\} | \mathbf{x}, \{\boldsymbol{\theta}^{(k)} \boldsymbol{\lambda}^{(k)}\}\right) \right). \quad (1.4)$$

If all other parameters are known,  $\hat{\mathbf{x}}_{\text{ML}}$  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

$$\hat{\mathbf{x}}_{\text{ML}} = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T \mathbf{y}, \quad (1.5)$$

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

$$\hat{\mathbf{x}}_{\text{ML}} = \left( \sum_{k=1}^K \lambda_1^{(k)2} \mathbf{W}^{(k)T} \mathbf{W}^{(k)} \right)^{-1} \left[ \sum_{k=1}^K \lambda_1^{(k)} \mathbf{W}^{(k)T} \left( \mathbf{y}^{(k)} - \boldsymbol{\lambda}_2^{(k)} \right) \right] \quad (1.6)$$

Thus we can solve for  $\hat{\mathbf{x}}_{\text{ML}}$  directly if we know  $\{\mathbf{W}^{(k)}, \boldsymbol{\lambda}^{(k)}\}$  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) [10]. Such schemes take an objective function  $\mathcal{L}$ , and its derivative with respect to the current estimate of the high-resolution image,  $\frac{\partial \mathcal{L}}{\partial \mathbf{x}}$ , and find the optimal  $\mathbf{x}$  using an iterative scheme. For the ML solution,

**FIGURE 1.2**

**The synthetic graffiti dataset.** Left: ground truth graffiti wall image. Right: four of the 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 2.

the expressions of interest are therefore

$$\mathcal{L} = \frac{1}{2} \sum_{k=1}^K \left\| \mathbf{y}^{(k)} - \lambda_1^{(k)} \mathbf{W}^{(k)} \mathbf{x} - \lambda_2^{(k)} \right\|_2^2 \quad (1.7)$$

$$= \frac{1}{2} \sum_{k=1}^K \left\| \mathbf{r}^{(k)} \right\|_2^2 \quad (1.8)$$

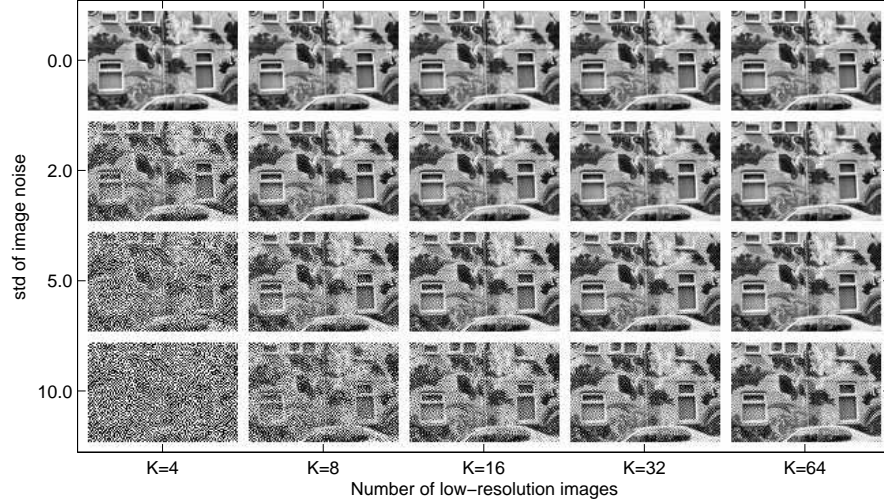
$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \sum_{k=1}^K -\lambda_1^{(k)} \mathbf{W}^{(k)T} \mathbf{r}^{(k)}. \quad (1.9)$$

When this is initialised with a reasonable estimate of the super-resolution image, this scheme can be used to improve the super-resolution estimate 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 one 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 required to obtain a good image. Using SCG, small super-resolution images (under  $200 \times 200$  pixels) tend to require fewer than 50 iterations before the super-resolution image intensity values change by less than a grey level per iteration.

### 1.1.2.2 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 *Graffiti* datasets is introduced in Figure 1.2, where the number

**FIGURE 1.3**

**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.

of images and the amplitude of the additive Gaussian noise is varied, and the registration parameters are determined randomly under a planar projective motion model. The noise amplitude is measured here in *grey levels*, with one grey level being one 255<sup>th</sup> 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 1.3. Here pixels are represented by values in the range  $[-\frac{1}{2}, \frac{1}{2}]$ , but the ML noise pattern has values many orders of magnitude higher than this (though rounded to  $\pm\frac{1}{2}$  for viewing purposes). There are four input images in the datasets in the left column, going up in powers of two to 64 images for each output in the right column. The standard deviation of the noise goes zero for the top row to 2, 5 and 10 grey levels proceeding down the figure. The more images there are in the dataset, the less the oscillations dominate the output, though even with 64 input images, visible degradation is still evident in the output for cases with 5 and 10 grey levels of noise.

#### 1.1.2.3 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. The *Maximum a Posteriori* (MAP) approach is explained here in terms of the generative model and its probabilistic interpretation. We go on to cover a few of the general image priors commonly selected for image super-resolution.

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

$$p(x|d) = \frac{p(d|x)p(x)}{p(d)}. \quad (1.10)$$

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

$$p(d) = \int p(d|x)p(x)dx. \quad (1.11)$$

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

$$p(\mathbf{x} | \{\mathbf{y}^{(k)}, \boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}\}) = \frac{p(\{\mathbf{y}^{(k)}\} | \mathbf{x}, \{\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}\}) p(\mathbf{x})}{p(\{\mathbf{y}^{(k)}\} | \{\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}\})} \quad (1.12)$$

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,  $\mathbf{x}_{\text{MAP}}$ , can be found by maximizing the numerator with respect to  $\mathbf{x}$ , giving

$$\hat{\mathbf{x}}_{\text{MAP}} = \underset{\mathbf{x}}{\operatorname{argmax}} p(\{\mathbf{y}^{(k)}\} | \mathbf{x}, \{\boldsymbol{\theta}^{(k)}\}) p(\mathbf{x}). \quad (1.13)$$

We take the objective function  $\mathcal{L}$  to be the negative log of the numerator of (1.12), and minimize  $\mathcal{L}$  with respect to  $\mathbf{x}$ . The objective function and its gradient are

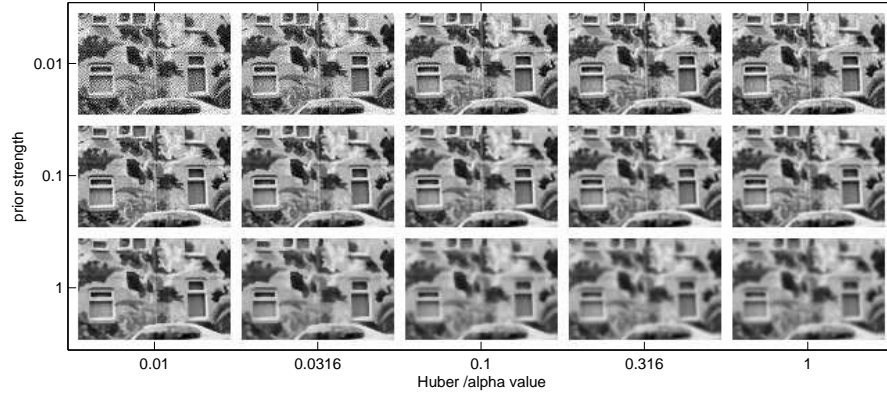
$$\mathcal{L} = -\log(p(\mathbf{x})) + \frac{\beta}{2} \sum_{k=1}^K \|\mathbf{r}^{(k)}\|_2^2 \quad (1.14)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} [-\log(p(\mathbf{x}))] - \sum_{k=1}^K \lambda_1^{(k)} \mathbf{W}^{(k)T} \mathbf{r}^{(k)}. \quad (1.15)$$

In order to solve this, one still requires a form for the image prior  $p(\mathbf{x})$ .

In general the prior should favour smoother solutions than the ML approach typically yields, so it is usual 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 [10] can be used along with (1.14) and (1.15) 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 1.4 shows the improvement in super-resolution image estimates that can be achieved using a very simple prior on the super-resolution image,  $\mathbf{x}$ .

**FIGURE 1.4**

***Maximum a Posteriori* super-resolution images.** This figure uses the same input data as 1.3, but uses the *Maximum a Posteriori* method to infer the super-resolution images using an array of prior parameter settings. With only 9 images and 5 grey levels of noise, the corresponding ML output would be swamped with noise. However, in all but the top left case (weakest prior), the MAP images clearly show the details of the underlying scene.

The super-resolution images were reconstructed using exactly the same input datasets as Figure 1.3, with 9 images and 5 grey levels of noise, but this time a Huber prior was used on image gradients, and all of the noise present in the ML solutions is gone. A few simple forms of prior will be considered next.

### 1.1.3 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.

#### GMRF image priors

Gaussian Markov Random Field (GMRF) priors arise from a formulation where the gradient of the super-resolution solution is penalized, and corre-

spond to specifying a Gaussian distribution over  $\mathbf{x}$ :

$$p(\mathbf{x}) = (2\pi)^{-\frac{N}{2}} |\mathbf{Z}_x|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \mathbf{x}^T \mathbf{Z}_x^{-1} \mathbf{x} \right\}, \quad (1.16)$$

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

$$p(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \mathbf{Z}_x). \quad (1.17)$$

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

$$\mathcal{L} = \beta \|\mathbf{r}\|^2 + \mathbf{x}^T \mathbf{Z}_x^{-1} \mathbf{x} \quad (1.18)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = -2\beta \lambda_1 \mathbf{W}^T \mathbf{r} + 2\mathbf{Z}_x^{-1} \mathbf{x}, \quad (1.19)$$

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:

$$\mathbf{x}_{\text{GMRF}} = \beta \mathbf{\Sigma} \left( \sum_{k=1}^K \lambda_1^{(k)} \mathbf{W}^{(k)T} (\mathbf{y}^{(k)} - \boldsymbol{\lambda}_2^{(k)}) \right) \quad (1.20)$$

$$\mathbf{\Sigma} = \left[ \mathbf{Z}_x^{-1} + \beta \left( \sum_{k=1}^K \lambda_1^{(k)2} \mathbf{W}^{(k)T} \mathbf{W}^{(k)} \right) \right]^{-1}, \quad (1.21)$$

where  $\mathbf{\Sigma}$  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. They are often expressed as the square of some linear operation on the pixels of  $\mathbf{x}$ , such as an approximation to the image gradient or image Laplacian. This gives a general form

$$p(\mathbf{x}) = \frac{1}{Z} \exp \left\{ -\gamma \|\mathbf{D}\mathbf{x}\|_2^2 \right\}, \quad (1.22)$$

so that

$$\mathbf{Z}_x^{-1} = \frac{\gamma}{2} \mathbf{D}^T \mathbf{D}. \quad (1.23)$$

In [3],  $\mathbf{D}$  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, giving a  $4N \times N$  sparse  $\mathbf{D}$  matrix with two non-zero elements per row. In [6],  $\mathbf{D}$  is 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  $i^{\text{th}}$  row has an entry of 1 at position  $i$ , and four entries of  $-\frac{1}{4}$  corresponding to the four cardinal neighbours of pixel  $i$ . In neither of these cases do  $\mathbf{Z}$  and  $\mathbf{Z}^{-1}$  need to be computed explicitly for the iterative solution to be found.

In their *Bayesian Image Super-resolution* work [13], 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

$$Z_x(i, j) = A \exp \left\{ -\frac{\|\mathbf{v}_i - \mathbf{v}_j\|^2}{r^2} \right\}, \quad (1.24)$$

where  $\mathbf{v}_i$  is the two-dimensional position of the pixel that is lexicographically  $i^{\text{th}}$  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.

### Image priors with heavier tails

The *Bilinear Total Variation* (BTV) prior is used by Farsiu *et al.* [4]. 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  $L_2$ .

### 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

$$p(\mathbf{x}) = \frac{1}{Z} \exp \left\{ -\nu \sum_{g \in \mathcal{D}(\mathbf{x})} \rho(g, \alpha) \right\}, \quad (1.25)$$

where  $\mathcal{D}$  is the same set of gradient estimates as (1.22), given by  $\mathbf{D}\mathbf{x}$ . The parameter  $\nu$  is a prior strength somewhat similar to a variance term,  $Z$  is the normalization constant, and  $\alpha$  is a parameter of the Huber function specifying the gradient value at which the penalty switches from being quadratic to being linear:

$$\rho(x, \alpha) = \begin{cases} x^2, & \text{if } |x| \leq \alpha \\ 2\alpha|x| - \alpha^2, & \text{otherwise.} \end{cases} \quad (1.26)$$

For a very simple 1D case, the probability function corresponding to the

Huber function is

$$p(x) = \frac{1}{Z} \exp \{-\nu \rho(x, \alpha)\}, \quad (1.27)$$

and one can verify by integration that

$$Z = \frac{1}{\nu \alpha} \exp \{-2\nu \alpha^2\} + \left(\frac{\pi}{\nu}\right)^{\frac{1}{2}} \operatorname{erf}\{\alpha \nu\}. \quad (1.28)$$

Unfortunately the prior over a whole image is much harder to normalize, as the structure 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  $\alpha$  and  $\nu$ , or writing down an explicit form of  $p(\alpha, \nu | \mathbf{x})$ . However, there are still methods available that allow values for  $\alpha$  and  $\nu$  to be learnt, which we will return to in section 1.3.

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

$$\mathcal{L} = \sum_{k=1}^K \beta \|\mathbf{r}^{(k)}\|_2^2 + \nu \sum_{g \in \mathcal{D}\mathbf{x}} \rho(g, \alpha) \quad (1.29)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = -2\beta \sum_{k=1}^K \lambda_1^{(k)} \mathbf{W}^{(k)T} \mathbf{r}^{(k)} + \nu \mathbf{D}^T \rho'(\mathbf{D}\mathbf{x}, \alpha) \quad (1.30)$$

where

$$\rho'(x) = \begin{cases} 2x, & \text{if } |x| \leq \alpha \\ 2\alpha \operatorname{sign}(x), & \text{otherwise.} \end{cases} \quad (1.31)$$

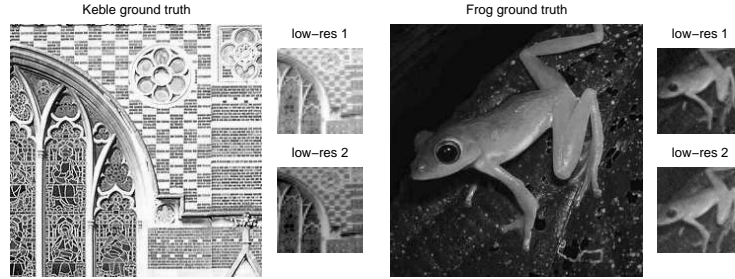
and  $\mathbf{D}$  is again a  $N \times 4N$  matrix giving first order approximations of the gradients in four directions in the image space. This has the advantage over the TV prior that unless  $\alpha \rightarrow 0$ , the function and its gradient with respect to  $\mathbf{x}$  are continuous as well as convex, and can be solved easily using gradient-descent methods like SCG.

---

## 1.2 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. In particular, the super-resolution problem involves



**FIGURE 1.5**

**Keble and Frog datasets.** Ground truth images for the synthetic Keble and Frog datasets, along with two low-resolution images of each (not to scale).

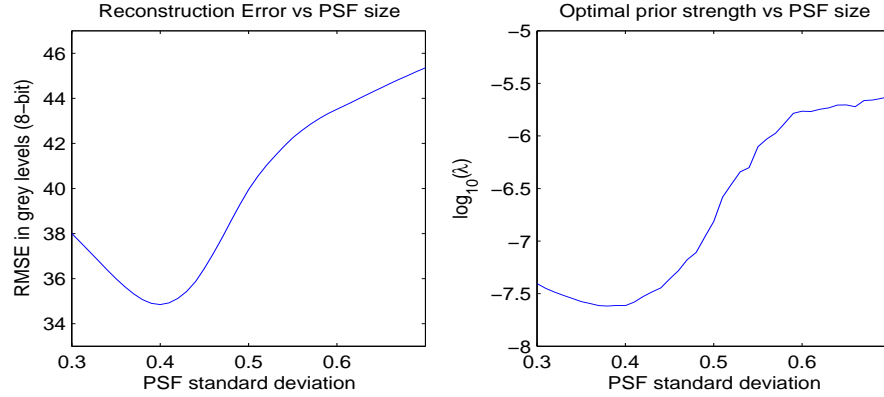
several closely-interrelated components: geometric registration, photometric registration, parameters for the prior, noise estimates and the point-spread function, not to mention the estimates of the values of the high-resolution image pixels themselves. If one component is estimated badly, there can 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.

### 1.2.1 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 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, 16 images from the synthetic “Keble” dataset (see Figure 1.5) with a noise standard deviation of 2.5 grey levels are super-resolve at a zoom factor of 4 using the known geometric (8Dof) and photometric (2DoF) registration values, and a Huber prior with  $\alpha = 0.05$ . The PSF standard deviation,  $\gamma$ , is varied, and for each value the prior strength ratio ( $\nu/\beta$  in (1.29)) which minimises the RMS error with respect to the ground-truth image is found.

The results are plotted in Figure 1.6, which clearly shows that as  $\gamma$  moves away 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 1.6**

Reconstructing the Keble dataset with various PSF estimates: errors. When  $\gamma$  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.

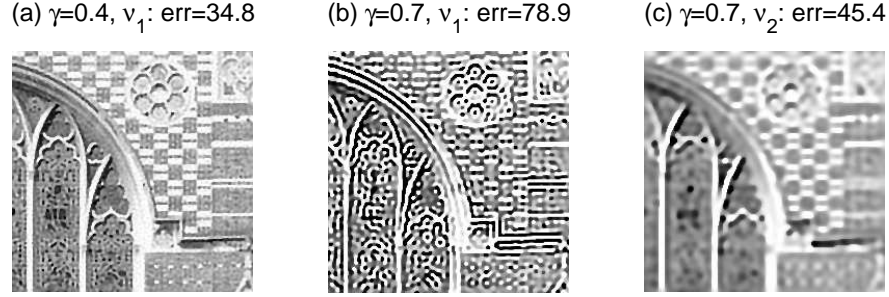
Figure 1.7 shows three images from the results. The first is the image reconstructed with the true  $\gamma$ , showing a very good super-resolution result. The next is the image reconstructed with  $\gamma = 0.7$ , and is a very poor super-resolution result. The final image of the three shows the super-resolution image reconstructed with the same poor value of  $\gamma$ , 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  $\gamma$  and  $\nu$  were varied. The consequence of this kind of relationship is that even when it is possible to make a *reasonably* accurate estimate of each of the hyperparameters needed for super-resolution, the values themselves must be selected together in order to guarantee a good-looking super-resolution image.

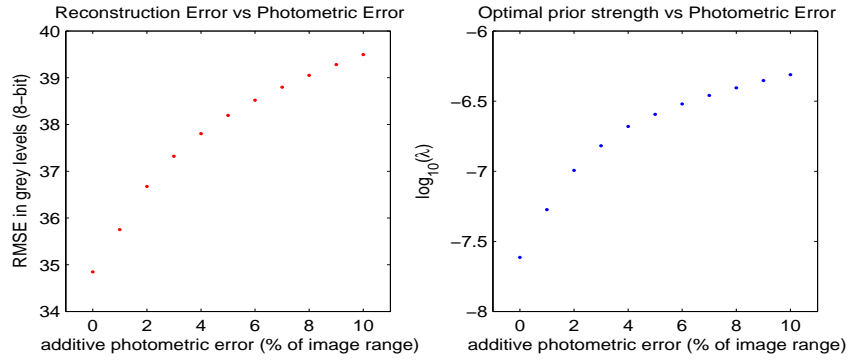
### 1.2.2 Photometric registration example

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, some errors may be expected 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, several

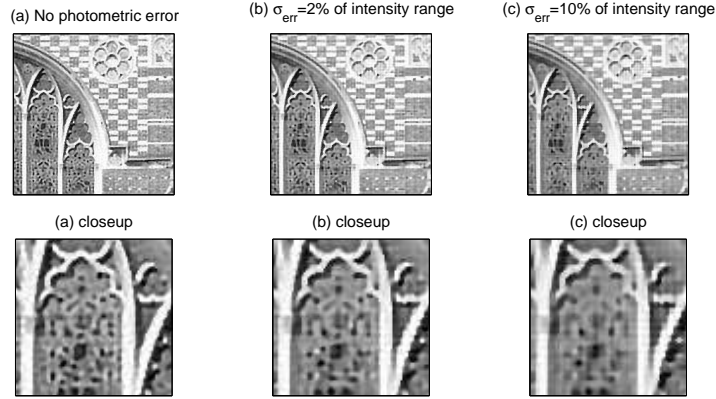
**FIGURE 1.7**

**Reconstructing the Keble dataset with various PSF estimates: images.** (a) The best reconstruction achieved at the correct value,  $\gamma = 0.4$ . (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 almost 100 times stronger than for the first image, even though the input images themselves are identical.

**FIGURE 1.8**

**Reconstructing the Keble dataset with photometric error: errors.** Left: RMSE of the reconstruction increases with the uncertainty in the photometric shift parameter (0 to 10%). Right: the prior strength setting necessary to achieve the best reconstruction for each setting of the photometric parameters. The prior strength increases by well over an order of magnitude between the no-error case and the 10% case.

super-resolution reconstructions of the Keble dataset are made 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

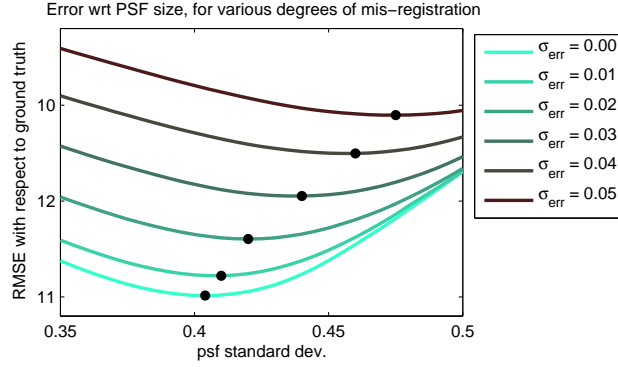
**FIGURE 1.9****Reconstructing the Keble dataset with photometric error: images.**

Top: full super-resolution image; Bottom: close-up of the main window where the different levels of detail are very noticeable. Left-to-right: reconstruction with additive errors of 0%, 2% and 10% on the photometric shift parameters. As the error increases, the edges still remain well-localized, but finer details are smoothed out due to the necessary increase in prior strength.

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 1.8 show the lowest error (left) and prior strength ratio,  $\log_{10}(\nu/\beta)$  (right) for each case. Figure 1.9 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 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.

**FIGURE 1.10**

**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  $\gamma$  was varied about its ground truth value of 0.4. The larger the registration error, the bigger the error in  $\gamma$  is in order to optimize the result.

### 1.2.3 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  $\theta^{(k)}$  can to a small extent be mitigated by a small increase in the size of the blur kernel.

Sixteen images from the synthetic Frog dataset of Figure 1.5 with a zoom factor of 4, a PSF width of 0.4 low-resolution pixels, and various levels of *i.i.d.* Gaussian noise are taken as a starting point. 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  $\theta$  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, *i.e.* a very small amount. For each of these six registrations of the input data, a set of super-resolution images is recovered as the PSF standard deviation,  $\gamma$  is varied, and for each setting of  $\gamma$ , the prior strength ratio giving the best reconstruction is found.

Figure 1.10 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 exceedingly accurately, the minimum falls at the true value of  $\gamma$ . 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

1. **Initialize PSF, image registrations, super-resolution image and prior parameters.**
2. (a) **(Re)-sample the set of validation pixels**, selecting them from across all low-resolution images.  
 (b) **Update  $\alpha$  and  $\nu$  (prior parameters)**. Perform gradient descent on the cross-validation error to improve the values of  $\alpha$  and  $\nu$ .  
 (c) **Update the super-resolution image and the registration parameters**. Optimize  $\mathcal{L}$  (equation 1.33) 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  $\alpha$ ,  $\nu$ , or any element of  $\mathbf{x}$ ,  $\boldsymbol{\lambda}$  or  $\boldsymbol{\phi}$  is above preset convergence thresholds, return to 2.

**FIGURE 1.11**  
**Basic structure of the simultaneous algorithm.**

tends to spread the assumed influence of each high-resolution pixel over a larger area of the collection of low-resolution images.

---

### 1.3 Simultaneous super-resolution

In the preceding section, we saw that the the problems of determining image registration or motion estimation, low-resolution image blur estimation, selection of a suitable prior, and super-resolution image estimation are seldom truly independent. In addition, it is also expected 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.

The basic structure of the simultaneous super-resolution algorithm is given in Figure 1.11, and its components are each outlined in turn in the rest of this section, which concludes with examples on synthetic and real super-resolution datasets.

### 1.3.1 Super-resolution with registration

Standard approaches to super-resolution *first* determine the registration, *then* fix it and optimize a function like the MAP objective function of (1.14) 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 image  $\mathbf{x}$ , the set of geometric registration parameters,  $\boldsymbol{\theta}$  (which parameterize  $\mathbf{W}$ ), and the photometric parameters  $\boldsymbol{\lambda}$  (composed of the  $\lambda_1$  and  $\lambda_2$  values), at the same time, i.e. we determine the point at which

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = 0. \quad (1.32)$$

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. 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, *e.g.* by assuming a very broad Gaussian prior distribution over relevant components of the geometric registration.

The simultaneous super-resolution and image registration problem closely resembles the well-studied problem of Bundle Adjustment [14], 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 [14].

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.*

$$\mathcal{L} = \sum_{k=1}^K \beta \|\mathbf{r}^{(k)}\|_2^2 + \nu \sum_{g \in \mathcal{D}\mathbf{x}} \rho(g, \alpha) \quad (1.33)$$

$$[\mathbf{x}_{\text{MAP}}, \boldsymbol{\theta}_{\text{MAP}}, \boldsymbol{\lambda}_{\text{MAP}}] = \underset{\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}}{\operatorname{argmax}} \mathcal{L}, \quad (1.34)$$

where (1.33) is the same as (1.29), repeated here for convenience, though any reasonable image prior can be used in place of the Huber-MRF here.

Using the Scaled Conjugate Gradients (SCG) implementation from Netlab [10], 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 (1.30), and the gradients with respect to  $\mathbf{W}^{(k)}$  and the photometric registration parameters are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(k)}} = -2\beta \lambda_1^{(k)} \mathbf{r}^{(k)} \mathbf{x}^T \quad (1.35)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_1^{(k)}} = -2\beta \mathbf{x}^T \mathbf{W}^{(k)T} \mathbf{r}^{(k)} \quad (1.36)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_2^{(k)}} = -2\beta \sum_{i=1}^M r_i^{(k)}. \quad (1.37)$$

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 affects the entire normalisation of  $\mathbf{W}^{(k)}$ , requiring a great deal of computation to find the exact derivatives of each individual matrix element.

### 1.3.2 Learning prior strength parameters from data

For most forms of MAP super-resolution, one must determine values for free parameters like the prior strength, and prior-specific additional parameters like the Huber-MRF's  $\alpha$  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  $\nu$  and  $\alpha$ . For example, the expression for the Huber-MRF is

$$p(\mathbf{x}) = \frac{1}{Z(\nu, \alpha)} \exp \left\{ -\nu \sum_{g \in \mathcal{D}\mathbf{x}} \rho(g, \alpha) \right\}, \quad (1.38)$$

and so the full negative log likelihood function is

$$\mathcal{L} = \frac{1}{2} \sum_{k=1}^K \beta \|\mathbf{r}^{(k)}\|_2^2 - \log p(\mathbf{x}) \quad (1.39)$$

$$= \frac{1}{2} \sum_{k=1}^K \beta \|\mathbf{r}^{(k)}\|_2^2 + \nu \sum_{g \in \mathcal{D}\mathbf{x}} \rho(g, \alpha) - \log Z(\nu, \alpha). \quad (1.40)$$

For a ML solution to  $\nu$  and  $\alpha$ , this should be optimized with respect to those two variables, and in fact the entire data-error term could be neglected, since it does not depend on these variables. However, the partition function  $Z(\nu, \alpha)$  for these sorts of edge-based priors is not generally easy to compute.

Rather than setting the prior parameters using an ML or MAP technique, therefore, cross-validation is chosen 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 error in the super-resolution estimate 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  $\alpha$  and  $\nu$  should minimize this cross-validation error.

In determining a search direction in  $(\nu, \alpha)$ -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  $(\nu, \alpha)$ -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.

### 1.3.3 Scaling and convergence

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.

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. Thresholds are defined differently depending on the nature of the parameter. For  $\mathbf{x}$ , 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) is chosen. The same threshold number (*i.e.* 0.3) is used for the registration

parameter values, after they have been shifted and scaled as described above. For  $\nu$  and  $\alpha$ , 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 of 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.

### 1.3.4 Initialization

In our experiments, input images are assumed to be pre-registered by a standard algorithm [7] (*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 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, they are represented as log values throughout. A candidate PSF is selected in order to compute the *average image*,  $\mathbf{a}$ , which is a stable though excessively smooth approximation to  $\mathbf{x}$ . 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

$$\mathbf{a} = \mathbf{S}^{-1}\mathbf{W}^T\boldsymbol{\Lambda}_1^{-1}(\mathbf{y} - \boldsymbol{\lambda}_2), \quad (1.41)$$

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  $\mathbf{W}$ . Notice that both inverted matrices are diagonal, so  $\mathbf{a}$  is simple to compute. The resulting images are generally much smoother than the corresponding high-resolution frame calculated from the same input images will be, but are very robust to noise on the low-resolution images.

In order to get a registration estimate, it is possible to optimize  $\mathcal{L}$  of 1.33 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 (1.8) and (1.9) 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 [3]. 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.

### 1.3.5 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 (1.1) 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 algo

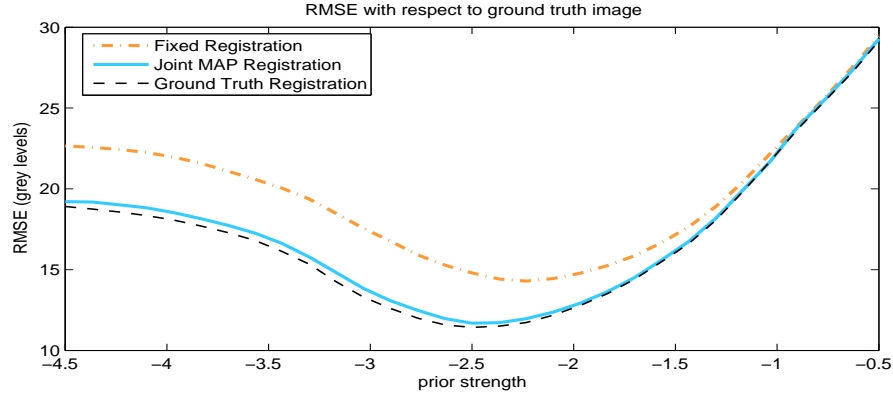
#### Registration only

In order to compare the simultaneous super-resolution algorithm to one with a fixed pre-determined image registration, synthetic low-resolution images were generated from a simple “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  $[-\frac{1}{2}, \frac{1}{2}]$  for the experiments), though the majority of the pixels are black or white. The low-resolution images are  $30 \times 30$  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.

An initial registration was then carried out using the *average image registration* technique described above. 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.

Two sets of super-resolution images were computed: one set with a fixed registration, and one set using the simultaneous approach. Within each set, the value for the prior strength parameter  $\nu$  was varied while keeping the Huber parameter  $\alpha$  set to 0.04. The noise precision parameter  $\beta$  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.

The RMSE compared to the ground truth image for both the fixed registration and the joint MAP approach are plotted, in Figure 1.12, along with a curve representing the performance if the ground truth registration is known. The joint MAP curve is very close to that of the true registrations, showing

**FIGURE 1.12**

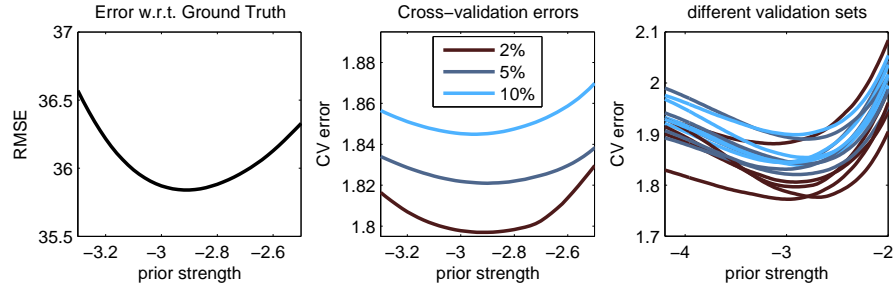
**Synthetic data results.** RMSE compared to ground truth for the “eyechart” text image, plotted for the fixed and joint MAP algorithms, and for the Huber super-resolution image found using the ground truth registration.

that allowing the super-resolution image to inform an update to the registrations does indeed yield an improvement in super-resolution performance.

### Cross-validation example

A second synthetic-data experiment shows that the cross-validation-based prior learning phase is effective. The cross-validation error is measured by holding a percentage of 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, and the results are plotted in Figure 1.13, 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  $-2.92$ . In the cross-validation plots shown in the centre of the figure, the curves’ minima 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 straight-forward to achieve.

**FIGURE 1.13**

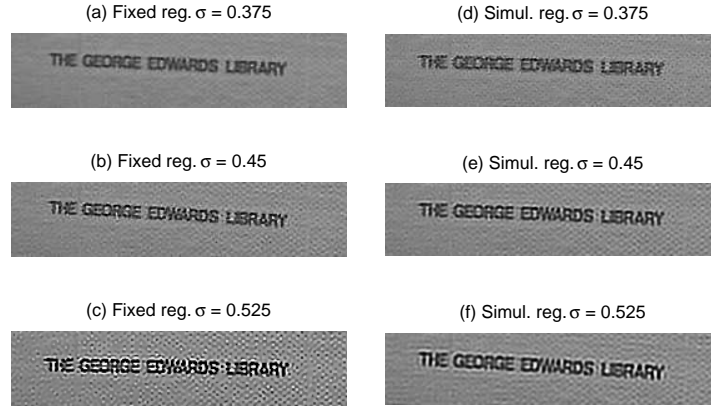
**Cross-validation errors on synthetic data.** Left: Error with respect to ground truth on the Keble dataset for this noise level. Centre: Three cross-validation curves, corresponding to 2%, 5% and 10% of pixels being selected. Right: More cross-validation curves at each ratio.

### 1.3.6 Tests on real data

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  $3.5 \times 10^9$  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 1.15 show that while both algorithms perform well with the middle PSF size, the simultaneous-registration algorithm handles the worse PSF estimates more gracefully.

**FIGURE 1.14**

**The ‘Surrey library’ real dataset:** close-ups of text from 9 of the 30 images across the low-resolution sequence, recorded with a real camera panning over a scene including the external wall of a library.

**FIGURE 1.15**

**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.

### Colour Sequence:

Finally, the simultaneous super-resolution method is demonstrated on a publically-available colour super-resolution dataset<sup>1</sup>. The dataset consists of 30 low-resolution images captured with a colour camera following a global translation model. The PSF standard deviation was chosen to be 0.4 low-resolution pixels, the desired zoom factor was set to  $\pi$ , and the simultaneous algorithm was run as described above.

In generalising the basic algorithm to handle colour images, a very simple approach is taken. The three colour channels are treated separately, though the geometric and photometric components are shared between the two. The Huber prior and data error terms are summed over the three channels, and all other algorithm components remain the same. Note that no colour demosaicing step has been introduced for the purpose of this example.

Figure 1.16 shows 9 of the 30 low-resolution input images along with the super-resolution result in this case. This result compares well to other results for algorithms which do not employ a colour demosaicing step in their models; the “jagging” artifacts visible at book edges, especially on the left-hand side of the image, result from this colour channel alignment, but overall the prior term has adapted effectively to keep the overall image natural-looking at most of the edges, and book titles such as “Kalman Filtering” and “French Review

<sup>1</sup>Data from <http://www.soe.ucsc.edu/~milanfar/software/sr-datasets.html>

**FIGURE 1.16**

**Results on a colour sequence.** Left: nine of the 30 colour low-resolution images. Right: the super-resolution image when the zoom factor was chosen arbitrarily to be  $\pi$ .

and Practice” are clearly legible, which is not the case in the input images. Other than specifying the PSF and the desired zoom factor, this is achieved entirely automatically without any parameter-tuning necessary.

## 1.4 Bayesian marginalization

This section describes 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 by Bayesian marginalization. Such parameters are sometimes 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 also describe the alternative Bayesian approach of Tipping and Bishop [13], 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, 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.

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.

#### 1.4.1 Marginalizing over registration parameters

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

If these parameters are collected together into a single vector,  $\phi$ , this marginalizations can be expressed simply as

$$p(\mathbf{x} | \{\mathbf{y}^{(k)}\}) = \int p(\mathbf{x}, \phi | \{\mathbf{y}^{(k)}\}) d\phi \quad (1.42)$$

$$= \int \frac{p(\mathbf{x}, \phi)}{p(\{\mathbf{y}^{(k)}\})} p(\{\mathbf{y}^{(k)}\} | \mathbf{x}, \phi) d\phi \quad (1.43)$$

$$= \frac{p(\mathbf{x})}{p(\{\mathbf{y}^{(k)}\})} \int p(\{\mathbf{y}^{(k)}\} | \mathbf{x}, \phi) p(\phi) d\phi. \quad (1.44)$$

Notice that  $p(\mathbf{x}, \phi) = p(\mathbf{x}) p(\phi)$  (because the super-resolution image and registration parameters are independent) and that  $p(\mathbf{x})$  and  $p(\{\mathbf{y}^{(k)}\})$  can be taken outside the integral. This leaves one free to choose any suitable super-resolution image prior  $p(\mathbf{x})$ , rather than being constrained to picking a Gaussian merely to make the integral tractable, as in the image-marginalizing case discussed later.

A prior distribution over  $\phi$ , which appears within the integral, must be specified. We assume that a preliminary image registration (geometric and photometric) and an estimate of the PSF are available, and also that these registration values are related to the ground truth registration values by unknown zero-mean Gaussian-distributed additive noise. The registration estimate can be obtained using classical registration methods, either intensity-based [8] or estimation from image points [7]. There is a rich literature of *Blind Image Deconvolution* concerned with estimating an unknown blur on an image [9].

We introduce a vector  $\delta$  to represent the perturbations from ground truth in the initial parameter estimate. For the parameters of a single image, this gives

$$\begin{bmatrix} \theta^{(k)} \\ \lambda_1^{(k)} \\ \lambda_2^{(k)} \end{bmatrix} = \begin{bmatrix} \bar{\theta}^{(k)} \\ \bar{\lambda}_1^{(k)} \\ \bar{\lambda}_2^{(k)} \end{bmatrix} + \delta^{(k)} \quad (1.45)$$



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

$$\boldsymbol{\delta}^T = \left[ \boldsymbol{\delta}^{(1)T}, \boldsymbol{\delta}^{(2)T}, \dots, \boldsymbol{\delta}^{(K)T}, \delta_\gamma \right] \quad (1.46)$$

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

$$\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \mathbf{V}). \quad (1.47)$$

The final step before the integral of (1.44) can be evaluated is to bring out the dependence on  $\boldsymbol{\phi}$  in  $p(\{\mathbf{y}^{(k)}\} | \mathbf{x}, \boldsymbol{\phi})$ . Starting with

$$p(\{\mathbf{y}^{(k)}\} | \mathbf{x}, \boldsymbol{\phi}) = \left( \frac{\beta}{2\pi} \right)^{-\frac{KM}{2}} \exp \left\{ -\frac{\beta}{2} \mathbf{e}(\boldsymbol{\delta}) \right\} \quad (1.48)$$

where

$$\mathbf{e}(\boldsymbol{\delta}) = \sum_{k=1}^K \left\| \mathbf{y}^{(k)} - \lambda_1^{(k)} \mathbf{W}(\boldsymbol{\theta}^{(k)}, \gamma) \mathbf{x} - \lambda_2^{(k)} \right\|_2^2 \quad (1.49)$$

and where  $\boldsymbol{\theta}$ ,  $\boldsymbol{\lambda}$  and  $\gamma$  are functions of  $\boldsymbol{\delta}$  and the initial registration values, we can then expand the integral in (1.44) to an integral over  $\boldsymbol{\delta}$ ,

$$\begin{aligned} p(\mathbf{x} | \{\mathbf{y}^{(k)}\}) &= \frac{p(\mathbf{x}) |\mathbf{V}^{-1}|^{1/2} \beta^{KM/2}}{p(\{\mathbf{y}^{(k)}\}) (2\pi)^{(KM+Kn+1)/2}} \\ &\quad \times \int \exp \left\{ -\frac{\beta}{2} \mathbf{e}(\boldsymbol{\delta}) - \frac{1}{2} \boldsymbol{\delta}^T \mathbf{V}^{-1} \boldsymbol{\delta} \right\} d\boldsymbol{\delta}. \end{aligned} \quad (1.50)$$

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

$$e(\boldsymbol{\delta}) = f + \mathbf{g}^T \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^T \mathbf{H} \boldsymbol{\delta}. \quad (1.51)$$

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  $\{\mathbf{y}^{(k)}, \boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}\}$ .

We are now in a position to evaluate the integral in (1.50) using the identity [2]

$$\int \exp \left\{ -\mathbf{b}\mathbf{x} - \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} \right\} d\mathbf{x} = 2\pi^{\frac{d}{2}} |\mathbf{A}|^{-\frac{1}{2}} \exp \{ \mathbf{b}^T \mathbf{A}^{-1} \mathbf{b} \}, \quad (1.52)$$

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

The exponent in the integral in (1.50), becomes

$$a = -\frac{\beta}{2}\mathbf{e}(\boldsymbol{\delta}) - \frac{1}{2}\boldsymbol{\delta}^T \mathbf{V}^{-1} \boldsymbol{\delta} \quad (1.53)$$

$$= -\frac{\beta}{2}f - \frac{\beta}{2}\mathbf{g}^T \boldsymbol{\delta} - \frac{1}{2}\boldsymbol{\delta}^T \left[ \frac{\beta}{2}\mathbf{H} + \mathbf{V}^{-1} \right] \boldsymbol{\delta}. \quad (1.54)$$

so that

$$\int \exp\{a\} d\boldsymbol{\delta} = \exp\left\{-\frac{\beta}{2}f\right\} \int \exp\left\{-\frac{\beta}{2}\mathbf{g}^T \boldsymbol{\delta} - \frac{1}{2}\boldsymbol{\delta}^T \mathbf{S} \boldsymbol{\delta}\right\} d\boldsymbol{\delta} \quad (1.55)$$

$$= \exp\left\{-\frac{\beta}{2}f\right\} (2\pi)^{\frac{nK+1}{2}} |\mathbf{S}|^{-\frac{1}{2}} \exp\left\{\frac{\beta^2}{8}\mathbf{g}^T \mathbf{S}^{-1} \mathbf{g}\right\} \quad (1.56)$$

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 (1.50), the final expression for our conditional distribution of the super-resolution image is

$$p(\mathbf{x} | \{\mathbf{y}^{(k)}\}) = \frac{p(\mathbf{x})}{p(\{\mathbf{y}^{(k)}\})} \left( \frac{\beta^{KM} |\mathbf{V}^{-1}|}{(2\pi)^{KM} |\mathbf{S}|} \right)^{\frac{1}{2}} \times \exp\left\{-\frac{\beta}{2}f + \frac{\beta^2}{8}\mathbf{g}^T \mathbf{S}^{-1} \mathbf{g}\right\}. \quad (1.57)$$

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 (1.1.3), this gives

$$\mathcal{L} = \frac{\nu}{2}\rho(\mathbf{D}\mathbf{x}, \alpha) + \frac{\beta}{2}f + \frac{1}{2}\log |\mathbf{S}| - \frac{\beta^2}{8}\mathbf{g}^T \mathbf{S}^{-1} \mathbf{g}. \quad (1.58)$$

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 summarised

$$f(\mathbf{x}) = \sum_{k=1}^K \left\| \mathbf{y}^{(k)} - \lambda_1^{(k)}(\boldsymbol{\delta}) \mathbf{W}^{(k)}(\boldsymbol{\delta}) \mathbf{x} - \lambda_2^{(k)}(\boldsymbol{\delta}) \right\|_2^2 \quad (\text{scalar}) \quad (1.59)$$

$$\mathbf{g}(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \boldsymbol{\delta}} \quad (p \times 1 \text{ gradient vector}) \quad (1.60)$$

$$\mathbf{H}(\mathbf{x}) = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \boldsymbol{\delta}} \quad (p \times p \text{ Hessian matrix}) \quad (1.61)$$

$$\mathbf{S}(\mathbf{x}) = \frac{\beta}{2}\mathbf{H}(\mathbf{x}) + \mathbf{V}^{-1} \quad (p \times p \text{ matrix}), \quad (1.62)$$

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  $\mathbf{V}$ . A detailed derivation of the gradient of (1.58) can be found in [12].

### 1.4.2 Marginalizing over the super-resolution image

We will outline the marginalization method used in [13] 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 [13] 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 1.1.3 (equations (1.16) and (1.24)), but if we consider the exact form of the probability and its normalizing constant, it is

$$p(\mathbf{x}) = (2\pi)^{-\frac{N}{2}} |\mathbf{Z}_x|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \mathbf{x}^T \mathbf{Z}_x^{-1} \mathbf{x} \right\}. \quad (1.63)$$

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

$$\mathbf{y} = \mathcal{N}(\mathbf{y} | \boldsymbol{\lambda}_2, \mathbf{Z}_y), \quad (1.64)$$

where

$$\mathbf{Z}_y = \beta^{-1} \mathbf{I} + \boldsymbol{\Lambda}_1 \mathbf{W} \mathbf{Z}_x \mathbf{W}^T \boldsymbol{\Lambda}_1^T. \quad (1.65)$$

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.

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

$$\mathcal{L} = \frac{1}{2} \left[ \beta \sum_{k=1}^K \left\| \mathbf{y}^{(k)} - \lambda_1^{(k)} \mathbf{W}^{(k)} \boldsymbol{\mu} - \boldsymbol{\lambda}_2^{(k)} \right\|_2^2 + \boldsymbol{\mu}^T \mathbf{Z}_x^{-1} \boldsymbol{\mu} - \log |\boldsymbol{\Sigma}| \right] \quad (1.66)$$

where

$$\boldsymbol{\Sigma} = \left[ \mathbf{Z}_x^{-1} + \beta \sum_{k=1}^K \lambda_1^{(k)2} \mathbf{W}^{(k)T} \mathbf{W}^{(k)} \right]^{-1} \quad (1.67)$$

$$\boldsymbol{\mu} = \beta \boldsymbol{\Sigma} \left( \sum_{k=1}^K \lambda_1^{(k)} \mathbf{W}^{(k)T} \left( \mathbf{y}^{(k)} - \boldsymbol{\lambda}_2^{(k)} \right) \right). \quad (1.68)$$

The expression for the posterior mean  $\boldsymbol{\mu}$  is the closed form of the overall MAP solution for the super-resolution image. However, in [13], the optimization over registration and blur parameters is carried out with low-resolution

image patches of just  $9 \times 9$  pixels, rather than the full low-resolution images, because of the computational cost involved in computing the terms in (1.66) — even for a tiny  $50 \times 50$ -pixel high-resolution image, the  $\mathbf{Z}_x$  and  $\mathbf{\Sigma}$  matrices are  $2500 \times 2500$ . 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  $9 \times 9$  patches. This is exactly equivalent to solving the usual MAP super-resolution approach of Section 1.1.2.3, with  $p(\mathbf{x})$  defined as in (1.16), using the covariance of (1.24).

In comparison, the dimensionality of the matrices in the terms comprising the registration-marginalizing objective function (1.58) is in most cases much lower than those in (1.66). 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.

### 1.4.3 Implementation notes

The objective function (1.58) can be optimized using *Scaled Conjugate Gradients* (SCG) [10], noting that the gradient can be expressed

$$\begin{aligned} \frac{d\mathcal{L}}{d\mathbf{x}} = & \frac{\nu}{2} \mathbf{D}^T \frac{d}{d\mathbf{x}} \rho(\mathbf{D}\mathbf{x}, \alpha) + \frac{\beta}{2} \frac{df}{d\mathbf{x}} - \frac{\beta^2}{4} \boldsymbol{\xi}^T \frac{d\mathbf{g}}{d\mathbf{x}} \\ & + \left[ \frac{\beta}{4} \text{vec} \left( \mathbf{S}^{-1} + \frac{\beta^2}{8} \boldsymbol{\xi} \boldsymbol{\xi}^T \right)^T \right] \frac{d\text{vec}(\mathbf{H})}{d\mathbf{x}}, \end{aligned} \quad (1.69)$$

where

$$\boldsymbol{\xi} = \mathbf{S}^{-1} \mathbf{g}, \quad (1.70)$$

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  $e^{(k)}$  ( $\boldsymbol{\delta}^{(k)}$ ) with respect to  $\mathbf{x}$ ) with respect to the geometric parameters.

The upper part of  $\mathbf{H}$  is block-diagonal  $nK \times nK$  sparse matrix, and the final  $(nK + 1)^{\text{th}}$  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 (1.58) is simply the reprojection error of the current estimate of  $\mathbf{x}$  at the mean registration parameter values, *i.e.* the value of (1.49) 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  $k^{\text{th}}$  image is used.

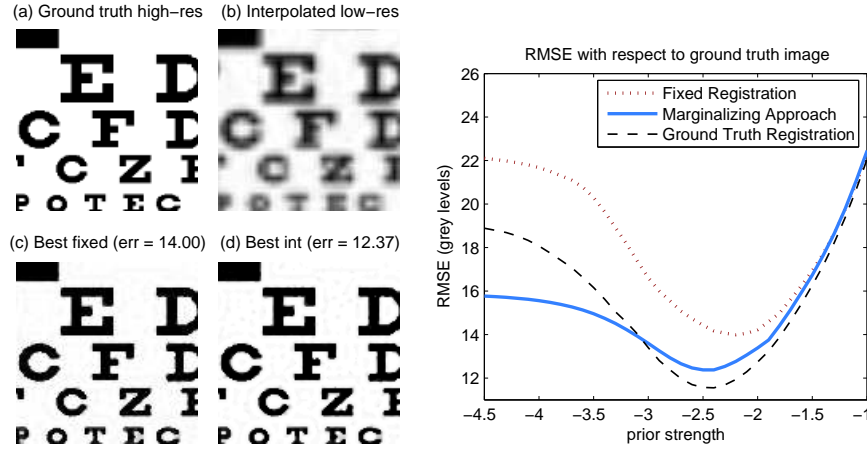


FIGURE 1.17

**Super-resolving the synthetic eyechart dataset.** (a) ground truth image; (b) interpolated low-resolution image; (c) best (minimum MSE) image from the regular Huber-MAP algorithm; and (d) best result using our approach of integrating over  $\theta$  and  $\lambda$ . As well as having a lower RMSE, note the improvement in black-white edge detail on some of the letters on the bottom line. Right: variation of RMSE with prior strength for the standard Huber-prior MAP super-resolution method and our approach integrating over  $\theta$  and  $\lambda$ .

#### 1.4.4 Experimental Evaluation

Results from two experiments show the marginalization methods working on synthetic and real data. The first example uses a synthetic dataset to allow a quantitative measure of performance to be taken with respect to known ground truth high-resolution images. The second example shows the full system working on real data, and compares the results to the standard Huber-MAP method, and to the approach of [13].

##### Synthetic eyechart sequence

Sixteen low-resolution “eyechart” inputs were created from the ground truth image of an eye chart, as used for Figure 1.12. Each image is generated at a zoom factor of 4, again using the model with 2 translational degrees of freedom and two photometric degrees of freedom. A Gaussian point-spread function with a standard deviation of 0.4 low-resolution pixels is used, and Gaussian noise (30dB; standard deviation equivalent to approximately 3.4 grey levels) is added to the intensity of each low-resolution pixel independently.

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-



**FIGURE 1.18**  
Two of the ten input images in the real dataset.

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  $\alpha$  was fixed at 0.01 for all runs, and  $\nu$  was varied over a range of possible values representing ratios between  $\nu$  and the image noise precision  $\beta$ .

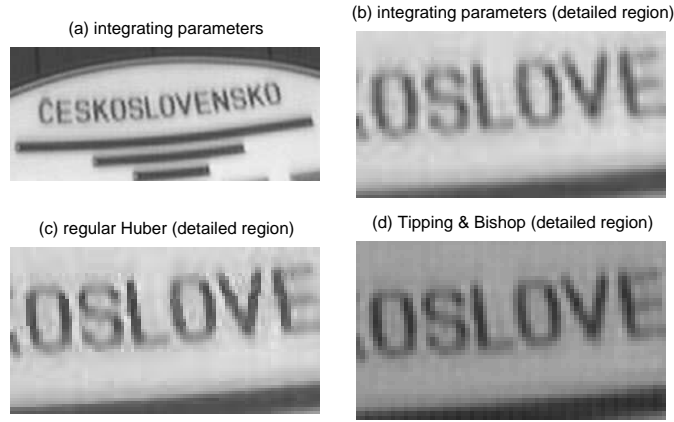
The images giving lowest RMS error from each set are displayed Figure 1.17. Visually, the differences between the images are subtle, though the bottom row of letters is better defined in the output from our algorithm. Plotting the RMSE as a function of  $\nu$ , it is clear that the registration-marginalizing 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  $\nu$  and  $\beta$  are free parameters in the algorithm, it is an advantage that the marginalizing approach is less sensitive to variation in their values.

### Real data

The final example 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 1.18. 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 [3] 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, extended to cover the illumination model, was 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 1.19. 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 regulariza-

**FIGURE 1.19**

**Super-resolving the “Československo” sequence** (a) Full output from our algorithm. (b) Detail of the central letters, again with our algorithm. (c) Detail with the regular Huber-MAP super-resolution image. (d) Detail 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.

tion in the constant background regions is slightly more successful. Also note that the Gaussian prior on the image-marginalizing method is zero-mean (see equation (1.24)), 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.

#### 1.4.5 Discussion

It is possible to interpret the extra terms introduced into the objective function in the registration-marginalizing method as an extra regularizer term or image prior. Considering (1.58), 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 (1.58) will favour image solutions which are not acutely sensitive to minor adjustments in the image registration. Since the chequer-board pattern in ML super-resolution images die to ill-conditioning *is* very sensitive to the exact registration, this component of the super-resolution image is penalised.

---

## 1.5 Concluding remarks

In this chapter we have highlighted the importance of considering latent quantities such as image registration or point-spread function size as part of the super-resolution problem instead of estimating and fixing them in advance. Within a probabilistic framework based on a generative model of the image formation process, two different algorithms were described: one which optimizes the latent variables at the same time as the super-resolution image, and one which marginalizes them out of the 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 it operates. Note that while the examples in the marginalization section concentrated on a translation-only 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.

The simultaneous super-resolution algorithm which was presented earlier is conceptually simpler to understand and implement than the marginalization approach, but likewise demonstrated a quantitative improvement in super-resolution image quality. While a combination of the two approaches may yield even more accurate results at a higher computational cost, the efficacy of the simultaneous approach makes it a reliable choice of super-resolution algorithm when a high degree of reconstruction accuracy is required.



---

## Bibliography

- [1] S. Baker and T. Kanade. Limits on super-resolution and how to break them. 24(9):1167–1183, 2002.
- [2] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [3] D. P. Capel. *Image Mosaicing and Super-resolution (Distinguished Dissertations)*. Springer, ISBN: 1852337710, 2004.
- [4] 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.
- [5] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. 13(10):1327–1344, October 2004.
- [6] R. C. Hardie, K. J. Barnard, and E. E. Armstrong. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. 6(12):1621–1633, 1997.
- [7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [8] M. Irani and S. Peleg. Improving resolution by image registration. *Graphical Models and Image Processing*, 53:231–239, 1991.
- [9] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–46, May 1996.
- [10] I. Nabney. *Netlab algorithms for pattern recognition*. Springer, 2002.
- [11] N. Nguyen, P. Milanfar, and G. Golub. Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement. 10(9):1299–1308, September 2001.
- [12] L. C. Pickup. *Machine Learning in Multi-frame Image Super-resolution*. PhD thesis, University of Oxford, February 2008.
- [13] M. E. Tipping and C. M. Bishop. Bayesian image 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.

- [14] 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.