

Chapter 1

SUPER-RESOLUTION: LIMITS AND BEYOND

Simon Baker
and Takeo Kanade

Abstract

A variety of super-resolution algorithms have been described in this book. Most of them are based on the same source of information however; that the super-resolution image should generate the lower resolution input images when appropriately warped and down-sampled to model image formation. (This information is usually incorporated into super-resolution algorithms in the form of reconstruction constraints which are frequently combined with a smoothness prior to regularize their solution.) In this final chapter, we first investigate how much extra information is actually added by having more than one image for super-resolution. In particular, we derive a sequence of analytical results which show that the reconstruction constraints provide far less useful information as the decimation ratio increases. We validate these results empirically and show that for large enough decimation ratios any smoothness prior leads to overly smooth results with very little high-frequency content however many (noiseless) low resolution input images are used. In the second half of this chapter, we propose a super-resolution algorithm which uses a completely different source of information, in addition to the reconstruction constraints. The algorithm recognizes local “features” in the low resolution images and then enhances their resolution in an appropriate manner, based on a collection of high and low-resolution training samples. We call such an algorithm a *hallucination* algorithm.

Keywords: Super-resolution, analysis of limits, learning, faces, text, hallucination.

1. INTRODUCTION

A large number of super-resolution algorithms have been described in this book. Most of them, however, are based on the same source of information; specifically, that the super-resolution image, when appropriately warped and down-sampled to model the image formation process, should yield the low resolution images. This information is typically embedded in a set of reconstruction constraints, first introduced by (Peleg et al., 1987; Irani and Peleg, 1991).

These reconstruction constraints can be embedded in a Bayesian framework incorporating a prior on the super-resolution image (Schultz and Stevenson, 1996; Hardie et al., 1997; Elad and Feuer, 1997). Their solution can also be estimated either in batch mode or recursively using a Kalman filter (Elad and Feuer, 1999; Dellaert et al., 1998). Several other refinements have been proposed, including simultaneously computing 3D structure (Cheeseman et al., 1994; Shekarforoush et al., 1996; Smelyanskiy et al., 2000) and removing other degrading artifacts such as motion blur (Bascle et al., 1996).

In the first part of this chapter, we analyze the super-resolution reconstruction constraints. We derive three analytical results which show that the amount of information provided by having more than one image available for super-resolution becomes very much less as the decimation ratio q increases. Super-resolution therefore becomes inherently much more difficult as q increases. This reduction in the amount of information provided by the reconstruction constraints is traced to the fact that the pixel intensities in the input images take discrete values (typically 8-bit integers in the range 0–255). This causes a loss of information and imposes inherent limits on how well super-resolution can be performed from the reconstruction constraints (and other equivalent formulations based on the same underlying source of information.)

How, then, can high-decimation ratio super-resolution be performed? Our analytical results hold for an arbitrary number of images so using more low resolution images does not help. Suppose, however, that the input images contain printed text. Moreover, suppose that it is possible to perform optical character recognition (OCR) and recognize the text. If the font can also be determined, it would then be easy to perform super-resolution for *any decimation ratio*. The text could be reproduced at any resolution by simply rendering it from the script of the text and the definition of the font. In the second half of this chapter, we describe a super-resolution algorithm based on this idea which we call *hallucination* (Baker and Kanade, 1999; Baker and Kanade, 2000a). Our super-resolution hallucination algorithm is based, however, on the recognition of generic local “features” (rather than the characters detected by OCR). It can therefore be applied to other phenomena such as images of human faces.

2. THE RECONSTRUCTION CONSTRAINTS

Denote the low resolution input images by $x_L^{(k)}(i, j)$ where $k = 1, \dots, K$. The starting point in the derivation of the reconstruction constraints is then the continuous image formation equation (Horn, 1996):

$$x_L^{(k)}(i, j) = \left(I^{(k)} * h^{(k)} \right) (i, j) = \int_{x_L^{(k)}} I^{(k)}(x, y) \cdot h^{(k)}(x - i, y - j) \, dx \, dy \quad (1.1)$$

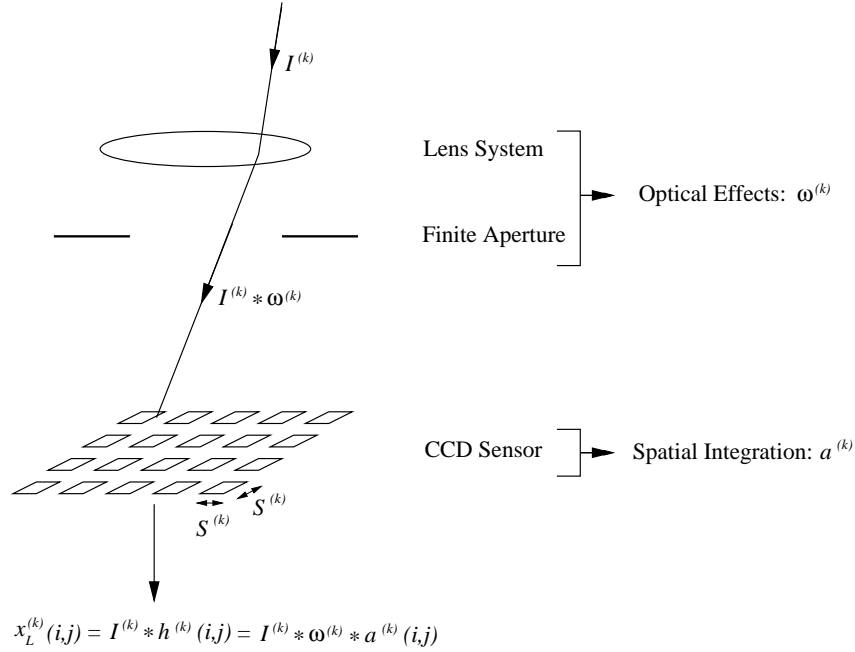


Figure 1.1 The low resolution input images $x_L^{(k)}$ are formed by the convolution of the irradiance $I^{(k)}$ with the camera point spread function $h^{(k)}$. We model the point spread function itself as the convolution of two terms: (1) $\omega^{(k)}$ models the optical effects caused by the lens and the finite aperture, and (2) $a^{(k)}$ models the spatial integration performed by the CCD sensor.

where $I^{(k)}(x, y)$ is the continuous irradiance function that would have reached the image plane of the k^{th} camera under the pinhole model, and $h^{(k)}$ is point spread function of the k^{th} camera. The (double) integration is performed over the image plane of $x_L^{(k)}$. See Figure 1.1 for an illustration.

2.1 MODELING THE POINT SPREAD FUNCTION

We decompose the point spread function into two parts (see Figure 1.1):

$$h^{(k)}(x, y) = \left(\omega^{(k)} * a^{(k)} \right) (x, y) \quad (1.2)$$

where $\omega^{(k)}(x, y)$ models the blurring caused by the optics and $a^{(k)}(x, y)$ models the spatial integration performed by the CCD sensor (Baker et al., 1998). The optical blurring $\omega^{(k)}$ is typically further split into a defocus factor that can be approximated by a pill-box function and a diffraction-limited optical transfer function that can be modeled by the square of the first-order Bessel function of the first kind (Born and Wolf, 1965). We aim to be as general as possible and

so avoid making any assumptions about $\omega^{(k)}$. Instead, (most of) our analysis is performed for arbitrary optical blurring functions. We do, however, assume a parametric form for $a^{(k)}$. We assume that the photo-sensitive areas of the CCD pixels are square and uniformly sensitive to light, as in (Baker et al., 1998; Barbe, 1980). If the length of the side of the square photosensitive area is $S^{(k)}$, the spatial integration function is then:

$$a^{(k)}(x, y) = \begin{cases} \frac{1}{S^{(k)} \times S^{(k)}} & \text{if } |x| \leq \frac{S^{(k)}}{2} \text{ and } |y| \leq \frac{S^{(k)}}{2} \\ 0 & \text{otherwise.} \end{cases} \quad (1.3)$$

In general the photosensitive area is not the entire pixel since space is needed for the circuitry to read out the charge. Therefore the only assumption we make about $S^{(k)}$ is that it lies in $[0, 1]$. Our analysis is then in terms of $S^{(k)}$ (rather than the inter-pixel distance which is assumed to define the unit distance.)

2.2 WHAT IS SUPER-RESOLUTION ANYWAY?

We wish to estimate a super-resolution image $x_H(i', j')$. Precisely what does this mean? Let us begin with the coordinate frame of x_H . The coordinate frame of a super-resolution image is typically defined relative to that of the corresponding low resolution input image. If the decimation ratio is q , the pixels in x_H will be q times closer to each other than those in the corresponding low resolution image, $x_L^{(k')}$ say. The coordinate frame of x_H can therefore be defined in terms of that for $x_L^{(k')}$ via:

$$(i', j') = \left(\frac{i}{q}, \frac{j}{q} \right). \quad (1.4)$$

In this chapter we assume that the input images have already been registered with each other and therefore with the coordinate frame of x_H . Then, denote the point in image $x_L^{(k)}$ (where k may or may not equal k') that corresponds to (x, y) in x_H by $\mathbf{r}^{(k)}(x, y)$. From now on we assume that $\mathbf{r}^{(k)}$ is known.

The integration in Equation (1.1) is performed over the low resolution image plane. Transforming to the super-resolution image plane of x_H gives:

$$x_L^{(k)}(i, j) = \int_{x_H} I^{(k)}(\mathbf{r}^{(k)}(x, y)) \cdot h^{(k)}(\mathbf{r}^{(k)}(x, y) - (i, j)) \cdot \left| \frac{\partial \mathbf{r}^{(k)}}{\partial x, y} \right| dx dy \quad (1.5)$$

where $\left| \frac{\partial \mathbf{r}^{(k)}}{\partial x, y} \right|$ is the determinant of the Jacobian of the registration $\mathbf{r}^{(k)}$.

Now, $I^{(k)}(\mathbf{r}^{(k)}(x, y))$ is the irradiance that would have reached the image plane of the k^{th} camera under the pinhole model, transformed onto the super-resolution image plane. Assuming that the registration is correct, and that the radiance of every point in the scene does change across k (a Lambertian-like

assumption), $I^{(k)}(\mathbf{r}^{(k)}(x, y))$ should be the same for all k . Moreover, it equals the irradiance that would have reached the super-resolution image plane of x_H under the pinhole model. Denoting this function by $I(x, y)$, we have:

$$x_L^{(k)}(i, j) = \int_{x_H} I(x, y) \cdot h^{(k)}(\mathbf{r}^{(k)}(x, y) - (i, j)) \cdot \left| \frac{\partial \mathbf{r}^{(k)}}{\partial x, y} \right| dx dy. \quad (1.6)$$

The goal of super-resolution is then to recover (a representation of) $I(x, y)$. Doing this requires both increasing the resolution and “deblurring” the image; i.e. removing the effects of the convolution with the point spread function $h^{(k)}$.

In order to proceed we need to specify which continuous function $I(x, y)$ is represented by the discrete image $x_H(i', j')$. For simplicity, we assume that $x_H(i', j')$ represents the piecewise constant function:

$$I(x, y) = x_H(i', j') \quad (1.7)$$

for all $x \in (i' - 0.5, i' + 0.5]$ and $y \in (j' - 0.5, j' + 0.5]$. Then, Equation (1.6) can be rearranged to give the super-resolution reconstruction constraints:

$$x_L^{(k)}(i, j) = \sum_{i', j'} W^{(k)}(i, j, i', j') \cdot x_H(i', j') \quad (1.8)$$

where $k = 1, \dots, K$ and:

$$W^{(k)}(i, j, i', j') = \int_{i'-0.5, j'-0.5}^{i'+0.5, j'+0.5} h^{(k)}(\mathbf{r}^{(k)}(x, y) - (i, j)) \cdot \left| \frac{\partial \mathbf{r}^{(k)}}{\partial x, y} \right| dx dy. \quad (1.9)$$

The super-resolution reconstruction constraints are therefore a set of linear constraints on the unknown super-resolution pixels $x_H(i', j')$ in terms of the known low resolution pixels $x_L^{(k)}(i, j)$ and the coefficients $W^{(k)}(i, j, i', j')$.

3. ANALYSIS OF THE CONSTRAINTS

The constant coefficients $W^{(k)}(i, j, i', j')$ in the reconstruction constraints depend on both the point spread function $h^{(k)}$ and the registration $\mathbf{r}^{(k)}$. Without some assumptions about these functions any analysis would be meaningless. If the point spread function is arbitrary, it can be chosen to simulate the “small pixels” of the super-resolution image. Similarly, if the registration is arbitrary, it can be chosen (in effect) to move the camera towards the scene and thereby directly capture the super-resolution image. We therefore have to make some (reasonable) assumptions about the imaging conditions.

Assumptions Made About the Point Spread Function

As mentioned above, we assume that the point spread function takes the form of Equation (1.3). Moreover, we assume that the width of the photosensitive area $S^{(k)}$ is the same for all of the images (and equals S .) In the first part

of our analysis, we also assume that $\omega^{(k)}(x, y) = \delta(x) \cdot \delta(y)$, where δ is the Dirac delta function. Afterwards, in the second and third parts of our analysis, we allow $\omega^{(k)}$ to be arbitrary; i.e. our analysis holds for *any* optical blurring.

Assumptions Made About the Registration

To outlaw motions which (effectively) allow the camera to be moved towards the scene, we assume that each registration takes the form:

$$\mathbf{r}^{(k)}(x, y) = \frac{1}{q}(x, y) + (c^{(k)}, d^{(k)}) \quad (1.10)$$

where $(c^{(k)}, d^{(k)})$ is a constant translation (which in general may be different for each low resolution image k) and the $\frac{1}{q}$ accounts for the change of coordinate frame from high to low resolution images. See also Equation (1.4).

Even given these assumptions, the performance of any super-resolution algorithm will depend upon the exact number of input images K , the values of $(c^{(k)}, d^{(k)})$, and, moreover, how well the algorithm can register the low resolution images to estimate the $(c^{(k)}, d^{(k)})$. Our goal is to show that super-resolution becomes fundamentally more difficult as the decimation ratio q increases. We therefore assume that the conditions are as favorable as possible and perform the analysis for an arbitrary number of input images K , with arbitrary translations $(c^{(k)}, d^{(k)})$. We also assume that the algorithm has estimated these values perfectly. Any results derived under these conditions will only be stronger in practice, where the registrations may be degenerate or inaccurate.

3.1 INVERTIBILITY ANALYSIS

We first analyze when the reconstruction constraints are invertible, and what the rank of the null space is when they are not. In order to get an easily interpretable result, the analysis in this section is performed under the scenario that the optical blurring can be ignored; i.e. $\omega^{(k)}(x, y) = \delta(x) \cdot \delta(y)$. (This assumption will be removed in the following two sections.) The expression for $W^{(k)}(i, j, i', j')$ in Equation (1.9) then simplifies to:

$$\frac{1}{q^2} \int_{i'-0.5}^{i'+0.5} \int_{j'-0.5}^{j'+0.5} a^{(k)} \left(\frac{1}{q}(x, y) + (c^{(k)}, d^{(k)}) - (i, j) \right) dx dy. \quad (1.11)$$

Using the definition of $a^{(k)}$ it can be seen that $W^{(k)}(i, j, i', j')$ is equal to $1/(q \cdot S)^2$ times the area of the intersection of the two squares in Figure 1.2 (the high resolution pixel $[i' - 0.5, i' + 0.5] \times [j' - 0.5, j' + 0.5]$ and the region where $a^{(k)}$ non-zero and equals $\frac{1}{S^2}$.) We then have:

Theorem 1 *If $q \cdot S$ is an integer greater than 1, then for all $(c^{(k)}, d^{(k)})$ the reconstruction constraints (Equations (1.8) and (1.11)) are not invertible. Moreover, the dimension of the null space is at least $(q \cdot S - 1)^2$. If $q \cdot S$ is not an integer, $c^{(k)}$ and $d^{(k)}$ always exist such that the constraints are invertible.*

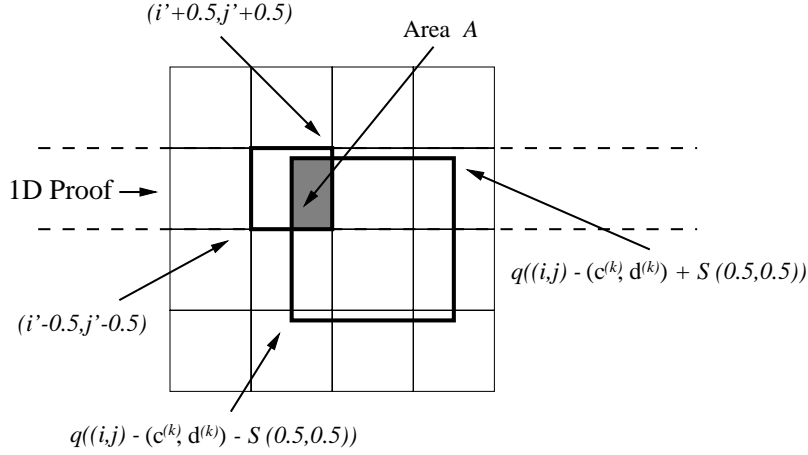


Figure 1.2 The high-resolution pixel (i', j') over which the integration is performed in Equation (1.11) is indicated by the small square at the upper middle left of the figure. The larger square towards the bottom right is the region in which $a^{(k)}$ is non-zero. Since $a^{(k)}$ takes the value $1/S^2$ in this region, the integral in Equation (1.11) equals A/S^2 , where A is the area of the intersection of the two squares. This figure is used to illustrate the 1D proof of Theorem 1.

Proof: We provide a proof for 1D images. (See Figure 1.2.) The extension to 2D is conceptually no more difficult and so is omitted for reasons of brevity.

The null space is defined by $\sum_{i', j'} \overline{W}^{(k)}(i, j, i', j') \cdot x_H^{(k)}(i', j') = 0$ where $\overline{W}^{(k)}(i, j, i', j') = (q \cdot S)^2 \cdot W^{(k)}(i, j, i', j')$ is the area of intersection of the 2 squares in Figure 1.2. Any element of the null space therefore corresponds to an assignment of values to the small squares such that their weighted sum (over the large square) equals zero, where the weights are the areas of intersection.

In 1D we just consider one row of the figure. Changing $c^{(k)}$ (and $d^{(k)}$) to slide the large square along the row by a small amount, we get a similar constraint on the elements in the null space. The only difference is in the left-most and right-most small squares. Subtracting these two constraints shows that the left-most square and the right-most square must have the same value.

If $q \cdot S$ is not an integer (or is 1), this proves that neighboring values of $x_H^{(k)}$ must be equal and hence 0. (Since $q \cdot S$ is not an integer, the big square slides out of one small square before the other and the result then follows by transitivity of equality.) Therefore, there exist values for the translations $c^{(k)}$ (and $d^{(k)}$) such that the null space only contains the zero vector; i.e. the reconstruction constraints are invertible in general if $q \cdot S$ is not an integer (or is 1).

If $q \cdot S$ is an integer greater than 1, this same constraint places an upper bound of $q \cdot S - 1$ on the maximum dimension of the null space computed over all possible translations $c^{(k)}$ (and $d^{(k)}$). The space of all assignments to $x_H^{(k)}$

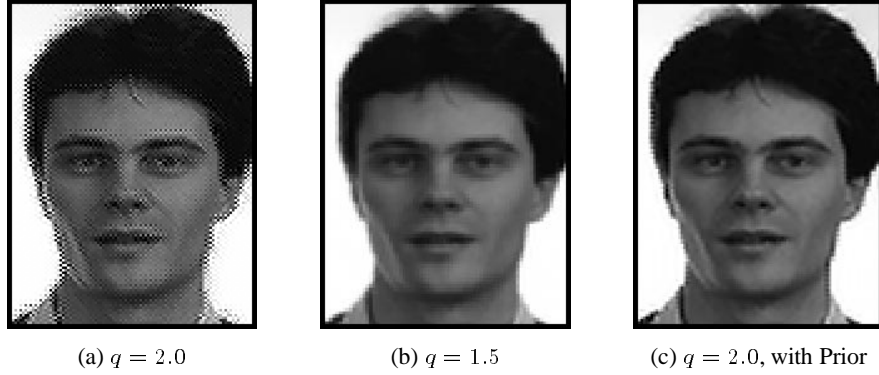


Figure 1.3 Validation of Theorem 1: The results of solving the reconstruction constraints using gradient descent for a square point spread function with $S = 1.0$. (a) When $q \cdot S$ is an integer, the equations are not invertible and so a random periodic image in the null space is added to the original image. (b) When $q \cdot S$ is not an integer, the reconstruction constraints are invertible (in general) and so a smooth solution is found, even without a prior. (The result for $q = 1.5$ was interpolated to make it the same size as that for $q = 2.0$.) (c) When a smoothness prior is added to the reconstruction constraints the difficulties seen in (a) disappear. (For larger values of q simply adding a smoothness prior does not solve this problem, as will be seen.)

that are periodic with period $q \cdot S$ and which have a zero mean can also easily be seen to always lie in the null space and so this value is also a lower bound on the dimension of the null space for any translations $c^{(k)}$ (and $d^{(k)}$). \square

To validate this theorem, we solved the reconstruction constraints using gradient descent for the two cases $q = 2.0$ and $q = 1.5$, (where $S = 1.0$.) The results are presented in Figure 1.3. In this experiment, no smoothness prior is used and gradient descent is run for a sufficiently long time that the (smooth) initial image does not bias the results. The input in both cases consisted of multiple down-sampled images of the face. Specifically, 1024 randomly translated images were used as input. Exactly the same inputs are used for the two experiments. The only difference is the decimation ratio. (The output for $q = 1.5$ is actually smaller than that for $q = 2.0$ and was interpolated to be the same size for display purposes. This is the reason it appears slightly smoother than (c).)

As can be seen in Figure 1.3, for $q = 2.0$ the (additive) error is approximately a periodic image with period 2 pixels. For $q = 1.5$ the equations are invertible and so a smooth solution is found, even though no smoothness prior was used. For $q = 2.0$, the fact that the problem is not invertible does not have any practical significance. Adequate solutions can be obtained by simply adding a smoothness prior to the reconstruction constraints, as shown in Figure 1.3(c). For $q \gg 2$ the situation is different, however. The rapid rate of increase of the dimension of null space (quadratic in $q \cdot S$) is the root cause of the problems, as will be seen in the next two sections.

3.2 CONDITIONING ANALYSIS

Most linear systems that are close to being not invertible are usually ill-conditioned. It is no surprise then that changing from a square point spread function to an arbitrary blurring function $h^{(k)} = \omega^{(k)} * a^{(k)}$ results in an ill-conditioned system, as we now show in the second part of our analysis:

Theorem 2 *If $\omega^{(k)}(x, y)$ is a function for which $\omega^{(k)}(x, y) \geq 0$ for all (x, y) and $\int \int \omega^{(k)}(x, y) dx dy = 1$, then the condition number of the reconstruction constraints (Equations (1.8) and (1.9)) grows at least as fast as $(q \cdot S)^2$.*

Proof: We first prove the theorem for the square point spread function $h^{(k)} = a^{(k)}$ (i.e. for Equations (1.8) and (1.11)) and then generalize. The condition number of a linear operator A can be written as:

$$\text{Cond}(A) = \frac{\sup_{\|\mathbf{x}\|_\infty=1} \|A\mathbf{x}\|_\infty}{\inf_{\|\mathbf{x}\|_\infty=1} \|A\mathbf{x}\|_\infty}. \quad (1.12)$$

It follows from Equations (1.8) and (1.11) that if $x_H(i', j') = 1$ for all (i', j') , then $x_L^{(k)}(i, j) = 1$ for all (i, j) . Hence the numerator in Equation (1.12) is at least 1. Setting $x_H(i', j')$ to be the checkerboard pattern (1 if $i' + j'$ is even, -1 if odd) we find that $|x_L^{(k)}(i, j)| \leq 1/(q \cdot S)^2$ since the integration of the checkerboard over any square in the real plane lies in the range $[-1, 1]$. (Proof omitted.) Hence the denominator is at most $1/(q \cdot S)^2$. The desired result for $h^{(k)} = a^{(k)}$ follows immediately.

For arbitrary point spread functions, note that Equations (1.8) and (1.9) can be combined and then rewritten as:

$$\begin{aligned} x_L^{(k)}(i, j) &= \int_{x_H} \frac{x_H(x, y)}{q^2} \cdot h^{(k)} \left(\frac{1}{q}(x, y) + (c^{(k)}, d^{(k)}) - (i, j) \right) dx dy \\ &= \left(h^{(k)} * \overline{x}_H \right) ((c^{(k)}, d^{(k)}) - (i, j)) \\ &= \left[\omega^{(k)} * \left(a^{(k)} * \overline{x}_H \right) \right] ((c^{(k)}, d^{(k)}) - (i, j)) \end{aligned} \quad (1.13)$$

where we have set $\overline{x}_H(x, y) = x_H(-qx, -qy)$ and changed variables $(x, y) \Rightarrow -\frac{1}{q}(x, y)$. Both of the properties of $x_L^{(k)}$ that we used to prove the result for square point spread functions therefore also hold with $a^{(k)}$ replaced by $h^{(k)} = \omega^{(k)} * a^{(k)}$ using standard properties of the convolution operator. Hence, the desired, more general, result follows immediately from Equation (1.13). \square

This theorem is more general than the previous one because it applies to arbitrary optical blurring functions. On the other hand, it is a weaker result (in some situations) because it only predicts that super-resolution is ill-conditioned (rather than not invertible.) This theorem on its own, therefore, does not entirely explain the poor performance of super-resolution. As we showed in Figure 1.3, problems that are ill-conditioned (or even not invertible, where the

condition number is infinite) can often be solved by simply adding a smoothness prior. (The not invertible super-resolution problem in Figure 1.3(a) is solved in Figure 1.3(c) in this way.) Several researchers have performed conditioning analysis of various forms of super-resolution, including (Elad and Feuer, 1997; Shekarforoush, 1999; Qi and Snyder, 2000). Although useful, none of these results fully explain the drop-off in performance with the decimation ratio q . The weakness of conditioning analysis is that an ill-conditioned system may be ill-conditioned because of a single “almost singular value.” As indicated by the rapid growth in the dimension of the null space in Theorem 1, super-resolution has a large number of “almost singular values” for large q . This is the real cause of the difficulties seen in Figure 1.4, as we now show.

3.3 ANALYSIS OF THE VOLUME OF SOLUTIONS

If we could work with noiseless, real-valued quantities and perform arbitrary precision arithmetic then the fact that the reconstruction constraints are ill-conditioned might not be a problem. In reality, however, images are always intensity discretized (typically to 8-bit values in the range 0–255 grey levels.) There will therefore always be noise in the measurements, even if it is only plus-or-minus half a grey-level. Suppose that $\text{int}[\cdot]$ denotes the operator which takes a real-valued irradiance measurement and turns it into an integer-valued intensity. If we incorporate this quantization into our image formation model, the reconstruction constraints in Equation (1.13) become:

$$x_L^{(k)}(i, j) = \text{int} \left[\int_{x_H} \frac{x_H(x, y)}{q^2} h^{(k)} \left(\frac{1}{q}(x, y) + (c^{(k)}, d^{(k)}) - (i, j) \right) dx dy \right]. \quad (1.14)$$

Suppose also that x_H is a finite size image with n pixels. We then have:

Theorem 3 *The volume of solutions of the intensity discretized reconstruction constraints in Equation (1.14) grows asymptotically at least as fast as $(q \cdot S)^{2 \cdot n}$.*

Proof: First note that the space of solutions is convex since integration is linear. Next note that one solution of Equation (1.14) is the solution of:

$$x_L^{(k)}(i, j) - 0.5 = \int_{x_H} \frac{x_H(x, y)}{q^2} h^{(k)} \left(\frac{1}{q}(x, y) + (c^{(k)}, d^{(k)}) - (i, j) \right) dx dy. \quad (1.15)$$

The definition of the point spread function as $h^{(k)} = \omega^{(k)} * a^{(k)}$ and the properties of the convolution give $0 \leq h^{(k)} \leq 1/S^2$. Therefore, adding $(q \cdot S)^2$ to any pixel in x_H is still a solution since the right hand side of Equation (1.15) increases by at most 1. (The integrand is increased by less than 1 grey-level in the pixel, which only has an area of 1 unit.) The volume of solutions of Equation (1.14) therefore contains an n -dimensional simplex, where the angles at

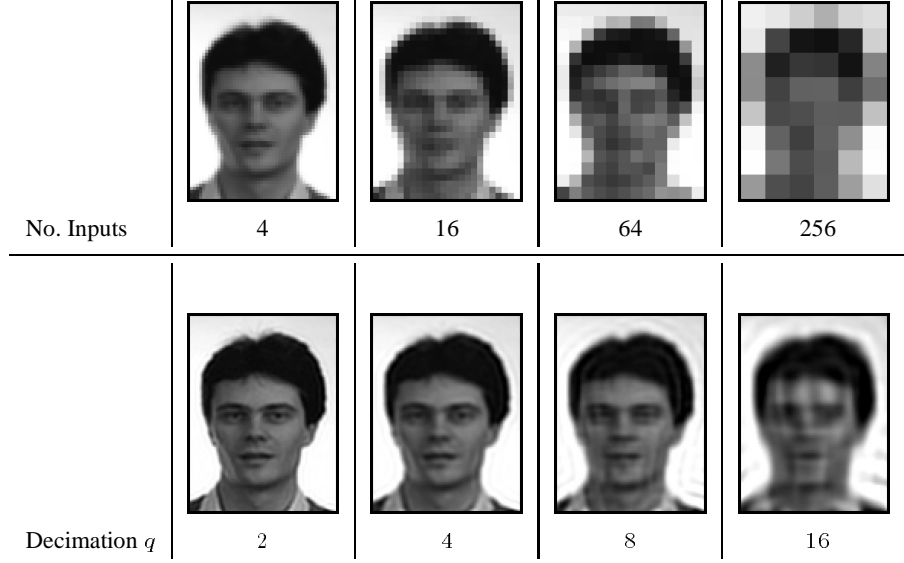


Figure 1.4 Results of the reconstruction-based super-resolution algorithm (Hardie et al., 1997) for various decimation ratios. A high high-resolution image of a face is translated multiple times by random sub-pixel amounts, blurred with a Gaussian, and then down-sampled. (The algorithm is provided with exact knowledge of the point spread function and the sub-pixel translations.) Comparing the images in the right-most column, we see that the algorithm does quite well given the very low resolution of the input. The degradation in performance as the decimation ratio increases from left to right is very dramatic, however.

one vertex are all right-angles, and the sides are all $(q \cdot S)^2$ units long. The volume of such a simplex grows asymptotically like $(q \cdot S)^{2n}$ (treating n as a constant and M and S as variables). The desired result follows. \square

In Figures 1.4 and 1.5 we present results to illustrate Theorems 2 and 3. We took a high resolution image of a face and translated it by random sub-pixel amounts, blurred it with a Gaussian, and then down-sampled it. We repeated this procedure for several decimation ratios; $q = 2, 4, 8$, and 16 . In each case, we generated multiple down-sampled images, each with a different translation. We generated enough images so that there were as many low resolution pixels in total as pixels in the original high resolution image. For example, we generated 4 half size images, 16 quarter size images, and so on. We then applied the algorithms of (Hardie et al., 1997) and (Schultz and Stevenson, 1996).

The results for (Hardie et al., 1997) are shown in the figure. The results for (Schultz and Stevenson, 1996) were very similar and are omitted. We provided the algorithms with exact knowledge of both the point spread function used in the down-sampling and the random sub-pixel translations. Restricting attention to the right-most column of Figure 1.4, the results look very good.

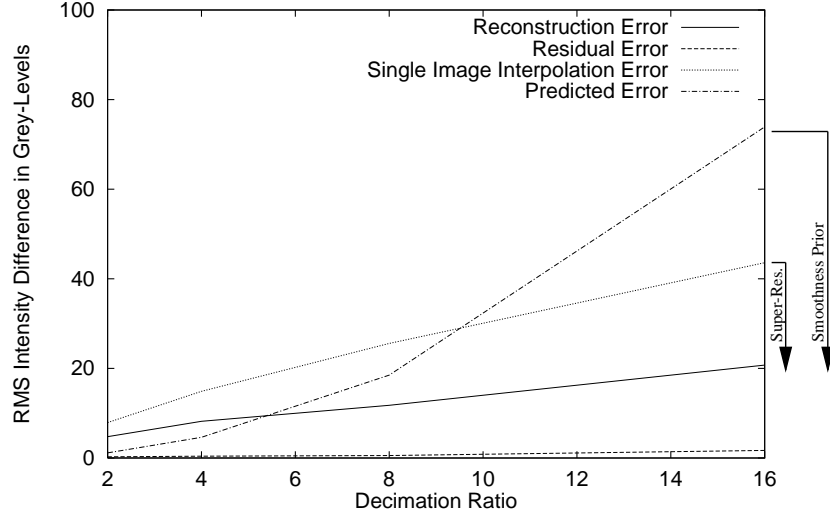


Figure 1.5 An illustration of Theorems 2 and 3 using the same inputs as in Figure 1.4. The reconstruction error is much higher than the residual, as would be expected for an ill-conditioned system. For low decimation ratios, the prior is unnecessary and so the results are worse than predicted. For high decimation ratios, the prior does help, but at the price of smooth results. (See Figure 1.4.) An estimate of the amount of information provided by the reconstruction constraints is given by the improvement of the reconstruction error over the interpolation error. Similarly, the improvement from the predicted error to the reconstruction error is an estimate of the amount of information provided by the smoothness prior. By this measure, the smoothness prior provides more information than the reconstruction constraints for $q = 16$.

The algorithm is able to do a decent job of reconstructing the face from input images which barely resemble faces. On the other hand, the performance gets much worse as the decimation ratio increases (from left to right.)

Our third and final theorem provides the best explanation of these results. For large decimation ratios $q = 8$ and 16 , there is a huge volume of solutions to the discretized reconstruction constraints in Equation (1.14). The smoothness prior which is added to resolve this ambiguity simply ensures that it is one of the overly smooth solutions that is chosen. (Of course, without the prior any solution might be chosen which would generally be even worse.)

Using the same inputs as Figure 1.4, we plot the reconstruction error against the decimation ratio in Figure 1.5; i.e. the difference between the reconstructed high resolution image and the original. We compare this error with the residual error; i.e. the difference between the low resolution inputs and their predictions from the reconstructed high resolution image. As expected for an ill-conditioned system, the reconstruction error is much higher than the residual. We also compare with a rough prediction of the reconstruction error obtained

by multiplying the lower bound on the condition number $(q \cdot S)^2$ by an estimate of the expected residual assuming that the grey-levels are discretized from a uniform distribution. For low decimation ratios, this estimate is an under-estimate because the prior is unnecessary for noise free data; i.e. better results would be obtained without the prior. For high decimation ratios the prediction is an over-estimate because the local smoothness assumption does help the reconstruction (albeit at the expense of overly smooth results.)

We also plot interpolation results in Figure 1.5; i.e. just using the reconstruction constraints for one image (as was proposed, for example, in (Schultz and Stevenson, 1994).) The difference between this curve and the reconstruction error curve is a measure of how much information the reconstruction constraints provide. Similarly, the difference between the predicted error and the reconstruction error is a measure of how much information the smoothness prior provides. For a decimation ratio of 16, we see that the prior provides more information than the super-resolution reconstruction constraints.

4. SUPER-RESOLUTION BY HALLUCINATION

How then is it possible to perform super-resolution with a high decimation ratio without the results looking overly smooth? As we have just shown, the required high-frequency information was lost from the reconstruction constraints when the input images were discretized to 8-bit values. Smoothness priors may help regularize the problem, but cannot replace the missing information.

Our goal in this section is to develop a super-resolution algorithm which uses the information contained in a collection of recognition decisions (in addition to the reconstruction constraints.) Our approach (which we call *hallucination*) is to embed the results of the recognition decisions in a *recognition-based prior* on the solution of the reconstruction constraints, thereby hopefully resolving the inherent ambiguity in their solution.

Our approach is somewhat related to that of (Freeman and Pasztor, 1999) who recently, and independently, proposed a learning framework for low-level vision, one application of which is image interpolation. Besides being applicable to an arbitrary number of images, the other major advantage of our approach is that it uses a prior which is specific to the class of object (in the “class-based” sense of (Riklin-Raviv and Shashua, 1999)) and a set of local recognition decisions. Our algorithm is also related to (Edwards et al., 1998), in which active-appearance model are used for model-based super-resolution.

4.1 BAYESIAN MAP FORMULATION

We use a Bayesian formulation of super-resolution (Cheeseman et al., 1994; Schultz and Stevenson, 1996; Hardie et al., 1997; Elad and Feuer, 1997). In this approach, super-resolution is posed as finding the maximum *a posteriori*

(or MAP) super-resolution image x_H : i.e. estimating $\arg \max_{x_H} P[x_H | x_L^{(k)}]$. Bayes law for this estimation problem is:

$$P[x_H | x_L^{(k)}] = \frac{P[x_L^{(k)} | x_H] \cdot P[x_H]}{P[x_L^{(k)}]}. \quad (1.16)$$

Since $P[x_L^{(k)}]$ is a constant because the images $x_L^{(k)}$ are (known) inputs, and since the logarithm function is a monotonically increasing function, we have:

$$\arg \max_{x_H} P[x_H | x_L^{(k)}] = \arg \min_{x_H} \left(-\ln P[x_L^{(k)} | x_H] - \ln P[x_H] \right). \quad (1.17)$$

The first term in this expression $-\ln P[x_L^{(k)} | x_H]$ is the (negative log) probability of reconstructing the low resolution images $x_L^{(k)}$ given that the super-resolution image is x_H . It is therefore normally set to be a quadratic (i.e. energy) function of the error in the reconstruction constraints:

$$-\ln P[x_L^{(k)} | x_H] = \frac{1}{2\sigma_\eta^2} \sum_{i,j,k} \left[x_L^{(k)}(i,j) - \sum_{i',j'} W^{(k)}(i,j,i',j') \cdot x_H(i',j') \right]^2 \quad (1.18)$$

where $W^{(k)}(i,j,i',j')$ is defined in Equation (1.9). In this expression, we are implicitly assuming that the noise is independently and identically distributed (across both the images and the pixels) and is Gaussian with covariance σ_η^2 .

4.2 RECOGNITION-BASED PRIORS

The second term on the right-hand side of Equation (1.17) is (the negative logarithm of) the prior $-\ln P[x_H]$. Usually the prior is chosen to be a simple smoothness prior (Cheeseman et al., 1994; Schultz and Stevenson, 1996; Hardie et al., 1997; Elad and Feuer, 1997). Instead, we would like it to depend upon the results of a set of recognition decisions. Suppose the outputs of the recognition decisions partition the inputs (i.e. the low resolution input images $x_L^{(k)}$) into a set of subclasses $\{C_{k,l} | l = 1, 2, \dots\}$. We then define a *recognition-based prior* as one that can be written in the following form:

$$P[x_H] = \sum_l P[x_H | x_L^{(k)} \in C_{k,l}] \cdot P[x_L^{(k)} \in C_{k,l}]. \quad (1.19)$$

Essentially there is a separate prior $P[x_H | x_L^{(k)} \in C_{k,l}]$ for each possible partition $C_{k,l}$ of the input space. Once the low resolution input images $x_L^{(k)}$ are available, the various recognition algorithms can be applied, and it can be determined which partition the inputs lie in. The recognition-based prior $P[x_H]$ then reduces to the more specific prior $P[x_H | x_L^{(k)} \in C_{k,l}]$. This prior can be made more powerful than the overall prior because it can be tailored to $C_{k,l}$.

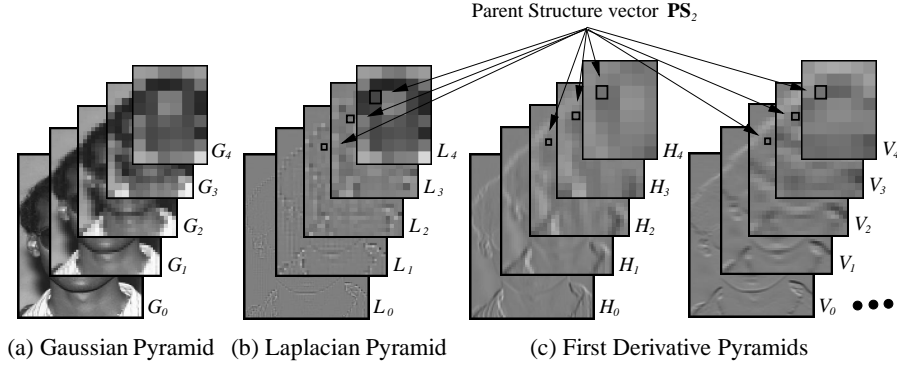


Figure 1.6 The Gaussian, Laplacian, and first derivative pyramids of an image of a face. (We also use two second derivatives but omit them from the figure.) We combine these pyramids into a single multi-valued pyramid, where we store a vector of the Laplacian and the derivatives at each pixel. The Parent Structure vector $\mathbf{PS}_l(i, j)$ of a pixel (i, j) in the l^{th} level of the pyramid consists of the vector of values for that pixel, the vector for its parent in the $l + 1^{\text{th}}$ level, the vector for its parent's parent, etc (De Bonet, 1997). The Parent Structure vector is therefore a high-dimensional vector of derivatives computed at various scales. In our algorithms, recognition means finding the training sample with the most similar Parent Structure vector.

4.3 MULTI-SCALE DERIVATIVE FEATURES

We decided to try to recognize generic local image features (rather than higher level concepts such as ASCII characters) because we want to apply our algorithm to a variety of phenomena. Motivated by (De Bonet, 1997), we also decided to use multi-scale features. In particular, given an image x , we first form its Gaussian pyramid $G_0(x), \dots, G_N(x)$ (Burt, 1980). Afterwards, we also form its Laplacian pyramid $L_0(x), \dots, L_N(x)$ (Burt and Adelson, 1983), the horizontal $H_0(x), \dots, H_N(x)$ and vertical $V_0(x), \dots, V_N(x)$ first derivatives of the Gaussian pyramid, and the horizontal $H_0^2(x), \dots, H_N^2(x)$ and vertical $V_0^2(x), \dots, V_N^2(x)$ second derivatives of the Gaussian pyramid. (See Figure 1.6 for examples of these pyramids.) Finally, we form a feature pyramid:

$$\mathbf{F}_j(x) = \left(L_l(x), H_l(x), V_l(x), H_l^2(x), V_l^2(x) \right) \quad \text{for } l = 0, \dots, N. \quad (1.20)$$

The pyramid $\mathbf{F}_0(x), \dots, \mathbf{F}_N(x)$ is a pyramid where there are 5 values stored at each pixel, the Laplacian and the 4 derivatives.

Then, given a pixel in the low resolution image that we are performing super-resolution on, we want to find (i.e. recognize) a pixel in a collection of training data that is locally “similar.” By similar, we mean that both the Laplacian and the image derivatives are approximately the same, at all scales. To capture this notion, we define the Parent Structure vector (De Bonet, 1997) of a pixel (i, j) in the l^{th} level of the feature pyramid $\mathbf{F}_0(x), \dots, \mathbf{F}_N(x)$ to be:

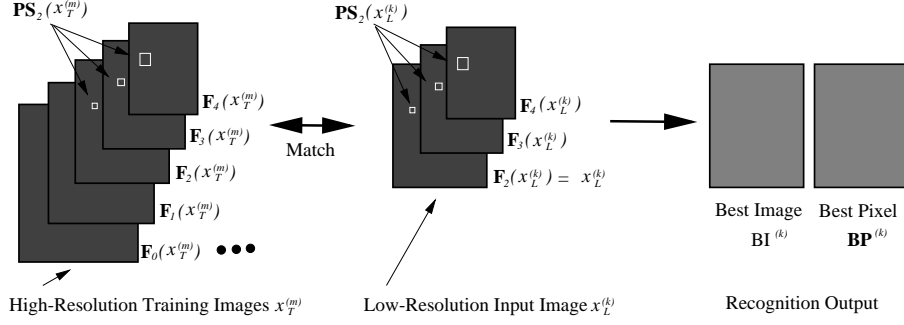


Figure 1.7 We compute the feature pyramids $\mathbf{F}_0(x_T^{(m)}), \dots, \mathbf{F}_N(x_T^{(m)})$ for the training images $x_T^{(m)}$ and the feature pyramids $\mathbf{F}_l(x_L^{(k)}), \dots, \mathbf{F}_N(x_L^{(k)})$ for the low resolution input images $x_L^{(k)}$. For each pixel in the low resolution images, we find (i.e. recognize) the closest matching Parent Structure in the high resolution data. We record and output the best matching image $\mathbf{BI}^{(k)}$ and the pixel location of the best matching Parent Structure $\mathbf{BP}^{(k)}$. Note that these data structures are both defined independently for each pixel (i, j) in each image $x_L^{(k)}$.

$$\mathbf{PS}_l(x)(i, j) =$$

$$\left(\mathbf{F}_l(x)(i, j), \mathbf{F}_{l+1}(x)\left(\left\lfloor \frac{i}{2} \right\rfloor, \left\lfloor \frac{j}{2} \right\rfloor\right), \dots, \mathbf{F}_N(x)\left(\left\lfloor \frac{i}{2^{N-l}} \right\rfloor, \left\lfloor \frac{j}{2^{N-l}} \right\rfloor\right) \right) \quad (1.21)$$

As illustrated in Figure 1.6, the Parent Structure vector at a pixel in the pyramid consists of the feature vector at that pixel, the feature vector of the parent of that pixel, the feature vector of its parent, and so on. Exactly as in (De Bonet, 1997), our notion of two pixels being similar is then that their Parent Structure vectors are approximately the same (measured by some norm.)

4.4 FINDING THE CLOSEST PARENT STRUCTURE

Suppose we have a set of high resolution training images $x_T^{(m)}$ where $m = 1, 2, \dots, M$. We first form the feature pyramids $\mathbf{F}_0(x_T^{(m)}), \dots, \mathbf{F}_N(x_T^{(m)})$. Also suppose that the input image $x_L^{(k)}$ is at a resolution that is $q = 2^l$ times smaller than the training samples. (The image may have to be interpolated to make this ratio exactly a power of 2.) We can then compute the feature pyramid for the input image from level l and upwards $\mathbf{F}_k(x_L^{(k)}), \dots, \mathbf{F}_N(x_L^{(k)})$. Figure 1.7 shows an illustration of this scenario for $l = 2$.

Independently for each pixel (i, j) in the input $x_L^{(k)}$, we compare its Parent Structure vector $\mathbf{PS}_l(x_L^{(k)})(i, j)$ against all of the training Parent Structure vectors at the same level l ; i.e. we compare against $\mathbf{PS}_l(x_T^{(m)})(i', j')$ for all m and for all (i', j') . The best matching image $\mathbf{BI}^{(k)}(i, j) = m$ and the best matching pixel $\mathbf{BP}^{(k)}(i, j) = (i', j')$ are stored as the output of the recognition deci-

sion, independently for each pixel (i, j) in $x_L^{(k)}$. (We found the performance to be largely independent of the distance function used to determine the best matching Parent Structure vector. We actually used a weighted L^2 -norm, giving the derivative components half as much weight as the Laplacian values and reducing the weight by a factor of 2 for each increase in the pyramid level.)

Recognition in our hallucination algorithm therefore means finding the closest matching pixel in the training data in the sense that the Parent Structure vectors of the two pixels are the most similar. This search is, in general, performed over all pixels in all of the images in the training data. If we have frontal images of faces, however, we restrict this search to consider only the corresponding pixels in the training data. In this way, we treat each pixel in the input image differently, depending on its spatial location, similarly to the “class-based” approach of (Riklin-Raviv and Shashua, 1999).

4.5 THE RECOGNITION-BASED GRADIENT PRIOR

For each pixel (i, j) in the input image $x_L^{(k)}$, we have recognized the pixel that is the most similar in the training data, specifically, the pixel $\mathbf{BP}^{(k)}(i, j)$ in the l^{th} level of the pyramid for training image $x_T^{(\mathbf{BI}^{(k)}(i, j))}$. These recognition decisions partition the inputs into a collection of subclasses, as required by the recognition-based prior described in Section 4.2. If we denote the subclasses by $C_{k, \mathbf{BP}^{(k)}, \mathbf{BI}^{(k)}}$ (i.e. using a multi-dimensional index rather than l) Equation (1.19) can be rewritten as:

$$P[x_H] = \sum_{\mathbf{BP}^{(k)}, \mathbf{BI}^{(k)}} P[x_H | x_L^{(k)} \in C_{k, \mathbf{BP}^{(k)}, \mathbf{BI}^{(k)}}] \cdot P[x_L^{(k)} \in C_{k, \mathbf{BP}^{(k)}, \mathbf{BI}^{(k)}}] \quad (1.22)$$

where $P[x_H | x_L^{(k)} \in C_{k, \mathbf{BP}^{(k)}, \mathbf{BI}^{(k)}}]$ is the probability that the super-resolution image is x_H , given that the inputs $x_L^{(k)}$ lie in the subclass that will be recognized to have $\mathbf{BP}^{(k)}$ as the best matching pixel in training image $x_T^{(\mathbf{BI}^{(k)}(i, j))}$.

We now need to define $P[x_H | x_L^{(k)} \in C_{k, \mathbf{BP}^{(k)}, \mathbf{BI}^{(k)}}]$. We decided to make this recognition-based prior a function of the gradient because the base, or average, intensities in the super-resolution image are defined by the reconstruction constraints. It is the high-frequency gradient information that is missing. So, we want to define the prior to encourage the gradient of the super-resolution image to be close to the gradient of the closest matching training samples.

Each low resolution input image $x_L^{(k)}$ has a (different) closest matching (Parent Structure) training sample for each pixel. Moreover, each such Parent Structure corresponds to a number of different pixels in the 0^{th} level of the pyramid, (2^l of them to be precise. See also Figure 1.7.) We therefore impose a separate gradient constraint for each pixel (i, j) in the 0^{th} level of the pyra-

mid (and for each input image $x_L^{(k)}$.) The best matching pixel $\mathbf{BP}^{(k)}$ is only defined on the l^{th} level of the pyramid. For notational convenience, therefore, given a pixel (i, j) on the 0^{th} level of the pyramid, define the best matching pixel on the 0^{th} level of the pyramid to be:

$$\overline{\mathbf{BP}}^{(k)}(i, j) \equiv 2^l * \mathbf{BP}^{(k)}\left(\left\lfloor \frac{i}{2^l} \right\rfloor, \left\lfloor \frac{j}{2^l} \right\rfloor\right) + (i, j) - 2^l * \left(\left\lfloor \frac{i}{2^l} \right\rfloor, \left\lfloor \frac{j}{2^l} \right\rfloor\right). \quad (1.23)$$

Also define the best matching image as $\overline{\mathbf{BI}}^{(k)}(i, j) \equiv \mathbf{BI}^{(k)}\left(\left\lfloor \frac{i}{2^l} \right\rfloor, \left\lfloor \frac{j}{2^l} \right\rfloor\right)$.

If (i, j) is a pixel in the 0^{th} level of the pyramid for image $x_L^{(k)}$, the corresponding pixel in the super-resolution image x_H is $(\mathbf{r}^{(k)})^{-1}\left(\frac{i}{2^l}, \frac{j}{2^l}\right)$. We therefore want to impose the constraint that the first derivatives of x_H at this point should equal the derivatives of the closest matching pixel (Parent Structure) in the training data. Parametric expressions for $H_0(x_H)$ and $V_0(x_H)$ at $(\mathbf{r}^{(k)})^{-1}\left(\frac{i}{2^l}, \frac{j}{2^l}\right)$ can easily be derived as linear functions of the unknown pixels in the high resolution image x_H . We assume that the errors in the gradient values between the recognized training samples and the super-resolution image are independently and identically distributed and moreover that they are Gaussian with covariance σ_{∇}^2 . Therefore: $P[x_H | x_L^{(k)} \in C_{k, \mathbf{BP}^{(k)}, \mathbf{BI}^{(k)}}] =$

$$\frac{1}{2\sigma_{\nabla}^2} \left(\sum_{i,j,k} \left[H_0(x_H)\left((\mathbf{r}^{(k)})^{-1}\left(\frac{i}{2^l}, \frac{j}{2^l}\right)\right) - H_0(x_T^{\overline{\mathbf{BI}}^{(k)}(i,j)})\left(\overline{\mathbf{BP}}^{(k)}(i, j)\right) \right]^2 \right. \\ \left. \sum_{i,j,k} \left[V_0(x_H)\left((\mathbf{r}^{(k)})^{-1}\left(\frac{i}{2^l}, \frac{j}{2^l}\right)\right) - V_0(x_T^{\overline{\mathbf{BI}}^{(k)}(i,j)})\left(\overline{\mathbf{BP}}^{(k)}(i, j)\right) \right]^2 \right). \quad (1.24)$$

This prior enforces the constraints that the gradient of the super-resolution image x_H should equal to the gradient of the best matching training image.

4.6 ALGORITHM PRACTICALITIES

Equations (1.17), (1.18), (1.22), and (1.24) form a high-dimensional linear least squares problem. The constraints in Equation (1.18) are the standard super-resolution reconstruction constraints. Those in Equation (1.24) are the recognition-based prior. The relative weights of these constraints are defined by the noise covariances σ_{η}^2 and σ_{∇}^2 . We assume that the reconstruction constraints are the more reliable ones and so set $\sigma_{\eta}^2 \ll \sigma_{\nabla}^2$.

The number of unknowns is equal to the number of pixels in x_H . Inverting a linear system of such a size can prove problematic. We therefore implemented

a gradient descent algorithm using the standard diagonal approximation to the Hessian (Press et al., 1992) to set the step size in a similar way to (Szeliski and Golland, 1998). Since the error function is quadratic, the algorithm converges to the (single) global minimum without any problem.

4.7 EXPERIMENTAL RESULTS ON HUMAN FACES

Our experiments for human face images were conducted with a subset of the FERET dataset (Philips et al., 1997) consisting of 596 images of 278 individuals (92 women and 186 men). Most people appear twice, with the images taken on the same day under similar illumination conditions, but with different expressions (one expression is neutral, the other typically a smile.) A small number of people appear 4 times, with the images separated by several months.

The images in the FERET dataset are 256×384 pixels, however the area occupied by the face varies considerably, but most of the faces are around 96×128 pixels or larger. In the class-based approach (Riklin-Raviv and Shashua, 1999), the input images (which are all frontal) need to be aligned so that we can assume that the same part of the face appears in roughly the same part of the image every time. This alignment was performed by hand marking the location of 3 points, the centers of the two eyes and the lower tip of the nose. These 3 points define an affine warp (Bergen et al., 1992), which was used to warp the images into a canonical form. These canonical 96×128 pixel images were then used as the training samples $x_T^{(m)}$ where $m = 1, \dots, 596$.

We used a “leave-one-out” methodology to test our algorithm. To test on any particular person, we removed all occurrences of that individual from the training set. We then trained the algorithm on the reduced training set, and tested on the images of the individual that had been removed. Because this process is quite time consuming, we used a test set of 100 randomly selected images of 100 different individuals rather than the entire training set.

Comparison with Existing Super-Resolution Algorithms

We initially restrict attention to the case of enhancing 24×32 pixel images four times to give 96×128 pixel images. Later we will consider the variation in performance with the decimation ratio. We simulate the multiple slightly translated images required for super-resolution using the FERET database by randomly translating the original FERET images multiple times by sub-pixel amounts before down-sampling them to form the low resolution input images.

In our first set of experiments we compare our algorithm with those of (Hardie et al., 1997) and (Schultz and Stevenson, 1996). In Figure 1.8(a) we plot the RMS pixel error against the number of low resolution inputs, computed over the 100 image test set. (We compute the RMS error using the original high resolution image used to synthesize the inputs from.) We also plot results for cubic B-spline interpolation (which only uses one image) for comparison.

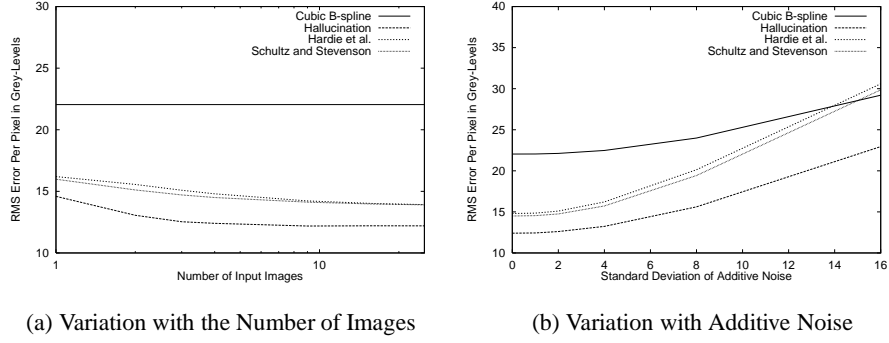


Figure 1.8 A comparison of our hallucination algorithm with the reconstruction-based super-resolution algorithms of (Schultz and Stevenson, 1996) and (Hardie et al., 1997). In (a) we plot the RMS pixel intensity error computed across the 100 image test set against the number of low resolution input images. Our algorithm outperforms the traditional super-resolution algorithms across the entire range. In (b) we vary the amount of additive noise. Again we find that our algorithm does better than the traditional super-resolution algorithms.

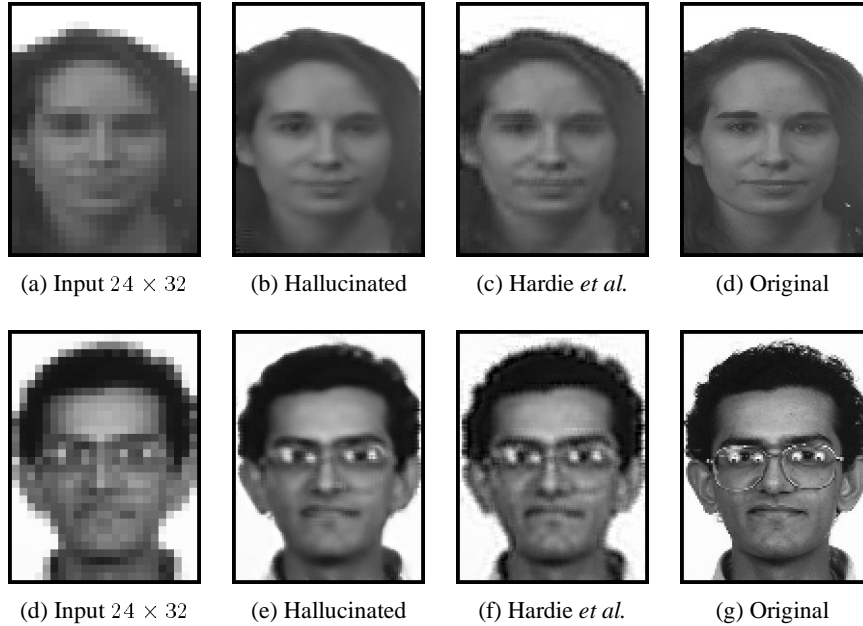


Figure 1.9 The best and worst results in Figure 1.8(a) in terms of the RMS error of the hallucination algorithm for 9 input images. In (a)–(d) we display the results for the best performing image in the 100 image test set. The results for the worst image are presented in (e)–(g). (The results for Schultz and Stevenson are similar to those for Hardie et al. and are omitted.) There is little difference in image quality between the best and worst hallucinated results.

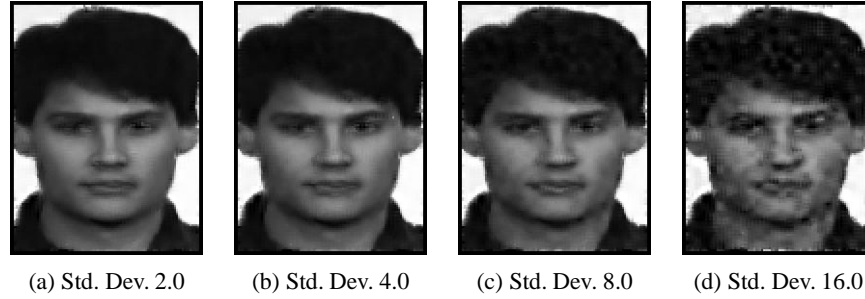


Figure 1.10 An example from Figure 1.8(b) of the variation in the performance of the hallucination algorithm with additive zero-mean, white Gaussian noise. As can be seen, the output is hardly affected until around 4-bits of intensity noise have been added to the inputs. The reason the hallucination algorithm is so robust to noise is that it uses the strong recognition-based face prior to generate smooth, face-like images however noisy the inputs are.

In Figure 1.8(a) we see that our hallucination algorithm does outperform the reconstruction-based super-resolution algorithms, from one input image to 25. The improvement is consistent across the number of input images and is around 20%. The improvement is also largely independent of the actual input. In particular, Figure 1.9 contains the best and worst results obtained across the entire test set in terms of the RMS error of the hallucination algorithm for 9 low resolution inputs. As can be seen, there is little difference between the best results in Figure 1.9(a)–(d) and the worst ones in (e)–(g). Notice, also, how the hallucinated results are a dramatic improvement over the low resolution input, and moreover are visibly sharper than the results for Hardie *et al.*.

Robustness to Additive Intensity Noise

Figure 1.8(b) contains the results of an experiment investigating the robustness of the 3 super-resolution algorithms to additive noise. In this experiment, we added zero-mean, white Gaussian noise to the low resolution images before passing them as inputs to the algorithms. In the figure, the RMS pixel intensity error is plotted against the standard deviation of the additive noise. The results shown are for 4 low resolution input images, and again, the results are an average over the 100 image test set. As might be expected, the performance of all 4 algorithms gets much worse as the standard deviation of the noise increases. The hallucination algorithm and cubic B-spline interpolation, however, seem somewhat more robust than the reconstruction-based super-resolution algorithms. The reason for this increased robustness is probably that the hallucination algorithm always tends to generate smooth, face-like images (because of the strong recognition-based prior) however noisy the inputs are. One example of how the hallucination algorithm degrades with the amount of additive noise is presented in Figure 1.10.



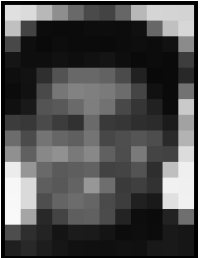
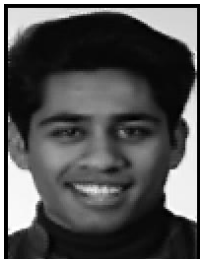


			
Input	48×64	24×32	12×16
			
Output	$\times 2$	$\times 4$	$\times 8$
Reduction in RMS error vs. cubic B-spline	77% (9.2 vs. 11.9)	56% (12.4 vs. 22.2)	57% (19.5 vs. 33.9)

Figure 1.11 The variation in the performance of our hallucination algorithm with the input image size. We see that the algorithm works well down to 12×16 pixel images. It begins to break down for 6×8 pixel images. See (Baker and Kanade, 1999) for examples.

Variation in Performance with the Input Image Size

We do not expect our hallucination algorithm to work for all sizes of input. Once the input gets too small, the recognition decisions will be based on essentially no information. In the limit that the input image is just a single pixel, the algorithm will always generate the same face (for a single input image), but with different average grey levels. We therefore investigated the lowest resolution at which our hallucination algorithm works reasonable well.

In Figure 1.11 we show example results for one face in the test set for 3 different input sizes. (All of the results use just 4 input images.) We see that the algorithm works reasonably well down to 12×16 pixels. (For 6×8 pixel images it produces a face that appears to be a pieced-together combination of a variety of faces. See (Baker and Kanade, 1999) for examples.)

In the last row of Figure 1.11, we give numerical results of the average improvement in the RMS error over cubic B-spline interpolation (computed



Figure 1.12 Selected results for 12×16 pixel images, the smallest input size for which our hallucination algorithm works reliably. (The input consists of only 4 low resolution input images.) Notice how sharp the hallucinated results are. See (Baker and Kanade, 1999) for the results of (Hardie et al., 1997) which are omitted due to lack of space.

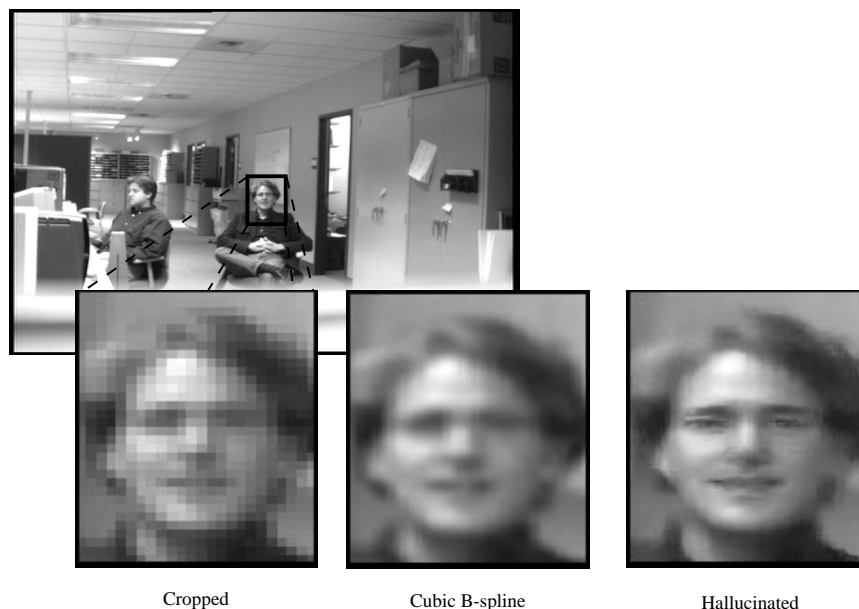


Figure 1.13 Example results on a face not in the FERET dataset. The facial features, such as eyes, nose, and mouth, which are blurred and unclear in the original cropped face, are enhanced and appear much sharper in the hallucinated image. The cubic B-spline result is overly smooth.

over the 100 image test set.) We see that for 24×32 and 12×16 pixel images, the reduction in the error is very dramatic. It is roughly halved. For 48×64 pixel images, the RMS is only cut by about 25% because cubic B-spline does so well it is hard to do much better.

The results for the 12×16 pixel image are excellent, however. (Also see Figure 1.12 which contains several more examples.) The input images are barely recognizable as faces and the facial features such as the eyes, eye-brows, and mouths only consist of a handful of pixels. The outputs, albeit slightly noisy, are clearly recognizable to the human eye. The facial features are also clearly discernible. The hallucinated results are also a huge improvement over (Hardie et al., 1997) and (Schultz and Stevenson, 1996). See (Baker and Kanade, 1999) for these results which are omitted due to a lack of space.

Results on Non-FERET Test Images

In our final experiment for human faces, we tried our algorithm on an image not in the FERET dataset. The results in Figure 1.13 give a big improvement over the cubic B-spline interpolation algorithm. The facial features, such as the eyes, nose, and mouth are all enhanced and appear much sharper in the hallucinated result than either in the input or in the interpolated image.

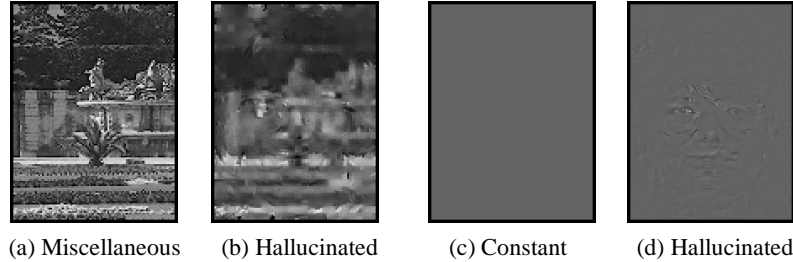


Figure 1.14 The results of applying our algorithm to images not containing faces. (We have omitted the low resolution input and just display the high resolution one.) A face is hallucinated by our algorithm even when none is present, hence the term “hallucination.”

Results on Images Not Containing Faces

In Figure 1.14 we briefly present a few results on images that do not contain faces, even though the algorithm has been trained on the FERET dataset. (Figure 1.14(a) is a miscellaneous image and Figure 1.14(c) is a constant image.) As might be expected, our algorithm hallucinates an outline of a face in both cases, even though there is no face in the input. This is the reason we called our algorithm a “hallucination algorithm.”

4.8 EXPERIMENTAL RESULTS ON TEXT DATA

We also applied our algorithm to text data. In particular, we grabbed an image of an window displaying one page of a letter and used the bit-map as the input. The image was split into disjoint training and test samples. The results are presented in Figures 1.15. The input in Figure 1.15(a) is half the resolution of the original in Figure 1.15(f). The hallucinated result in Figure 1.15(c) is the best reconstruction of the text, both visually and in terms of the RMS intensity error. For example, compare the appearance of the word “was” in the second sentence in Figures 1.15(b)–(f). The hallucination algorithm also has an RMS error of only 24.5 grey levels, compared to over 48.0 for the other algorithms.

5. SUMMARY

In the first half of this chapter we showed that the super-resolution reconstruction constraints provide less and less useful information as the decimation ratio increases. The major cause of this phenomenon is the spatial averaging over the photosensitive area; i.e. the fact that S is non-zero. The underlying reason that there are limits on reconstruction-based super-resolution is therefore the simple fact that CCD sensors must have a non-zero photosensitive area in order to be able to capture a non-zero number of light photons.

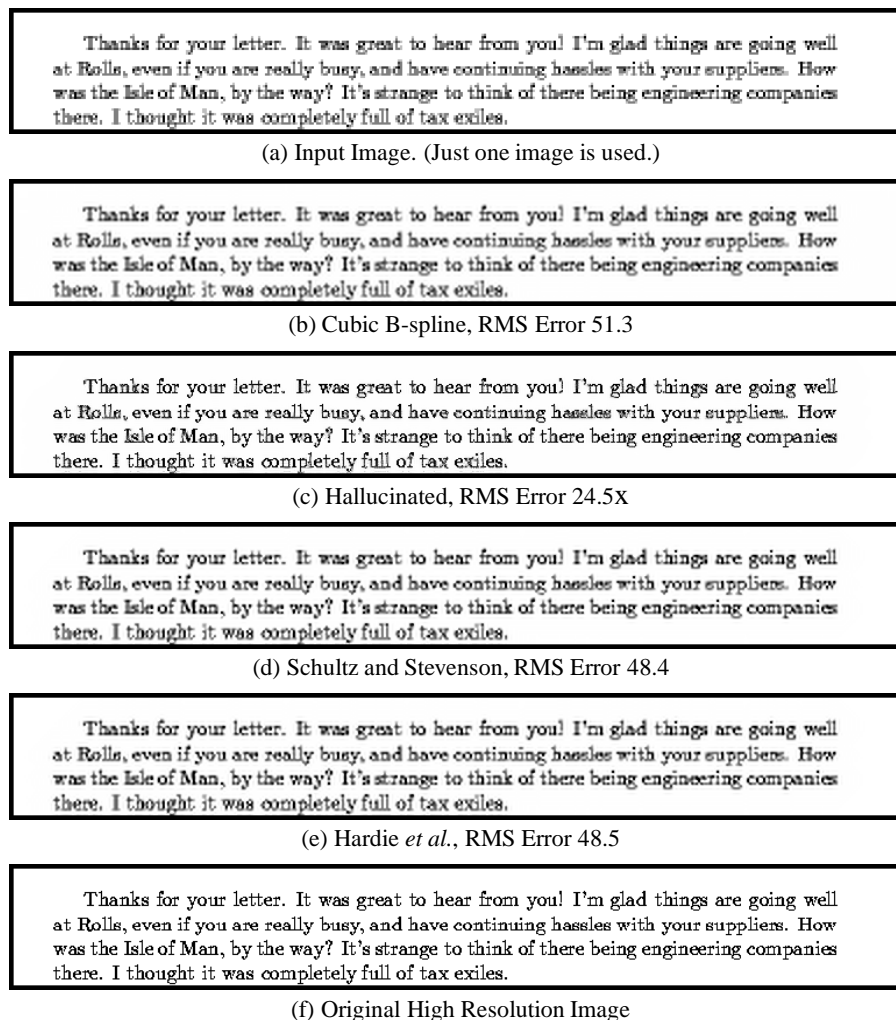


Figure 1.15 The results of enhancing the resolution of a piece of text by a factor of 2. Our hallucination algorithm produces a clear, crisp image using no explicit knowledge that the input contains text. In particular, look at the word “was” in the second sentence. The RMS pixel intensity error is also almost a factor of 2 improvement over the other algorithms.

Our analysis assumes quantized noiseless images; i.e. the intensities are 8-bit values, created by rounding noiseless real-valued numbers. (It is this quantization that causes the loss of information, which when combined with spatial averaging, means that high decimation ratio super-resolution is not possible from the reconstruction constraints.) Without this assumption, however, it might be possible to increase the number of bits per pixel by averaging a collection of quantized noisy images (in an intelligent way). In practice, taking advantage of such information is very difficult. This point also does not affect another outcome of our analysis which was to show that reconstruction-based super-resolution inherently trades-off intensity resolution for spatial resolution.

In the second half of this chapter we showed that recognition processes may provide an additional source of information for super-resolution algorithms. In particular, we developed a “hallucination” algorithm and demonstrated that this algorithm can obtain far better results than existing reconstruction-based super-resolution algorithms, both visually and quantitatively.

6. DISCUSSION

In the past 10-15 years or so much of the research on super-resolution has focused on the reconstruction constraints, and various way of incorporating simple smoothness priors to allow the constraints to be solved. It is a major accomplishment that most of this area is now fairly well understood. This does not mean that super-resolution is now a “solved” problem. As we have shown in this chapter, simply writing down the reconstruction constraints, adding a smoothness prior, and solving the resulting linear system does not necessarily mean that a good solution will be found. There are therefore a number of wide open areas for future super-resolution research:

- One such area involves conducting detailed analysis of the reconstruction constraints, when they provide additional information, how much additional information they provide, and how sensitive the information is to the signal to noise ratio of the input images. Some preliminary work has been done in this area, including (Elad and Feuer, 1997; Shekarforoush, 1999; Qi and Snyder, 2000; Baker and Kanade, 2000b). However, many issues are still a long way from being fully understood.
- Much of the work on super-resolution assumes a fairly simple image formation model. For example, there is almost no modeling of the effect of non-Lambertian surfaces and varying illumination. As a result, many algorithms (including the one described in this chapter) are very sensitive to illumination effects such as shadowing. Although some illumination invariant super-resolution algorithms have been proposed (Chiang and Boulton, 1997), much more work remains to be done.

- In the second half of this chapter we proposed a hallucination algorithm. This algorithm is an instance of a model-based algorithm. Other examples include (Edwards et al., 1998; Freeman and Pasztor, 1999; Baker and Kanade, 2000a). These approaches appear very promising, however the area of model-based super-resolution is in its infancy and a great deal of work remains to be done for completely exploit the idea.
- Other areas which have been largely overlooked include the investigation of applications of super-resolution and the evaluation of the utility of super-resolution algorithms for those applications. There are two types of applications: (1) those where the enhanced image will be shown to a human, and (2) those where the enhanced image will be further processed by a machine. The evaluation of these two types of applications will be very different. The first will need to be done using rigorous subjective studies of how humans can make use of the super-resolution images. The second use of super-resolution is best evaluated in terms of the performance of the algorithms that will actually use the enhanced images. Both of these areas have barely been touched, even though they are vital for proving the utility of super-resolution as a whole.

Acknowledgements

We wish to thank Harry Shum for pointing out the reference (Freeman and Pasztor, 1999), Iain Matthews for pointing out (Edwards et al., 1998), and Henry Schneiderman for suggesting the conditioning analysis in Section 3.2. We would also like to thank a number of people for comments and suggestions, including Terry Boulton, Peter Cheeseman, Michal Irani, Shree Nayar, Steve Seitz, Sundar Vedula, and everyone in the Face Group at CMU. The research described in this chapter was supported by US DOD Grant MDA-904-98-C-A915. A preliminary version of this chapter appeared in the IEEE Conference on Computer Vision and Pattern Recognition (Baker and Kanade, 2000b). More experimental results can be found in (Baker and Kanade, 1999).

References

- Baker, S. and Kanade, T. (1999). Hallucinating faces. Technical Report CMU-RI-TR-99-32, The Robotics Institute, Carnegie Mellon University.
- Baker, S. and Kanade, T. (2000.a). Hallucinating faces. In *Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition*, Grenoble, France.
- Baker, S. and Kanade, T. (2000b). Limits on super-resolution and how to break them. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina.

- Baker, S., Nayar, S., and Murase, H. (1998). Parametric feature detection. *International Journal of Computer Vision*, 27(1):27–50.
- Barbe, D. (1980). *Charge-Coupled Devices*. Springer-Verlag.
- Bascle, B., Blake, A., and Zisserman, A. (1996). Motion deblurring and super-resolution from an image sequence. In *Proceedings of the Fourth European Conference on Computer Vision*, pages 573–581, Cambridge, England.
- Bergen, J. R., Anandan, P., Hanna, K. J., and Hingorani, R. (1992). Hierarchical model-based motion estimation. In *Proceedings of the Second European Conference on Computer Vision*, pages 237–252, Santa Margherita Liguere, Italy.
- Born, M. and Wolf, E. (1965). *Principles of Optics*. Pergamon Press.
- Burt, P. (1980). Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16:20–51.
- Burt, P. and Adelson, E. (1983). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540.
- Cheeseman, P., Kanefsky, B., Kraft, R., Stutz, J., and Hanson, R. (1994). Super-resolved surface reconstruction from multiple images. Technical Report FIA-94-12, NASA Ames Research Center, Moffet Field, CA.
- Chiang, M.-C. and Boulton, T. (1997). Local blur estimation and super-resolution. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pages 821–826, San Juan, Puerto Rico.
- De Bonet, J. (1997). Multiresolution sampling procedure for analysis and synthesis of texture images. In *Computer Graphics Proceedings, Annual Conference Series, (SIGGRAPH '97)*, pages 361–368.
- Dellaert, F., Thrun, S., and Thorpe, C. (1998). Jacobian images of super-resolved texture maps for model-based motion estimation and tracking. In *Proceedings of the Fourth Workshop on Applications of Computer Vision*, pages 2–7, Princeton, NJ.
- Edwards, G., Taylor, C., and Cootes, T. (1998). Learning to identify and track faces in image sequences. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pages 260–265, Nara, Japan.
- Elad, M. and Feuer, A. (1997). Restoration of single super-resolution image from several blurred, noisy and down-sampled measured images. *IEEE Transactions on Image Processing*, 6(12):1646–58.
- Elad, M. and Feuer, A. (1999). Super-resolution reconstruction of image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):817–834.
- Freeman, W. and Pasztor, E. (1999). Learning low-level vision. In *Proceedings of the Seventh International Conference on Computer Vision*, Corfu, Greece.

- Hardie, R., Barnard, K., and Armstrong, E. (1997). Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Transactions on Image Processing*, 6(12):1621–1633.
- Horn, B. (1996). *Robot Vision*. McGraw Hill.
- Irani, M. and Peleg, S. (1991). Improving resolution by image restoration. *Computer Vision, Graphics, and Image Processing*, 53:231–239.
- Peleg, S., Keren, D., and Schweitzer, L. (1987). Improving image resolution using subpixel motion. *Pattern Recognition Letters*, pages 223–226.
- Philips, P., Moon, H., Rauss, P., and Rizvi, S. (1997). The FERET evaluation methodology for face-recognition algorithms. In *CVPR '97*.
- Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipes in C*. Cambridge University Press, second edition.
- Qi, H. and Snyder, Q. (2000). Conditioning analysis of missing data estimation for large sensor arrays. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina.
- Riklin-Raviv, T. and Shashua, A. (1999). The Quotient image: Class based recognition and synthesis under varying illumination. In *Proceedings of the 1999 Conference on Computer Vision and Pattern Recognition*, pages 566–571, Fort Collins, CO.
- Schultz, R. and Stevenson, R. (1994). A Bayesian approach to image expansion for improved definition. *IEEE Transactions on Image Processing*, 3(3):233–242.
- Schultz, R. and Stevenson, R. (1996). Extraction of high-resolution frames from video sequences. *IEEE Transactions on Image Processing*, 5(6):996–1011.
- Shekarforoush, H. (1999). Conditioning bounds for multi-frame super-resolution algorithms. Technical Report CAR-TR-912, Computer Vision Laboratory, Center for Automation Research, University of Maryland.
- Shekarforoush, H., Berthod, M., Zerubia, J., and Werman, M. (1996). Sub-pixel bayesian estimation of albedo and height. *International Journal of Computer Vision*, 19(3):289–300.
- Smelyanskiy, V., Cheeseman, P., Maluf, D., and Morris, R. (2000). Bayesian super-resolved surface reconstruction from images. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina.
- Szeliski, R. and Golland, P. (1998). Stereo matching with transparency and matting. In *Sixth International Conference on Computer Vision (ICCV'98)*, pages 517–524, Bombay.