To Appear in the IEEE Transactions on Pattern Analysis and Machine Intelligence

Limits on Super-Resolution and How to Break Them

Simon Baker and Takeo Kanade

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Nearly all super-resolution algorithms are based on the fundamental constraints that the super-resolution image should generate the low resolution input images when appropriately warped and down-sampled to model the image formation process. (These reconstruction constraints are normally combined with some form of smoothness prior to regularize their solution.) In the first part of this paper, we derive a sequence of analytical results which show that the reconstruction constraints provide less and less useful information as the magnification factor increases. We also validate these results empirically and show that for large enough magnification factors any smoothness prior leads to overly smooth results with very little high-frequency content (however many low resolution input images are used.) In the second part of this paper, we propose a super-resolution algorithm that uses a different kind of constraint, in addition to the reconstruction constraints. The algorithm attempts to recognize local features in the low resolution images and then enhances their resolution in an appropriate manner. We call such a super-resolution algorithm a *hallucination* or *recogstruction* algorithm. We tried our hallucination algorithm on two different datasets, frontal images of faces and printed Roman text. We obtained significantly better results than existing reconstruction-based algorithms, both qualitatively and in terms of RMS pixel error.

Keywords: Super-resolution, analysis of reconstruction constraints, learning, faces, text, hallucination, recogstruction.

1 Introduction

Super-resolution is the process of combining multiple low resolution images to form a higher resolution one. Numerous super-resolution algorithms have been proposed in the literature [39, 32, 51, 33, 29, 31, 53, 30, 34, 37, 13, 9, 35, 47, 49, 7, 38, 26, 21, 15, 23, 18] dating back to the frequency domain approach of Huang and Tsai [28]. Usually it is assumed that there is some (small) relative motion between the camera and the scene, however motionless super-resolution is indeed possible if other imaging parameters (such as the amount of defocus blur) vary instead [21]. If there is relative motion between the camera and the scene, then the first step to super-resolution is to register or align the images; i.e. compute the motion of pixels from one image to the others. The motion fields are typically assumed to take a simple parametric form, such as a translation or a projective warp [8], but instead could be dense optical flow fields [20, 2]. We assume that image registration has already been performed, and concentrate on the second half of super-resolution which consists of fusing the multiple (aligned) low resolution images into a higher resolution one.

The second, fusion step is usually based on the constraints that the super-resolution image, when appropriately warped and down-sampled to take into account the alignment and to model the image formation process, should yield the low resolution input images. These reconstruction constraints have been used by numerous authors since first studied by Peleg *et al.* [39, 29]. The constraints can easily be embedded in a Bayesian framework incorporating a prior on the high resolution image [47, 26, 21][1]. Their solution can be estimated either in batch mode or recursively using a Kalman filter [22, 18]. Several refinements have been proposed, including simultaneously computing structure [13, 49, 50] and removing other degrading effects such as motion blur [7].

In practice the results obtained using these reconstruction-based algorithms are mixed. While the super-resolution images are generally an improvement over the inputs, the high frequency components of the images are generally not reconstructed very well. To illustrate this point, we conducted an experiment, the results of which are included in Figure 1. We took a high resolution image of a face (shown in the top left of the figure) and synthetically translated it by random sub-pixel amounts, blurred it with a Gaussian, and then down-sampled it. We repeated this procedure for several different (linear) down-sampling factors; 2, 4, 8, and 16. In each case, we generated multiple down-sampled images, each with a different random translation. We generated enough

[1]The three papers [47, 26, 21] all use slightly different priors. Roughly speaking though, the priors are all smoothness priors that encourage each pixel to take on the average of its neighbors. In our experience with the first two algorithms, the exact details of the prior make fairly little difference to the super-resolution results. To our knowledge, however, there is no paper that rigorously compares the performance of different priors for super-resolution.

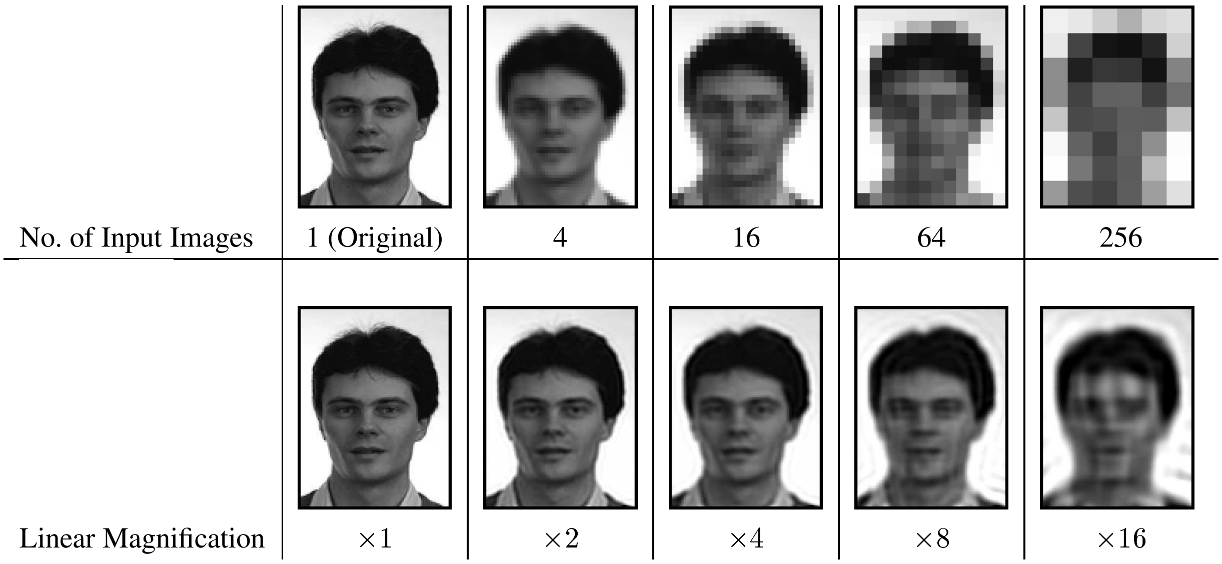


Figure 1: Results of the reconstruction-based super-resolution algorithm [26] for various magnification factors. The original high-resolution image (shown in the top left) is translated multiple times by random sub-pixel amounts, blurred with a Gaussian, and then down-sampled. (The algorithm is provided 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 magnification increases from left to right is very dramatic, however.

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 Schultz and Stevenson [47] and Hardie *et al.* [26]. The results for Hardie *et al.* [26] are shown in the figure. The results for Schultz and Stevenson [47] 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 (although a minor modification of the iterative algorithm in [26] does a very good job of estimating the translation even for the very low resolution 6×8 pixel images in the right-most column.) Restricting attention to the right-most column of Figure 1, the results look very good. 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 magnification increases (from left to right.)

This paper is divided into two parts. In the first part we analyze the super-resolution reconstruction constraints. We derive three results which all show that super-resolution becomes much more difficult as the magnification factor increases. First we show that for square point spread functions (and integer magnification factors), the reconstruction constraints are not even invertible and, moreover, the dimension of the null space grows as a quadratic function of the linear magnification. In the second result we show that even though the constraints are generally invertible for other point spread functions, the condition number also always grows at least as fast as a quadratic. (This second result is proved for a large class of point spread functions including all point spread functions which reasonably model CCD sensors.) This second result, however, does not entirely explain the results shown in Figure 1. It is frequently possible to invert an ill-conditioned problem by simply imposing a smoothness prior on the solution. In our third result we use the fact that the pixels in the input images take values in a finite set (typically integers in the range 0–255) to show that the volume of solutions to the discretized reconstruction constraints grows at an extremely fast rate. This, then, is the underlying reason for the results in Figure 1. For large magnification factors, there are a huge number of solutions to the reconstruction constraints, including numerous very smooth solutions. The smoothness prior that is typically added to resolve the ambiguity in the large solution space simply ensures that it is one of the overly smooth solutions that is chosen. (Strictly, the final solution to the overall problem is only an approximate solution of the reconstruction constraints since both sets of constraints are added as least squares constraints.)

How, then, can high-magnification super-resolution be performed? Our analytical results hold for an arbitrary number of images. Using more low resolution images therefore does not help, at least beyond a point (which in practice is determined by a wide range of factors. See the discussion at the end of the paper for more details.) The additional (8-bit, say) input images simply do not provide any more information because the information was lost when they were quantized to 8-bits. Suppose, however, that the input images contain 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 be perform super-resolution. The text could be reproduced at any resolution by simply rendering it from the recognized text and the definition of the font. In the second half of this paper, we describe a super-resolution algorithm based on this principle, which we call *recognition-based super-resolution* or *hallucination* [1, 3]. More generally, we propose the term *recogstruction* for recognition-based reconstruction techniques. Our hallucination algorithm, however, is based on the recognition of generic local "features" (rather than the characters detected by OCR) so that it can be applied to other phenomena. The recognized local features are used to predict a *recognition-based* prior which replaces the smoothness priors used in existing algorithms such as [13, 47, 26, 21]. We trained our hallucination algorithm separately on both frontal images of faces and computer generated text. We obtained significantly better results than traditional reconstruction-based super-resolution, both visually and in terms of RMS pixel intensity error.

Our algorithm is closely related to the independent work of [25] in which a learning framework for low-level vision was proposed, one application of which is image interpolation. In their paper, Freeman *et al.* learn a prior on the higher resolution image using a belief propagation network. Our algorithm has the advantage of being applicable to an arbitrary number of images. Our algorithm is also closely related to [19] in which the parameters of an "active-appearance" model are used for super-resolution. This algorithm can also be interpreted as having a strong, learned face prior.

The remainder of this paper is organized as follows. We begin in Section 2 by deriving the super-resolution reconstruction constraints, before analyzing them in Section 3. We present our hallucination algorithm (with results) in Section 4. We end with a discussion in Section 5.

2 The Super-Resolution Reconstruction Constraints

Denote the low resolution input images by $\mathrm{Lo}_i(\mathbf{m})$ where $i = 1, \ldots, N$ and $\mathbf{m} = (m, n)$ is a vector in $\mathbf{Z}^2$ containing the (column and row) pixel coordinates. The staring point in the derivation of the reconstruction constraints is then the continuous image formation equation (see Figure 2) [27, 36]:

$$\mathrm{Lo}_i(\mathbf{m}) \;=\; (E_i * \mathrm{PSF}_i)(\mathbf{m}) \;=\; \int_{\mathrm{Lo}_i} E_i(\mathbf{x}) \cdot \mathrm{PSF}_i(\mathbf{x} - \mathbf{m}) \, \mathrm{d}\mathbf{x} \tag{1}$$

where $E_i(\cdot)$ is the continuous irradiance light-field that would have reached the image plane of Lo_i under the pinhole model, $\mathrm{PSF}_i(\cdot)$ is the point spread function of the camera, and $\mathbf{x} = (x, y) \in \mathbf{R}^2$ are coordinates in the image plane of Lo_i (over which the integration is performed.) All that Equation (1) says is that the pixel intensity $\mathrm{Lo}_i(\mathbf{m})$ is the result of convolving the irradiance function $E_i(\cdot)$ with the point-spread function of the camera $\mathrm{PSF}_i(\cdot)$ and then sampling it at the discrete pixel locations $\mathbf{m} = (m, n) \in \mathbf{Z}^2$. (In a more general formulation, $E_i(\cdot)$ may also be a function of both time t and wavelength λ. Equation (1) would then also contain integrations over these two variables as well. We do not model these effects because they do not affect the spatial analysis.)

2.1 Modeling the Point Spread Function

We decompose the point spread function of the camera into two components (see Figure 2):

$$\mathrm{PSF}_i(\mathbf{x}) \;=\; (\omega_i * a_i)(\mathbf{x}) \tag{2}$$

where $\omega_i(\mathbf{x})$ models the blurring caused by the optics and $a_i(\mathbf{x})$ models the spatial integration performed by the CCD sensor [5]. The optical blurring $\omega_i(\cdot)$ is typically further split into a defocus factor that can be approximated by a pill-box function and a diffraction-limited optical transfer

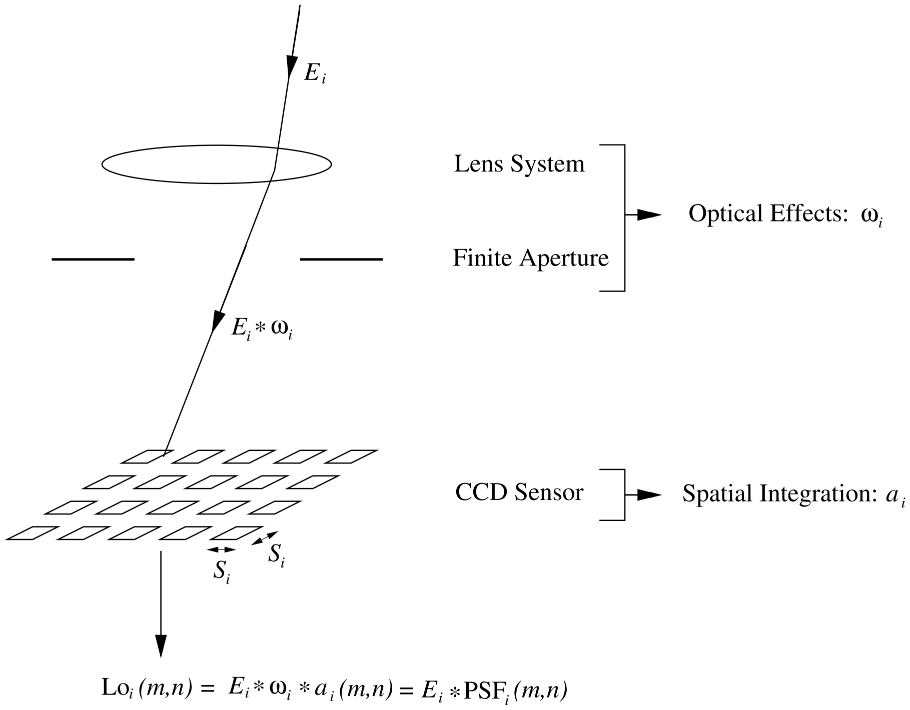


Figure 2: Our image formation model. We assume that the low resolution input images $\mathrm{Lo}_i(\mathbf{m})$ are formed by the convolution of the irradiance $E_i(\cdot)$ with the camera point spread function $\mathrm{PSF}_i(\cdot)$. We model the point spread function itself as the convolution of two terms: (1) ω_i models the optical effects caused by the lens and the finite aperture, and (2) a_i models the spatial integration performed by the CCD sensor.

function that can be modeled by the square of the first-order Bessel function of the first kind [10]. We aim to be as general as possible and so avoid making any assumptions about $\omega_i(\mathbf{x})$. Instead, (most of) our analysis is performed for arbitrary functions $\omega_i(\mathbf{x})$. We do, however, assume a parametric form for a_i. We assume that the the photo-sensitive areas of the CCD pixels are square and uniformly sensitive to light, as in [5, 6]. If the length of the side of the square photosensitive area is S_i, the spatial integration function is then:

$$a_i(\mathbf{x}) \;=\; \begin{cases} \frac{1}{S_i^2} & \text{if } |x| \leq \frac{S_i}{2} \text{ and } |y| \leq \frac{S_i}{2} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

In general the photosensitive area is not the entire pixel since space is need for the circuitry to read out the charge. Therefore S_i is just assumed to take some value in the range $[0, 1]$. Our analysis of the super-resolution problem is then in terms of this parameter (not the inter-pixel distance.)

(Although a detailed model of the point spread function is needed to analyze the limits of super-resolution, typically this modeling is not performed for super-resolution algorithms because the point spread function is a very complex function which depends upon a large number of parameters. In practice, a simple parametric form is assumed for $\mathrm{PSF}_i(\cdot)$, more often than not, that it is Gaussian. The parameter (sigma) is then estimated empirically. Since the point spread function describes "the image of an isolated point object located on a uniformly black background" [36], the parameter(s) can be estimated from the image of a light placed a large distance from the camera.)

2.2 What is Super-Resolution Anyway?

We wish to estimate a super-resolution image $\mathrm{Su}(\mathbf{p})$ where $\mathbf{p} = (p, q) \in \mathbf{Z}^2$ are pixel coordinates. Precisely what does this mean? Let us begin with the coordinate frame of $\mathrm{Su}(\mathbf{p})$. The coordinate frame of Su is typically defined by that of one of the low resolution input images, say $\mathrm{Lo}_1(\mathbf{m})$. If the linear magnification of the super-resolution process is M, the pixels in Su will be M times closer to each other than those in Lo_1. The coordinate frame of Su can therefore be defined in terms of that for Lo_1 using the equation:

$$\mathbf{p} \;=\; \frac{1}{M}\,\mathbf{m}. \tag{4}$$

In the introduction we said that we would assume that the input images Lo_i have been registered with each other. We can therefore assume that they have been registered with the coordinate frame of the super-resolution image Su defined by Equation (4). Then, denote the pixel in image Lo_i that corresponds to pixel $\mathbf{p}$ in Su by $\mathbf{r}_i(\mathbf{p})$. From now on we assume that $\mathbf{r}_i(\cdot)$ is known.

The integration in Equation (1) is performed over the low resolution image plane. Transforming to the super-resolution image plane of Su using the registration $\mathbf{x} = \mathbf{r}_i(\mathbf{z})$ gives:

$$\mathrm{Lo}_i(\mathbf{m}) \;=\; \int_{\mathrm{Su}} E_i(\mathbf{r}_i(\mathbf{z})) \cdot \mathrm{PSF}_i(\mathbf{r}_i(\mathbf{z}) - \mathbf{m}) \cdot \left|\frac{\partial \mathbf{r}_i}{\partial \mathbf{z}}\right| \,\mathrm{d}\mathbf{z} \tag{5}$$

where $\left|\frac{\partial \mathbf{r}_i}{\partial \mathbf{z}}\right|$ is the determinant of the Jacobian of the registration transformation $\mathbf{r}_i(\cdot)$. (Note that we have assumed here that $\mathbf{r}_i$ is invertible. A similar analysis, albeit approximate, can be conducted wherever $\mathbf{r}_i$ is locally invertible by truncating the point spread function.)

Now, $E_i(\mathbf{r}_i(\mathbf{z}))$ is the irradiance that would have reached the image plane of the i^{th} camera under the pinhole model, transformed onto the super-resolution image plane. Assuming the registration $\mathbf{r}_i(\cdot)$ is correct, and that the radiance of the scene does change, $E_i(\mathbf{r}_i(\mathbf{z}))$ should be the same for all $i = 1, \ldots N$ and, moreover, equal to the irradiance that would have reached the super-resolution image plane of Su under a pinhole model. Denoting this value by $E(\mathbf{z})$, we have:

$$\mathrm{Lo}_i(\mathbf{m}) \;=\; \int_{\mathrm{Su}} E(\mathbf{z}) \cdot \mathrm{PSF}_i(\mathbf{r}_i(\mathbf{z}) - \mathbf{m}) \cdot \left|\frac{\partial \mathbf{r}_i}{\partial \mathbf{z}}\right| \,\mathrm{d}\mathbf{z}\,. \tag{6}$$

Given this equation we distinguish two processes:

Deblurring is estimating a representation of $E(\mathbf{z})$ (that is as opposed to estimating $E * \mathrm{PSF}_i$); i.e. deblurring is removing the effects of the convolution with the point spread function $\mathrm{PSF}_i(\cdot)$. Deblurring is independent of whether the representation of $E(\mathbf{z})$ is on a denser grid than that of the input images. The resolution may or may not change during deblurring.

Resolution Enhancement consists of estimating either of the irradiance functions (E or $E_i * \mathrm{PSF}_i$) on a denser grid than that of the input image(s). For example, enhancing the resolution by the linear magnification factor M consists of estimating the irradiance function on the grid defined by Equation (4). If the number of input images is 1, resolution enhancement is known as *interpolation*. If there is more than one input image, resolution enhancement is known as *super-resolution*. Resolution is therefore synonymous with pixel grid density.

In this paper we study the most general case; i.e. the combination of super-resolution and deblurring. We estimate $\mathrm{Su}(\mathbf{p})$, a representation of $E(\mathbf{z})$ on the grid defined by Equation (4).

2.3 Representing Continuous Images

In order to proceed we need to specify which continuous function $E(\mathbf{z})$ is represented by the discrete image $\mathrm{Su}(\mathbf{p})$. The simplest case is that $\mathrm{Su}(\mathbf{p})$ represents the piecewise constant function:

$$E(\mathbf{z}) \; = \; \mathrm{Su}(\mathbf{p}) \tag{7}$$

for all $\mathbf{z} \in (p-0.5, p+0.5] \times (q-0.5, q+0.5]$ and where $\mathbf{p} = (p, q) \in \mathbf{Z}^2$ are the coordinates of a pixel in Su. Then, Equation (5) can be rearranged to give:

$$\mathrm{Lo}_i(\mathbf{m}) \; = \; \sum_{\mathbf{p}} \mathrm{Su}(\mathbf{p}) \cdot \int_{\mathbf{p}} \mathrm{PSF}_i(\mathbf{r}_i(\mathbf{z}) - \mathbf{m}) \cdot \left| \frac{\partial \mathbf{r}_i}{\partial \mathbf{z}} \right| \, \mathrm{d}\mathbf{z} \tag{8}$$

where the integration is performed over the pixel $\mathbf{p}$; i.e. over $(p-0.5, p+0.5] \times (q-0.5, q+0.5]$. The super-resolution reconstruction constraints are therefore:

$$\mathrm{Lo}_i(\mathbf{m}) \; = \; \sum_{\mathbf{p}} \mathrm{W}_i\left(\mathbf{m}, \mathbf{p}\right) \cdot \mathrm{Su}(\mathbf{p}) \quad \text{where} \quad \mathrm{W}_i\left(\mathbf{m}, \mathbf{p}\right) \; = \; \int_{\mathbf{p}} \mathrm{PSF}_i(\mathbf{r}_i(\mathbf{z}) - \mathbf{m}) \cdot \left| \frac{\partial \mathbf{r}_i}{\partial \mathbf{z}} \right| \, \mathrm{d}\mathbf{z}. \tag{9}$$

for $i = 1, \ldots N$; i.e. a set of linear constraints on the unknown super-resolution pixels $\mathrm{Su}(\mathbf{p})$, in terms of the known low resolution pixels $\mathrm{Lo}_i(\mathbf{m})$. The constant coefficients $W_i(\mathbf{m}, \mathbf{p})$ depend on both the point spread function $\mathrm{PSF}_i(\cdot)$ and on the registration $\mathbf{r}_i(\cdot)$. (Similar derivations can be performed for other representations of $E(\mathbf{z})$, such as piecewise linear or quadratic ones [14].)

3 Analysis of the Reconstruction Constraints

We now analyze the super-resolution reconstruction constraints defined by Equation (9). As can be seen, the equations depend upon two imaging properties: (1) the point spread function $\mathrm{PSF}_i(\cdot)$, and (2) the registration $\mathbf{r}_i(\cdot)$. 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 $\mathbf{r}_i(\cdot)$ 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

We assume that the point spread function is the same for all of the images Lo_i and takes the form:

$$\mathrm{PSF}_i(\mathbf{x}) \;=\; (\omega_i * a_i)(\mathbf{x}) \qquad \text{where} \qquad a_i(\mathbf{x}) \;=\; \begin{cases} \frac{1}{S^2} & \text{if } |x| \leq \frac{S}{2} \text{ and } |y| \leq \frac{S}{2} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

In particular, we assume that the width of the photosensitive area S is the same for all images. In the first part of the analysis, we also assume that $\omega_i(\mathbf{x}) = \delta(\mathbf{x})$, the Dirac delta function. Afterwards we allow $\omega_i(\mathbf{x})$ to be an arbitrary function; i.e. the analysis holds for arbitrary optical blurring.

Assumptions Made About the Registration

To outlaw motions which (effectively) allow the camera to be moved towards the scene, we assume that the registration between each pair of low resolution images is a translation. When combined with the super-resolution coordinate frame, as defined in Equation (4), this assumption means that each registration takes the form:

$$\mathbf{r}_i(\mathbf{z}) \;=\; \frac{1}{M}\mathbf{z} + \mathbf{c}_i \tag{11}$$

where $\mathbf{c}_i = (c_i, d_i) \in \mathbf{R}^2$ is a constant (which is different for each low resolution image Lo_i), and $M > 0$ is the *linear magnification* of the super-resolution problem.

Even given these assumptions, the performance of any super-resolution algorithm will depend upon the number of input images N, the exact values of $\mathbf{c}_i$, and, moreover, how well the algorithm can register the low resolution images to estimate the $\mathbf{c}_i$. Our goal is to show that super-resolution becomes fundamentally more difficult as the linear magnification M increases. We therefore assume that the conditions are as favorable as possible and perform the analysis for an arbitrary number of input images N, with arbitrary values of $\mathbf{c}_i$. Moreover, we assume that the algorithm has estimated these values perfectly. Any results derived under these conditions will only be stronger in practice, where the $\mathbf{c}_i$ might take degenerate values, or might be estimated inaccurately.

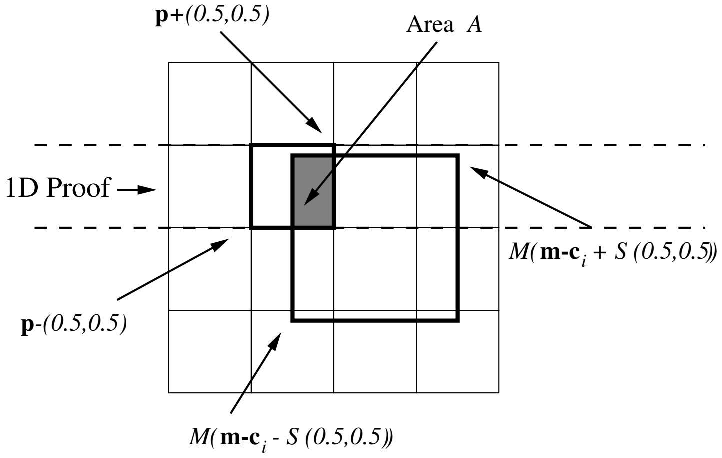


Figure 3: The pixel $\mathbf{p}$ over which the integration is performed in Equation (12) is indicated by the small square at the top left of the figure. The larger square on the bottom right is the region in which $a_i(\cdot)$ is non-zero. Since a_i takes the value $1/S^2$ in this region, the integral in Equation (12) equals A/S^2, where A is the area of the intersection of the two squares. This figure is used to illustrate the proof of Theorem 1.

3.1 Invertibility Analysis for Square Point Spread Functions

We analyze the reconstruction constraints in three different ways. The first analysis is concerned with when the constraints are invertible, and what the rank of the null space is when they are not invertible. In order to get an easily interpretable result, the analysis in this section is performed under the simplified scenario that the optical blurring can be ignored and so $\omega_i(\mathbf{x}) = \delta(\mathbf{x})$, the Dirac delta function. This assumption will be removed in the following two sections, where the analysis is for arbitrary optical blurring models $\omega_i(\mathbf{x})$. Assuming a square point spread function $\mathrm{PSF}_i(\mathbf{x}) = a_i(\mathbf{x})$ (and that the registration $\mathbf{r}_i(\cdot)$ is a translation) Equation (9) simplifies to:

$$\mathrm{Lo}_i(\mathbf{m}) \;=\; \sum_{\mathbf{p}} \mathrm{W}_i\left(\mathbf{m},\mathbf{p}\right)\cdot\mathrm{Su}(\mathbf{p}) \quad \text{where} \quad \mathrm{W}_i\left(\mathbf{m},\mathbf{p}\right) \;=\; \frac{1}{M^2}\cdot\int_{\mathbf{p}} a_i\left(\frac{1}{M}\mathbf{z}+\mathbf{c}_i-\mathbf{m}\right)\,\mathrm{d}\mathbf{z} \quad (12)$$

where the integration is performed over the pixel $\mathbf{p}$; i.e. over $(p-0.5, p+0.5] \times (q-0.5, q+0.5]$. Using the definition of $a_i(\cdot)$ it is easy to see that $\mathrm{W}_i\left(\mathbf{m},\mathbf{p}\right)$ is equal to $1/(M\cdot S)^2$ times the area of the intersection of the two squares in Figure 3. We then have:

Theorem 1 *If $M\cdot S$ is an integer greater than 1, then for all choices of $\mathbf{c}_i$ the set of Equations (12) is not invertible. Moreover, the minimum achievable dimension of the null space is $(M\cdot S-1)^2$. If $M\cdot S$ is not an integer, $\mathbf{c}_i$'s can always be chosen so that the set of Equations (12) are invertible.*

Proof: We provide a proof for 1D images. (See Figure 3.) The extension to 2D is straight-forward.

The null space of Equation (12) is defined by the constraints $\sum_{\mathbf{p}} W_i'(\mathbf{m}, \mathbf{p}) \cdot \mathrm{Su}(\mathbf{p}) = 0$ where $W_i'(\cdot, \cdot)$ is the area of intersection of the two squares in Figure 3. For 1D we just consider one row of the figure. Any element of the null space therefore corresponds to an assignment of values to the small squares in a way that their weighted sum (over the large square) equals zero, where the weights are the areas of intersection with the large square. (To be able to conduct this argument for every pixel in the super-resolution image, we need to assume that the number of pixels in every row and every column of the super-resolution image is greater than $2 \cdot M \cdot S$. This is a minor assumption since it corresponds to assuming that the low resolution images are bigger than 2×2 pixels. This follows from the fact that S is physically constrained to be less than 1.)

Changing $\mathbf{c}_i$ 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 squares. Subtracting these two constraints shows that the left-most square and the right-most square must have the same value. This means that $\mathrm{Su}(\mathbf{p})$ must equal both $\mathrm{Su}(\mathbf{p} + (\lceil M \cdot S \rceil, 0))$ and $\mathrm{Su}(\mathbf{p} + (\lfloor M \cdot S \rfloor, 0))$, if the assignment is to lie in the null space.

If $M \cdot S$ is not an integer (or is 1), this proves that neighboring values of $\mathrm{Su}(\mathbf{p})$ must be equal and hence 0. Therefore, values can always be chosen for the $\mathbf{c}_i$ so that the null space only contains the zero vector; i.e. the linear system is, in general, invertible. (The equivalence of the null space being non-trivial and the linear system being not invertible requires the assumption that the number of pixels in the super-resolution image is finite; i.e. the super-resolution image is bounded.)

If $M \cdot S$ is an integer, this constraint places an upper bound of $M \cdot S - 1$ on the dimension of the null space (since the null space is contained in the set assignments to Su that are periodic with period $M \cdot S$.) This value can also be shown to be a lower bound on the dimension of the null space by the space of period assignments for which $\sum_{i=0}^{M \cdot S - 1} \mathrm{Su}(\mathbf{p} + (i, 0)) = 0$. All of these assignments can easily be seen to lie in the null space (for any choice of the translations $\mathbf{c}_i$). $\square$

To validate this theorem, we solved the reconstruction constraints using gradient descent for the two cases $M = 2.0$ and $M = 1.5$, where $S = 1.0$. The results are presented in Figure 4. In this experiment, no smoothness prior is used and gradient decent is run for a sufficiently long time that the starting image (which is smooth) does not bias the results. The input in both cases consisted of multiple down-sampled images similar to the one at the top of the second column in Figure 1. Specifically, 1024 randomly translated images were used as input. Exactly the same inputs are used for the two experiments. The only difference is the magnification factor in the super-resolution algorithm. The output for $M = 1.5$ is therefore actually smaller than that for $M = 2.0$ (and was enlarged to the same size in Figure 4 for display purposes only.)

(a) $M = 2.0$ (b) $M = 1.5$ (c) $M = 2.0$, with Prior

Figure 4: 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 $M \cdot S$ 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 M is not an integer, the reconstruction constraints are invertible and so a smooth solution is found, even without a prior. (The result for $M = 1.5$ has been interpolated to make it the same size as that for $M = 2.0$.) (c) When a smoothness prior is added to the reconstruction constraints the difficulties seen in (a) disappear.

As can be seen in Figure 4, for $M = 2.0$ the (additive) error is approximately a periodic image with period 2 pixels. For $M = 1.5$ the equations are invertible and so a smooth solution is found, even though no smoothness prior was used. For $M = 2.0$ 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 4(c). For $M \gg 2$ the situation is different, however. As will be shown in the third part of our analysis, it is the rapid rate of increase of the dimension of null space that is the root cause of the problems for large M.

3.2 Conditioning Analysis for Arbitrary Point Spread Functions

Any linear system that is close to being not invertible is usually ill-conditioned. It is no surprise then that changing from a square point spread function to an arbitrary function $\mathrm{PSF}_i = \omega_i * a_i$ results in an ill-conditioned system, as we now show in the second part of our analysis:

Theorem 2 *Suppose* $\omega_i(\mathbf{x})$ *is any function for which* $\omega_i(\mathbf{x}) \geq 0$ *for all* $\mathbf{x}$ *and* $\int \omega_i(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$. *Then, the condition number of the following linear system grows at least as fast as* $(M \cdot S)^2$*:*

$$\mathrm{Lo}_i(\mathbf{m}) \;=\; \sum_{\mathbf{p}} \mathrm{W}_i\left(\mathbf{m},\mathbf{p}\right) \cdot \mathrm{Su}(\mathbf{p}) \quad \text{where} \quad \mathrm{W}_i\left(\mathbf{m},\mathbf{p}\right) \;=\; \frac{1}{M^2} \cdot \int_{\mathbf{p}} \mathrm{PSF}_i\left(\frac{1}{M}\mathbf{z} + \mathbf{c}_i - \mathbf{m}\right)\,\mathrm{d}\mathbf{z} \tag{13}$$

where $\mathrm{PSF}_i = \omega_i * a_i$.

Proof: We first prove the theorem for the square point spread function $a_i(\cdot)$ (i.e. for Equation 12) and then generalize. The condition number of an $m \times n$ matrix A is defined [42] as:

$$\mathrm{Cond}(A) \;=\; \frac{w_1}{w_n} \tag{14}$$

where $w_1 \geq \ldots \geq w_n \geq 0$ are the singular values of A. The one property of singular values that we need is that if $\mathbf{x}$ is any vector:

$$w_1 \;\geq\; \frac{||A\mathbf{x}||_2}{||\mathbf{x}||_2} \;\geq\; w_n \tag{15}$$

where $||\cdot||_2$ is the L2 norm. (This result follows immediately from the SVD $A = USV^{\mathrm{T}}$. The matrices U and V^{T} do not affect the L2 norm of a vector since their columns are orthonormal. Equation (15) clearly holds for S.) It follows immediately that if $\mathbf{x}$ and $\mathbf{y}$ are any two vectors then:

$$\mathrm{Cond}(A) \;\geq\; \frac{||\mathbf{x}||_2 ||A\mathbf{y}||_2}{||\mathbf{y}||_2 ||A\mathbf{x}||_2}. \tag{16}$$

It follows from Equation (12) that if $\mathrm{Su}(\mathbf{p}) = 1$ for all $\mathbf{p}$, then $\mathrm{Lo}_i(\mathbf{m}) = 1$ for all $\mathbf{m}$. Setting $\mathrm{Su}(\mathbf{p}) \;=\; \mathrm{Su}(p, q)$ to be the checkerboard pattern (1 if $p + q$ is even, -1 if odd) we find that $|\mathrm{Lo}_i(\mathbf{m})| \leq 1/(M \cdot S)^2$ since the integration of the checkerboard over any square in the real plane lies in the range $[-1, 1]$. (Proof omitted.) By setting $\mathbf{y}$ to be the first of these vectors and $\mathbf{x}$ the second, it follows immediately from Equation (16) that $\mathrm{Cond}(A) \geq (M \cdot S)^2$.

To generalize to arbitrary point spread functions, note that Equation (13) can be rewritten as:

$$\begin{aligned}
\mathrm{Lo}_i(\mathbf{m}) \;&=\; \int_{\mathrm{Su}} \frac{\mathrm{Su}(\mathbf{z})}{M^2} \cdot \mathrm{PSF}_i \left(\frac{1}{M}\mathbf{z} + \mathbf{c}_i - \mathbf{m} \right) \mathrm{d}\mathbf{z} \\
&=\; \left(\mathrm{PSF}_i * \overline{\mathrm{Su}} \right) (\mathbf{c}_i - \mathbf{m}) \\
&=\; \left[\omega_i * \left(a_i * \overline{\mathrm{Su}} \right) \right] (\mathbf{c}_i - \mathbf{m})
\end{aligned} \tag{17}$$

where we have changed variables $\mathbf{x} = -\frac{1}{M}\mathbf{z}$ and set $\overline{\mathrm{Su}}(\mathrm{x}) = \mathrm{Su}(-M \cdot \mathrm{x})$. The example vectors $\mathbf{x}$ and $\mathbf{y}$ used above can still be used to prove the same result with a_i replaced by $\omega_i * a_i$ using standard properties of the convolution operator: (1) the convolution of a function that takes the value 1 everywhere with a function that is positive and has unit area is also 1 everywhere, and (2) the maximum absolute value of the convolution of a function with a positive function that has unit area cannot increase during the convolution. Hence, the desired (more general) result follows immediately from the last line of Equation (17) and the properties of $\mathbf{x}$ and $\mathbf{y}$ used above. $\square$

This theorem is more general than the previous one because it applies to (essentially) arbitrary point spread 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 4, 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 4(a) is solved in Figure 4(c) in this way. Several researchers have performed conditioning analysis of various forms of super-resolution, including [21, 48, 43]. Although useful, none of these results fully explain the drop-off in performance with the magnification M. 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 magnifications. This is the real cause of the difficulties seen in Figure 1. One way to show this is to derive the volume of solutions, as we now do in the third part of our analysis.

3.3 Volume of Solutions for Arbitrary Point Spread Functions

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 $\mathrm{int}[\cdot]$ 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 then Equation (17) becomes:

$$\mathrm{Lo}_i(\mathbf{m}) = \mathrm{int}\left[\int_{\mathrm{Su}} \frac{\mathrm{Su}(\mathbf{z})}{M^2} \cdot \mathrm{PSF}_i\left(\frac{\mathbf{z}}{M} + \mathbf{c}_i - \mathbf{m}\right) \mathrm{d}\mathbf{z}\right] . \tag{18}$$

Suppose that Su is a fixed size image [2] with n pixels. We then have:

[2] There are at least two ways we could analyze Equation (18). One is to assume that the super-resolution image is of fixed size and fixed resolution. It is the resolution of the inputs that varies. The other way is to assume that the input images are fixed and the resolution of the super-resolution image varies. We chose to analyze Equation (18) in the first case. We assume that the super-resolution image is fixed and that we are given a sequence of super-resolution tasks each with different input images and different pixel sizes. The advantage of this approach is that the quantity that we are trying to estimate stays the same. Moreover, the size of the space of all super-resolution images stays the same.

Theorem 3 *If* $\mathrm{int}[\cdot]$ *is the standard rounding operator which replaces a real number with the nearest integer, then the volume of the set of solutions of Equation (18) grows asymptotically at least as fast as* $(M \cdot S)^{2 \cdot n}$ *(treating* n *as a constant and* M *and* S *as variables.)*

Proof: First note that the space of solutions is convex since integration is a linear operation. Next note that one solution of Equation (18) is the solution to:

$$\mathrm{Lo}_i(\mathbf{m}) - 0.5 = \int_{\mathrm{Su}} \frac{\mathrm{Su}(\mathbf{z})}{M^2} \cdot \mathrm{PSF}_i\left(\frac{\mathbf{z}}{M} + \mathbf{c}_i - \mathbf{m}\right) \mathrm{d}\mathbf{z}. \tag{19}$$

The definition of the point spread function as $\mathrm{PSF}_i = \omega_i * a_i$ and the properties of the convolution give $0 \leq \mathrm{PSF}_i \leq 1/S^2$. Therefore, adding $(M \cdot S)^2$ to any pixel in Su is still a solution since the right hand side of Equation (19) 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 (18) therefore contains an n-dimensional simplex, where the angles at one vertex are all right-angles, and the sides are all $(M \cdot S)^2$ units long. The volume of such a simplex grows asymptotically like $(M \cdot S)^{2n}$ (treating n as a constant and M and S as variables). The desired result follows. $\square$

This third and final theorem provides the best explanation of the super-resolution results presented in Figure 1. For large magnification factors M, there is a huge volume of solutions to the discretized reconstruction constraints in Equation (18). 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. As mentioned in the introduction, the final solution is really only an approximate solution of the reconstruction constraints since both sets of constraints are added as least squares constraints.)

In Figure 5 we present quantitative results to illustrate Theorems 2 and 3. We again used the reconstruction-based algorithm [26]. We verified our implementation in two ways: (1) we checked that for small magnification factors and no prior our implementation does yield (essentially) perfect reconstructions, and (2) for magnifications of 4 we checked that our numerical results are consistent with those in [26]. We also tried the related algorithm of [47] and obtained very similar results.

Using the same inputs as Figure 1, we plot the reconstruction error against the magnification; 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

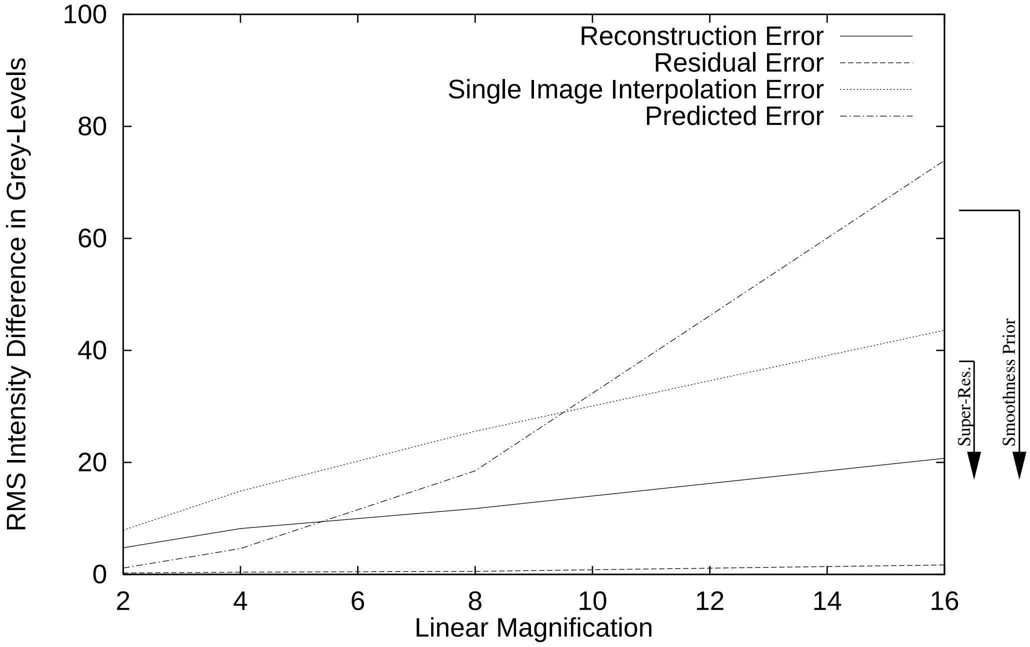


Figure 5: An illustration of Theorems 2 and 3 using the same inputs as in Figure 1. The reconstruction error is much higher than the residual, as would be expected for an ill-conditioned system. For low magnifications, the prior is unnecessary and so the results are worse than predicted. For high magnifications, the prior does help, but at the price of overly smooth results. (See Figure 1.) A rough estimate of the amount of information provided by the reconstruction constraints is given by the improvement of the reconstruction error over the single image 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 a magnification of 16.

$(M \cdot S^2)$ by an estimate of the expected residual assuming that the grey-levels are discretized from a uniform distribution. For low magnification factors, 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 magnifications 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 5; i.e. just using the reconstruction constraints for one image (as was proposed, for example, in [46].) 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 magnification of 16, we see that the prior provides more information than the super-resolution reconstruction constraints. This, then, is an alternative interpretation of why the results in Figure 1 are so smooth.

4 Recognition-Based Super-Resolution or Hallucination

How then is it possible to perform high magnification super-resolution 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. Generic smoothness priors may help regularize the problem, but cannot replace the missing information.

As outlined in the introduction, our goal in this section is to develop a super-resolution algorithm that uses the information contained in a collection of recognition decisions (in addition to the reconstruction constraints.) Our approach is to embed the results of the recognition decisions in a *recognition-based prior* on the solution of the reconstruction constraints, thereby resolving the inherent ambiguity in their solution (see Section 3.3).

4.1 Bayesian MAP Formulation of Super-Resolution

We begin with the (standard) Bayesian formulation of super-resolution [13, 47, 26, 21]. In this approach, super-resolution is posed as finding the maximum *a posteriori* (or MAP) super-resolution image Su: i.e. estimating $\arg\max_{\mathrm{Su}} \Pr[\mathrm{Su} \mid \mathrm{Lo}_i]$. Bayes law for this estimation problem is:

$$\Pr[\mathrm{Su} \mid \mathrm{Lo}_i] \;=\; \frac{\Pr[\mathrm{Lo}_i \mid \mathrm{Su}] \cdot \Pr[\mathrm{Su}]}{\Pr[\mathrm{Lo}_i]}. \tag{20}$$

Since $\Pr[\mathrm{Lo}_i]$ is a constant because the images Lo_i are inputs (and so are "known"), and since the logarithm function is a monotonically increasing function, we have:

$$\arg\max_{\mathrm{Su}} \Pr[\mathrm{Su} \mid \mathrm{Lo}_i] \;=\; \arg\min_{\mathrm{Su}} \left(-\ln \Pr[\mathrm{Lo}_i \mid \mathrm{Su}] - \ln \Pr[\mathrm{Su}]\right). \tag{21}$$

The first term in this expression $-\ln \Pr[\mathrm{Lo}_i \mid \mathrm{Su}]$ is the (negative log) probability of reconstructing the low resolution images Lo_i, given that the super-resolution image is Su. It is therefore normally set to be a quadratic (i.e. energy) function of the error in the reconstruction constraints:

$$-\ln \Pr[\mathrm{Lo}_i \mid \mathrm{Su}] \;=\; \frac{1}{2\sigma_\eta^2} \sum_{\mathbf{m},i} \left[\mathrm{Lo}_i(\mathbf{m}) - \sum_{\mathbf{p}} \mathrm{Su}(\mathbf{p}) \cdot \int_{\mathbf{p}} \mathrm{PSF}_i(\mathbf{r}_i(\mathbf{z}) - \mathbf{m}) \cdot \left| \frac{\partial \mathbf{r}_i}{\partial \mathbf{z}} \right| \, \mathrm{d}\mathbf{z} \right]^2 \tag{22}$$

In using this expression, we are implicitly assuming that the noise is independently and identically distributed (across both the images Lo_i and the pixels $\mathbf{m}$) and is Gaussian with covariance σ_η^2. (All of these assumptions are standard [13, 47, 26, 21].) Minimizing the expression in Equation (22) is then equivalent to finding the (unweighted) least-squares solution of the reconstruction constraints.

4.2 Recognition-Based Priors for Super-Resolution

The second term on the right-hand side of Equation (21) is (the negative logarithm of) the prior $-\ln \Pr[\mathrm{Su}]$. Usually this prior on the super-resolution image is chosen to be a simple smoothness prior [13, 47, 26, 21]. Instead we would like to choose it so that it depends upon a set of recognition decisions. Suppose that the outputs of the recognition decisions partition the set of inputs (i.e. the low resolution input images Lo_i) into a set of subclasses $\{C_{i,k} \mid k = 1, 2, \ldots\}$. We then define a *recognition-based prior* as one that can be written in the following form:

$$\Pr[\mathrm{Su}] \;=\; \sum_k \Pr[\mathrm{Su} \mid \mathrm{Lo}_i \in C_{i,k}] \cdot \Pr[\mathrm{Lo}_i \in C_{i,k}]. \tag{23}$$

Essentially, there is a separate prior $\Pr[\mathrm{Su} \mid \mathrm{Lo}_i \in C_{i,k}]$ for each possible partition $C_{i,k}$. Once the low resolution input images Lo_i are available, the various recognition algorithms can be applied, and it can be determined which partition the inputs lie in. The recognition-based prior $\Pr[\mathrm{Su}]$ then reduces to the more specific prior $\Pr[\mathrm{Su} \mid \mathrm{Lo}_i \in C_{i,k}]$. This prior can be made more powerful than the overall prior $\Pr[\mathrm{Su}]$ because it can be tailored to the (smaller) subset of the input domain $C_{i,k}$.

4.3 Multi-Scale Derivative Features: The Parent Structure

We decided to try to recognize generic local image features (rather than higher level concepts such as human faces or ASCII characters) because we want to apply our algorithm to a variety of phenomena. Motivated by [16, 17] we also decided to use multi-scale features. In particular, given an image I, we first form its Gaussian pyramid $G_0(I), \ldots, G_N(I)$ [11]. Afterwards, we also form its Laplacian pyramid $L_0(I), \ldots, L_N(I)$ [12], the horizontal $H_0(I), \ldots, H_N(I)$ and vertical $V_0(I), \ldots, V_N(I)$ first derivatives of the Gaussian pyramid, and the horizontal $H_0^2(I), \ldots, H_N^2(I)$ and vertical $V_0^2(I), \ldots, V_N^2(I)$ second derivatives of the Gaussian pyramid [1]. (See Figure 6 for examples of these pyramids for an image of a face.) Finally, we form a pyramid of features:

$$\mathbf{F}_j(I) \;=\; \Big(L_j(I), H_j(I), V_j(I), H_j^2(I), V_j^2(I)\Big) \quad \text{for } j = 0, \ldots, N. \tag{24}$$

The pyramid $\mathbf{F}_0(I), \ldots, \mathbf{F}_N(I)$ is a pyramid where there are 5 values stored at each pixel, the Laplacian and the 4 derivatives, rather than the single value typically stored in most pyramids. (The choice of the features in Equation (24) is an instance of the "feature selection" problem. For example, steerable filters [24] could be used instead, or the second derivatives could be dropped if they are too noisy. We found the performance of our algorithms to be largely independent of the choice of features. The selection of the optimal features is outside the scope of this paper.)

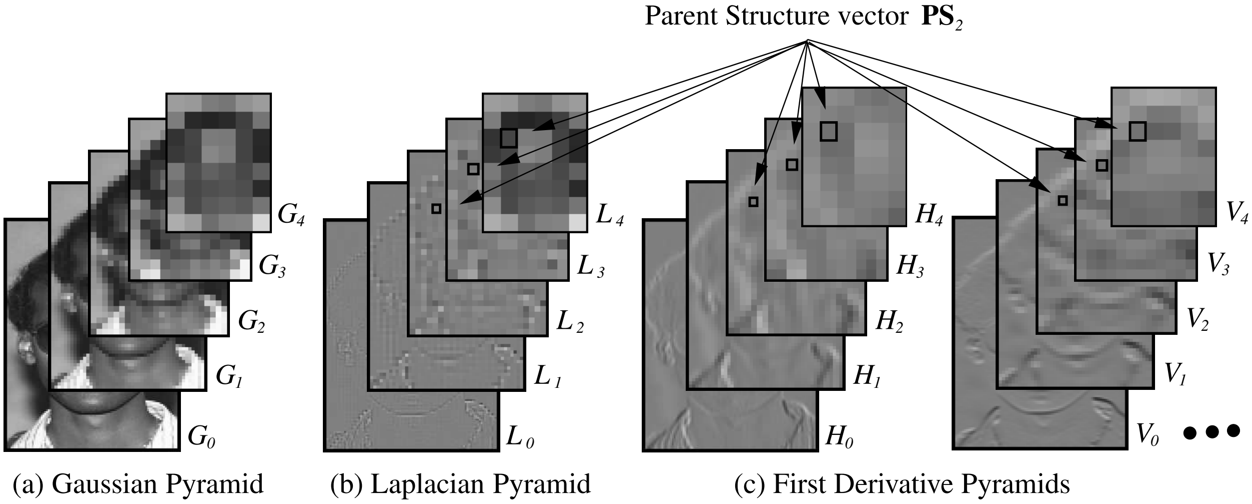


Figure 6: The Gaussian, Laplacian, and first derivative pyramids of an image of a face. (We also use two second derivatives but omit them from the figure.) We combine these pyramids into a single multi-valued pyramid, where we store a vector of the Laplacian and the derivatives at each pixel. The Parent Structure vector $\mathbf{PS}_l(m, n)$ of a pixel (m, n) 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 [16]. 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.

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 [16] of a pixel (m, n) in the l^{th} level of the feature pyramid $\mathbf{F}_0(I), \ldots, \mathbf{F}_N(I)$ to be:

$$\mathbf{PS}_l(I)(m, n) \;=\; \Big(\mathbf{F}_l(I)(m, n), \mathbf{F}_{l+1}(I)(\left\lfloor \frac{m}{2} \right\rfloor, \left\lfloor \frac{n}{2} \right\rfloor), \ldots, \mathbf{F}_N(I)(\left\lfloor \frac{m}{2^{N-l}} \right\rfloor, \left\lfloor \frac{n}{2^{N-l}} \right\rfloor).\Big) \qquad (25)$$

As illustrated in Figure 6, the Parent Structure vector at any particular 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 [16], our notion of two pixels being similar is then that their Parent Structure vectors are approximately the same (as measured by some norm.)

4.4 Recognition as Finding the Nearest-Neighbor Parent Structure

Suppose we have a set of high resolution training images T_j. We can then form all of their feature pyramids $\mathbf{F}_0(T_j), \ldots, \mathbf{F}_N(T_j)$. Also suppose that we are given a low resolution input image Lo_i. Finally, suppose that this image is at a resolution that is $M = 2^k$ times smaller than the training

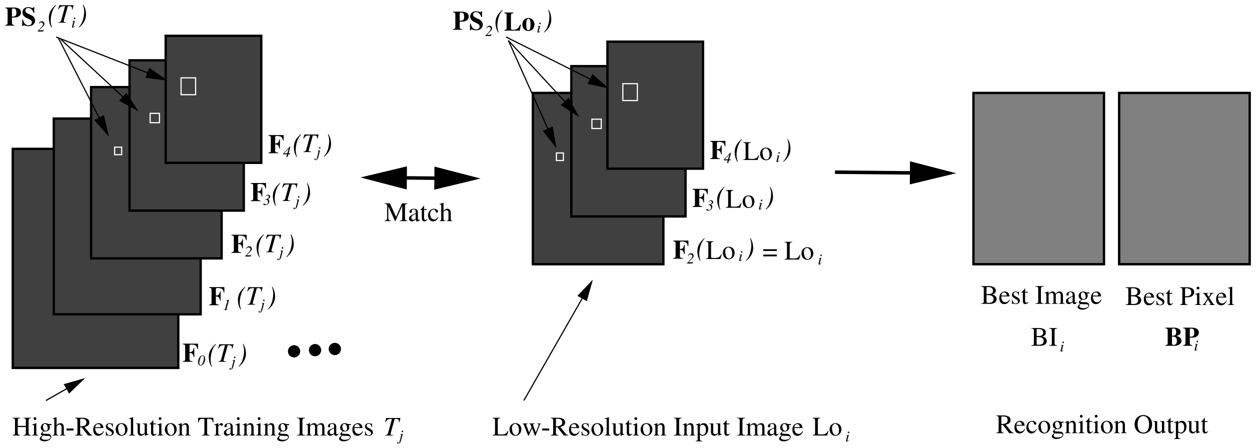


Figure 7: We compute the feature pyramids $\mathbf{F}_0(T_j), \ldots, \mathbf{F}_N(T_j)$ for the training images T_j and the feature pyramid $\mathbf{F}_k(\mathrm{Lo}_i), \ldots, \mathbf{F}_N(\mathrm{Lo}_i)$ for the low resolution input image Lo_i. For each pixel in the low resolution image, we find (i.e. recognize) the closest matching Parent Structure in the high resolution data. We record and output the best matching image BI_i and the pixel location of the best matching Parent Structure $\mathbf{BP}_i$. Note that these data structures are both defined independently for each pixel (m, n) in the images Lo_i.

samples. (The image may have to be interpolated to make this ratio exactly a power of 2. Since the interpolated image is immediately downsampled to create the pyramid it is only the lowest level of the pyramid features that are affected by this interpolation. The overall effect on the prior is therefore very small.) We can then compute the feature pyramid for the input image from level k and upwards $\mathbf{F}_k(\mathrm{Lo}_i), \ldots, \mathbf{F}_N(\mathrm{Lo}_i)$. Figure 7 shows an illustration of this scenario for $k = 2$.

For each pixel (m, n) in the input Lo_i independently, we compare its Parent Structure vector $\mathbf{PS}_k(\mathrm{Lo}_i)(m, n)$ against all of the training Parent Structure vectors at the same level k; i.e. we compare against $\mathbf{PS}_k(T_j)(p, q)$ for all j and for all (p, q). The best matching image $\mathrm{BI}_i(m, n) = j$ and the best matching pixel $\mathbf{BP}_i(m, n) = (p, q)$ are stored as the output of the recognition decision, independently for each pixel (m, n) in Lo_i. (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 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 [44].

4.5 A Recognition-based Gradient Prior

For each pixel (m, n) in the input image Lo_i, we have recognized the pixel that is the most similar in the training data, specifically, the pixel $\mathbf{BP}_i(m, n)$ in the k^{th} level of the pyramid for training image $T_{\mathrm{BI}_i(m,n)}$. These recognition decisions partition the inputs Lo_i into a collection of subclasses, as required by the recognition-based prior described in Section 4.2. If we denote the subclasses by $C_{i,\mathbf{BP}_i,\mathrm{BI}_i}$ (i.e. using a multi-dimensional index rather than k) Equation (23) can be rewritten as:

$$\Pr[\mathrm{Su}] \;=\; \sum_{\mathbf{BP}_i,\mathrm{BI}_i} \Pr[\mathrm{Su} \,|\, \mathrm{Lo}_i \in C_{i,\mathbf{BP}_i,\mathrm{BI}_i}] \cdot \Pr[\mathrm{Lo}_i \in C_{i,\mathbf{BP}_i,\mathrm{BI}_i}] \tag{26}$$

where $\Pr[\mathrm{Su} \,|\, \mathrm{Lo}_i \in C_{i,\mathbf{BP}_i,\mathrm{BI}_i}]$ is the probability that the super-resolution image is Su, given that the input images Lo_i lie in the subclass that will be recognized to have $\mathbf{BP}_i$ as the closest matching pixel in the training image T_{BI_i} (in the k^{th} level of the pyramid.)

We now need to define $\Pr[\mathrm{Su} \,|\, \mathrm{Lo}_i \in C_{i,\mathbf{BP}_i,\mathrm{BI}_i}]$. 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. Specifically, we want to define $\Pr[\mathrm{Su} \,|\, \mathrm{Lo}_i \in C_{i,\mathbf{BP}_i,\mathrm{BI}_i}]$ 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 Lo_i 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^k of them to be precise. See also Figure 7.) We therefore impose a separate gradient constraint for each pixel (m, n) in the 0^{th} level of the pyramid (and for each Lo_i.) Now, the best matching pixel $\mathbf{BP}_i$ is only defined on the k^{th} level of the pyramid. For notational convenience, therefore, given a pixel (m, n) on the 0^{th} level of the pyramid, define the best matching pixel on the 0^{th} level of the pyramid to be:

$$\overline{\mathbf{BP}}_i(m, n) \;\equiv\; 2^k * \mathbf{BP}_i(\left\lfloor \frac{m}{2^k} \right\rfloor, \left\lfloor \frac{n}{2^k} \right\rfloor) + (m, n) - 2^k * (\left\lfloor \frac{m}{2^k} \right\rfloor, \left\lfloor \frac{n}{2^k} \right\rfloor). \tag{27}$$

Also for notational convenience, define the best matching image as $\overline{\mathrm{BI}}_i(m, n) \equiv \mathrm{BI}_i(\left\lfloor \frac{m}{2^k} \right\rfloor, \left\lfloor \frac{n}{2^k} \right\rfloor)$.

If (m, n) is a pixel in the 0^{th} level of the pyramid for image Lo_i, the corresponding pixel in the super-resolution image Su is $\mathbf{r}_i^{-1}(\frac{m}{2^k}, \frac{n}{2^k})$. We therefore want to impose the constraint that the first derivatives of Su at this point should equal the derivatives of the closest matching pixel (Parent Structure) in the training data. Parametric expressions for $H_0(\mathrm{Su})$ and $V_0(\mathrm{Su})$ at $\mathbf{r}_i^{-1}(\frac{m}{2^k}, \frac{n}{2^k})$ can easily be derived as linear functions of the unknown pixels in the high resolution image Su. We assume that the errors in the gradient values between the recognized training samples and the super-resolution image are independently and identically distributed (across both the images Lo_i and the pixels (m, n)) and moreover that they are Gaussian with covariance σ^2_∇. Therefore:

$$\begin{aligned}\Pr[\mathrm{Su} \mid \mathrm{Lo}_i \in C_{i,\mathbf{BP}_i,\mathrm{BI}_i}] \;=\; & \frac{1}{2\sigma^2_\nabla}\sum_{i,m,n}\left[H_0(\mathrm{Su})(\mathbf{r}_i^{-1}(\frac{m}{2^k},\frac{n}{2^k})) - H_0(T_{\overline{\mathrm{BI}}_i(m,n)})(\overline{\mathbf{BP}}_i(m,n))\right]^2 \\ + \; & \frac{1}{2\sigma^2_\nabla}\sum_{i,m,n}\left[V_0(\mathrm{Su})(\mathbf{r}_i^{-1}(\frac{m}{2^k},\frac{n}{2^k})) - V_0(T_{\overline{\mathrm{BI}}_i(m,n)})(\overline{\mathbf{BP}}_i(m,n))\right]^2\end{aligned} \quad (28)$$

This expression enforces the constraints that the gradient of the super-resolution image Su should be equal to the gradient of the best matching training image (separately for each pixel (m, n) in each input image Lo_i.) These constraints are also linear in the unknown pixels of Su.

4.6 Algorithm Practicalities

Equations (21), (22), (26), and (28) form a high-dimensional linear least squares problem. The constraints in Equation (22) are the standard super-resolution reconstruction constraints. Those in Equation (28) are the recognition-based prior. The relative weights of these constraints are defined by the noise covariances σ^2_η and σ^2_∇. We assume that the reconstruction constraints are the more reliable ones and so set $\sigma^2_\eta \ll \sigma^2_\nabla$ (typically $\sigma^2_\nabla = 20 \cdot \sigma^2_\eta$) to make them almost hard constraints.

The number of unknowns in the linear system is equal to the total number of pixels in the super-resolution image Su. Directly inverting a linear system of such size can prove problematic. We therefore implemented a gradient descent algorithm (using a diagonal approximation to the Hessian [42] to set the how step size in a similar way to [52].) 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 faces were conducted with a subset of the FERET dataset [40] consisting of 596 images of 278 individuals (92 women and 186 men). Each person appears between 2 and 4 times. Most people appear twice, with the images taken on the same day under approximately the same illumination conditions, but with different facial expressions (one image is usually neutral, the other typically a smile.) A small number of people appear 4 times, with the images taken over two different days, separated by several months.

The images in the FERET dataset are 256×384 pixels, however the area of the image occupied by the face varies considerably. Most of the faces are around 96×128 pixels or larger. In the class-based approach [44], 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 allows us to obtain the best results. 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 [8], which was used to warp the images into a canonical form. The canonical image is 96×128 pixels with the right eye at $(31, 63)$, the left eye at $(63, 63)$, and the lower tip of the nose at $(47, 83)$. These 96×128 pixel images were then used as the training samples T_j. (In most of our experiments, we also added 8 synthetic variations of each image to the training set by translating the image 8 times, each time by a small amount. This step enhances the performance of our algorithm slightly, although it is not vital to obtain good performance.)

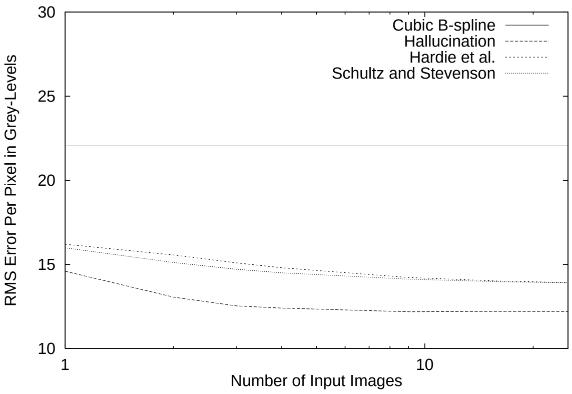
We used a "leave-one-out" methodology to test our algorithm. To test on any particular person, we removed all occurrences of that individual from the training set. We then trained the algorithm on the reduced training set, and tested on the images of the individual that had been removed. Because this process is quite time consuming, we used a test set of 100 images of 100 different individuals rather than the entire training set. The test set was selected at random from the training set. As will be seen, the test set spans both sex and race reasonably well.

4.7.1 Comparison with Existing Super-Resolution Algorithms

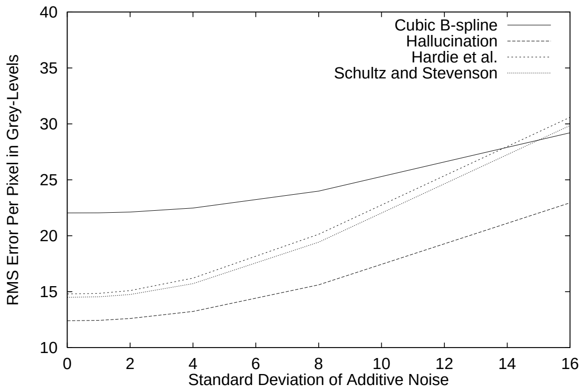
We initially restrict attention to the case of enhancing 24×32 pixel images four times to give 96×128 pixel images. Later we will consider the variation in performance with the magnification factor. We simulate the multiple slightly translated images required for super-resolution using the FERET database 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.* [26] and Schultz and Stevenson [47]. In Figure 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 [41] for comparison. Since this algorithm is an interpolation algorithm, only one image is ever used and so the performance is independent of the number of inputs.

In Figure 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

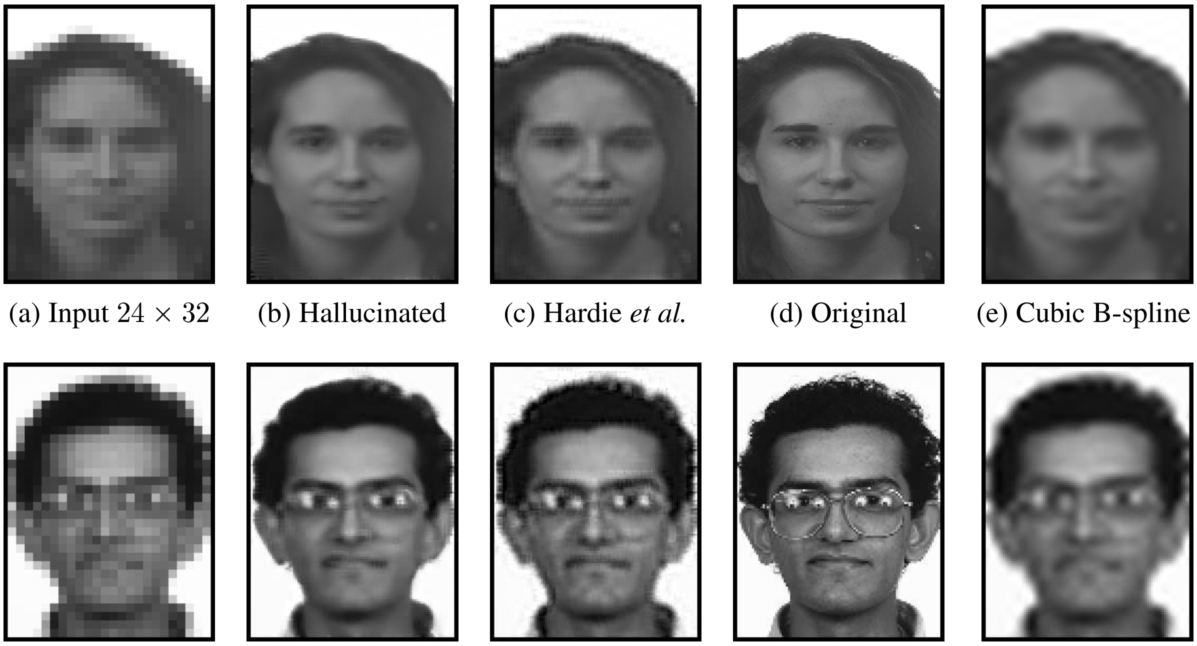


(a) Variation with the Number of Images



(b) Variation with Additive Noise

Figure 8: A comparison of our hallucination algorithm with the reconstruction-based super-resolution algorithms of Schultz and Stevenson [47] and Hardie *et al.* [26]. 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 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, especially as the standard deviation of the noise increases.



(a) Input 24×32 (b) Hallucinated (c) Hardie *et al.* (d) Original (e) Cubic B-spline

(f) Input 24×32 (g) Hallucinated (h) Hardie *et al.* (i) Original (j) Cubic B-spline

Figure 9: The best and worst results in Figure 8(a) in terms of the RMS error of the hallucination algorithm for 9 input images. In (a)–(e) we display the results for the best performing image in the 100 image test set. The results for the worst image are presented in (f)–(j). (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. The hallucinated results are also visibly better than those for Hardie *et al.*

(a) Std. Dev. 1.0 (b) Std. Dev. 2.0 (c) Std. Dev. 4.0 (d) Std. Dev. 8.0 (e) Std. Dev. 16.0

Figure 10: An example from Figure 8(b) of the variation in the performance of the hallucination algorithm with additive zero-mean, white Gaussian noise. The outputs of the hallucination algorithm are shown for various levels of noise. As can be seen, the output is hardly affected until around 4-bits of intensity noise have been added to the inputs. This is because the hallucination algorithm uses the strong recognition-based face prior to generate smooth, face-like images however noisy the input images are. At around 4-bits of noise, the recognition decisions begin to fail and the performance of the algorithm begins to drop off.

number of input images and is around 20%. The improvement is also largely independent of the actual input. In particular, Figure 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 9(a)–(e) and the worst ones in (f)–(j). 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.*.

4.7.2 Robustness to Additive Intensity Noise

Figure 8(b) contains the results of an experiment investigating the robustness of the 3 super-resolution algorithms to additive intensity 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. (The results for cubic B-spline interpolation just use one input image, of course.) As would 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 traditional 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. So long as the recognition decision are

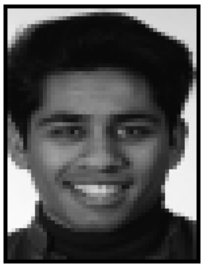
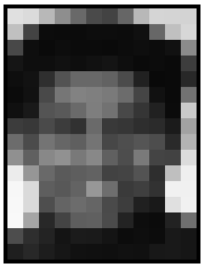
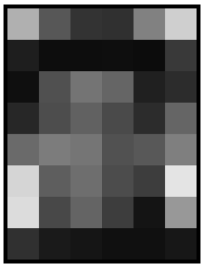
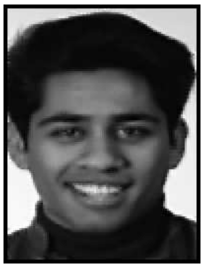
				
Input	48×64	24×32	12×16	6×8
				
Output	$\times 2$	$\times 4$	$\times 8$	$\times 16$
Reduction in RMS error vs. cubic B-spline	77% (9.2 vs. 11.9)	56% (12.4 vs. 22.2)	57% (19.5 vs. 33.9)	73 % (33.3 vs. 45.4)

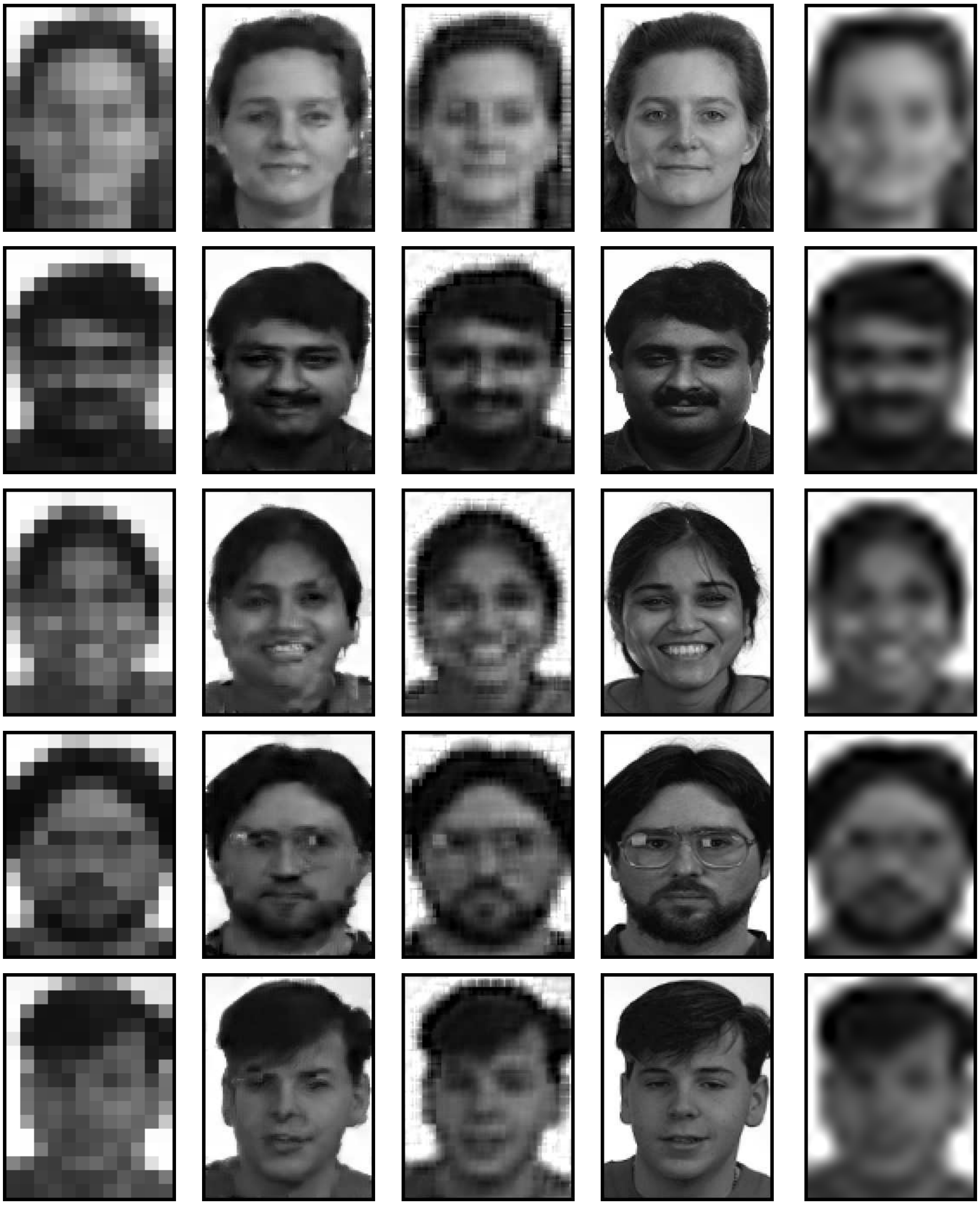
Figure 11: The variation in the performance of our hallucination algorithm with the input image size. From the example in the top two rows, we see that the algorithm works well down to 12×16 pixel images, but not for 6×8 pixel images. (See also Figure 12.) The improvement in the RMS error over the 100 image test set in the last row confirms the fact that the algorithm begins to break down between these two image sizes.

not affected too much, the results should look reasonable. One example of how the output of the hallucination algorithm degrades with the amount of additive noise is presented in Figure 10.

4.7.3 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 11 we show example results for one face in the test set for 4 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, but for 6×8 pixel images it produces a face that appears to be a pieced-together combination of a variety of faces. This is not too surprising because the 6×8 pixel input image is not even clearly an image of a face. (Many face detectors such as [45] use input windows of



(a) Input 12×16 (1 of 4 images) (b) Hallucinated (c) Schultz and Stevenson (d) Original (e) Cubic B-spline

Figure 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 compared to the input and the results for the Schultz and Stevenson [47] algorithm. (The results for Hardie *et al.* [26] are similar to those for Schultz and Stevenson and so are omitted.)

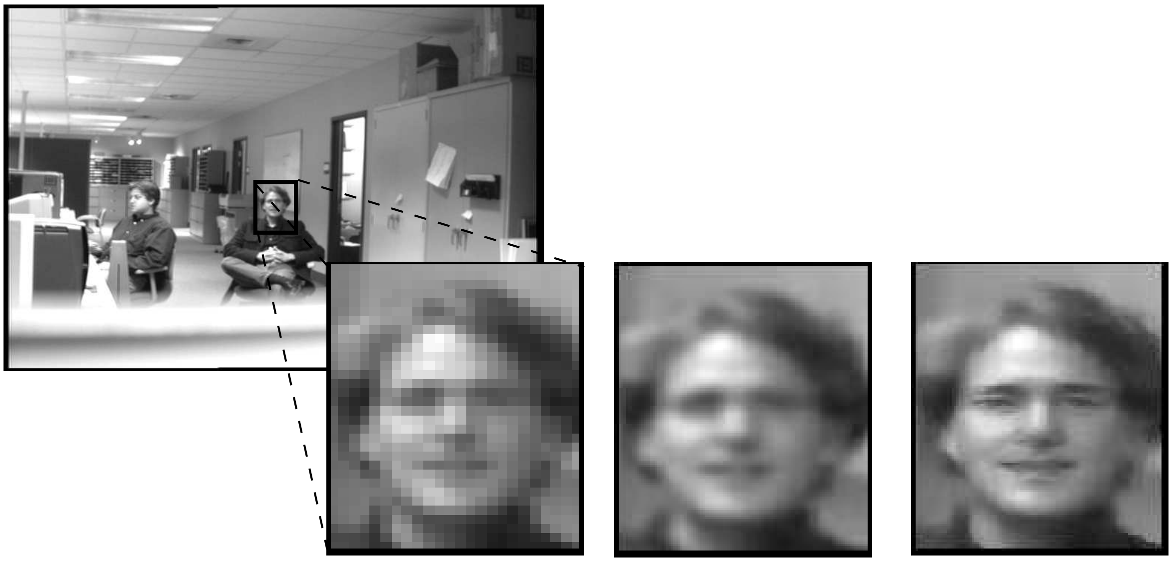


Figure 13: Example results on a single image not in the FERET dataset. The facial features, such as eyes, nose, and mouth, which are blurred and unclear in the original cropped face, are enhanced and appear much sharper in the hallucinated image. In comparison, cubic B-spline interpolation gives overly smooth results.

around 20×20 pixels so it is unlikely that the 6×8 pixel image would be detected as a face.)

In the last row of Figure 11, we give numerical results of the average improvement in the RMS error over cubic B-spline interpolation (computed 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 the other sizes, the results are less impressive, with the RMS error being cut by about 25%. For 6×8 pixel images, the reason is that the hallucination algorithm is being to break down. For 48×64 pixel images, the reason is that cubic B-spline does so well that it is hard to do much better.

The results for the 12×16 pixel image are excellent, however. (Also see Figure 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 Schultz and Stevenson [47].

4.7.4 Results on Non-FERET Test Images

In our final experiment for human faces, we tried our algorithm on a number of images not in the FERET dataset. In Figure 13 we present hallucination results just using a single input image. As can be seen, the hallucinated results are a big improvement over cubic B-spline interpolation. The

Cropped Schultz and Stevenson Hallucinated

Figure 14: Example results on a short video of eight frames. (Only one of the input images and the cropped low resolution face region are shown. The other seven input images are similar except that the camera is slightly translated.) The results of the hallucination algorithm are slightly better than those of the Schultz and Stevenson algorithm, for example, around the eye-brows, around the face contour, and around the hairline. This improvement is only marginal because of the harsh illumination conditions. At present, the performance of our hallucination algorithm is very dependent upon such effects.

facial features, such as the eyes, nose, and mouth are all enhanced and appear much sharper in the hallucinated result than either in the low resolution input or in the interpolated image.

In Figure 14 we present results on a short eight frame video. The face region is marked by hard in the first frame and then tracked over the remainder of the sequence. Our algorithm is compared with that of Schultz and Stevenson. (The results for Hardie *et al.* are similar and so are omitted.) Our algorithm marginally outperforms both reconstruction-based algorithms. In particular, the eye-brows, the face contour, and the hairline are all a little sharper in the hallucinated result. The improvement is quite small, however. This is because the hallucination algorithm is currently very sensitive to illumination conditions and other photometric effects. We are working on making our algorithm more robust to such effects, as well as on several other refinements.

4.7.5 Results on Images Not Containing Faces

In Figure 15 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 15(a) is a random image, (b) is a miscellaneous image, and (c) is a constant image.) As might be expected, our algorithm hallucinates an outline of a face in all three cases, even though there is no face in the input. This is the reason we

(a) Random Hallucinated (b) Misc. Hallucinated (c) Constant Hallucinated

Figure 15: The results of applying our hallucination algorithm to images not containing faces. (We have omitted the low resolution input and have just displayed the original high resolution image.) As is evident, a face is hallucinated by our algorithm even when none is present, hence the term "hallucination algorithm."

called our algorithm a "hallucination algorithm." (The hallucination algorithm naturally performs worse on images that is was not trained for than reconstruction-based algorithms do.)

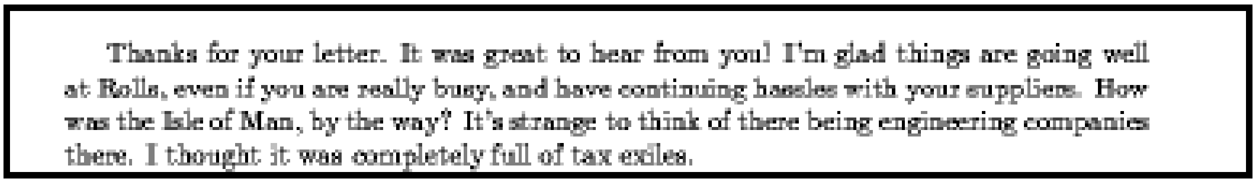
4.8 Experimental Results on Text Data

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 training and test data therefore contain the same font, are at the same scale, and the data is noiseless. The training and test data are not registered in anyway however.) The results are presented in Figures 16. The input in Figure 16(a) is half the resolution of the original in Figure 16(f). The hallucinated result in Figure 16(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 of the text in Figures 16(b)–(f). The hallucination algorithm also has an RMS error of only 24.5 grey levels, compared to over 48.0 for the three other algorithms, almost a factor of two improvement.

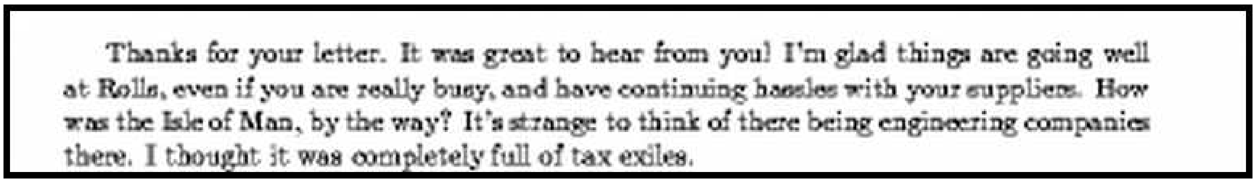
5 Discussion

In the first half of this paper we showed that the super-resolution reconstruction constraints provide less and less useful information as the magnification factor 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 photons of light.

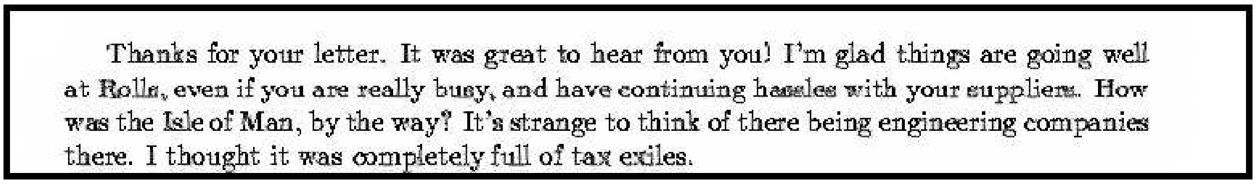
Our analysis assumes quantized noiseless images; i.e. the intensities are 8-bit values, created



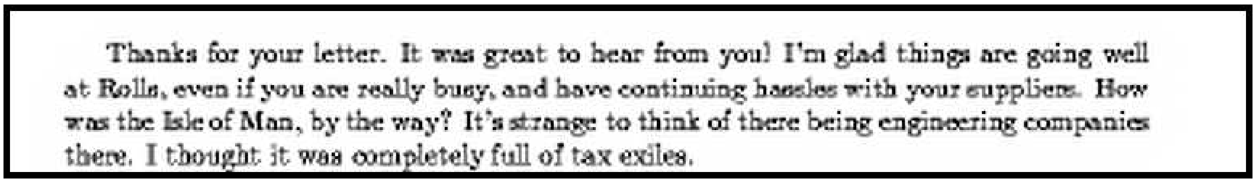
(a) Input Image. (Just one image is used.)



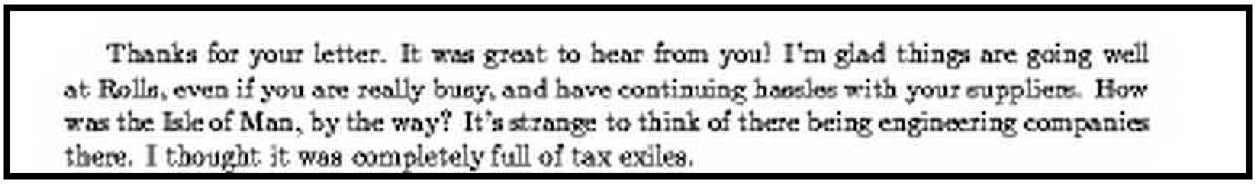
(b) Cubic B-spline, RMS Error 51.3



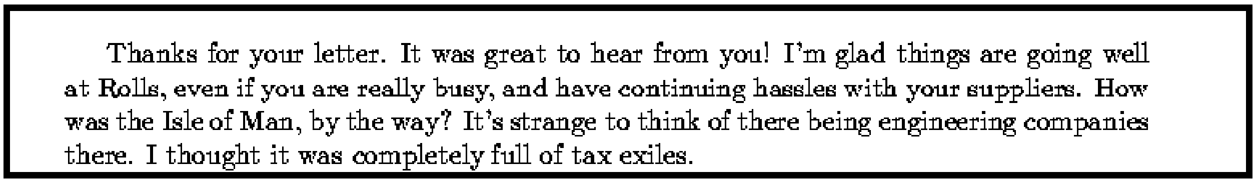
(c) Hallucinated, RMS Error 24.5



(d) Schultz and Stevenson, RMS Error 48.4



(e) Hardie *et al.*, RMS Error 48.5



(f) Original High Resolution Image

Figure 16: The results of enhancing the resolution of a piece of text by a factor of 2. (Just a single input image is used.) 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.

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 magnification 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 paper we showed that recognition processes may provide an additional source of information for super-resolution algorithms. In particular, we developed a "hallucination" algorithm for super-resolution and demonstrated that this algorithm can obtain far better results than existing reconstruction-based super-resolution algorithms, both visually and in terms of RMS pixel intensity error. Similar approaches may aid other (ie. 3D) reconstruction tasks.

At this time, however, our hallucination algorithm is not robust enough to be used on typical surveillance video. Besides integrating it with a 3D head tracker to avoid the need for manual registration and to remove the restriction to frontal faces, the robustness of the algorithm to illumination conditions must be improved. This lack of robustness to illumination can be seen in Figure 14 where the performance of our algorithm on images captured outdoors and in novel illumination conditions results in significantly less improvement over existing reconstruction-based algorithms than that seen in some of our other results. (The most appropriate figure to compare Figure 14 with is Figure 9.) We are currently working on these and other refinements.

The two halves of this paper are related in the following sense. Both halves are concerned with where the information comes from when super-resolution is performed and how strong that information is. The first half investigates how much information is contained in the reconstruction constraints and shows that the information content is fundamentally limited by the dynamic range of the images. The second half demonstrates that strong class-based priors can provide far more information than the simple smoothness priors that are used in existing super-resolution algorithms.

Acknowledgements

We wish to thank Harry Shum for pointing out the work of Freeman and Pasztor [25], Iain Matthews for pointing out the work of Edwards *et al.* [19], and Henry Schneiderman for suggesting we perform the conditioning analysis in Section 3.2. We would also like to thank numerous people for comments and suggestions, including Terry Boult, Peter Cheeseman, Michal Irani, Shree Nayar, Steve Seitz, Sundar Vedula, and everyone in the Face Group at CMU. Finally we would like to thank the anonymous reviewers for their comments and suggestions. The research described in this paper was supported by US DOD Grant MDA-904-98-C-A915. A preliminary version of this paper [4] appeared in June 2000 in the IEEE Conference on Computer Vision and Pattern Recognition. Additional experimental results can be found in the technical report [1].

References

[1] S. Baker and T. Kanade. Hallucinating faces. Technical Report CMU-RI-TR-99-32, The Robotics Institute, Carnegie Mellon University, 1999.

[2] S. Baker and T. Kanade. Super-resolution optical flow. Technical Report CMU-RI-TR-99-36, The Robotics Institute, Carnegie Mellon University, 1999.

[3] S. Baker and T. Kanade. Hallucinating faces. In *Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 2000.

[4] S. Baker and T. Kanade. 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, 2000.

[5] S. Baker, S.K. Nayar, and H. Murase. Parametric feature detection. *International Journal of Computer Vision*, 27(1):27–50, 1998.

[6] D.F. Barbe. *Charge-Coupled Devices*. Springer-Verlag, 1980.

[7] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *Proceedings of the Fourth European Conference on Computer Vision*, pages 573–581, Cambridge, England, 1996.

[8] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the Second European Conference on Computer Vision*, pages 237–252, Santa Margherita Liguere, Italy, 1992.

[9] M. Berthod, H. Shekarforoush, M. Werman, and J. Zerubia. Reconstruction of high resolution 3D visual information. In *Proceedings of the 1994 Conference on Computer Vision and Pattern Recognition*, pages 654–657, Seattle, WA, 1994.

[10] M. Born and E. Wolf. *Principles of Optics*. Permagon Press, 1965.

[11] P.J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16:20–51, 1980.

[12] P.J. Burt and E.H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communiations*, 31(4):532–540, 1983.

[13] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson. Super-resolved surface reconstruction from multiple images. Technical Report FIA-94-12, NASA Ames Research Center, Moffet Field, CA, 1994.

[14] M.-C. Chiang and T.E. Boult. Imaging-consistent super-resolution. In *Proceedings of the 1997 DARPA Image Understanding Workshop*, 1997.

[15] M.-C. Chiang and T.E. Boult. 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, 1997.

[16] J.S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Computer Graphics Proceedings, Annual Conference Series, (SIGGRAPH '97)*, pages 361–368, 1997.

[17] J.S. De Bonet and P. Viola. Texture recognition using a non-parametric multi-scale statistical model. In *Proceedings of the 1998 Conference on Computer Vision and Pattern Recognition*, pages 641–647, Santa Barbara, CA, 1998.

[18] F. Dellaert, S. Thrun, and C. Thorpe. 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, 1998.

[19] G.J. Edwards, C.J. Taylor, and T.F. Cootes. 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, 1998.

[20] M. Elad. *Super-Resolution Reconstruction of Image Sequences - Adaptive Filtering Approach*. PhD thesis, The Technion - Israel Institute of Technology, Haifa, Israel, 1996.

[21] M. Elad and A. Feuer. Restoration of single super-resolution image from several blurred, noisy and down-sampled measured images. *IEEE Transactions on Image Processing*, 6(12):1646–58, 1997.

[22] M. Elad and A. Feuer. Super-resolution reconstruction of image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):817–834, 1999.

[23] M. Elad and A. Feuer. Super-resolution restoration of an image sequence - adaptive filtering approach. *IEEE Transactions on Image Processing*, 8(3):387–395, 1999.

[24] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[25] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 20(1):25–47, 2000.

[26] R.C. Hardie, K.J. Barnard, and E.E. Armstrong. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Transactions on Image Processing*, 6(12):1621–1633, 1997.

[27] B.K.P. Horn. *Robot Vision*. McGraw Hill, 1996.

[28] T.S. Huang and R. Tsai. Multi-frame image restoration and registration. *Advances in Computer Vision and Image Processing*, 1:317–339, 1984.

[29] M. Irani and S. Peleg. Improving resolution by image restoration. *Computer Vision, Graphics, and Image Processing*, 53:231–239, 1991.

[30] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occulsion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, 1993.

[31] M. Irani, B. Rousso, and S. Peleg. Image sequence enhancement using multiple motions analysis. In *Proceedings of the 1992 Conference on Computer Vision and Pattern Recognition*, pages 216–221, Urbana-Champaign, Illinois, 1992.

[32] D. Keren, S. Peleg, and R. Brada. Image sequence enhancement using sub-pixel displacements. In *Proceedings of the 1988 Conference on Computer Vision and Pattern Recognition*, pages 742–746, Ann Arbor, Michigan, 1988.

[33] S. Kim, N. Bose, and H. Valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38:1013–1027, 1990.

[34] S. Kim and W.-Y. Su. Recursive high-resolution reconstruction of blurred multiframe images. *IEEE Transactions on Image Processing*, 2:534–539, 1993.

[35] S. Mann and R.W. Picard. Virtual bellows: Constructing high quality stills from video. In *Proceedings of the First International Conference on Image Processing*, pages 363–367, Austin, TX, 1994.

[36] V.S. Nalwa. *A Guided Tour of Computer Vision*. Addison-Wesley, 1993.

[37] T. Numnonda, M. Andrews, and R. Kakarala. High resolution image reconstruction by simulated annealing. *Image and Vision Computing*, 11(4):213–220, 1993.

[38] A. Patti, M. Sezan, and A. Tekalp. Superresolution video reconstruction with arbitrary sampling latices and nonzero aperture time. *IEEE Transactions on Image Processing*, 6(8):1064–1076, 1997.

[39] S. Peleg, D. Keren, and L. Schweitzer. Improving image resolution using subpixel motion. *Pattern Recognition Letters*, pages 223–226, 1987.

[40] P.J. Philips, H. Moon, P. Rauss, and S.A. Rizvi. The FERET evaluation methodology for face-recognition algorithms. In *CVPR '97*, 1997.

[41] W.K. Pratt. *Digital Image Processing*. Wiley-Interscience, 1991.

[42] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[43] H. Qi and Q. Snyder. 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, 2000.

[44] T. Riklin-Raviv and A. Shashua. 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, 1999.

[45] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.

[46] R. Schultz and R. Stevenson. A Bayseian approach to image expansion for improved definition. *IEEE Transactions on Image Processing*, 3(3):233–242, 1994.

[47] R. Schultz and R. Stevenson. Extraction of high-resolution frames from video sequences. *IEEE Transactions on Image Processing*, 5(6):996–1011, 1996.

[48] H. Shekarforoush. Conditioning bounds for multi-frame super-resolution algorithms. Technical Report CAR-TR-912, Computer Vision Laboratory, Center for Automation Research, University of Maryland, 1999.

[49] H. Shekarforoush, M. Berthod, J. Zerubia, and M. Werman. Sub-pixel bayesian estimation of albedo and height. *International Journal of Computer Vision*, 19(3):289–300, 1996.

[50] V. Smelyanskiy, P. Cheeseman, D. Maluf, and R. Morris. Bayesian super-resolved surface reconstruction from images. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina, 2000.

[51] H. Stark and P. Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. *Journal of the Optical Society of America A*, 6:1715–1726, 1989.

[52] R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *Sixth International Conference on Computer Vision (ICCV'98)*, pages 517–524, Bombay, 1998.

[53] H. Ur and D. Gross. Improved resolution from subpixel shifted pictures. *Computer Vision, Graphics, and Image Processing*, 54(2):181–186, 1992.