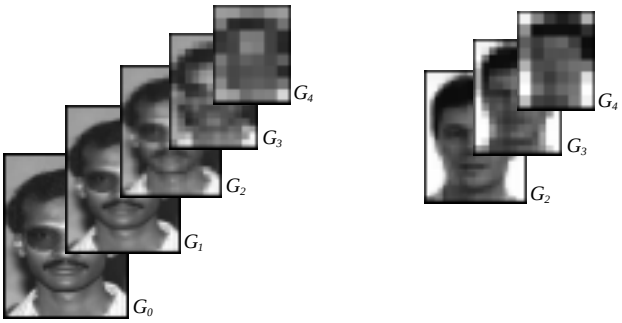
Hallucinating Faces

Simon Baker and Takeo Kanade
The Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213, USA
{simonb,tk}@cs.cmu.edu

Abstract

Faces often appear very small in surveillance imagery because of the wide fields of view that are typically used and the relatively large distance between the cameras and the scene. For tasks such as face recognition, resolution enhancement techniques are therefore generally needed. Although numerous resolution enhancement algorithms have been proposed in the literature, most of them are limited by the fact that they make weak, if any, assumptions about the scene. We propose an algorithm to learn a prior on the spatial distribution of the image gradient for frontal images of faces. We proceed to show how such a prior can be incorporated into a resolution enhancement algorithm to yield 4–8 fold improvements in resolution (i.e. 16–64 times as many pixels). The additional pixels are, in effect, hallucinated.



(a) High Resolution Pyramid (b) Low Resolution Pyramid

Figure 1. (a) A Gaussian pyramid is created by repeated smoothing and down-sampling. (b) The pyramid of a lower resolution image. Resolution enhancement can be formulated as estimating the bottom level of the pyramid.

1. Introduction

Most approaches to resolution enhancement make weak assumptions about the image(s). A common assumption is that the images are Markov Random Fields [9, 14]. While this assumption does aid resolution enhancement, it is a generic assumption applicable to most images. The class of images of human faces is a relatively small subset. It should therefore be possible to develop algorithms specifically for human faces that outperform these generic algorithms.

In Figure 1(a) we illustrate the Gaussian pyramid [3, 4] of a fairly high resolution image of a face. In Figure 1(b) we include a similar pyramid for a lower resolution image. The second pyramid has been started at a higher level so that the resolutions at the G_2 level and above match. In this setting, resolution enhancement can be posed as predicting the lowest level of the Gaussian pyramid G_0.

We propose an algorithm to learn the resolution enhancement function, say from G_2 to G_0, for frontal images of faces. Specifically, we use a pyramid based algorithm to learn a prior on the derivatives of the high resolution image G_0 as a function of the spatial location in the image, and the information in the higher levels of the pyramid. This prior is then incorporated into the function from G_2 to G_0.

We demonstrate that our algorithm does learn to enhance the resolution of faces, and in fact only faces. We also show that it outperforms the cubic B-spline interpolation algorithm by a wide margin. Our algorithm can enhance images of faces which are barely detectable (12×16 pixels) by a factor of 8 in each direction (to give 96×128 pixel images.)

1.1. Related Work

Freeman and Pasztor [8] recently proposed a learning framework for low-level vision, one application of which is image interpolation. They only applied their algorithms to generic images however, and so have not obtained as good results as we have. Our algorithm is closely related to the "class-based" illumination normalization algorithm [13] in this respect. Riklin-Raviv and Shashua showed that illumination normalization is possible for frontal images of faces, where it is not possible for generic objects. Our algorithm might therefore be called a class-based resolution enhancement algorithm. Another major advantage of our approach over [8] is that we are able to use multiple images from a video stream, if available, as in traditional super-resolution.

A resolution enhancement algorithm for human faces is proposed in [7]. As a face is tracked, the parameters of an "active-appearance" model are estimated and used to predict what a high resolution version of the face would look like. It is unlikely that such an algorithm could ever work on images as small as 12×16 pixels. Active appearance

models are based on the location of around 50 points on the face. When the image itself only contains 100-200 pixels, the triangulated elements in the model essentially become degenerate points.

As will be described in the next section, our approach uses similar techniques to the super-resolution algorithms of Schultz and Stevenson [15] and Hardie *et al.* [9], and the multi-resolution texture synthesis algorithms of De Bonet and Viola [5, 6] and Heeger and Bergen [10]. The application to class-based enhancement is, of course, different.

2. Theory and Algorithms

2.1. Gaussian Pyramids

The Gaussian pyramid [3, 4] starting at level $l = k$ of an image I is the set of images $G_k(I), G_{k+1}(I), \ldots, G_N(I)$, where:

$$G_l(I) = \begin{cases} I & \text{if } \; l = k \\ \text{REDUCE}(G_{l-1}(I)) & \text{if } \; k < l \leq N \end{cases} \quad (1)$$

and N is chosen so that $G_N(I)$ is (smaller than) some fixed size. The operator REDUCE$(\cdot)$ combines a (Gaussian) smoothing step and a down-sampling step. The details of this operator vary somewhat from author to author. We found the performance of our algorithms to be largely independent of the choice. We actually chose REDUCE$(\cdot)$ to be the pixel averaging function:

$$\text{REDUCE}(I)(m,n) = \frac{1}{4}\sum_{i=0}^{1}\sum_{j=0}^{1} I(2*m+i, 2*n+j) \quad (2)$$

because this definition is more consistent with image formation as integration over the pixel. A "Gaussian" pyramid computed using this definition is illustrated in Figure 1(a).

In terms of the Gaussian pyramid, a resolution enhancement algorithm is a function from $G_l(I)$ to $G_0(I)$ where $l > 0$. Given a lower resolution image we can create a Gaussian pyramid starting at a higher level. For example, if the image is 2^k times smaller (in each direction), we start at level $l = k$, as is illustrated in Figure 1(b) for $k = 2$. Resolution enhancement then consists of estimating $G_0(I)$.

2.2. Observation Model

We assume that the low resolution image $G_k(I)$ actually captured by the camera is perturbed with additive i.i.d. Gaussian noise $\eta(m,n)$. We therefore have:

$$G_k(I)(m,n) = \sum_{(p,q)} W(m,n,p,q) G_0(I)(p,q) + \eta(m,n). \quad (3)$$

Each low resolution pixel $G_k(I)(m,n)$ is the weighted sum of the high resolution pixels $G_0(I)(p,q)$ and the additive noise $\eta(m,n)$. The weights $W(\cdot)$ relating the pixels are a function of how much the low resolution pixels (m,n) and the high resolution pixels (p,q) overlap. Using the definition of REDUCE$(\cdot)$ in Equation (2), an Expression for $W(\cdot)$ can be derived:

$$W(m,n,p,q) = \begin{cases} \frac{1}{2^{2k}} & \text{if } \; p \in [m\cdot 2^k, (m+1)\cdot 2^k) \\ & \;\; \& \; q \in [n\cdot 2^k, (n+1)\cdot 2^k) \\ 0 & \text{Otherwise.} \end{cases} \quad (4)$$

Equation (3) is an implicit expression for the unknown high resolution image $G_0(I)$ in terms of the known low resolution image $G_k(I)$. Equation (3) cannot be used to solve directly for the unknown high resolution image $G_0(I)$ for two reasons: (1) the noise $\eta(m,n)$ is unknown, and (2) there are more unknowns in $G_0(I)$ than equations. One common approach [9, 15] in this situation is to solve for the maximum *a posteriori* (MAP) solution using Bayes Law.

2.3. Bayesian MAP Formulation

The maximum *a posteriori* (MAP) estimate of the high resolution image G_0 is $\arg\max_{G_0} \Pr[G_0 \mid G_k]$. Bayes law for this estimation problem is:

$$\Pr[G_0 \mid G_k] = \frac{\Pr[G_k \mid G_0] \cdot \Pr[G_0]}{\Pr[G_k]}. \quad (5)$$

Since $\Pr[G_k]$ is a constant because G_k is known already, and since the logarithm function is a monotonically increasing function, we have: $\arg\max_{G_0} \Pr[G_0 \mid G_k] =$

$$\arg\min_{G_0} \left(-\ln \Pr[G_k \mid G_0] - \ln \Pr[G_0]\right). \quad (6)$$

The first term in this expression $-\ln \Pr[G_k \mid G_0]$ is the (negative log) probability of getting the low resolution image G_k, given that the high resolution image is G_0. It depends upon the distribution of the noise η. Since we assume that the noise $\eta(m,n)$ is i.i.d. and Gaussian, with covariance σ_η^2, we therefore have: $-\ln \Pr[G_k \mid G_0] = C_1 +$

$$\frac{1}{2\sigma_\eta^2}\sum_{m,n}\left[G_k(m,n) - \sum_{(p,q)} W(m,n,p,q) G_0(p,q)\right]^2 \quad (7)$$

where C_1 is a constant that only depends upon σ_η^2. Hence C_1 can be ignored in Equation (6).

2.4. Predicting a Gradient Prior

We now discuss the prior term $-\ln \Pr[G_0]$. Schultz and Stevenson [15] and Hardie *et al.* [9] both used Markov Random Field priors in their super-resolution algorithms. We wish to use a prior predicted from training samples.

Suppose we have a collection of high resolution training images of human faces T_i. We can compute their Gaussian pyramids $G_0(T_i), \ldots, G_N(T_i)$. We can also compute their Laplacian pyramids $L_0(T_i), \ldots, L_N(T_i)$ [4], the horizontal $H_0(T_i), \ldots, H_N(T_i)$ and vertical $V_0(T_i), \ldots, V_N(T_i)$ first derivatives of the Gaussian pyramids, and the horizontal

$H_0^2(T_i), \ldots, H_N^2(T_i)$ and vertical $V_0^2(T_i), \ldots, V_N^2(T_i)$ second derivatives of the Gaussian pyramids. See [1] for more details. We can then form a pyramid of feature vectors:

$$\mathbf{F}_j(T_i) = \big(L_j(T_i), H_j(T_i), V_j(T_i), H_j^2(T_i), V_j^2(T_i)\big) \quad (8)$$

for $j = 0, \ldots, N$.

Given a low resolution image I that is 2^k times smaller than the training samples, we can compute the Gaussian pyramid from level k and upwards $G_k(I), \ldots, G_N(I)$, as is illustrated in Figure 1(b). Similarly, we can compute the feature pyramids for those levels $\mathbf{F}_k(I), \ldots, \mathbf{F}_N(I)$. We know nothing, however, about the lower levels of the feature pyramid $\mathbf{F}_0(I), \ldots, \mathbf{F}_{k-1}(I)$, and in particular $\mathbf{F}_0(I)$.

To describe the algorithm we use to predict $\mathbf{F}_0(I)$, we need one further piece of notation. If (m, n) is a pixel in the l^{th} level of a pyramid, its parent at the $l + 1^{\text{th}}$ level is $(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor)$. We therefore define the Parent Structure vector of a pixel (m, n) in the l^{th} level to be: $\mathbf{PS}_l(I)(m, n) =$

$$\Big(\mathbf{F}_l(I)(m, n), \ldots, \mathbf{F}_N(I)(\Big\lfloor \frac{m}{2^{N-1}} \Big\rfloor, \Big\lfloor \frac{n}{2^{N-1}} \Big\rfloor)\Big). \quad (9)$$

We then use the following algorithm to predict $\overline{\mathbf{F}}_0(I)$. (We use over-line to denote predicted values.)

Gradient Prior Prediction Algorithm

For each pixel (m, n) in the high resolution image to be predicted $G_0(I)$, do:

1. Create space for $\overline{\mathbf{F}}_0(I)(m, n)$
2. Find $j = \arg\min_i$

$$\Big\| \mathbf{PS}_k(I)(\Big\lfloor \frac{m}{2^k} \Big\rfloor, \Big\lfloor \frac{n}{2^k} \Big\rfloor) - \mathbf{PS}_k(T_i)(\Big\lfloor \frac{m}{2^k} \Big\rfloor, \Big\lfloor \frac{n}{2^k} \Big\rfloor) \Big\|$$

3. Copy $\mathbf{F}_0(T_j)(m, n)$ into $\overline{\mathbf{F}}_0(I)(m, n)$.

The distance function $\|\cdot\|$ is a weighted L^2 norm. We found the performance to be largely independent of the weights, but eventually gave the derivative components half as much weight as the Laplacian values and reduced the weight by a factor of 2 for each increase in the pyramid level.

This algorithm is similar to the random texture synthesis algorithm of De Bonet and Viola [5, 6]. One difference is that it is deterministic. It chooses the most likely values for the Parent Structure vector rather than randomly sampling from a set of values. Another difference is that the decision is spatially variant. At each pixel (m, n) the algorithm only looks at the corresponding pixels in the training samples.

2.5. Incorporation in the MAP Framework

Once $\overline{\mathbf{F}}_0(I)$ has been estimated, the horizontal and vertical derivatives of the high resolution image ($\overline{H}_0(I)$ and $\overline{V}_0(I)$) can be extracted from it. (See Equation (8).) The derivatives of $G_0 = G_0(I)$ should equal these values. Parametric expressions for $H_0(G_0)$ and $V_0(G_0)$ can be derived in terms of the unknown pixels in the high resolution image $G_0 = G_0(I)$. We assume that the errors between the predicted and actual derivatives are i.i.d. and Gaussian with covariance σ_∇^2. Therefore we set: $-\ln \Pr[G_0] = C_2 +$

$$\frac{1}{2\sigma_\nabla^2} \sum_{m,n} \big[H_0(G_0)(m, n) - \overline{H}_0(I)(m, n)\big]^2 + \frac{1}{2\sigma_\nabla^2} \sum_{m,n} \big[V_0(G_0)(m, n) - \overline{V}_0(I)(m, n)\big]^2 \quad (10)$$

where C_2 is a constant that only depends upon σ_∇^2 (and therefore can be ignored.)

Note that $-\ln \Pr[G_0]$ is a function of $G_k(I) = I$. This is legitimate for the following reason. The Gradient Prior Prediction algorithm divides the set of images I into a collection of subclasses, based on the decisions made in Step 2. If these subclasses are denoted K_i, then $\Pr[G_0] = \sum_i \Pr[G_0 \mid I \in K_i] \cdot \Pr[I \in K_i]$. Once I is known, it can be determined which subclass it is in. If this class is K_k, then the expression for $\Pr[G_0]$ simplifies to $\Pr[G_0 \mid I \in K_k]$. It is really this function that is contained in Equation (10).

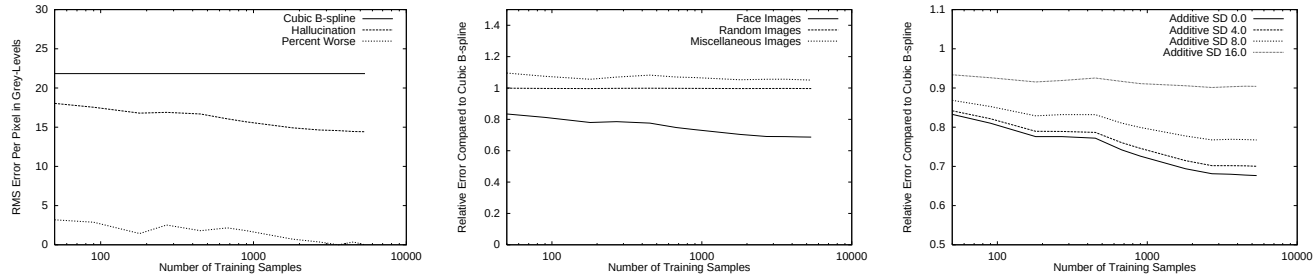
2.6. Optimization by Gradient Descent

The expressions for the unknown high resolution image derivatives $H_0(G_0)(m, n)$ and $V_0(G_0)(m, n)$ are linear in the unknown pixels $G_0(I)(m, n)$. Equations (6), (7), and (10) therefore form a weighted least squares problem in the unknown high resolution image pixels $G_0(I)(m, n)$. This problem, however, can be very high dimensional. The number of unknowns is the number of pixels in the high resolution image $G_0(I)$. Directly solving a linear system of such size can prove problematic. We therefore used a gradient descent algorithm using the standard diagonal approximation to the Hessian [12] to determine how large the step size should be. Since the error function is quadratic, the algorithm converges to the single global minimum.

3. Experimental Results

Our experiments were conducted with a subset of the FERET data set [11] consisting of 596 images of 278 individuals (92 women and 186 men). The images (which are all frontal) need to be aligned in the class-based approach so we can assume that the same part of the face appears in roughly the same part of the image [13]. This alignment was performed by hand marking the location of 3 points; the centers of the eyes and the lower tip of the nose. These 3 points define an affine warp [2], which was used to warp the images into canonical 96×128 pixel images. These images were then down-sampled by pixel averaging to give 48×64, 24×32, and 12×16 pixel input images.

We used a "leave-one-out" methodology to test our algorithm. 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. We also added 8 synthetic variations of each image to the training set by translating



(a) Performance Vs. No. of Training Samples (b) Performance Vs. Type of Image (c) Robustness Against Noise

Figure 2. (a) Validation that our hallucination algorithm learns how to enhance the resolution of faces, and (b) only faces. (c) The robustness of our algorithm to additive pixel intensity noise.

it 8 times by a small amount. This step enhances the performance of our algorithms slightly, although it is not vital. Finally, note that due to lack of space the results contained in this section a just a illustrative subset of those in [1].

3.1. Demonstration of Learning

Initially we consider the case of enhancing 24×32 pixel images fourfold to give 96×128 pixel images. Our first set of experiments are designed to show that our algorithm does learn how to enhance the resolution. First we varied the number of training samples. We graph the results in Figure 2(a). The average (RMS) pixel error is plotted against the number of training samples. Two curves are plotted, one for our face hallucination algorithm, and one for the cubic B-spline algorithm [16]. As might be expected, our algorithm does perform better than cubic B-spline interpolation, which incorporates no knowledge of the type of image being used. The other important point to note is that, as expected, the performance of our algorithm does improve as the number of training samples increases.

The results in Figure 2(a) are an average over the 100 images in the test set. To get an idea of the variation in the results across the test set, we also plot in Figure 2(a) the percentage of times that the hallucination algorithm does worse than cubic B-spline. By around 5000 training samples, this percentage has dropped to almost zero. Therefore, given enough training samples, we can be reasonably sure that the hallucination algorithm will perform better than cubic B-spline, and most of the time much better.

As further justification that our algorithm performs well for any frontal face image, in Figure 3 we present the results for both the best and worst performing images in the 100 image test set (by RMS error). As can be seen, there is little qualitative variation in the performance between these two images. Also note how the hallucinated image in the second column is much higher resolution than the input in the first, and also how it contains much more high resolution detail than the cubic B-spline result in the third column.

In Figure 2(b) we present similar results for images that do not contain faces. For comparison across different types of image, we plot the relative RMS pixel error compared to the cubic B-spline algorithm, instead of plotting the absolute RMS pixel error itself. We plot curves of the relative RMS pixel error for faces, random images, and 50 miscellaneous images from an image database (mostly consisting of images of outdoor scenes.) We find that the hallucination algorithm is an improvement only for faces. For random images there is little difference, and for the miscellaneous image set the hallucination algorithm actually does worse.

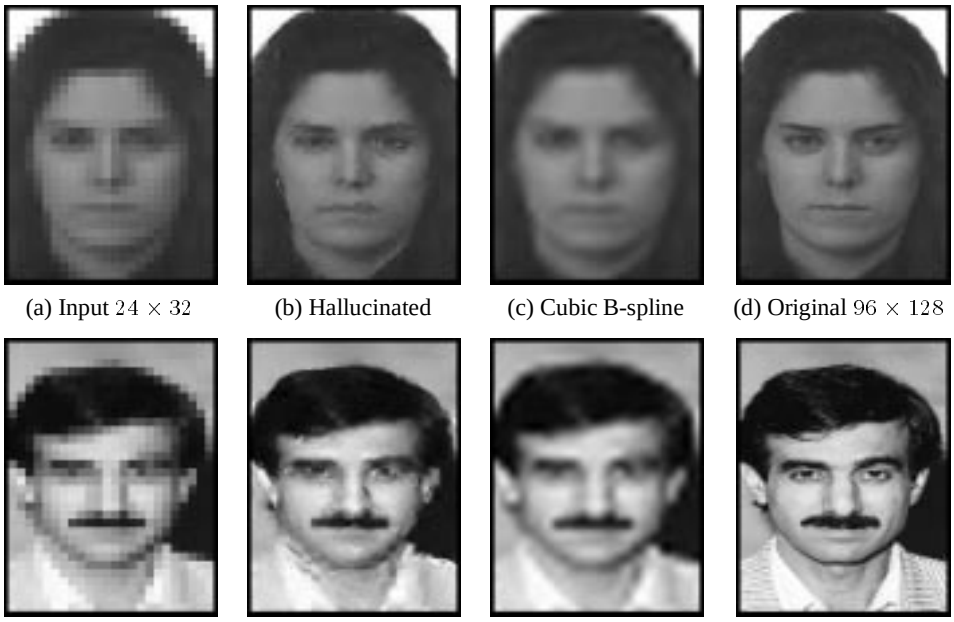
3.2. Robustness to Additive Noise

Next we investigated how robust the performance is in the presence of additive intensity noise. The results are presented in Figure 2(c). We ran exactly the same experiments as in Figure 2(a), but before applying our algorithm we added Gaussian noise with various standard deviations to the down-sampled images. For standard deviation 0.0 the results are the same, however in Figure 2(c) we plot the relative RMS pixel error rather than the absolute value. We also plot curves for standard deviations of 2.0, 4.0, 8.0, and 16.0 grey levels. The results show that for standard deviations up to around 4.0–8.0 the performance is relatively unaffected by the noise, but around standard deviation 16.0 the performance drops off very quickly. Hence, our algorithm is reasonably robust to this type of noise. It can tolerate 2-3 bits of noise without much degradation in performance. Images similar to those in Figure 3 validate this result [1].

3.3. Multiple Image Results

We now present results for multiple 12×16 pixel images. In the traditional super-resolution manner [9, 15] we assume that we have a video of the face. Hence, multiple slightly translated images are available. We simulate this using the FERET database by randomly translating the original FERET images multiple times by sub-pixel amounts to form the inputs. In Section 2 we described our algorithms in terms of a single image. Extending them to allow multiple slightly translated images is straightforward [1]. Multiple copies of Equations (7) and (10) are needed, one for each input image. These equations also need to be modified slightly to allow for the relative translations of the images. Otherwise the formulation stays essentially the same.

The results are included in Figure 4. The input consists



(a) Input 24×32 (b) Hallucinated (c) Cubic B-spline (d) Original 96×128

(e) Input 24×32 (f) Hallucinated (g) Cubic B-spline (h) Original 96×128

Figure 3. The best (a)–(d) and worst (e)-(h) results in Figure 2(a) in terms of the RMS pixel error.

of 3 versions of the 12×16 pixel image in the leftmost column, translated by random sub-pixel amounts. The inputs are aligned using a standard parametric motion algorithm [2] and then the multiple image hallucination algorithm [1] applied. The results in the second column are a huge improvement over both cubic B-spline interpolation and the Schultz and Stevenson algorithm [15]. In particular, note how high resolution features such as the eyebrows and lips are recovered, even though there is little evidence for them in the input. Also try squinting at the images, a standard test of enhancement quality. Unlike the results in [8], a marked difference can be seen between the input and the output.

4. Discussion

We have presented an algorithm to predict a gradient prior and shown how to incorporate this prior into a resolution enhancement algorithm. We have shown our algorithms to be a huge improvement over existing interpolation and super-resolution algorithms. A small number of 12×16 pixel images of a human face can be fused into a single 96×128 pixel image that closely resembles the original face. The one factor that most contributes to the high performance is that the algorithms are class-based [13].

4.1. Recognition Vs. Enhancement

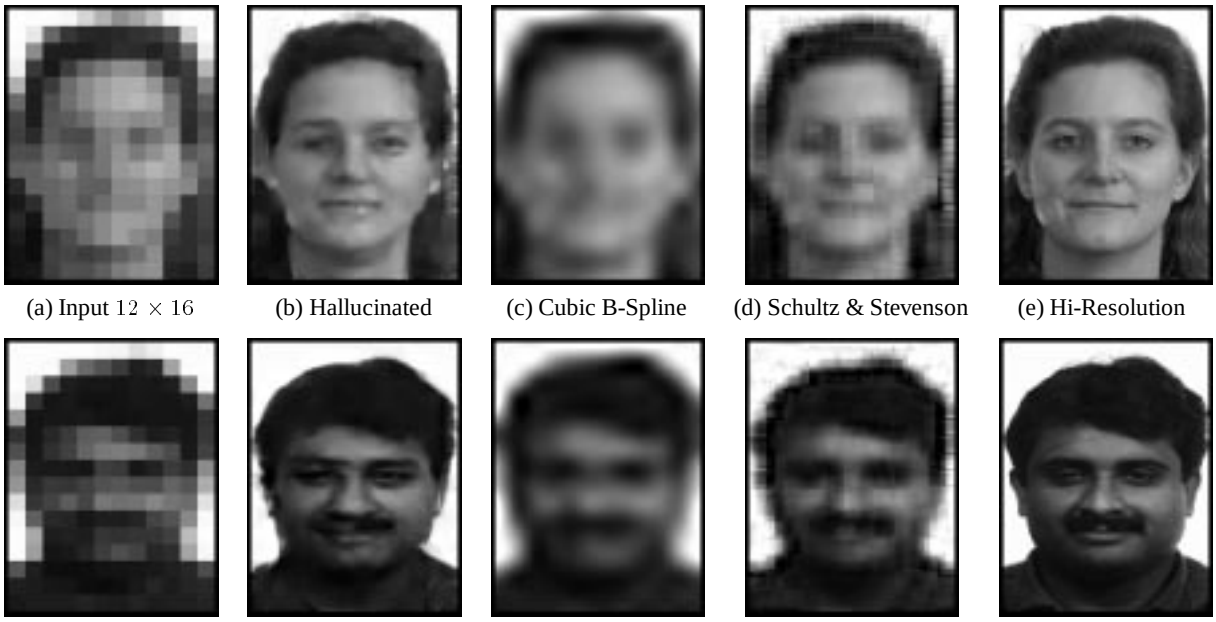
We have not had time to demonstrate that our algorithms improve face recognition performance. Although we leave this task as future work, we would like to mention a couple of points. No new information has been added during resolution enhancement. Theoretically, therefore, face recognition algorithms could be developed that work as well on the low resolution images, as they do on the output of our hallucination algorithm. What, then, is the utility of our approach? (1) Our algorithms should make the development of low resolution face recognition algorithms easier since researchers will not have to worry about the additional complications introduced by the low resolution images. (2) Our algorithms are useful for humans. If someone were shown Figures 4(a) and (b) and asked whether they had seen the person before, they would be much more confident in their response when shown the 96×128 pixel image.

There is a great deal in common between resolution (and illumination) enhancement and recognition. First, our approach works using a form of recognition. A discrete recognition decision is made in Step 2. of the gradient prediction algorithm to determine which of the training samples looks most like the input at the low resolution. In a way, a local feature detector is applied, and how the resolution is enhanced depends upon which feature is detected. (Edwards *et al.* [7] use a related approach.) At the other extreme, a face recognition algorithm could be used for enhancement. If the person can be recognized from the low resolution data, the image can be enhanced by looking up the person in the database. The major difference between these extremes is the "scale" at which the recognition decision is made.

4.2. Future Work

All of our results are on hand-registered images from the FERET data set [11]. To show our approach is useful in real surveillance scenarios, we need to try out our algorithms on data captured using surveillance cameras. To build an automatic system we will also need to implement face tracking, pose estimation, and feature localization algorithms.

The learning algorithm at the heart of our approach is a



(a) Input 12×16 (b) Hallucinated (c) Cubic B-Spline (d) Schultz & Stevenson (e) Hi-Resolution

(f) Input 12×16 (g) Hallucinated (h) Cubic B-Spline (i) Schultz & Stevenson (j) Hi-Resolution

Figure 4. Selected results for multiple 12 x 16 pixel images. The input consists of 3 versions of the image in the leftmost column, translated by random sub-pixel amounts. Note how high frequency features such as the eyes are reconstructed even though there almost no evidence for them in the input.

nearest neighbor algorithm. Many other learning algorithms could be used instead. We would like to perform a systematic comparison of these techniques. We would also like to explore the use of different feature spaces. (See Equation (8).) In particular, one of the most interesting questions is how "local" the features should be.

Acknowledgements

The research described in this paper was supported by US DOD Grant MDA-904-98-C-A915. We also wish to thank Harry Shum for pointing out the work of Freeman and Pasztor, Iain Matthews for pointing out the work of Edwards *et al.*, and everyone in the Face Group at CMU for their comments and suggestions.

References

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

[2] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of Second ECCV*, pages 237–252, 1992.

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

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

[5] J. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of SIGGRAPH '97*, pages 361–368, 1997.

[6] J. De Bonet and P. Viola. A non-parametric multi-scale statistical model for natural images. *Advances in Neural Information Processing*, 10, 1997.

[7] G. Edwards, C. Taylor, and T. Cootes. Learning to identify and track faces in image sequences. In *Proceedings of the Third ICAFGR*, pages 260–265, 1998.

[8] W. Freeman and E. Pasztor. Learning low-level vision. In *Proceedings of the Seventh ICCV*, 1999.

[9] R. Hardie, K. Barnard, and 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.

[10] D. Heeger and J. Bergen. Pyarmid based texture analysis/synthesis. In *Proceedings of SIGGRAPH '95*, pages 229–238, 1995.

[11] P. Philips, H. Moon, P. Rauss, and S. Rizvi. The FERET evaluation methodology for face-recognition algorithms. In *Proceedings of CVPR '97*, pages 137–143, 1997.

[12] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Second edition, 1992.

[13] T. Riklin-Raviv and A. Shashua. The Quotient image: Class based recognition and synthesis under varying illumination. In *Proceedings of CVPR '99*, pages 566–571, 1999.

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

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

[16] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, 1992.