# OPTIMAL IMAGE SCALING USING PIXEL CLASSIFICATION


*C. Brian Atkins*[†]

Hewlett-Packard Laboratories
1501 Page Mill Rd., M/S 4U-6
Palo Alto, CA 94304

*Charles A. Bouman, Jan P. Allebach*

Purdue University
1285 Electrical Engineering Bldg.
W. Lafayette, IN 47907



## ABSTRACT

We introduce a new approach to optimal image scaling called Resolution Synthesis (RS). In RS, the pixel being interpolated is £rst classi£ed in the context of a window of neighboring pixels; and then the corresponding high-resolution pixels are obtained by £ltering with coef£cients that depend upon the classi£cation. RS is based on a stochastic model explicitly re¤ecting the fact that pixels falls into different classes such as edges of different orientation and smooth textures. We present a simple derivation to show that RS generates the minimum mean-squared error (MMSE) estimate of the high-resolution image, given the low-resolution image. The parameters that specify the stochastic model must be estimated beforehand in a training procedure that we have formulated as an instance of the well-known expectation-maximization (EM) algorithm. We demonstrate that the model parameters generated during the training may be used to obtain superior results even for input images that were not used during the training.


## 1. INTRODUCTION

Image scaling is an important application in the £eld of image processing; and a critical one when it comes to displaying or printing an image at a resolution higher than that at which the image is available. In this paper, we present a new approach to optimal image scaling called Resolution Synthesis (RS) [1, 2, 3]. Formally, what sets RS apart from prior approaches to optimal image scaling is that RS is based on a stochastic model explicitly re¤ecting the fact that pixels falls into different classes such as edges and smoother gradients of different orientation. By an analysis of this model, we show that RS image scaling generates the minimum mean-squared error (MMSE) estimate of the high-resolution image, given the low-resolution image. A model which is similar but different from the one we present here has been used by Popat and Picard [4] for image restoration, compression, and synthesis.

In RS image scaling, the pixel being interpolated is £rst classi£ed in the context of a window of neighboring pixels; and then the corresponding high-resolution pixels are obtained by £ltering with coef£cients that are optimal for the selected class. The result of this is that edges are interpolated using £lters which do not smooth out their sharp transitions.

The parameters that specify the stochastic model behind RS must be estimated in a training procedure that we have formulated as an instance of the well-known expectation-maximization (EM) algorithm. While the training is computationally intensive, it only needs to be performed once. We demonstrate that the resulting model parameters may be used to obtain superior results for input images that were not used during the training.

[†]This work was performed while C. Brian Atkins was a graduate student at Purdue University.

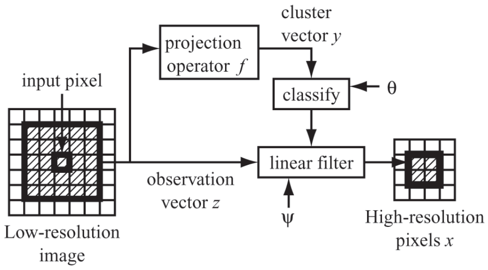


**Fig. 1**. Structure of the RS predictor.

RS is different from various image scaling approaches previously proposed in the literature. These include the class of B-spline interpolators [5, 6, 7]; cubic convolution interpolation [8]; edge-directed methods [9], [10]; and a maximum *a priori* (MAP) estimation approach [11].

In [12], we describe a separate approach to MMSE image scaling implemented as a regression tree. The present approach is different in that pixel classi£cation is based on class membership in a Gaussian mixture model. We have found that the present approach leads to better interpolation quality.

The remainder of this paper is organized as follows. In Sec. 2, we introduce RS by describing how RS image scaling is carried out, and how the parameters for the RS model are obtained. Finally, in Secs. 3 and 4, we present results and conclusions.

## 2. RESOLUTION SYNTHESIS

We begin by training on example low- and high-resolution image pairs to estimate the required prediction parameters. The resulting prediction parameter estimates may be used to scale images that were not even in the training set. The structure of the RS predictor is illustrated in Fig. 1. The example here is for a scaling factor of 2; however, the derivation we present only requires that the scaling factor be a positive integer.

For each input pixel in the low-resolution image, the prediction task is to compute the corresponding block of high-resolution pixels. The predictor uses all the pixels in a neighborhood centered at the input pixel. In this example, the neighborhood is a $5 \times 5$–pixel window. For the analysis, we will write the input pixel neighborhood as the elements of an observation vector $z$, and the estimated high-resolution pixels in a vector $x$. More generally, we will be modeling both these vectors as random quantities, and we will follow the convention of using upper-case letters for random variables and lower-case letters for their realizations.

The form of our predictor is based on the assumption that each input pixel is from some class, with the different classes representing image features like edges and smoother gradient transitions of different orientation. Further, we assume we have an optimal prediction £lter for each class, so that if we knew the class of the input pixel, we could generate an optimal estimate of $X$ by applying the £lter to $z$. The only problem is that we can not ascertain the class of the input pixel.

In RS, the prediction process is to associate the input pixel with each of the classes, to different degrees, with a mixture of proportions. We estimate $X$ by £rst £ltering $z$ with the optimal predictors for the individual classes, and then combining the results with the proportion mixture. We refer to the £rst step as "classi£cation," and we represent the classi£er parameters with the symbol $\theta$. We refer to the second step as "£ltering," and we represent collectively the prediction £lters for the different classes with the symbol $\psi$.

Rather than classifying the input pixel based on the entire observation vector $z$, we use a "cluster vector" $y$ extracted from the observation vector using a nonlinear transformation $f(\cdot)$:

$$y = f(z) .$$

Generally $y$ is of lower dimension than $z$, and we refer to this transformation as a "projection." The projection operation is critical, because ultimately it in¤uences which classes are de£ned during the training, which in turn controls the quality of overall scaling process.

In the remainder of this section, we formally derive the scaling procedure, while demonstrating its optimality under certain assumptions. Then we describe the training process. Finally, we discuss brie¤y the role of the projection operator $f(\cdot)$. Note that here we assume images are grayscale. One way to achieve color image scaling is to treat the red, green, and blue planes as if they were individual grayscale images.

### 2.1. Optimal Image Scaling

In this subsection, we derive RS image scaling as an MMSE predictor, assuming we know the values of the classi£er parameters $\theta$ and the £lter parameters $\psi$. The training process, where we compute estimates of $\theta$ and $\psi$, is given in the following subsection.

Our derivation is based on three assumptions.

**Assumption 1:** The cluster vector $Y$ is distributed as a multivariate Gaussian mixture. That is, the probability density function $p_y(y)$ has the form

$$p_y(y) \quad = \quad \sum_{j=1}^{M} p_{y|j}(y \mid j)\pi_j \ , \tag{1}$$

where $j$ is a random variable representing the class; $p_{y|j}(y \mid j)$ is a multivariate Gaussian density for each $j$, and $\pi_j$ is the probability of class $j$. This assumption is not very strong, since we may closely approximate the true distribution by using a large number of classes. Note, assuming there are $M$ classes, $J$ takes discrete values between 1 and $M$.

**Assumption 2:** The conditional distribution of $X$ given $Z$ and $J$ is multivariate Gaussian, with mean $A_J Z + \beta_J$. The £lter parameters represented by $\psi$ are obtained directly from the parameters for these distributions, so we write

$$\psi \quad = \quad (\{A_j\}_{j=1}^{M}, \{\beta_j\}_{j=1}^{M}) \ . \tag{2}$$

This assumption is similar to the previous one since it requires that a suf£ciently large number of classes be chosen so that for each class, $J$, the high-resolution pixels are distributed as a linear transformation of the low-resolution pixels plus multivariate Gaussian noise.

**Assumption 3:** The class $J$ is conditionally independent of the high- and low-resolution vectors $X$ and $Z$, given the cluster vector $Y$. Formally, this means that

$$p_{j|x,z}(j \mid x, z) \quad = \quad p_{j|y}(j \mid y) \ . \tag{3}$$

Intuitively, this assumption means that the cluster vector $Y$ gives us all possible information required to determine the class $J$. This is a strong assumption since $Z$ and $X$ may contain additional information which might be useful in determining the class. However, this assumption will lead to important algorithmic simpli£cations that we have found improve the quality of the resulting scaled images.

Given these assumptions, we may compute the MMSE [13] estimate as

$$\begin{aligned} \hat{X} \quad &= \quad E[X \mid Z] && (4)\\ &= \quad \sum_{j=1}^{M} E[X \mid Z, J = j] p_{j|z}(j \mid Z) && (5)\\ &= \quad \sum_{j=1}^{M} (A_j Z + \beta_j) p_{j|y}(j \mid Y) \ . && (6) \end{aligned}$$

To obtain the £rst and second terms in the argument of the last summation, we have invoked assumptions 2 and 3, respectively.

In order to compute $\hat{X}$, we need to compute $p_{j|y}(j \mid y)$, the probability that the cluster vector $y$ belongs to class $j$. By assumption 1, the conditional probability of $Y$ given $J$ is a multivariate Gaussian. We will denote the mean of the $j$–th Gaussian distribution as $\mu_j$. In general, the covariance could be chosen to vary with each class $j$. However, we have found that choosing the covariance for all classes to be $\sigma^2 I$, where $I$ is the identity matrix, actually improves the quality of the scaled images. While this may seem counter-intuitive, it is quite reasonable, since the reduced number of parameters makes training more effective and accurate. Putting this together we write the parameters of $p_y(y)$ as $\theta$:

$$\theta \quad = \quad \left\{ \{\mu_j, \pi_j\}_{j=1}^{M}, \sigma \right\} \ . \tag{7}$$

Hence by an application of Bayes' rule, we can compute

$$p_{j|y}(j \mid y, \theta) \quad = \quad \frac{\exp\left(\frac{-1}{2\sigma^2}\|y - \mu_j\|^2\right)\pi_j}{\sum_{l=1}^{M} \exp\left(\frac{-1}{2\sigma^2}\|y - \mu_l\|^2\right)\pi_l} \ . \tag{8}$$

Inserting (8) into (6), we obtain the equation for optimal image scaling, in terms of the unknown parameters:

$$\hat{X} \quad = \quad \sum_{j=1}^{M} (A_j Z + \beta_j) \frac{\exp\left(\frac{-1}{2\sigma^2}\|y - \mu_j\|^2\right)\pi_j}{\sum_{l=1}^{M} \exp\left(\frac{-1}{2\sigma^2}\|y - \mu_l\|^2\right)\pi_l} \ . \tag{9}$$

The optimal output pixels are computed as a combination of the outputs of all $M$ of the interpolating £lters.

### 2.2. Estimating Predictor Parameters

Our objective here is to compute maximum likelihood (ML) estimates [13] of $\psi$ and $\theta$ from training images by extracting example realizations of the pair $(Z, X)$, which we assume are independent. This is generally difficult since the data does not reveal realizations of the class label $J$. This is known as the incomplete data problem, where observations of the triplet $(Z, X, J)$ would be complete data. To address this we have used the expectation-maximization (EM) algorithm [14, 15, 16]. A formal derivation of the training as an instantiation of the EM algorithm is given in [1]. Here, because of space limitation, we only list the steps in the process.

1. Build the training set. Procure example high-resolution images. Create the low-resolution images by lowpass filtering and downsampling. For scaling factor $L$, doing this by computing averages of non-overlapping $L \times L$ blocks works well because it guarantees perfect registration between the low- and high-resolution images. To build sharpening in to the scaling procedure, sharpen the high-resolution images after generating the low resolution images from them.

2. Extract training vectors from the training set. A training vector is a pair of low- and high-resolution vectors $(z, x)$ corresponding to any one of the pixels in the low-resolution images. We will denote the set of low-resolution pixels in the training set as $S$, and we will denote the extracted training vectors as $\{(z_s, x_s)\}_{s \in S}$. By writing $y_s$, we refer to the cluster vector extracted from vector $z_s$, computed as $f(z_s)$, with the operation described in Sec. 2.3. We will denote as $N$ the number of training vectors. The more, the better; but it is best to use at least $100,000$.

3. Select a value for the number $M$ of classes. In our experiments we used 100; however, satisfactory results can be obtained with as few as 25 classes.

4. Initialize estimate of $\theta$. For each $j \in \{1, \ldots, M\}$, set $\pi_j^{(0)}$ to $\frac{1}{M}$, and set each cluster mean $\mu_j^{(0)}$ to equal one of the cluster vectors from the training vectors. (To ensure the cluster means represent the training data comprehensively, acquire them from throughout the training images rather than from, say, just one of the training images.) Set ${\sigma^2}^{(0)}$ as the average of the sample variances of the elements in the cluster vectors:
$$ {\sigma^2}^{(0)} = \frac{1}{d} \sum_{i=1}^{d} r_i \ , $$
where $d$ is the number of elements in the cluster vector, and $r_i$ is the sample variance of the $i^{th}$ cluster vector component computed over the available cluster vectors.

5. Iterate the update equations (10)–(13) for $1 \le j \le M$ to estimate $\theta$.
$$ \begin{aligned} N_j^{(k+1)} &= \sum_{s \in S} p_{j|y}(j \mid y_s, \theta^{(k)}) \ ; & (10) \\ \pi_j^{(k+1)} &= \frac{N_j^{(k+1)}}{N} \ ; & (11) \\ \mu_j^{(k+1)} &= \frac{1}{N_j^{(k+1)}} \sum_{s \in S} y_s p_{j|y}(j \mid y_s, \theta^{(k)}) \ ; & (12) \end{aligned} $$
and
$$ {\sigma^2}^{(k+1)} = \frac{1}{d} \sum_{j=1}^{M} \left[ \pi_j^{(k+1)} \Xi_j \right] \ , \qquad (13) $$
where
$$ \Xi_j = \frac{1}{N_j^{(k+1)}} \sum_{s \in S} \| y_s - \mu_j^{(k+1)} \|^2 p_{j|y}(j \mid y_s, \theta^{(k)}) \ . $$
Stop when the quantity $\sum_{s \in S} \log p_y(y_s | \theta^{(k)})$ ceases to increase appreciably from one update to the next.

6. Estimate $\psi$. Compute
$$ \begin{aligned} \hat{A}_j &= \Sigma_{xz|j} \Sigma_{zz|j}^{-1} \ ; & (14) \\ \hat{\beta}_j &= \nu_{x|j} - \Sigma_{xz|j} \Sigma_{zz|j}^{-1} \nu_{z|j} \ ; & (15) \end{aligned} $$
for $1 \le j \le M$, where
$$ \begin{aligned} N_j &= \sum_{s \in S} p_{j|y}(j \mid y_s, \hat{\theta}) \ ; & (16) \\ \nu_j &\stackrel{\text{def}}{=} \begin{pmatrix} \nu_{x|j} \\ \nu_{z|j} \end{pmatrix} & (17) \\ &= \frac{1}{N_j} \sum_{s \in S} b_s p_{j|y}(j \mid y_s, \hat{\theta}) \ ; & (18) \end{aligned} $$
and
$$ \begin{aligned} \Sigma_j &\stackrel{\text{def}}{=} \begin{pmatrix} \Sigma_{xx|j} & \Sigma_{xz|j} \\ \Sigma_{xz|j}^t & \Sigma_{zz|j} \end{pmatrix} & (19) \\ &= \frac{1}{N_j} \sum_{s \in S} b_s b_s^t p_{j|y}(j \mid y_s, \hat{\theta}) \ , & (20) \end{aligned} $$
where
$$ b_s \stackrel{\text{def}}{=} \begin{pmatrix} x_s \\ z_s \end{pmatrix} \ . $$

### 2.3. The Projection Operator $f(\cdot)$

To extract a cluster vector from the observation vector, we first define the vector $y'$ by stacking the 8 pixels adjacent to the input pixel into a vector, and subtracting the value of the central pixel from each of these 8 elements. Then
$$ y = f(z) = \begin{cases} y' \, \|y'\|^{-\frac{3}{4}} & \text{if } y' \neq 0 \\ 0 & \text{else} \end{cases} \ . \qquad (21) $$
We experimented with a variety of projection operators and this one yielded the best results. It was chosen to accentuate edge features during clustering. To see this, observe that the vector $y'$ contains differences between the input pixel and its 8 nearest neighbors. In the $y'$ sample space, pixels from uniform, flat image regions fall near the origin and edge pixels fall away from the origin. The nonlinear transformation of (21) pushes nonzero $y'$ vectors away from the origin, which encourages the formation of more edge classes.

## 3. RESULTS

In Figs. 2 and 3, we present image scaling results for a factor of $L = 4$, using Photoshop bicubic interpolation and RS. We have found that RS compares very favorably with other scaling methods; but we have included Photoshop bicubic interpolation because it is common. We feel that this comparison shows that RS renders edges and detail relatively continuous and sharp.

## 4. CONCLUSIONS

We have introduced the image scaling algorithm known as Resolution Synthesis. We have provided an analytical derivation of the RS scaling procedure. Under certain assumptions, RS generates a MMSE estimate, in a closed-form computation, of the high-resolution image, given the source image. Finally, we have presented results which demonstrate that both RS and ERS yield results of very satisfactory subjective quality.

## 5. REFERENCES


[1] C. B. Atkins, *Classi£cation-Based Methods in Optimal Image Interpolation*, Ph.D. thesis, Purdue University, 1998, available at www.ece.purdue.edu/˜bouman/publications.

[2] C. B. Atkins, J. P. Allebach, C. A. Bouman, J. S. Gondek, M. T. Schramm, and F. W. Sliz, "Computerized method for improving data resolution," U.S. Patent No. 6,058,248, May 2000.

[3] C. B. Atkins, J. P. Allebach, C. A. Bouman, J. S. Gondek, M. T. Schramm, and F. W. Sliz, "Computerized method for improving data resolution," U.S. Patent No. 6,075,926, June 2000.

[4] Kris Popat and Rosalind W. Picard, "Cluster-based probability model and its application to image and texture processing," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 268–284, February 1997.

[5] H. S. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital £ltering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 508–517, December 1978.

[6] M. Unser, A. Aldroubi, and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 277–285, March 1991.

[7] M. Unser, A. Aldroubi, and M. Eden, "Enlargement or reduction of digital images with minimum loss of information," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 247–258, March 1995.

[8] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, December 1981.

[9] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 285–295, March 1995.

[10] J. P. Allebach and P. W. Wong, "Edge-directed interpolation," in *Proc. of the International Conference on Image Processing*, 1996, pp. 707 – 710.

[11] R. R. Schultz and R. L. Stevenson, "A Bayesian approach to image expansion for improved de£nition," *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 233–242, May 1994.

[12] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Tree-based Resolution Synthesis," in *Proc. of the Image Processing, Image Quality, Image Capture Systems Conference*, Savannah, GA, 1999.


**Fig. 2**. 4X scaling by Photoshop's bicubic scaling.

**Fig. 3**. 4X scaling by RS.


[13] L. L. Scharf, *Statistical Signal Processing*, Addison-Wesley, Reading, MA, 1991.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Society B*, vol. 39, pp. 1–38, 1977.

[15] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[16] R. A. Boyles, "On the convergence of the EM algorithm," *J. Roy. Stat. Society B*, vol. 45, no. 1, pp. 47–50, 1983.