Data Driven Image Models through Continuous Joint Alignment

Erik G. Learned-Miller

Abstract—This paper presents a family of techniques that we call *congealing* for modeling image classes from data. The idea is to start with a set of images and make them appear as similar as possible by removing variability along the known axes of variation. This technique can be used to eliminate "nuisance" variables such as affine deformations from handwritten digits or unwanted bias fields from magnetic resonance images. In addition to separating and modeling the latent images—i.e., the images without the nuisance variables—we can model the nuisance variables themselves, leading to factorized generative image models. When nuisance variable distributions are shared between classes, one can share the knowledge learned in one task with another task, leading to efficient learning. We demonstrate this process by building a handwritten digit classifier from just a single example of each class. In addition to applications in handwritten character recognition, we describe in detail the application of bias removal from magnetic resonance images. Unlike previous methods, we use a separate, nonparametric model for the intensity values at each pixel. This allows us to leverage the data from the MR images of *different* patients to remove bias from each other. Only very weak assumptions are made about the distributions of intensity values in the images. In addition to the digit and MR applications, we discuss a number of other uses of congealing and describe experiments about the robustness and consistency of the method.

Index Terms—Alignment, artifact removal, bias removal, congealing, clustering, correspondence, density estimation, entropy, maximum likelihood, medical imaging, magnetic resonance imaging, nonparametric statistics, registration, unsupervised learning.

1 Introduction

The adoption of classical statistical modeling techniques has revolutionized computer vision in the last two decades. The well-developed tools from probability and statistics are essential for dealing with the uncertainty encountered in real-world vision problems and provide solid principles on which to build robust and adaptive systems. These tools, however, were not generally developed for problems like those encountered in computer vision. Vision problems have peculiarities not well handled by classical methods, including measurements with millions of components (images), strong nonlinear dependencies, and problems due to lack of alignment, causing image values to have different meanings in different images. In the last few years, a variety of modeling techniques that address the specific challenges of computer vision have been proposed. In this paper, we introduce another such technique, which we call *congealing*, for model building in computer vision.

Congealing is a nonparametric technique for factoring, or separating, a set of images into sets of approximately independent "ingredients" or causes. It can be applied to problems in which there is shape variability within and between classes (e.g., binary digit recognition), when variability occurs mostly in brightness or color or when any other form of structured but *continuous* variation of a parameter occurs. While we focus mostly on using pixel values directly as features, the techniques are easily extended to arbitrary features of an image.

With respect to other hidden variable models like transformed mixtures of Gaussians (TMG, discussed in detail below) [1], this paper makes the following contributions: First, we model pose and other hidden variables continuously rather than as a discrete set. This allows our models to ultimately be more precise, but it also provides an optimization challenge. This leads to the second difference, which is that we optimize our models by iteratively maximizing the joint likelihood of a set of "latent" images, using a joint gradient descent procedure,[1] by finding the transformations that make them most similar. A final difference is that we model the residual images nonparametrically rather than as a set of Gaussians or other parametric distributions. This has important algorithmic and modeling consequences which we shall discuss. We illustrate these ideas on two problems in detail: handwritten digit recognition and bias removal in MR images. The organization of the paper is as follows:

In Section 2, we introduce congealing and illustrate its use on the problem of handwritten digit recognition. Our goal is not to break the record for accuracy in handwritten digit recognition using large numbers of training examples. Rather, it is to separate approximately independent factors of an image set, latent images, and transformations from each other. We then show how such generic factors as distributions over transformations, when learned from one set of characters (*letters*), can be used to build fairly good models of characters from other classes (*digits*) using only a single example of each new character class. That is, we demonstrate the transferral of learning in one problem to learning in another problem. We describe a handwritten

• *The author is with the Department of Computer Science, University of Massachusetts, Amherst, 140 Governor's Drive, Amherst, MA 01003. E-mail: elm@cs.umass.edu.*

Manuscript received 30 Nov. 2004; revised 27 May 2005; accepted 7 June 2005; published online 13 Dec. 2005.
Recommended for acceptance by A. Rangarajan.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0640-1104.

1. "Joint gradient descent" means a group of gradient descents, one for each image, occurring simultaneously.

digit recognizer with near 90 percent accuracy based on only a single example of each digit.

Our binary digit example illustrates congealing on binary images using the set of nonreflecting affine transforms as the "corrupting" hidden variable. Congealing, however, is not restricted to binary-valued images, to affine transformations, or even to images themselves. In Section 3, we show that congealing naturally extends to nonspatial transformations, such as brightness distortions in magnetic resonance (MR) medical images. We report in detail on recent work using congealing to eliminate intensity bias fields in MR images.

In Section 4, we briefly review other approaches that use hidden variable models similar to our latent image-transform model. Then, in Section 5, we discuss some additional properties of congealing, including results of experiments on alignment to local minima and robustness to noise. Section 6 concludes with a mention of other data types to which we have applied congealing and to directions for future work.

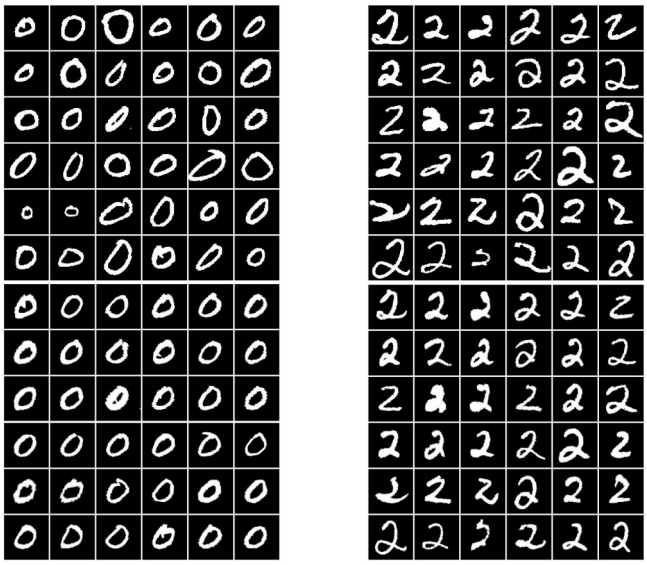

Fig. 1. (Upper left) Samples of zeros from the NIST database. (Upper right) Samples of twos from the NIST database. (Lower left) The zeros after congealing. (Lower right) The twos after congealing.

2 Congealing: Continuous Joint Alignment

Congealing is an algorithm for taking a set of images (or, in general, a set of arrays of arbitrary dimension) and transforming them according to a continuous set of allowable transformations to make them more similar, according to some measure of similarity [2]. Thus, there are three ingredients in any application of congealing:

- a set of arrays of measurements,
- a continuous set of allowable transformations, and
- a measure of the *joint* similarity of the arrays within the set.

For ease of explication, we introduce the congealing algorithm using binary images as our set of array measurements, the set of positive-determinant affine transformations as the set of allowable transformations, and a sum-of-entropies criterion as the measure of joint similarity of images within the set. However, each of the three choices above can be replaced by a variety of possible choices, leading to a wide range of potential applications of congealing fitting a variety of modeling assumptions. We explore some of these variations in Section 3.

The upper half of Fig. 1 shows two sets of handwritten digits from Special Database 19 created at the National Institute of Standards and Technology (NIST). The images are presented in the size, position, and orientation originally written into forms filled out by volunteers. They vary significantly in size, position, orientation, and degree of slant, and other characteristics. In the bottom half of the figure, each set of digits has been *congealed* to make the set as similar as possible. That is, each character has been transformed using an affine transformation (specific to that character) to make it as similar as possible to the other characters. For the zeros, it is clear that the variability in position, size, and other shape parameters has been dramatically reduced. While not as obvious, the same effect can be seen for the twos on the bottom right. Congealing data sets in this way has a variety of applications, as we shall see below. We now discuss the details of the method for this particular problem.

2.1 Transform Parameterization

As noted above, congealing is defined with respect to a set of transformations. These may be spatial transformations, brightness transformations, color transformations, or other operations on images or arrays that change some feature of interest. We introduce congealing in the context of spatial transformations since these are perhaps the simplest conceptually and most intuitive. There are many choices for sets of spatial transformations as well, but, to begin, we congeal using affine transformations.

We parameterize this set of transforms $\mathcal{T}$ by composing a transform from the following component transforms: x-translation, y-translation, rotation, x-scale, y-scale, x-shear, and y-shear. (We let the x-scale and y-scale parameters represent the logarithm of the actual scale change, which allows us to treat them as additive parameters.) Thus, given a parameter vector $\mathbf{v} = (t_x, t_y, \theta, s_x, s_y, h_x, h_y)$, a transformation matrix U is formed by multiplying the constituent matrices *in a fixed order* (needed to ensure a unique mapping between parameter vectors and the resulting matrices since matrix multiplication is noncommutative):

$$
\begin{aligned}
U &= F(t_x, t_y, \theta, s_x, s_y, h_x, h_y) \\
&= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&\quad \begin{bmatrix} e^{s_x} & 0 & 0 \\ 0 & e^{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & h_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
$$

Note that this is an overcomplete parameterization since there are seven parameters and only six degrees of freedom in the set of transforms.[2] The current goal is to investigate how to make a set of images more similar to each other by independently transforming each one of them in an affine manner. We now describe our similarity objective function.

2. Because we are using a coordinate descent algorithm, extra parameters can allow the algorithm to move "more directly" toward an optimum. For example, while a rotation is not strictly necessary as it can be written as two shearing operations followed by a scaling operation, a single step of the extra rotation parameter will move more directly toward a solution if a pure rotation is what is needed [3].

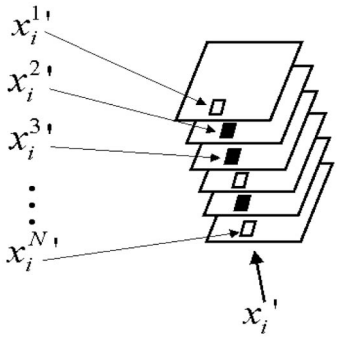

Fig. 2. A *pixel stack* is a collection of pixels drawn from the same location in each of a set of N images. Here, the ith pixel from each of six images forms a pixel stack. Since half of the pixels are black and half are white, this corresponds to a Bernoulli random variable with parameter $p = 0.5$. The entropy of such a random variable is $-(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$ bit.

2.2 Entropy Estimation and Pixel Stacks

Consider a set of N *observed* binary images of a particular class, each image having P pixels. Let the value of the ith pixel in the jth image be denoted x_i^j. Let the image created by transforming the jth image by transform U^j be denoted $I^{j'}$. We assume for the moment that this new image is still binary, which can be achieved, for example, by thresholding it after it is transformed. Let the ith pixel in this transformed image be denoted $x_i^{j'}$. Consider the set of N pixels at a particular image location after each image has undergone some transformation: $\{x_i^{1'}, x_i^{2'}, \ldots, x_i^{N'}\}$. We call this set of N pixels across the images a *pixel stack*. We denote the ith pixel stack of the original image set x_i and the ith pixel stack of a transformed image set x_i'. A pixel stack is illustrated in Fig. 2

This pixel stack can be viewed as a sample from a random variable or pixel process *at a particular location in the image*. We can estimate the entropy, or disorder, of this pixel process by calculating the entropy of the empirical distribution of values in the pixel stack. This is also referred to as the *empirical entropy* (see [4, p. 195]) of the set of values in the pixel stack:[3]

$$\hat{H}(x_i) = -\left(\frac{N_0}{N}\log_2\frac{N_0}{N} + \frac{N_1}{N}\log_2\frac{N_1}{N}\right), \quad (1)$$

where N_0 and N_1 are the number of occurrences of 0 (black) and 1 (white) in the binary-valued pixel stack.[4] The empirical entropy of the pixel stack shown in Fig. 2 is $\hat{H}(x_i') = 1$ bit since there are equal numbers of black and white pixels in the stack. We also refer to pixel stack entropies as pixelwise entropies.

2.3 Congealing as Joint Alignment

The main goal of congealing is to "align," or reduce the variability, in a set of images or other arrays. To achieve this, the idea is to minimize the quantity

$$\sum_{i=1}^{P} \hat{H}(x_i'), \quad (2)$$

the sum of the pixel-stack entropies, by transforming a set of images of a class. Each image is assigned its own vector $\mathbf{v}^j$ of parameters and the criterion (2) is minimized with respect to all components of all of these vectors.

The intuition behind this minimization is as follows: If a set of images *were* aligned as well as possible, we would expect most of the pixel stacks to have low entropy. That is, our notion of alignment can be expressed by the idea that the variability, or entropy, within a pixel stack is low. Thus, the idea of congealing is to drive a set of images, simultaneously, toward this state of low pixel-stack entropies. At convergence, assuming no problems with local minima, we expect the images to be as "aligned" as possible.

Fig. 3 shows the means of sets of twos and zeros before and after congealing. Fig. 3a and Fig. 3c show the mean "0" and mean "2" image at the beginning of the algorithm. The relative abundance of intermediate gray values indicates that many pixel stacks have high entropy since a middle gray value represents a pixel stack composed of half white and half black pixels, with maximum possible entropy. Fig. 3b and Fig. 3d show the pixel stack mean images at the end of the algorithm. Here, we can see that the pixels have distributions that are skewed much more heavily to pure black or pure white and, hence, are of lower entropy. Notice that there is greater entropy represented in the final mean "2" image than in the "0" image due to the fact that zeros can be better aligned through affine transforms than twos can.

One appealing property of congealing is that it defines a notion of central tendency of the data. Often, alignment algorithms proceed by choosing a particular example of a character and then aligning each character to that example. In this case, the aligned characters will be biased toward the original chosen character. Congealing avoids this phenomenon by aligning all characters to each other. Another approach is to align images to a "mean" image. While this can work well in some cases, it does not work well when pixel distributions are bimodal and the mean is not representative of the exemplars in the distribution. Again, congealing avoids this problem by maintaining the true values of the pixels in each image.

2.4 Coherent Transform "Drift" and Parameter Centering

There is a detail that must be addressed, however, to make this idea work in practice. In some cases in which congealing is applied, there may exist a set of transformations, one for each image, which reduces the pixel stack entropies, but, nevertheless, does not align the images as desired. If we restrict the set of transformations to *rigid* transformations, then there is no such degenerate set of transformations. However, for a set of nonrigid transforms, like the affine transforms, shrinking all of the images until the white digits in the center essentially disappear will reduce all of the pixel stack entropies to zero. That is, we are essentially eliminating the information in the images. Since all of the transforms have a shared component (shrinkage), we call this transform "drift."[5]

3. In this formula, $0 \log_2 0$ is interpreted to have value 0.

4. A pixel whose value is between 0 and 1 is interpreted as a mixture of underlying 0 and 1 "subpixels." To extend (1) to handle these pixels, we merely increment each count by the fraction of background and foreground in the mixture. For example, for a 50 percent gray value pixel, we would increment both N_0 and N_1 by 0.5.

5. When congealing using brightness or color transforms, a similar problem arises in which images tend to be darkened until they are black, resulting in low entropy image stacks. This can be avoided using the same techniques discussed in this section, i.e., by forcing the transformation parameters to stay balanced around zero.

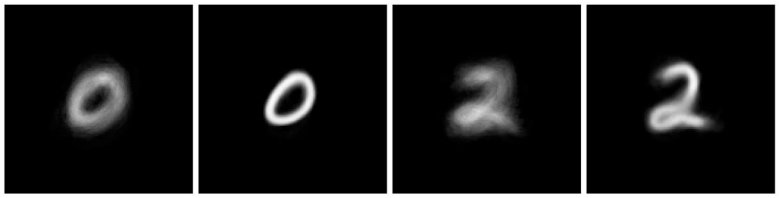

Fig. 3. Mean images during the congealing process. (a) The initial mean image for the set of zeros. The blurriness is due to the misalignment of the images. (b) The final mean image for the set of zeros. The coherence of the aligned images is indicated by the increased sharpness of the image. (c) The initial mean image for the set of twos. (d) The final mean image for the set of twos.

This degenerate case can be avoided in two related ways. One way is to add a term to our alignment criterion in (2) that penalizes large transformations away from the initial position. This can be done by adding a term to the criterion equal to the sum of the squared magnitudes of the transform parameter vectors $\mathbf{v}^j$:

$$E = \sum_{i=1}^{P} \hat{H}(x_i') + \sum_{j=1}^{N} |\mathbf{v}^j|^2. \quad (3)$$

We shall refer to this quantity as the penalized pixelwise entropy.

Using the penalized pixelwise entropies prevents the images from "collapsing" to size zero, but an algorithm based only on this function will still typically, on average, shrink the images somewhat since an average shrinking of the images tends to reduce pixel stack entropies simply by increasing the number of pixel stacks that are completely black. Thus, the penalty in (3) does not completely solve the drift problem.

A second technique for mitigating this shrinkage problem is to constrain each parameter (x-translation, y-log-scale, etc.) to have zero mean *across the image set*. That is, if some images are translated to the left, then other images should be translated to the right; if some images are shrunk, then others should be magnified; etc. This can be thought of as enforcing the property that the "mean transformation" should be the identity transformation.[6]

To incorporate this idea into the congealing algorithm for affine transformations, we "center" the parameter values periodically during the alignment process. That is, we subtract each parameter's mean value from the value of that parameter for each image, pinning the mean of each parameter to zero. This forces the average x-position change, the average shear, the average (log) scale, and the average rotation to be zero across the image sets.

The congealing algorithm for binary images using affine transformations, the penalized entropy (3), and parameter centering are detailed in Algorithm 1, which is shown in Fig. 4. Before moving on to applications of congealing, we consider a probabilistic interpretation of the algorithm.

2.5 Congealing and Factored Models

In this section, we discuss a slightly different interpretation of congealing, as the reversal of a certain generative process for images. In particular, we show how this interpretation leads to independent factor models for images, which in turn leads to a variety of applications. We have presented congealing as a means for removing certain modes of variability, like affine variation, from sets of image data. Rather than simply removing this variability, however, we can develop a model of the variability which can be used to improve simple probabilistic image models.

We start with the observation that most any handwritten digit, when modified by a small affine distortion, would still be interpreted as the same digit. This is true irrespective of the original "style" of the digit, i.e., whether it is a seven with or without a cross-bar, whether it is a two with or without a loop at the base. In other words, the variability of the position, size, orientation, and slant of a character tends to be independent of the particular style of the character.

We can take advantage of this approximate independence of style and "pose" by building a generative model in which these components are assumed to be independent for each given class of characters. Such a generative model is illustrated in Fig. 5.

This model shows the images generated by a process that composes some random "latent image" of a digit, which one can think of as a particular style of that character, with a random transformation. For the following discussion, we let I be an observed image, L be a latent image, and T be a transform. According to the model, the probability of a particular latent image-transform pair (L, T) for a specific class is given as $p(L, T) = p(L)p(T)$ since L and T are assumed to be independent conditioned on the class.

Given an observed image I, however, there are many choices for a latent image-transform pair that explain it. In fact, for any transform T^*, one could argue that the latent image-transform pair was $(T^{*-1}(I), T^*)$, i.e., the inverse of T^* applied to the observed image coupled with T^*.

If we define probability distributions over the latent images and transforms, however, we can ask what is the most likely pair (L, T) that explains a particular image I. Congealing can be seen as a way of simultaneously determining these pairs across an entire image set.

In particular, if we use an independent pixel model to evaluate the probability of an image and a uniform distribution over all transformations, then we can see that congealing finds the maximum likelihood latent images through the following derivation. Let $\mathbf{I}, \mathbf{L}, \mathbf{T}$ be sets of images, latent images, and transforms. Also, let $\mathcal{T}^N$ represent the N-fold product space of sets of transforms. Then, the most likely set of transforms is given by

6. Since the affine transformations do not form a vector space (because the *addition* of two affine transforms is not an affine transform), it is not immediately clear whether a concept such as the mean transformation is well-defined. There are a variety of ways of defining a mean on a space that is not a vector space, such as the Karcher mean.

Algorithm 1 Image Congealing with Affine Transforms

for $j = 1$ to N **do**
  $\mathbf{v}^j \leftarrow \mathbf{0}$ {Set transform parameters for image I^j to 0.}
  $U^j \leftarrow F(\mathbf{v}^j)$ {Set transforms to identity matrix.}
end for
$E \leftarrow \sum_{i=1}^{P} \hat{H}(x_i) + \sum_{j=1}^{N} |\mathbf{v}^j|^2$
$E^* \leftarrow \infty$
while $|E^* - E| > \varepsilon$ **do** {Until convergence...}
  $E^* \leftarrow E$
  for $j = 1$ to N **do** {For each image...}
    for $k = 1$ to 7 **do** {For each affine parameter...}
      $\mathbf{v}_k^{j'} \leftarrow \mathbf{v}_k^j + \delta(k)$ {Try new value of parameter.}
      $U^{j'} \leftarrow F(\mathbf{v}_k^{j'})$ {Compute new transform.}
      $I^{j'} \leftarrow U^{j'}(I^j)$ {Compute im. from new transform.}
      $E_{new} \leftarrow \sum_{i=1}^{P} \hat{H}(x_i') + \sum_{j=1}^{N} |\mathbf{v}^{j'}|^2$
      if $E_{new} < E$ **then**
        $E \leftarrow E_{new}$
        $\mathbf{v}_k^j \leftarrow \mathbf{v}_k^{j'}$
        $U^j \leftarrow U^{j'}$
      else
        $\mathbf{v}_k^{j'} \leftarrow \mathbf{v}_k^j - \delta(k)$ {Try other direction.}
        $U^{j'} \leftarrow F(\mathbf{v}_k^{j'})$
        $I^{j'} \leftarrow U^{j'}(I^j)$
        $E_{new} \leftarrow \sum_{i=1}^{P} \hat{H}(x_i') + \sum_{j=1}^{N} |\mathbf{v}^{j'}|^2$
        if $E_{new} < E$ **then**
          $E \leftarrow E_{new}$
          $\mathbf{v}_k^j \leftarrow \mathbf{v}_k^{j'}$
          $U^j \leftarrow U^{j'}$
        end if
      end if
    end for
  end for
  $\bar{\mathbf{v}} \leftarrow \frac{1}{N} \sum_{j=1}^{N} \mathbf{v}^j$
  for $j = 1$ to N **do**
    $\mathbf{v}^j \leftarrow \mathbf{v}^j - \bar{\mathbf{v}}$ {Adjust params to have 0 mean.}
  end for
end while

Fig. 4. Algorithm 1: Image congealing with affine transforms.

$$
\begin{aligned}
\arg\max_{\mathbf{T}\in\mathcal{T}^{\mathbf{N}}} p(\mathbf{T}|\mathbf{I}) &\overset{(a)}{=} \arg\max_{\mathbf{T}\in\mathcal{T}^{\mathbf{N}}} p(\mathbf{I}|\mathbf{T})p(\mathbf{T}) \\
&\overset{(b)}{=} \arg\max_{\mathbf{T}\in\mathcal{T}^{\mathbf{N}}} p(\mathbf{I}|\mathbf{T}) \\
&\overset{(c)}{=} \arg\max_{\mathbf{T}\in\mathcal{T}^{\mathbf{N}}} p(\mathbf{L}(\mathbf{I},\mathbf{T})) \\
&= \arg\max_{\mathbf{T}\in\mathcal{T}^{\mathbf{N}}} \prod_{i=1}^{P}\prod_{j=1}^{N} p(x_i^{j'}) \\
&= \arg\max_{\mathbf{T}\in\mathcal{T}^{\mathbf{N}}} \sum_{i=1}^{P}\sum_{j=1}^{N} \log p(x_i^{j'}) \\
&\overset{(d)}{\approx} \arg\min_{\mathbf{T}\in\mathcal{T}^{\mathbf{N}}} \sum_{i=1}^{P} H(p(x_i{}')) \\
&= \arg\min_{\mathbf{T}\in\mathcal{T}^{\mathbf{N}}} \sum_{i=1}^{P} H(U^1(x_i^1), U^2(x_i^2), \ldots, U^N(x_i^N)),
\end{aligned}
\tag{4}
$$

where U^i is again the inverse of the transform T^i. (a) follows from Bayes rule, (b) from the uniform prior on transformations, (c) because the latent image is a deterministic function of the observed image if the transform is given, and (d) is just a sample approximation of the entropy. The final expression is equivalent to (2).

Hence, the logarithm of the joint probability of a set of images as defined above is approximately equal to a positive constant times the negative of the sum of the pixelwise entropies. Hence, minimizing the summed pixelwise empirical entropies is approximately equivalent to maximizing the (latent) image probabilities under this model. It is interesting to note that the sum of pixel entropies is an *upper bound* on the entropy of the true image distribution, so minimizing this sum is minimizing an upper bound on the entropy of the images. This is equivalent to maximizing a lower bound on the true likelihood of the images.

Congealing works very well for aligning images even though the model it is based on, independent pixels and uniformly likely transformations, is very crude. One reason that the algorithm works well despite the poorness of the model is related to the concavity of entropy as a function of probability distributions. If a set of latent images are drawn i.i.d. from *any* distribution, then perturbing these images with *any* independent random permuting transformations, i.e., those transformations that only rearrange pixels, will *always* increase the pixel-stack entropies. This is because the distributions of the pixels in the transformed images will be convex combinations of the original pixel distributions and

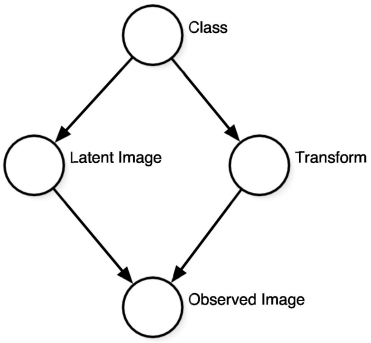

Fig. 5. A directed graphical model representing a general generative image model. The latent image and transform are both dependent on class and they are conditionally independent given the class.

the concavity of entropy guarantees that the resulting "mixed" pixels will have higher entropy. Thus, despite the poorness of the independent pixel model for representing images of digits, the essential property needed for congealing—that removing transform variability will reduce entropy—holds.

Despite the utility of the independent pixel model for "decoupling" the latent images and transforms, it will prove useful to employ more sophisticated models for the latent images and transforms once they have been separated. In the next section, we discuss how a classifier can be built in the congealing framework, using better models for the latent images and transforms once they have been separated.

2.6 A Product Density Classifier

Assume that we wish to build a classifier for handwritten digits, given a set of training examples for each class. To perform classification, we wish to estimate the model with the maximum posterior probability given a new observed image I. Assuming a uniform prior over the classes c_j, we have, using Bayes' rule:

$$\arg\max_j \; P(c_j|I) = \arg\max_j \; p(I|c_j). \tag{5}$$

We introduce the transformation variable T and integrate over it:

$$\arg\max_j \; p(I|c_j) = \arg\max_j \; \int_{T\in\mathcal{T}} p\big(T, I|c_j\big)\, dT \tag{6}$$

$$= \arg\max_j \; \int_{T\in\mathcal{T}} p\big(T|c_j\big)p\big(I|T, c_j\big)\, dT \tag{7}$$

$$= \arg\max_j \; \int_{T\in\mathcal{T}} p\big(T|c_j\big)p\big(L(I,T)|c_j\big)\, dT. \tag{8}$$

Here, dT is the invariant measure for affine transformations, which "measures" each transformation in the infinite set $\mathcal{T}$ equally. (See [5] for details.) Equation (8) follows since the latent image L corresponding to the image I is a deterministic function of I, given the transform T.

We now make the key simplifying assumption that, with high probability,

$$\begin{aligned}&\arg\max_j \; \int_{T\in\mathcal{T}} p\big(T|c_j\big)p\big(L|c_j\big)\, dT \\ &= \arg\max_j \; \max_{T\in\mathcal{T}} \; p\big(T|c_j\big)p\big(L|c_j\big).\end{aligned} \tag{9}$$

This is an approximation to the full Bayesian treatment of the problem and assumes that the joint distribution $p(T, L|c_j)$ peaks sharply around the maximum. This allows us to avoid the integral on the left of (9) and search a full space of possible transforms, rather than a discrete set as some authors have done [1].

To use (9) in a classifier, we must do two things. First, we must build models for the latent image and transform distributions for each class. To do this, we will use the latent images and associated transforms that come from congealing the set of training examples for each class. Second, we must find the maximum likelihood latent image-transform pair for a *test image*, conditioned on each class.

If we congeal a set of training images for a class, such as those in Fig. 1a or Fig. 1b, we get a set of latent images for the class, such as those in Fig. 1c or Fig. 1d. In our experiments, we defined a class conditional probability on latent images to be

$$p(L|c_j) = \frac{1}{Z}\exp\left(-\frac{D^2}{2}\right), \tag{10}$$

where D is the minimum symmetric Hausdorff distance [6] between the argument image L and the latent image set for the class c_j and Z is a normalization constant which is unknown, but is assumed to be approximately the same for each class and is, hence, ignored. Intuitively, the class conditional probability for a latent image is higher if it is closer (in the Hausdorff sense) to some latent image of the training set.

Note that congealing produces not only a sample of latent images, but also, implicitly, a sample of transformations. The transformations used in the congealing process to align images can be thought of as the inverses of the transforms that produce the observed images from the latent images in the generative model. Thus, by taking the matrix inverses of the transforms used in congealing to align images, we get a sample of transforms of the generative process. From this set of transforms—we shall refer to them as T_js—we can estimate a probability density over transforms using a Parzen windows style estimate

$$f(U; T_1, T_2, \ldots, T_N) = \frac{1}{N}\sum_{j=1}^{N} K(U; T_j).$$

The estimate uses a somewhat unusual kernel [5]:

$$K(U;T) = \frac{1}{C(h)}\exp\left(-\frac{1}{2h}\|\log(U^{-1}T)\|_F^2\right),$$

where h is a kernel bandwidth parameter, C is a normalization constant that depends upon the bandwidth, $\log$ is a *matrix logarithm*, and $\|\cdot\|_F$ is the Frobenius norm, the square root of the sum of the products of the matrix components by their complex conjugates. One appealing property of this kernel is that it takes on its maximum value when U is equal to T since $U^{-1}T$ is the identity matrix, which has the minimum matrix log. The kernel is also symmetric in the arguments and invariant when both arguments are transformed by the same affine transformation [5].

Fig. 6. A set of latent image estimates for an observed "8" image. The eight is aligned with each model in turn. Notice that alignment to the "1" model produces a latent image that is a plausible "1." Rejecting this model will depend upon penalizing the highly unlikely transformation that would be paired with such a latent image to produce the observed image.

Armed with a density for latent images and transforms, we now only need to find the latent image and transform of a test image, assuming it comes from a particular class, to evaluate the likelihood of a class. Given a test image of unknown class, we wish to separate it into a latent image and transform. However, this separation will depend upon the class that we assume the image is in. In other words, we will have a separate factorization of a test image into latent image and transform for each class. Fig. 6 shows the latent image estimates for the image of an eight for each different class.

One way to obtain these latent image-transform estimates for a test image is to insert it into the training set for each class. Thus, to get the best latent image-transform for an image assuming it is a zero, insert it into the training set of zeros and congeal the zeros. While this process has a certain elegance about it, it is extremely time-consuming as we must run the congealing algorithm on the *entire training set* for each class for every test character. Fortunately, there is a much more efficient way to "congeal" a test character.

The idea for factoring a test image efficiently is as follows: During the original congealing of the training set for a class, save out the data sets at each stage of the congealing. Then, when congealing a test image, the only image that needs to be updated at each step is the test image. Additional efficiencies in aligning a test example are described in detail here [3].

Given a procedure for splitting a test image into latent image-transform pairs for each class and latent image and transform densities for each class, it is easy to make a classifier for handwritten digits. Such a classifier has reasonably good performance (about 98 percent correct) when trained on 1,000 images of each class. However, the purpose of congealing was not to do digit classification in the presence of abundant training data, but rather to separate observed images into approximately independent factors. We now discuss how this automatic factorization of images can be used to make a digit classifier from just a single example of each class.

2.7 A One Example Classifier

While handwritten digit classifiers based on large training sets are getting very close to human performance, if we examine the performance of classifiers using a small amount of data, in the limit, one example per class, there still seems to be a large gap between the capabilities of machines and humans. Consider the symbol for the new European currency, the "Euro," shown in Fig. 7. After seeing a single example of such a character, humans can recognize the character in a wide variety of contexts, styles, and positions.

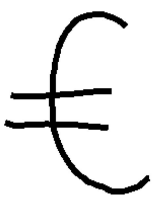

Fig. 7. The new symbol for the standard European currency. After seeing a single example, people typically have no difficulty recognizing the symbol in a wide variety of styles and variations.

Clearly, this is due at least in part to generalization based on previously learned classes. That is, our knowledge about handwriting in general allows us to bring prior knowledge to the formation of our model for a new character based on a single example. A long-standing question in computer vision, and in AI in general, is what form this prior knowledge takes. The original idea behind congealing was to try to extract generic forms of variation from one class or set of classes and to use that knowledge of variation in developing models for other classes from a small number of examples.

To test this idea, we congealed 10 sets of 100 handwritten *letters* from the NIST database. This results in 10 sets of latent images and 10 sets of transforms, leading to 10 separate distributions over transformations. We compared each of these letter transform distributions to another distribution, which was a Gaussian over the coordinates of affine transformations with mean at the identity. We found that each of the letter transform distributions was closer to the other letter transform distributions (in the estimated Kullback-Leibler divergence) than to the Gaussian transform distribution. This suggests (although it is certainly not a rigorous proof) that the letter transform distributions are quite similar to each other [3]. This suggests that the variability described by transformation densities, at least for handwritten characters, is *generic* and can be shared across classes.

Subsequently, we combined transform densities learned from *letters* with a single hand-chosen example of each handwritten *digit* (acting as a crude latent image model for that digit) to produce a model for each digit class. This process is illustrated in Fig. 8. We then ran a handwritten digit

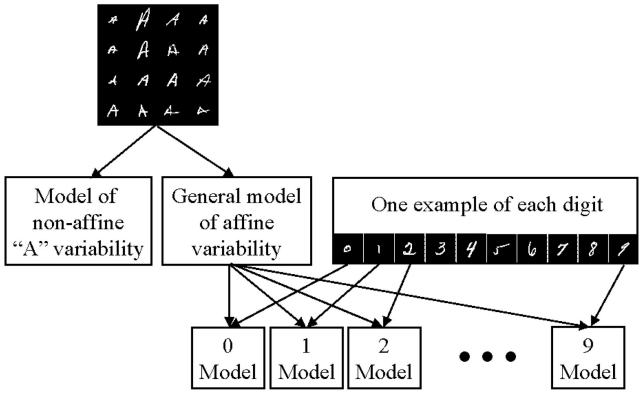

Fig. 8. Diagram of our method for sharing models of affine variability. A support set (shown as a set of "A"s) is given to the learner. From this support set, a general model of affine variability is derived. Combining the model of affine variability with a single example of a handwritten digit, a digit model is constructed.

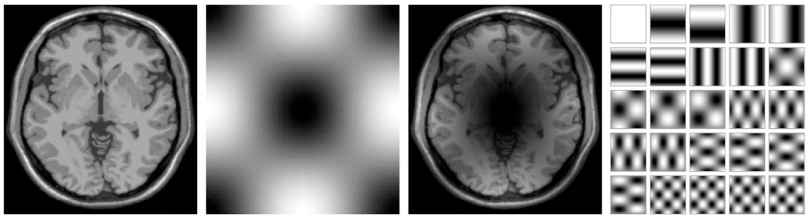

Fig 9. On the left is an idealized mid-axial MR image of the human brain with little or no bias field. The second image is a simulated low-frequency bias field. It has been exaggerated for ease of viewing. The third image is the result of pixelwise multiplication of the image by the bias field. The goal of MR bias correction is to recover the low-bias image on the left from the biased image on the right. On the right is the sine/cosine basis, used to construct band-limited bias fields (see text).

classification task using the single-example models of the digits. The classification accuracy, using the transform densities described in [5], was 89.3 percent. This is a huge improvement over the best alternative method (a nearest neighbor method using the Hausdorff distance) tried which gave an accuracy of only 48.2 percent. Methods that require training to set parameters (neural networks, SVMs, etc.) did substantially worse, severely overfitting to the single training example. Even when the single digit example was chosen randomly, the average performance of our classifier was still 74.7 percent, compared to only 32.6 percent for the nearest neighbor Hausdorff. Additional details of this work can be found in [3].

In summary, the main contribution of congealing to the area of handwriting recognition is *not* record breaking performance on large data sets, but rather the idea that, by modeling independent and generic modes of variability (like affine deformation), new learning tasks can be made more efficient based on previous learning tasks. Next, we use congealing to solve a completely different problem, removing bias from magnetic resonance images.

3 Congealing with Brightness Transformations

In the last section, we introduced congealing by discussing in detail the joint alignment of a set of binary digit images via affine transformations. As mentioned in Section 1, however, congealing can be applied to a wide range of data types using a variety of transformations.

3.1 Bias Correction in MR Images

Our second example of congealing is applied to magnetic resonance images, which are scalar-valued images of human anatomy. The transformations used in this application will be brightness transformations—none of the pixels will be moved spatially—chosen again to minimize the entropies of the pixel stacks in the image set.

The problem of bias fields in magnetic resonance (MR) images is an important problem in medical imaging. This problem is illustrated in Fig. 9. When a patient is imaged in the MR scanner, the goal is to obtain an image which is a function solely of the underlying tissue (left of Fig. 9). However, typically, the desired anatomical image is corrupted by a multiplicative bias field (second image of Fig. 9) that is caused by engineering issues such as imperfections in the radio frequency coils used to record the MR signal. The result is a corrupted image (third image of Fig. 9). (See [7] for background information.) The goal of MR bias correction is to estimate the uncorrupted image from the corrupted image.

A variety of statistical methods have been proposed to address this problem. Wells et al. [8] developed a statistical model using a discrete set of tissues, with the brightness distribution for each tissue type (in a bias-free image) represented by a one-dimensional Gaussian distribution. An expectation-maximization (EM) procedure was then used to simultaneously estimate the bias field, the tissue type, and the residual noise. While this method works well in many cases, it has several drawbacks: 1) Models must be developed a priori for each type of acquisition (for each different setting of the MR scanner), for each new area of the body, and for different patient populations (like infants and adults). 2) Models must be developed from "bias-free" images, which may be difficult or impossible to obtain. 3) The model assumes a fixed number of tissues, which may be inaccurate. For example, during development of the human brain, there is continuous variability between gray matter and white matter. In addition, a discrete tissue model does not handle so-called partial volume effects in which a pixel represents a combination of several tissue types. This occurs frequently since many pixels occur at tissue boundaries.

Nonparametric approaches have also been suggested, as, for example, by Viola [9]. In that work, a nonparametric model of the tissue was developed from a single image. Using the observation that the entropy of the pixel brightness distribution for a *single image* is likely to increase when a bias field is added, Viola's method postulates a bias-correction field by minimizing the entropy of the resulting pixel brightness distribution. This approach addresses several of the problems of fixed-tissue parametric models, but has its own drawbacks: 1) The statistical model may be weak since it is based on data from only a single image. 2) There is no mechanism for distinguishing between certain low-frequency image components and a bias field. That is, the method may mistake signal for noise when removal of the true signal reduces the entropy of the brightness distribution. We shall show that this is a problem in real medical images.

Congealing can overcome or improve upon problems associated with both of these methods and their many variations (see, e.g., [7] for recent techniques). It models tissue brightness nonparametrically, but uses data from multiple images to provide improved distribution estimates and alleviate the need for bias-free images for making a

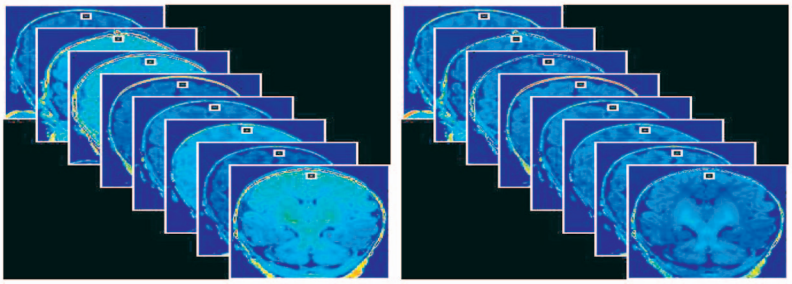

Fig. 10. **On the left** are a set of mid-coronal brain images from eight different infants, showing clear signs of bias fields. A *pixel-stack*, a collection of pixels at the same point in each image, is represented by the small square near the top of each image. Although there are probably no more than two or three tissue types represented by the pixel-stack, the brightness distribution through the pixel-stack has high empirical entropy due to the presence of different bias fields in each image. **On the right** are a set of images that have been corrected using our bias field removal algorithm. While the images are still far from identical, the pixel-stack entropies have been reduced by mapping similar tissues to similar values in an "unsupervised" fashion, i.e., without knowing or estimating the tissue types.

model. It conditions on spatial location, taking advantage of a rich information source ignored in other methods. Experimental results demonstrate the effectiveness of our method.

3.2 The Image Model and Problem Formulation

We assume we are given a set $(I^1, \ldots, I^N)$ of observed images, as shown on the left side of Fig. 10. Each of these images is assumed to be the product of some bias-free image L^j and a smooth bias field $B^j \in \mathcal{B}$.[7] We shall refer to the bias free images as latent images, as before with the affine congealing. The set of all latent images shall be denoted $\mathbf{L}$ and the set of unkown bias fields $\mathbf{B}$. Then, each observed image can be written as the product $I^j(x,y) = L^j(x,y) * B^j(x,y)$, where (x,y) gives the pixel coordinates of each point, with N pixels per image.

Consider again Fig. 10. A pixel-stack through each image set is shown as the set of pixels corresponding to a particular location in each image (not necessarily the same tissue type). Our method again relies on the principle that the pixel-stack values will have lower entropy when the bias fields have been removed. Fig. 11 shows the simulated effect, on the distribution of values in a pixel-stack, of adding different bias fields to each image.

The latent image generation model assumes that each pixel is drawn independently from a fixed distribution, $p_{x,y}(\cdot)$, which gives the probability of each gray value at the the location (x,y) in the image. It is also assumes that the bias fields for each image are chosen independently from some fixed distribution over bias fields. Unlike most models for this problem, which rely on statistical regularities *within* an image, we take a completely orthogonal approach by assuming that pixel values are independent given their image locations, but that pixel-stacks, in general, have low entropy when bias fields are removed.

We formulate the problem as a maximum a posteriori (MAP) problem, searching for the most probable bias fields given the set of observed images. Letting $\mathcal{B}^N$ represent the N-fold product space of smooth bias fields, we wish to find

$$
\begin{aligned}
&\arg\max_{\mathbf{B}\in\mathcal{B}^N} p(\mathbf{B}|\mathbf{I}) \overset{(a)}{=} \arg\max_{\mathbf{B}\in\mathcal{B}^N} p(\mathbf{I}|\mathbf{B})p(\mathbf{B}) \\
&\overset{(b)}{=} \arg\max_{\mathbf{B}\in\mathcal{B}^N} p(\mathbf{I}|\mathbf{B}) \\
&\overset{(c)}{=} \arg\max_{\mathbf{B}\in\mathcal{B}^N} p(\mathbf{L}(\mathbf{I},\mathbf{B})) \\
&= \arg\max_{\mathbf{B}\in\mathcal{B}^N} \prod_{x,y}\prod_{j=1}^{N} p_{x,y}(L^j(x,y)) \\
&= \arg\max_{\mathbf{B}\in\mathcal{B}^N} \sum_{x,y}\sum_{j=1}^{N} \log p_{x,y}(L^j(x,y)) \\
&\overset{(d)}{\approx} \arg\min_{\mathbf{B}\in\mathcal{B}^N} \sum_{x,y} H(p_{x,y}) \\
&\overset{(e)}{\approx} \arg\min_{\mathbf{B}\in\mathcal{B}^N} \sum_{x,y} \hat{H}_{\text{Vasicek}}(L^1(x,y),\ldots,L^N(x,y)) \\
&= \arg\min_{\mathbf{B}\in\mathcal{B}^N} \sum_{x,y} \hat{H}_{\text{Vasicek}}\left(\frac{I^1(x,y)}{B^1(x,y)},\ldots,\frac{I^N(x,y)}{B^N(x,y)}\right).
\end{aligned}
\tag{11}
$$

Here, H is the Shannon entropy ($-E(\log p(x))$) and $\hat{H}_{\text{Vasicek}}$ is a sample-based entropy estimator. This estimator allows the rapid estimation of entropy from a sample of a random variable, without first estimating the distribution itself from the data, as is often done.[8] (a), (b), and (c) are equivalent to the steps taken in derivation (4). The approximation (d) replaces the empirical mean of the log probability at each pixel with the negative entropy of the underlying distribution at that pixel. This entropy is in turn estimated, (e), using the entropy estimator of Vasicek [11] directly from the samples in the pixel-stack.

Equation (11) is once again the basic optimization of congealing, as in (2), although this time we are calculating the entropy of a continuous distribution of gray values, rather than the entropy of a distribution with only two values. To do congealing with this criterion, we simply need to parameterize the smooth bias fields and to optimize the criterion with respect to these parameters.

7. We assume that the images have been brought into *approximate* correspondence using spatial congealing (spatial congealing is relatively robust to bias fields) or any other method of alignment, such as [10].

8. The entropy estimator used is similar to Vasicek's estimator [11], given (up to minor details) by

$$\hat{H}_{\text{Vasicek}}(Z^1,\ldots,Z^N) = \frac{1}{N-m}\sum_{i=1}^{N-m}\log\left(\frac{N}{m}(Z^{(i+m)} - Z^{(i)})\right),$$

where Z^is represent the values in a pixel-stack, $Z^{(i)}$s represent those same values in rank order, N is the number of values in the pixel-stack, and m is a function of N (like $N^{0.5}$) such that m/N goes to 0 as m and N go to infinity. These entropy estimators are discussed at length elsewhere [12].

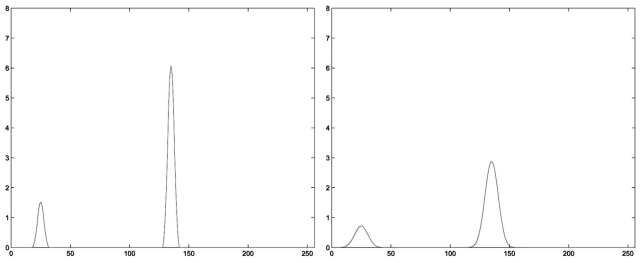

Fig. 11. **On the left** is a simulated distribution from a pixel-stack taken through a particular set of bias-free mid-axial MR images. The two sharp peaks in the brightness distribution represent two tissues which are commonly found at that particular pixel location. **On the right** is the result of adding an independent bias field to each image. In particular, the spread, or entropy, of the pixel distribution increases. In this work, we seek to remove bias fields by seeking to reduce the entropy of the pixel-stack distribution to its original state.

3.3 The Algorithm

Using these ideas, it is straightforward to construct algorithms for joint bias field removal. As mentioned above, we chose to optimize (11) over the set of band-limited bias fields. To do this, we parameterize the set of bias fields using the sine/cosine basis images ϕ_k shown on the right of Fig. 9:

$$B^j(x,y) = \sum_{k=1}^{25} \alpha_k^j \phi_k(x,y).$$

We optimize (11) by *simultaneously* updating the bias field estimates (taking a step along the numerical gradient) for each image to reduce the overall entropy. That is, at time step t, the coefficients α_k^js for each bias field are updated using the latent image estimates and entropy estimates from time step $t-1$. After all αs have been updated, a new set of latent images and pixel-stack entropies is calculated and another gradient step is taken.

Though it is possible to do a full gradient descent to convergence by optimizing one image at a time, the optimization landscape tends to have more local minima for the last few images in the process. The appeal of our joint gradient descent method, on the other hand, is that the ensemble of images provides a natural smoothing of the optimization landscape in the joint process. Algorithm 2, shown in Fig. 12, summarizes the process.

Upon convergence, it is assumed that the entropy has been reduced as much as possible by changing the bias fields unless one or more of the gradient descents is stuck in a local minimum. Empirically, the likelihood of sticking in local minima is dramatically reduced by increasing the number of images N in the optimization. In our experiments described below, with only 21 real infant brains, the algorithm appears to have found a global minimum of all bias fields, at least to the extent that this can be discerned visually.

Note that, for a set of *identical* images, the pixel-stack entropies are not increased by multiplying each image by the same bias field (since all images will still be the same). More generally, when images are approximately equivalent, their pixel-stack entropies are not significantly affected by a "common" bias field, i.e., one that occurs in all of the images.[9] This means that the algorithm cannot, in general, eliminate all bias fields from a set of images, but can only *set all of the bias fields to be equivalent.* We refer to any constant bias field remaining in all of the images after convergence as the *residual bias field*.

Fortunately, there is an effect that tends to minimize the impact of the residual bias field in many test cases. In particular, the residual bias field tends to consist of components for each α_j that approximate the mean of that component across images. For example, if half of the observed images have a positive value for a particular component's coefficient and half have a negative coefficient for that component, the residual bias field will tend to have a coefficient near zero for that component. Hence, the algorithm naturally eliminates bias field effects that are nonsystematic, i.e., that are not shared across images.

If the same type of bias field component occurs in a majority of the images, then the algorithm will not remove it as the component is indistinguishable, under our model, from the underlying anatomy. In such a case, one could resort to within-image methods to further reduce the entropy. However, there is a risk that such methods will remove components that actually represent smooth gradations in the anatomy. This can be seen in the bottom third of Fig. 13 and will be discussed in more detail below.

3.4 Experiments

To test our bias removal algorithm, we ran two sets of experiments, the first on synthetic images for validation and the second on real brain images. We obtained synthetic brain images from the BrainWeb project [13] such as the one shown on the left of Fig. 9. These images can be considered "idealized" MR images in the sense that the brightness values for each tissue are constant (up to a small amount of manually added isotropic noise). That is, they contain no bias fields. The initial goal was to ensure that our algorithm could remove synthetically added bias fields in which the bias field coefficients were known. Using N copies of a single "latent" image, we added known but different bias fields to each one. For as few as five images, we could reliably recover the known bias field coefficients, up to a fixed offset for each image, to within 1 percent of the power of the original bias coefficients.

More interesting are the results on real images, in which the latent images come from different patients. We obtained 21 preregistered[10] infant brain images (top of Fig. 13) from Brigham and Women's Hospital in Boston, Massachusetts. Large bias fields can be seen in many of the images. Probably the most striking is a "ramp-like" bias field in the sixth image of the second row. (The top of the brain is too bright, while the bottom is too dark.) Because the brain's white matter is not fully developed in these infant scans, it is difficult to categorize tissues into a fixed number of classes as is typically done for adult brain images; hence, these images are not amenable to methods based on specific tissue models developed for adults (e.g., [8]).

The middle third of Fig. 13 shows the results of our algorithm on the infant brain images. (These results must be

9. Actually, multiplying each image by a bias field of small magnitude can artificially reduce the entropy of a pixel-stack, but this is only the result of the brightness values shrinking toward zero. Such artificial reductions in entropy can be avoided by normalizing a distribution to unit variance between iterations of computing its entropy, as is done in this work.

10. It is interesting to note that registration is not strictly necessary for this algorithm to work. The proposed MAP method works under very broad conditions, the main condition being that the bias fields do not span the same vector space as parts of the actual medical images. It is true, however, that, as the latent images become less registered or differ in other ways, a much larger number of images is needed to get good estimates of the pixel-stack distributions.

Algorithm 2 Congealing with Brightness Transformations

for $j = 1$ to N **do**
  $\alpha^j \leftarrow \mathbf{0}$ {Set transform parameters for image I^j to 0.}
  $\alpha_1^j \leftarrow 1$ {Set DC offset parameter to 1 for each image.}
end for
$E \leftarrow \sum_{(x,y)} \hat{H}_{Vasicek} L(x,y)$
$E^* \leftarrow \infty$
while $|E^* - E| > \varepsilon$ **do** {Until convergence...}
  $E^* \leftarrow E$
  for $j = 1$ to N **do** {For each image...}
    Compute $\nabla_{\alpha^j} E$.
    $\alpha^j \leftarrow \alpha^j + \delta \nabla_{\alpha^j} E$
  end for
  $\bar{\alpha} \leftarrow \frac{1}{N} \sum_{j=1}^{N} \alpha^j$
  for $j = 1$ to N **do**
    $\alpha^j \leftarrow \alpha^j - \bar{\alpha}$ {Adjust params to have 0 mean.}
    $\alpha_1^j \leftarrow \alpha_1^j + 1$ {DC bias field maintains avg. of 1.}
  end for
end while

Fig. 12. Algorithm 2: Congealing with brightness transformations.

viewed in color on a good monitor to fully appreciate the results.) While a trained technician can see small imperfections in these images, the results are remarkably good. All major bias artifacts have been removed.

It is interesting to compare these results to a method that reduces the entropy of each image individually, without using constraints between images. Using the results of our algorithm as a starting point, we continued to reduce the entropy of the pixels *within* each image (using a method akin to Viola's [9]), rather than across images. These results are shown in the bottom third of Fig. 13.[11] Carefully comparing the central brain regions in the middle section of the figure and the bottom section of the figure, one can see that the butterfly shaped region in the middle of the brain, which represents developing white matter, has been suppressed in the lower images. This is most likely because the entropy of the pixels *within a particular image* can be reduced by increasing the bias field "correction" in the central part of the image. In other words, the algorithm strives to make the image more uniform by removing the bright part in the middle of the image. However, our algorithm, which compares pixels across images, does not suppress these real structures since they occur across images. Hence, coupling across images can produce superior results.

We have now seen congealing in two very different scenarios. This second scenario differs from the first in two significant ways. First, the transformations considered to "align" images are brightness transformations rather than spatial transformations. Second, the images are scalar-valued instead of binary-valued, requiring a different entropy estimator than used for the binary digits.

One more difference between the algorithms is worth mentioning. The binary digit algorithm used a coordinate descent method in the optimization where the optimization criterion was explicitly evaluated for small perturbations of the parameters in both the positive and negative directions. The reason for this is that the optimization function is not a very smooth function of the parameters since a small change in the scale or rotation of an image can produce a discontinuous jump in the sum of entropies function. The reason for this is that the binary digit images have sharp edges and, so, a small transformation can cause a discrete jump in the value of a pixel at a particular location. The brightness-based congealing of the MR images, however, produces a much smoother optimization landscape and, so, somewhat faster gradient descent methods can be used in the optimization. Other than these somewhat minor differences, the basic algorithms are almost identical, however.

This work uses information unused in other methods, i.e., information across images. This suggests an iterative scheme in which both types of information, both within and across images, are used. Local models could be based on weighted neighborhoods of pixels, *pixel cylinders*, rather than single pixel-stacks, in sparse data scenarios. For "easy" bias correction problems, such an approach may be overkill, but, for difficult problems in bias correction, where the bias field is difficult to separate from the underlying tissue, as discussed in [7], such an approach could produce critical extra leverage.

4 Additional Related Work

Although we have generalized the scope of congealing, it was originally proposed [2] as a way of dealing with spatial variability in images. There have been numerous other efforts to address shape variability in images using a latent image-transform type of approach.

Much work in handwritten character recognition has explicitly addressed the issue of shape deformations and modeling. Revow et al. [14] developed a model for characters by hand-specifying and then adapting a set of "control points," defining the probability of an observed character as a function of the deviation of the character from these predefined control points. The authors discounted the affine component of such deviations, achieving an affine

11. This particular result is not visible on a grayscale printout of the paper and needs to be viewed in color on a high-fidelity monitor.

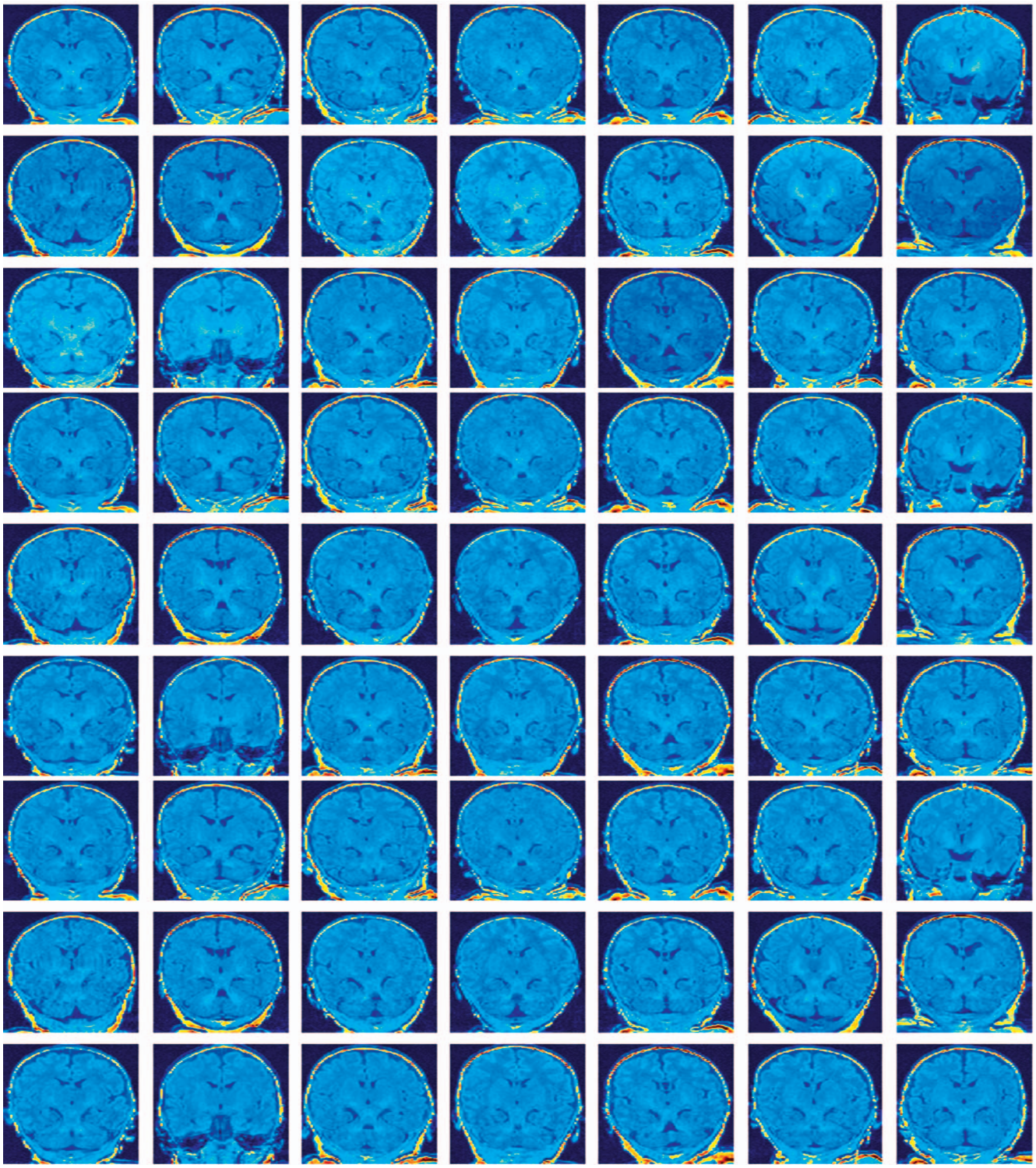

Fig. 13. **NOTE**: This image must be viewed in color (preferably on a bright display) for full effect. (Images can be viewed online at http://www.cs.umass.edu/elm/congealing.) **Top.** Original infant brain images. **Middle.** The same images after bias removal with our algorithm. Note that developing white matter (butterfly-like structures in middle brain) is well-preserved. **Bottom.** Bias removal using a single image based algorithm. Notice that white matter structures are repressed.

invariant character recognition model. In active appearance models [15], the authors describe a system which models observed images as a combination of "shape" variations and "appearance" variations. These terms correspond to the transform and latent image components of the model we use. While these models have a number of interesting potential applications in face recognition, tracking, and other vision tasks, the models are built upon manually identified correspondences. We shall focus our attention on models that can be learned automatically.

Jones and Poggio have presented models for images that consist of separate "texture" and "shape" components [16].

These parts correspond to the latent image and transform components used in this paper. The shape components in these models are linear models. That is, deformation vector fields are combined linearly to produce variations in the shape of latent images. Separating an observed image into its shape and texture parts is termed vectorization by the authors. These models were initially developed for images of faces by manually identifying correspondences among all of the prototype faces. The "flows" that put these faces in correspondence then provide the model of deformation with which one can do synthesis and analysis.

In [17], the authors introduce a technique for "bootstrapping" these correspondences by iteratively estimating more and more refined models of texture and shape with the aid of an optical flow algorithm. The joint nature of this optimization makes it quite similar to congealing. There are several important differences as well. Unlike in congealing, the deformation model is unconstrained. That is, virtually any flow field can arise from the algorithm. This makes the procedure both more flexible and more difficult to get working correctly. Another difference is that the authors do not explicitly define an optimization criterion other than the subjective one of good visual alignment. Nevertheless, the algorithm is ultimately very similar to one which minimizes the entropies of the final latent images, like congealing.

The work of Frey and Jojic [18], [19] has the greatest similarity to congealing. In these papers and in more recent work, the authors use the generative latent image-transform model of image production. They produce models of latent images by simultaneously maximizing the posterior likelihood of a set of latent images under a fixed set of transformations.

The authors use the EM algorithm to maximize the likelihood of the latent images simultaneously under a set of models. One key difference with our own work is that the authors entertain a finite set of transformations rather than a continuous set. This allows the authors to perform a full Bayesian analysis at each step, calculating the likelihood of an observed image under a particular model by integrating over all possible latent images and transforms. Hence, they do not need to resort to using the "peakiness assumption" of (9).

However, with a fixed set of transforms, the number of modes of spatial deformations that can be modeled is limited. The complexity of their algorithm is linear in the number of possible transformations, whereas congealing is linear in the *number of parameters*. This means congealing can be used with much larger sets of transforms and with unbounded resolution within each parameter. This means that congealing can achieve potentially more accurate alignments.

Another key difference between our work and the work of Frey and Jojic is that we model pixel distributions nonparametrically. This allows us to define accurate models of latent image pixels even when they may be bimodal or even more complicated. For example, in doing alignment or bias removal in MR images, pixel distributions are likely to have many modes, even after alignment, due to the different tissues that may appear at the same location in different brains.

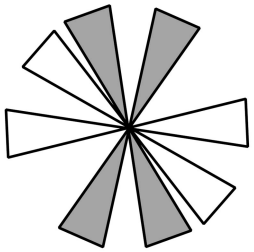

Fig. 14. An example of two images that will not congeal to the global minimum of the fitness function. This problem can usually be alleviated in practice by congealing a large number of samples simultaneously.

5 Other Properties of Congealing

Congealing has a number of appealing properties. We discuss below the issues of convergence to an optimum of the alignment function and robustness to noise in the images.

One problem with iterative methods such as congealing is that an image may fail to achieve the global minimum of the objective function.[12] This can be caused by the so-called "zero-gradient" problem or the existence of local minima in the objective function. An example of the zero-gradient problem is shown in Fig. 14. Note that the gray "X" and the white "X" do not overlap at all, despite the fact that their centroids are aligned. Thus, a differential change in the relative rotation of the two characters will not improve alignment according to the minimum entropy cost function. The figure illustrates a local minimum problem as well. It arises when one leg of an "X" overlaps a leg of the other "X," while the other legs do not overlap. In such a scenario, any perturbation of the rotation parameter would only increase the entropy. This scenario thus represents a local minimum of the entropy function.

The congealing process has a serendipitous advantage in that it often circumvents these two types of optimization problems. Because the alignment process is done over an ensemble of images which has a data-dependent smoothing effect, these two issues arise infrequently. This can be understood by reexamining the average observed images of Fig. 3, which show the relatively smooth "landscape" for hill-climbing in the congealing setting.

We note that, for aligning a pair of images, a strategy of blurring one of the images is commonly used [21], [22]. For binary images, this can be thought of as a type of implicit congealing over horizontal and vertical translations since convolving an image with a circular Gaussian distribution is equivalent to averaging a set of equivalent latent images that have been shifted horizontally and vertically according to a Gaussian distribution. Congealing improves upon this method by using the true distribution over transforms as a

12. Methods which optimize over a discrete set of transforms (e.g., [20], [1]) of course are guaranteed to reach a minimum over their set of defined transformations, but the best transform may not be close to the true global optimum. A combined approach optimization approach can be used in which a gradient descent is continued from the best discrete transform.

TABLE 1
Percentages of Images that Do *Not* Reach Global Minimum of the Probability Function

Number of Images	Percent trapped in local minimum									
	0	1	2	3	4	5	6	7	8	9
2	50	0	0	0	0	0	50	0	0	0
10	0	0	10	0	0	0	0	0	10	0
100	0	0	0	1	0	4	1	1	4	1
1000	0.6	0	0.9	0.7	0.3	1.1	0.3	0.0	0.0	0.6

(a) (b) (c) (d)

Fig. 15. (a) and (b) Two distinct centers of convergence for a set of rotated "4" images. The algorithm aligned the horizontal part of some fours with the vertical part of others and got stuck in this local minimum. However, since any test character which happens to be a four should rotate to one of these two positions, this can still make a good model for classification. (c) and (d) Two centers for different "8" images.

"convolution kernel." In addition, a blurring technique used on scalar-valued, color-valued, or feature-valued images may not achieve the desired effect since the average of two values may not be a good representation of a bimodal distribution. Thus, for anything other than binary images, it is not clear how to interpret a blurring operation. A limitation of congealing, however, is that enough images must be present to form a good approximation of the distribution.

To study the problems of local minima and zero gradients, the following experiments were performed: Training sets of four different sizes were congealed for each digit. The number of characters which failed to converge to the best alignment was evaluated. This judgment was made subjectively, based upon whether a human observer (the author) could find a better alignment of the character. The results are reported in Table 1. When a small number of examples is used in congealing, the lack of sufficient smoothing causes a greater number of local minima problems, as shown in the first two rows of the table.

Another phenomenon may occur when the observed data points are spread widely apart. In this case, congealing may produce multiple convergence centers rather than a single center. The following experiment was done to examine this issue more systematically: Starting with a single image of a "4" from the NIST database, we generated a sequence of 100 images rotated at uniform intervals from $-\frac{\theta}{2}$ to $\frac{\theta}{2}$. For $\theta < 68$ degrees, the images congealed to a unique position, but, when $\theta > 68$ degrees, two "centers" emerged. This is due to a local minimum in the congealing process, as illustrated in Fig. 15. Although this lack of convergence to a single global "center" is not ideal, it does not preclude us from using the resulting density model, which has relatively low entropy. That is, even in the presence of multiple convergent "centers," we are performing an important dimensionality reduction in the data by congealing. The key property is that a test character will be congealed to a predictable location for comparison with the model, without losing information about the character. Such multiple convergence centers were seen in the actual training data in the case of the class of eights. An example of each latent image is shown in Fig. 15c and Fig. 15d. A number of other experiments were conducted to compare the congealing algorithm to other preprocessing algorithms for aligning digits. For the sake of brevity, we refer the reader to [3] and do not replicate it here. The same reference contains a discussion of the robustness of congealing to noise, as well.

6 Additional Applications

In Section 2, we defined congealing as a general procedure for jointly aligning arrays of data with respect to a set of continuous transformations. This broad definition encompasses a wide range of data types and applications. It is straightforward to apply the ideas of congealing to time sequence data, like electroencephalograms (EEGs), to eliminate location and scale parameters. In this case, transformations could be linear or nonlinear remappings of the time coordinate. Another example of time-sequence data is data taken from a pen-tablet as an ordered sequence of coordinate pairs. One problem with modeling characters such as these is that different writers may draw characters that appear very similar, but at different speeds. If data is stored and indexed with a time coordinate, then the similarity of characters may be obscured. Congealing can be used to "realign" the time coordinates by minimizing the entropies of sample locations, across characters, by "warping" time according to either a linear or nonlinear transformation.

As we have seen already, congealing can be used to align two-dimensional images and to remove brightness variations from two-dimensional images. Both of these techniques can be applied to full three-dimensional volumes, although the computational challenges become greater. In [3], we report on the alignment of three-dimensional binary brain volumes using congealing. The same techniques can be applied to gray-valued 3D medical volumes.

There are three primary directions for our work in extending these techniques. One of these is to apply congealing to color images. To do this, we must consider the entropy of pixel stacks in which each pixel is a three-dimensional random variable, rather than simply a one-dimensional variable. We have recently implemented fast algorithms for estimating the entropy of three-dimensional distributions from samples, so this is now within reach.

Second, we wish to extend congealing to feature spaces. In particular, we aim to apply these ideas to images of edge-strengths and other filter outputs. Part of the problem in the

past with making this work is that slight misalignments in edge images would cause the congealing to get stuck. In other words, edges are so thin and their overlap from image to image is so small that congealing didn't work well. This can be addressed by our third major line of investigation.

The third idea we hope to investigate was mentioned in the context of the MR bias removal problem. In order to form better estimates of pixel stack distributions when few images are available, estimates could be built from the *neighborhood* surrounding a pixel, i.e., by using *pixel cylinders* rather than *pixel stacks*. Integration over neighborhoods provides a gradient for the alignment algorithm without destroying information, as would be done by blurring images. We believe these augmentations of our ideas will increase the applicability of congealing algorithms. Our current goal is to apply these ideas to alignment and structure learning in face images.

ACKNOWLEDGMENTS

T. Inder and S. Warfield provided brain images for this work (US National Institutes of Health grant P41 RR13218).

REFERENCES

[1] B.J. Frey and N. Jojic, "Transformation-Invariant Clustering and Dimensionality Reduction Using EM," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 1, pp. 1-17, Jan. 2003.
[2] E. Miller, N. Matsakis, and P. Viola, "Learning from One Example through Shared Densities on Transforms," *Proc. IEEE Computer Vision and Pattern Recognition Conf.,* 2000.
[3] E.G. Miller, "Learning from One Example in Machine Vision by Sharing Probability Densities," PhD thesis, Massachusetts Inst. of Technology, 2002, http://www.cs.umass.edu/~elm/papers/thesis.pdf.
[4] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* John Wiley & Sons, 1991.
[5] E.G. Miller and C. Chefd'hotel, "Practical Non-Parametric Density Estimation on a Transformation Group for Vision," *Proc. IEEE Computer Vision and Pattern Recognition Conf.,* 2003.
[6] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *Trans. IEEE Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 850-863, Sept. 1993.
[7] A.C. Fan, "A Variational Approach to MR Bias Correction," MS thesis, Massachusetts Inst. of Technology, 2003.
[8] W.M. Wells, W.E.L. Grimson, R. Kikinis, and F. Jolesz, "Adaptive Segmentation of MRI Data," *IEEE Trans. Medical Imaging,* vol. 15, pp. 429-442, 1996.
[9] P.A. Viola, "Alignment by Maximization of Mutual Information," PhD thesis, Massachusetts Inst. of Technology, 1995, ftp://publications.ai.mit. edu/ai-publications/pdf/AITR-1548.pdf.
[10] P. Viola and W.M. Wells III, "Mutual Information: An Approach for the Registration of Object Models and Images," *Int'l J. Computer Vision,* 1997.
[11] O. Vasicek, "A Test for Normality Based on Sample Entropy," *J. Royal Statistical Soc.,* Series B, vol. 38, no. 1, pp. 54-59, 1976.
[12] E.G. Learned-Miller and J.W. Fisher, "ICA Using Spacings Estimates of Entropy," *J. Machine Learning Research,* vol. 4, pp. 1271-1295, 2003.
[13] D.L. Collins, A.P. Zijdenbos, J.G. Kollokian, N.J. Sled, C.J. Kabani, C.J. Holmes, and A.C. Evans, "Design and Construction of a Realistic Digital Brain Phantom," *IEEE Trans. Medical Imaging,* vol. 17, pp. 463-468, 1998.
[14] M. Revow, C. Williams, and G. Hinton, "Using Generative Models for Handwritten Digit Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 6, pp. 592-606, June 1996.
[15] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 681-685, June 2001.
[16] M. Jones and T. Poggio, "Model-Based Matching by Linear Combinations of Prototypes," Technical Report AI Memo 1583, Massachusetts Inst. of Technology, 1995, ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-1583.pdf.
[17] T. Vetter, M. Jones, and T. Poggio, "A Bootstrapping Algorithm for Learning Linear Models of Object Classes," *Proc. IEEE Computer Vision and Pattern Recognition Conf.,* pp. 40-46, 1997.
[18] B. Frey and N. Jojic, "Estimating Mixture Models of Images and Inferring Spatial Transformations Using the EM Algorithm," *Proc. IEEE Computer Vision and Pattern Recognition Conf.,* pp. 416-422, 1999.
[19] B. Frey and N. Jojic, "Transformed Component Analysis: Joint Estimation of Spatial Transformations and Image Components," *Proc. Int'l Conf. Computer Vision,* 1999.
[20] C. Bishop, M. Svensén, and C. Williams, "GTM: The Generative Topographic Mapping," *Neural Computation,* vol. 10, no. 1, pp. 215-234, 1998.
[21] P. Simard, Y. LeCun, and J. Denker, "Efficient Pattern Recognition Using a New Transformation Distance," *Proc. Advanced in Neural Information Processing Systems 5,* pp. 51-58, 1993.
[22] N. Vasconcelos and A. Lippman, "Multiresolution Tangent Distance for Affine-Invariant Classification," *Proc. Advanced in Neural Information Processing Systems 10,* 1998.

Erik G. Learned-Miller (previously Erik G. Miller) received the BA degree in psychology from Yale University (1988) and the MS (1997) and PhD (2002) degrees from the Massachusetts Institute of Technology in electrical engineering and computer science. He is an assistant professor of computer science at the University of Massachusetts, Amherst. He spent two years as a postdoctoral researcher at the University of California, Berkeley, in the Computer Science Division. He also has seven years of experience developing and marketing software for medical imaging and neurosurgery applications.