International Journal of Computer Vision 24(2), 137–154 (1997)

Alignment by Maximization of Mutual Information

PAUL VIOLA
Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA 02139
viola@ai.mit.edu

WILLIAM M. WELLS III
Massachusetts Institute of Technology, Artificial Intelligence Laboratory; and Harvard Medical School and Brigham and Women's Hospital, Department of Radiology
sw@ai.mit.edu

Received August, 1995; Accepted September, 1995

Abstract. A new information-theoretic approach is presented for finding the pose of an object in an image. The technique does not require information about the surface properties of the object, besides its shape, and is robust with respect to variations of illumination. In our derivation few assumptions are made about the nature of the imaging process. As a result the algorithms are quite general and may foreseeably be used in a wide variety of imaging situations.

Experiments are presented that demonstrate the approach registering magnetic resonance (MR) images, aligning a complex 3D object model to real scenes including clutter and occlusion, tracking a human head in a video sequence and aligning a view-based 2D object model to real images.

The method is based on a formulation of the mutual information between the model and the image. As applied here the technique is intensity-based, rather than feature-based. It works well in domains where edge or gradient-magnitude based methods have difficulty, yet it is more robust than traditional correlation. Additionally, it has an efficient implementation that is based on stochastic approximation.

1. Introduction

In object recognition and image registration there is a need to find and evaluate the alignment of model and image data. It has been difficult to find a suitable metric for this comparison. In other applications, such as medical imaging, data from one type of sensor must be aligned with that from another. We will present an information theoretic approach that can be used to solve such problems. Our approach makes few assumptions about the nature of the imaging process. As a result the algorithms are quite general and may foreseeably be used with a wide variety of sensors. We will show that this technique makes many of the difficult problems of model comparison easier, including accommodation of the vagaries of illumination and reflectance.

The general problem of alignment entails comparing a predicted image of an object with an actual image. Given an object model and a pose (coordinate transformation), a model for the imaging process could be used to predict the image that will result. The predicted image can then be compared to the actual image directly. If the object model and pose are correct the predicted and actual images should be identical, or close to it. Of course finding the correct alignment is still a remaining challenge.

The relationship between an object model (no matter how accurate) and the object's image is a complex one. The appearance of a small patch of a surface is a function of the surface properties, the patch's orientation, the position of the lights and the position of the observer. Given a model $u(x)$ and an image $v(y)$ we can

formulate an imaging equation,

$$v(T(x)) = F(u(x), q) + \eta \tag{1}$$

or equivalently,

$$v(y) = F(u(T^{-1}(y)), q) + \eta. \tag{2}$$

The imaging equation has three distinct components. The first component is called a transformation, or pose, denoted T. It relates the coordinate frame of the model to the coordinate frame of the image. The transformation tells us which point in the model is responsible for a particular point in the image. The second component is the imaging function, $F(u(x), q)$. The imaging function determines the value of image point $v(T(x))$. In general a pixel's value may be a function both of the model and other exogenous factors. For example an image of a three dimensional object depends not only on the object but also on the lighting. The parameter q collects all of the exogenous influences into a single vector. Finally, η is a random variable that models noise in the imaging process. In most cases the noise is assumed Gaussian.

Alignment can be a difficult problem for a number of reasons:

- F, the imaging function of the physical world, can be difficult to model.
- q, the exogenous parameters, are not necessarily known and can be difficult to find. For example computing the lighting in an image is a non-trivial problem.
- T, the space of transformations, which may have many dimensions, is difficult to search. Rigid objects often have a 6 dimensional transformation space. Non-rigid objects can in principle have an unbounded number of pose parameters.

One reason that it is, in principle, possible to define F is that the image does convey information about the model. Clearly if there were no mutual information between u and v, there could be no meaningful F. We propose to finesse the problem of finding and computing F and q by dealing with this mutual information directly. We will present an algorithm that aligns by maximizing the mutual information between model and image. It requires no a priori model of the relationship between surface properties and scene intensities—it only assumes that the model tells more about the scene when it is correctly aligned.

Though the abstract suggestion that mutual information plays a role in object recognition may not be new, to date no concrete representations or efficient algorithms have been proposed. This paper will present a new approach for evaluating entropy and mutual information called EMMA[1]. It is distinguished in two ways: 1) EMMA does not require a prior model for the functional form of the distribution of the data; 2) entropy can be maximized (or minimized) efficiently using stochastic approximation.

In its full generality, EMMA can be used whenever there is a need to align images from two different sensors, the so-called "sensor fusion" problem. For example, in medical imaging data from one type of sensor (such as magnetic resonance imaging—MRI) must be aligned to data from another sensor (such as computed tomography—CT).

2. An Alignment Example

One of the alignment problems that we will address involves finding the pose of a three-dimensional object that appears in a video image. This problem involves comparing two very different kinds of representations: a three-dimensional model of the shape of the object and a video image of that object. For example, Fig. 1 contains a video image of an example object on the left and a depth map of that same object on the right (the object in question is a person's head: RK). A depth map is an image that displays the depth from the camera to every visible point on the object model.

From the depth map alone it might be difficult to see that the image and the model are aligned. For a human observer, the task can be made much easier by simulating the imaging process and rendering an image from the 3D model. Figure 2 contains two renderings of the object model. These synthetic images are constructed assuming that the 3D model has a Lambertian surface and that the model is illuminated from the right. It is almost immediately obvious that the model on the left of the figure is more closely aligned to the video image than the model on the right. Unfortunately, what might seem like a trivial determination is difficult to reproduce with a computer. The task is made difficult because the intensities of the true video image and the synthetic images are quite different. In fact, the pixels of the real image and the correct model image are uncorrelated. Somehow, the human visual system is capable of ignoring the superficial differences that arise from changes in illumination and surface properties.

Figure 1. Two different views of RK. On the left is a video image. On the right is a depth map of a model of RK that describes the distance to each of the visible points of the model. Closer points are rendered brighter than more distant ones.

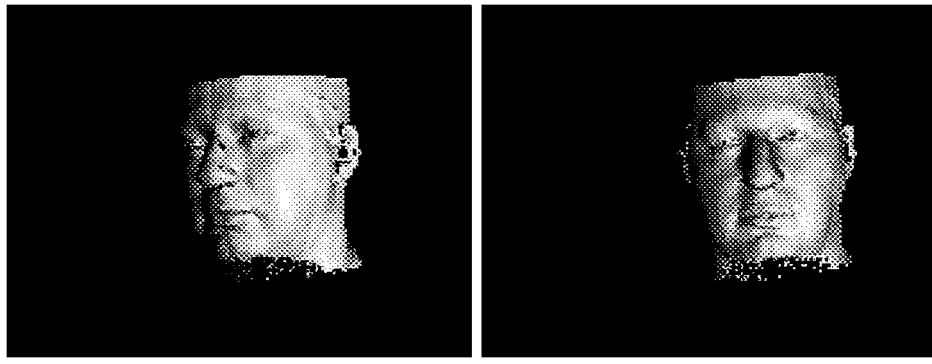

Figure 2. At left is a rendering of a 3D model of RK. The position of the model is the same as the position of the actual head. At right is a rendering of the head model in an incorrect pose.

A successful computational theory of object recognition must be similarly robust.

Lambert's law is perhaps the simplest model of surface reflectivity. It is an accurate model of the reflectance of a matte or non-shiny surface. Lambert's law states that the visible intensity of a surface patch is related to the dot product between the surface normal and the lighting. For a Lambertian object the imaging equation is:

$$v(T(x)) = \sum_i \alpha_i \vec{l}_i \cdot u(x), \tag{3}$$

where the model value $u(x)$ is the normal vector of a surface patch on the object, l_i is a vector pointing toward light source i, and α_i is proportional to the intensity of that light source ((Horn, 1986) contains an excellent review of imaging and its relationship to vision). As the illumination changes the functional relationship between the model and image will change.

Since we can not know beforehand what the imaging function will be, aligning a model and image can be quite difficult. These difficulties are only compounded if the surface properties of the object are not well understood. For example, many objects can not be modeled as having a Lambertian surface. Different surface finishes will have different reflectance functions. In general reflectance is a function of lighting direction, surface normal and viewing direction. The intensity of an observed patch is then:

$$v(T(x)) = \sum_i R(\alpha_i, \vec{l}_i, \vec{o}, u(x)), \tag{4}$$

where $\vec{o}$ is a vector pointing toward the observer from the patch and $R(\cdot)$ is the reflectance function of the surface. For an unknown material a great deal of experimentation is necessary to completely categorize the reflectance function. Since a general vision system should work with a variety of objects and under general illumination conditions, overly constraining assumptions about reflectance or illumination should be avoided.

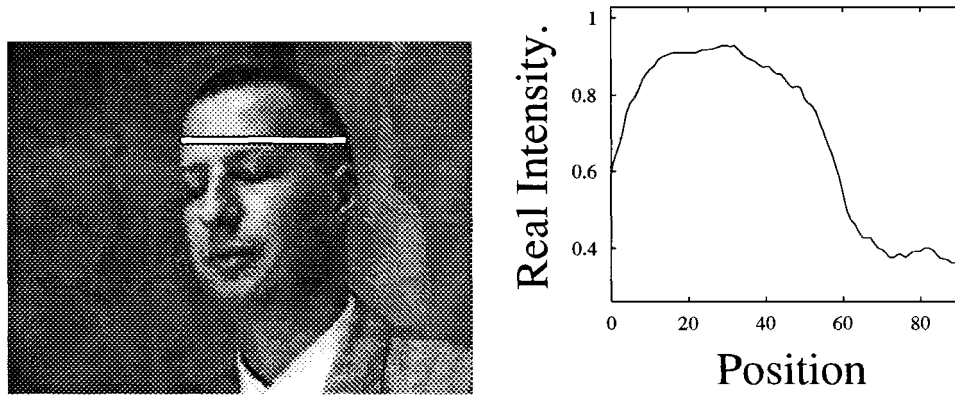

Figure 3. On the left is a video image of RK with the single scan-line highlighted. On the right is a graph of the intensities observed along this scan line.

Let us examine the relationship between a real image and model. This will allow us to build intuition both about alignment and image formation. Data from the real reflectance function can be obtained by aligning a model to a real image. An alignment associates points from the image with points from the model. If the alignment is correct, each pixel of the image can be interpreted as a sample of the imaging function $R(\cdot)$. The imaging function could be displayed by plotting intensity against lighting direction, viewing direction and surface normal. Unfortunately, because intensity is a function of so many different parameters the resulting plot can be prohibitively complex and difficult to visualize. Significant simplification will be necessary if we are to visualize any structure in this data.

In a wide variety of real images we can assume that the light sources are far from the object (at least in terms of the dimensions of the object). When this is true and there are no shadows, each patch of the object will be illuminated in the same way. Furthermore, we will assume that the observer is far from the object, and that the viewing direction is therefore constant throughout the image. The resulting relationship between normal and intensity is three dimensional. The normal vector has unit length and, for visible patches, is determined by two parameters: the x and y components. The image intensity is a third parameter. A three dimensional scatter plot of normal versus intensity is really a slice through the high dimensional space in which $R(\cdot)$ is defined. Though this graph is much simpler than the original, three dimensional plots are still quite difficult to interpret. We will slice the data once again so that all of the points have a single value for the y component of the normal.

Figure 3 contains a graph of the intensities along a single scan-line of the image of RK. Figure 4 shows similar data for the correctly aligned model of RK. Model normals from this scan-line are displayed in two graphs: the first shows the x component of the normal while the second shows the y component. Notice that we have chosen this portion of the model so that the y component of the normal is almost constant. As a result the relationship between normal and intensity can be visualized in only two dimensions. Figure 5 shows the intensities in the image plotted against the x component of the normal in the model. Notice that this relationship appears both consistent and functional. Points from the model with similar surface normals have very similar intensities. The data in this graph could be well approximated by a smooth curve. We will call an imaging function like this one *consistent*. Interestingly, we did not need any information about the illumination or surface properties of the object to determine that there is a consistent relationship between model normal and image intensity.

Figure 6 shows the relationship between normal and intensity when the model and image are no longer aligned. The only difference between this graph and the first is that the intensities come from a scan-line 3 centimeters below the correct alignment (i.e., the model is no longer aligned with the image, it is 3 centimeters too low). The normals used are the same. The resulting graph is no longer *consistent*. It does not look as though a simple smooth curve would fit this data well.

In summary, when model and image are aligned there will be a consistent relationship between image intensity and model normal. This is predicted by our assumption that there is an imaging function that relates

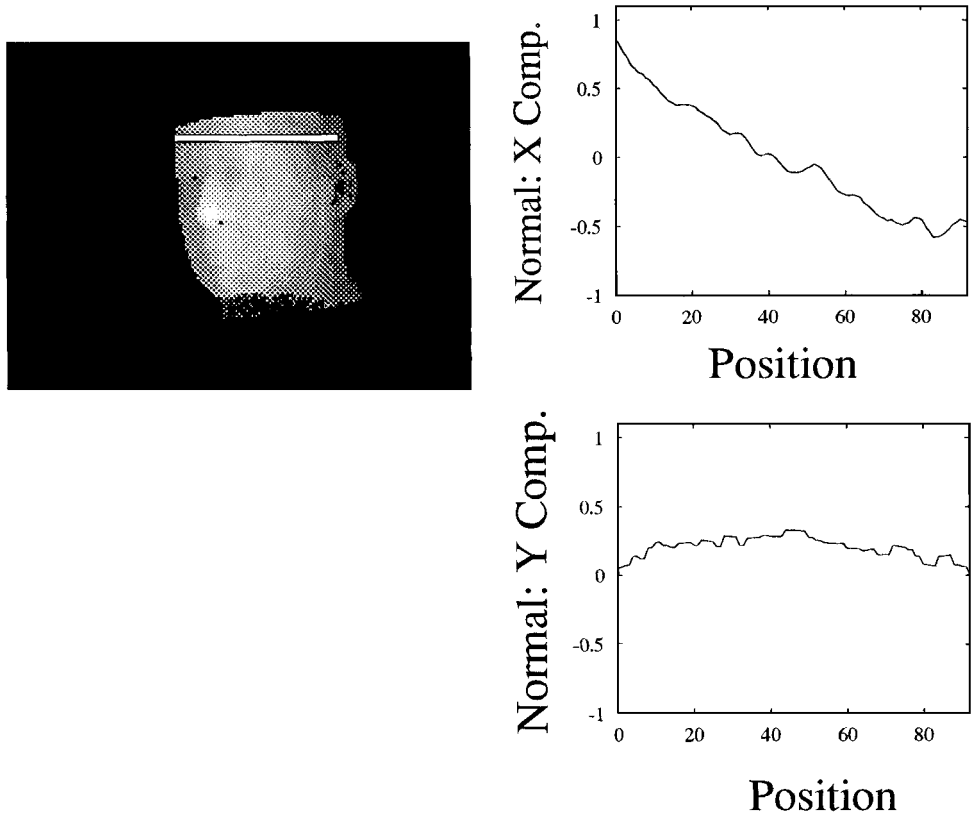

Figure 4. On the left is a depth map of RK with the single scan-line highlighted. At top right is a graph of the x component of the surface normal. On the bottom right is the y component of the normal.

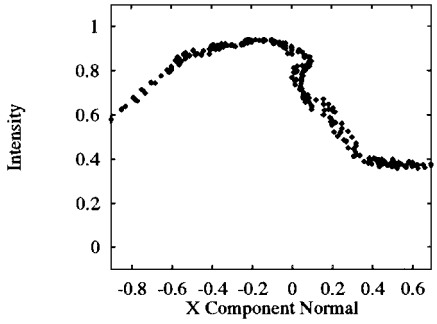

Figure 5. The aligned case: A scatter plot of the intensity of the video image versus the x component of the surface normal from the model. The image and model are correctly aligned.

models and images. While the actual form of this function depends on lighting and surface properties, a correct alignment will generally lead to a consistent relationship. Conversely, when model and image are misaligned the relationship between intensity and normal is inconsistent.

3. A Formal Definition of Consistency

Alignment can be performed by jointly searching over the space of possible imaging functions, exogenous parameters, and transformations. The principle of maximum likelihood can be used to motivate this procedure. The probability of an image given a model and transformation can be expressed as:

$$p(v \mid u, T) = \int\int \prod_{x_a} p(\eta = v(T(x_a)) - F(u(x_a), q)) \times p(F)\, p(q)\, dF\, dq, \quad (5)$$

where the product is computed over points from the model, x_a. This equation integrates over all possible imaging functions and all possible sets of exogenous variables. We are not aware of any approach that has come close to evaluating such an integral. It may not be feasible. Another possible approach is to find the imaging function and exogenous variables that make

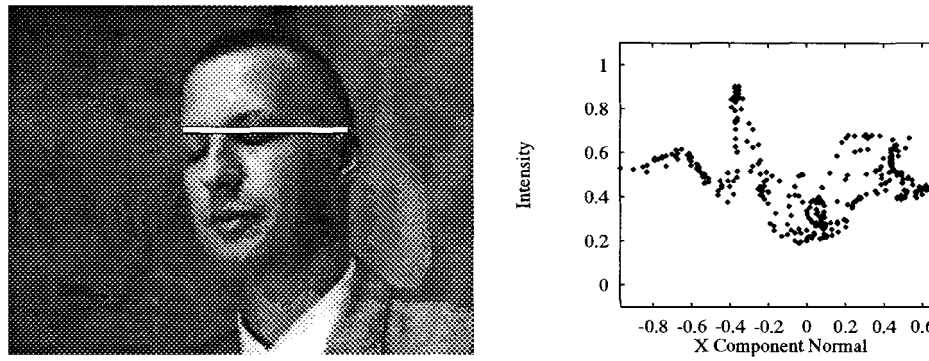

Figure 6. The misaligned case: on the left is the misaligned scan-line from the video image of RK. On the right is a scatter plot of the intensity of this part of the video image versus the x component of the surface normal from the model.

the image most likely,

$$p(v \mid u, T) \approx \max_{F,q} \prod_{x_a} p\Big(\eta = v(T(x_a)) - F(u(x_a), q)\Big) p(F) p(q). \quad (6)$$

This approximation is accurate whenever the integral in Eq. (5) is approximated by the component of the integrand that is maximal. The approximation is a good one when a particular F and q are much more likely than any other.

Using (6) we can define an alignment procedure as a nested search: i) given an estimate for the transformation, find F and q that make the image most likely; ii) given estimates for F and q, find a new transformation that makes the image most likely. Terminate when the transformation has stabilized. In other words, a transformation associates points from the model with points in the image; for every $u(x)$ there is a corresponding $v(T(x))$. A function F and parameter vector q are sought that best model the relationship between $u(x)$ and $v(T(x))$. This can be accomplished by "training" a function to fit the collection of pairs $\{v(T(x_a)), u(x_a)\}$.

The search for F is not a simple process. The range of possible imaging functions is of course infinite. In order to condition the search it is necessary to make a set of assumptions about the form of F. In addition some assumptions about the smoothness of F are necessary to insure convergence of the nested search for the maximum of (6). These assumptions can be enforced by formulating a strong prior probability over the space of functions, $p(F)$.

In many cases the search for an imaging function and exogenous parameters can be combined. For any particular F and q, another function $F_q(u(x)) = F(u(x), q)$ can be defined. The combined function is best thought of as a *reflectance map* (Horn, 1986). It maps the normals of an object directly into intensities. The three dimensional alignment procedure we will describe manipulates a similar combined function.

How might Eq. (6) be approximated efficiently? It seems reasonable to assume that for most real imaging functions similar inputs should yield similar outputs. In other words, the unknown imaging function is continuous and piecewise smooth. An efficient scheme for alignment could skip the step of approximating the imaging function and attempt to directly evaluate the *consistency* of a transformation. A transformation is considered consistent if points that have similar values in the model project to similar values in the image. By similar we do not mean similar in physical location, as in $|x_a - x_b|$, but similar in value, $|u(x_a) - u(x_b)|$ and $|v(T(x_a)) - v(T(x_b))|$. One ad-hoc technique for estimating consistency is to pick a similarity constant ψ and evaluate the following sum:

$$\text{Consistency}(T) = -\sum_{x_a \neq x_b} g_\psi(u(x_b) - u(x_a)) \times (v(T(x_b)) - v(T(x_a)))^2, \quad (7)$$

where g_ψ is a Gaussian with standard deviation ψ, and the sum is over points from the model, x_a and x_b. In order to minimize this measure, points that are close together in value must be more consistent, and those further apart less so.

An important drawback of consistency is that it is maximized by constancy. The most consistent transformation projects the points of the model onto a constant region of the image. For example, if scale is one of the transformation parameters, one entirely consistent transformation projects all of the points of the model down to a single point of the image.

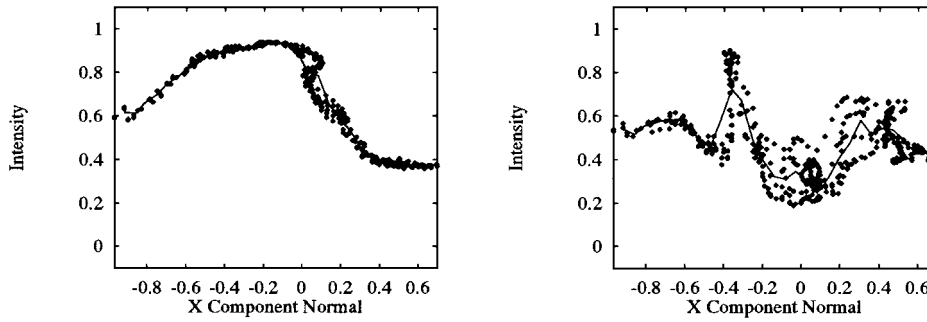

Figure 7. The joint distribution of data from the aligned and misaligned case above (left: aligned, right: misaligned). The weighted neighbor function approximation is show as a thin black line.

We now have two alternatives for alignment when the imaging function is unknown: a theoretical technique that may be intractable, and an outwardly efficient ad-hoc technique that has a number of important difficulties. One would like to find a technique that combines the best features from each approach. We propose that the complex search for the most likely imaging function, F_q, be replaced with a simpler search for the most consistent imaging function.

One type of function approximator that maximizes consistency is known as kernel regression or the weighted neighbor approximator:

$$F^*(u) = \frac{\sum_{x_a} R(u - u(x_a))v(T(x_a))}{\sum_{x_a} R(u - u(x_a))}. \qquad (8)$$

The weighting function R usually has a maximum at zero, and falls off asymptotically away from zero. F^* can be used to estimate the likelihood of a transformation as we did in (6). This formulation can be much more efficient than a naive implementation of (6) since there is no need to search for F_q. The model, image, and transformation define F^* directly.

Figure 7 shows the weighted neighbor approximation to the data from the RK alignments (in these graphs R is the Gaussian density function with variance 0.0003). Notice F^* fits the aligned model much better than the misaligned model. Assuming that the noise is Gaussian the log likelihood of the aligned model, 1079.49, is much larger than the log likelihood of the misaligned model, 537.34.[2]

4. From Likelihood to Entropy

The "classical" derivation of weighted neighbor likelihood provided a context in which insights could be developed and concrete representations described. Though weighted neighbor likelihood is a powerful technique, it has three significant drawbacks (see (Viola, 1995) for a more detailed discussion).

Firstly, it will only work when the image is a function of the model. Though this was assumed at the outset, in several important applications the image data may not be a function of the model. This is frequently the case in medical registration applications. For example, a CT scan is neither a function of an MR scan, nor is an MR scan a function of a CT scan. The second drawback of weighted neighbor log likelihood is that it can be susceptible to outliers. If one assumes, as is typical, that the image is conditionally Gaussian, occlusion and specularity can ruin an otherwise good match between model and image[3]. The third drawback arises from weighted neighbor likelihood's affinity for constant solutions.

Rather than require that the image be a function of the model, one natural generalization is to require that the image be predictable from the model. Predictability is closely related to the concept of entropy. A predictable random variable has low entropy, while an unpredictable random variable has high entropy. By moving to a formulation of alignment that is based on entropy many of the drawbacks of weighted neighbor likelihood can be circumvented.

The entropy of a random variable is defined as

$$h(y) \equiv -\int p(y) \ln p(y)\, dy. \qquad (9)$$

The joint entropy of two random variables x and y is

$$h(z, y) \equiv -\int p(z, y) \ln p(z, y)\, dz\, dy. \qquad (10)$$

Log likelihood and entropy are closely related (see (Cover and Thomas, 1991) for an excellent review of entropy and its relation to statistics). It can be shown that under certain conditions the conditional log

likelihood of the image given the model is a multiple of the conditional entropy of the image given the model:

$$\log p(v(T(x)) \mid u(x), T) = -Nh(v(T(x)) \mid u(x), T), \tag{11}$$

where N is the number of model points[4]. This is true only when the conditional distribution of v is the same as the assumed distribution for the noise, η. So if the noise is assumed Gaussian, equality holds when the conditional distribution of v is Gaussian. Note that while this is a restrictive assumption, it does not require either that the distribution of v be Gaussian, or that the joint distribution of v and u be Gaussian.

Both the constraint that v be a function of u and the constraint on the conditional distribution of v can be relaxed by estimating the conditional entropy directly:

$$h(v(T(x)) \mid u(X)) \equiv h(u(x)) - h(u(X), v(T(x))). \tag{12}$$

In the next section we will present an efficiently optimizable measure of entropy (EMMA) that can be used for this purpose. Nowhere in the derivation of EMMA will it be necessary to assume that v is a function of u. In addition, even in situations where v is a function of u, EMMA will frequently work better than weighted neighbor likelihood. While weighted neighbor likelihood requires restrictive assumptions about $p(\eta)$, EMMA can be used with a wide variety of densities. This make EMMA more robust to non-Gaussian errors.

In addition, the move from likelihood to entropy presents a principled mechanism for avoiding constant solutions. Conditional entropy, though it is more general than weighted neighbor likelihood, is still closely related to the consistency measure defined in Eq. (7). Like consistency, conditional entropy will accept a constant solution as optimal. Conditional entropy confounds two distinct situations: conditional entropy will be low when the image is predictable from the model, but it will also be low if the image by itself is predictable. Rather than conditional entropy we will estimate the *mutual information* between the model and the image:

$$I(u(x), v(T(x))) \equiv h(u(x)) + h(v(T(x))) - h(u(x), v(T(x))). \tag{13}$$

The mutual information defined in Eq. (13) has three components. The first term is the entropy in the model, and is not a function of T. The second term is the entropy of the part of the image into which the model projects. It encourages transformations that project u into complex parts of v. The third term, the (negative) joint entropy of u and v, contributes when u and v are functionally related. It encourages transformations where u explains v well. Together the last two terms identify transformations that find complexity and explain it well.

Why are weighted neighbor likelihood and conditional entropy related? Weighted neighbor likelihood measures the quality of the weighted neighbor function approximation. In the graph on the left of Fig. 7 the points of the sample lie near the weighted neighbor function approximation. In addition, the joint distribution of samples is tightly packed together. Points are not distributed throughout the space, but lie instead in a small part of the joint space. This is the hallmark of a low entropy distribution. In the graph on the right of Fig. 7 the weighted neighbor function approximation is a poor fit to the data and the data is more spread out. In general, aligned signals have low joint entropy and misaligned signals have high joint entropy.

5. EMMA Alignment

We seek an estimate of the transformation $\hat{T}$ that aligns the model u and image v by maximizing their mutual information over the transformations T,

$$\hat{T} = \arg\max_{T} I(u(x),\ v(T(x))). \tag{14}$$

Here we treat x as a random variable over coordinate locations in the model. In the alignment algorithm described below, we will draw samples from x in order to approximate I and its derivatives.

5.1. *EMMA and its Derivatives*

The entropies described above are defined in terms of integrals over the probability densities associated with the random variables u and v. When analyzing signals or images we will not have direct access to the densities. In this section we describe a differentiable estimate of the entropy of a random variable that is calculated from samples.

The entropy of a random variable z may be expressed as an expectation of the negative logarithm of the probability density:

$$h(z) = -E_z(\ln p(z)).$$

Our first step in estimating the entropies from samples is to approximate the underlying probability density $p(z)$ by a superposition of Gaussian densities centered on the elements of a sample A drawn from z:

$$p(z) \approx \frac{1}{N_A} \sum_{z_j \in A} G_\psi(z - z_j),$$

where

$$G_\psi(z) \equiv (2\pi)^{\frac{-n}{2}} |\psi|^{\frac{-1}{2}} \exp\left(-\frac{1}{2} z^T \psi^{-1} z \right).$$

This method of density estimation is widely known as the *Parzen Window* method. It is described in the textbook by Duda and Hart (1973). Use of the Gaussian density in the Parzen density estimate will simplify some of our subsequent analysis, but it is *not* necessary. Any differentiable function could be used. Another good choice is the Cauchy density.

Next we approximate statistical expectation with the sample average over another sample B drawn from z:

$$E_z(f(z)) \approx \frac{1}{N_B} \sum_{z_i \in B} f(z_i).$$

We may now write an approximation for the entropy of a random variable z as follows,

$$h(z) \approx \frac{-1}{N_B} \sum_{z_i \in B} \ln \frac{1}{N_A} \sum_{z_j \in A} G_\psi(z_i - z_j). \qquad (15)$$

The density of z may be a function of a set of parameters, T. In order to find maxima of mutual information, we calculate the derivative of entropy with respect to T. After some manipulation, this may be written compactly as follows,

$$\frac{d}{dT} h(z(T)) \approx \frac{1}{N_B} \sum_{z_i \in B} \sum_{z_j \in A} W_z(z_i, z_j)(z_i - z_j)^T \times \psi^{-1} \frac{d}{dT}(z_i - z_j), \qquad (16)$$

using the following definition:

$$W_z(z_i, z_j) \equiv \frac{G_\psi(z_i - z_j)}{\sum_{z_k \in A} G_\psi(z_i - z_k)}.$$

The weighting factor $W_z(z_i, z_j)$ takes on values between zero and one. It will approach one if z_i is significantly closer to z_j than it is to any other element of A. It will be near zero if some other element of A is significantly closer to z_i. Distance is interpreted with respect to the squared Mahalanobis distance (see (Duda and Hart, 1973))

$$D_\psi(z) \equiv z^T \psi^{-1} z.$$

Thus, $W_z(z_i, z_j)$ is an indicator of the degree of match between its arguments, in a "soft" sense. It is equivalent to using the "softmax" function of neural networks (Bridle, 1989) on the negative of the Mahalanobis distance to indicate correspondence between z_i and elements of A.

The summand in Eq. (16) may also be written as:

$$W_z(z_i, z_j) \frac{d}{dT} \frac{1}{2} D_\psi(z_i - z_j).$$

In this form it is apparent that to reduce entropy, the transformation T should be adjusted such that there is a reduction in the average squared distance between those values which W indicates are nearby, i.e., clusters should be tightened.

5.2. *Stochastic Maximization of Mutual Information*

The entropy approximation described in Eq. (15) may now be used to evaluate the mutual information of the model and image (Eq. (13)). In order to seek a maximum of the mutual information, we will calculate an approximation to its derivative,

$$\frac{d}{dT} I(u(x),\ v(T(x))) = \frac{d}{dT} h(v(T(x))) - \frac{d}{dT} h(u(x),\ v(T(x))).$$

Using Eq. (16), and assuming that the covariance matrices of the component densities used in the approximation scheme for the joint density are block diagonal: $\psi_{uv}^{-1} = \text{DIAG}(\psi_{uu}^{-1}, \psi_{vv}^{-1})$, we can obtain an estimate for the derivative of the mutual information as follows:

$$\widehat{\frac{dI}{dT}} = \frac{1}{N_B} \sum_{x_i \in B} \sum_{x_j \in A} (v_i - v_j)^T \times \left[W_v(v_i,\ v_j) \psi_v^{-1} - W_{uv}(w_i, w_j) \psi_{vv}^{-1} \right] \times \frac{d}{dT}(v_i - v_j). \qquad (17)$$

The weighting factors are defined as

$$W_v(v_i, v_j) \equiv \frac{G_{\psi_v}(v_i - v_j)}{\sum_{x_k \in A} G_{\psi_v}(v_i - v_k)}$$

and

$$W_{uv}(w_i, w_j) \equiv \frac{G_{\psi_{uv}}(w_i - w_j)}{\sum_{x_k \in A} G_{\psi_{uv}}(w_i - w_k)},$$

using the following notation (and similarly for indices j and k),

$$u_i \equiv u(x_i), \quad v_i \equiv v(T(x_i)), \quad \text{and} \quad w_i \equiv [u_i, v_i]^T.$$

If we are to increase the mutual information, then the first term in the brackets (of Eq. (17)) may be interpreted as acting to increase the squared distance between pairs of samples that are nearby in image intensity, while the second term acts to decrease the squared distance between pairs of samples that are nearby in *both* image intensity *and* the model properties. It is important to emphasize that distances are in the space of values (intensities, brightness, or surface properties), rather than coordinate locations.

The term $\frac{d}{dT}(v_i - v_j)$ will generally involve gradients of the image intensities and the derivative of transformed coordinates with respect to the transformation. In the simple case that T is a linear operator, the following outer product expression holds:

$$\frac{d}{dT} v(T(x_i)) = \nabla v(T(x_i)) x_i^T.$$

5.2.1. Stochastic Maximization Algorithm. We seek a local maximum of mutual information by using a stochastic analog of gradient descent. Steps are repeatedly taken that are proportional to the approximation of the derivative of the mutual information with respect to the transformation:

Repeat:

$A \leftarrow$ {sample of size N_A drawn from x}
$B \leftarrow$ {sample of size N_B drawn from x}
$T \leftarrow T + \lambda \widehat{\frac{dI}{dT}}$

The parameter λ is called the *learning rate*. The above procedure is repeated a fixed number of times or until convergence is detected.

A good estimate of the derivative of the mutual information could be obtained by exhaustively sampling the data. This approach has serious drawbacks because the algorithm's cost is quadratic in the sample size. For smaller sample sizes, less effort is expended, but additional noise is introduced into the derivative estimates.

Stochastic approximation is a scheme that uses noisy derivative estimate instead of the true derivative for optimizing a function (see (Widrow and Hoff, 1960; Ljung and Söderström, 1983; Haykin, 1994)). Convergence can be proven for particular linear systems, provided that the derivative estimates are unbiased, and the learning rate is annealed (decreased over time). In practice, we have found that successful alignment may be obtained using relatively small sample sizes, for example $N_A = N_B = 50$. We have proven that the technique will always converge to a pose estimate that is close to locally optimal (Viola, 1995).

It has been observed that the noise introduced by the sampling can effectively penetrate small local minima. Such local minima are often characteristic of continuous alignment schemes, and we have found that local minima can be overcome in this manner in these applications as well. We believe that stochastic estimates for the gradient usefully combine efficiency with effective escape from local minima.

5.2.2. Estimating the Covariance. In addition to λ, the covariance matrices of the component densities in the approximation method of Section 5.1 are important parameters of the method. These parameters may be chosen so that they are optimal in the maximum likelihood sense with respect to samples drawn from the random variables. This approach is equivalent to minimizing the cross entropy of the estimated distribution with the true distribution (Cover and Thomas, 1991). For simplicity, we assume that the covariance matrices are diagonal.

The most likely covariance parameters can be estimated on-line using a scheme that is almost identical in form to the scheme for maximizing mutual information.

6. Experiments

In this section we demonstrate alignment by maximization of mutual information in a variety of domains. In all of the following experiments, bi-linear interpolation was used when needed for non-integral indexing into images.

6.1. MRI Alignment

Our first and simplest experiment involves finding the correct alignment of two MR images (see Fig. 8).

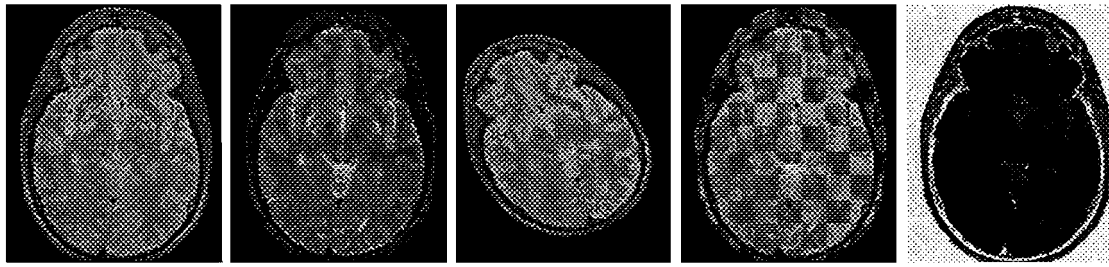

Figure 8. MRI alignment (from left to right): original proton-density image, original T2-weighted image, initial alignment, composite display of final alignment, intensity-transformed image.

The two original images are components of a double-echo MR scan and were obtained simultaneously, as a result the correct alignment should be close to the identity transformation. It is clear that the two images have high mutual information, while they are not identical.

A typical initial alignment appears in the center of Fig. 8. Notice that this image is a scaled, sheared, rotated and translated version of the original. A successful alignment is displayed as a checkerboard. Here every other 20×20 pixel block is taken either from the model image or target image. Notice that the boundary of the brain in the images is very closely aligned.

We represent the transformation by a 6 element affine matrix that takes two dimensional points from the image plane of the first image into the image plane of the second image. This scheme can represent any combination of scaling, shearing, rotation and translation. Before alignment the pixel values in the two MR images are pre-scaled so that they vary from 0 to 1. The component densities are $\psi_{uu} = \psi_w = 0.1$, and the random samples are of size 20. We used a learning rate of 0.02 for 500 iterations and 0.005 for 500 iterations. Total run time on a Sparc 10 was 12 seconds.

Over a set of 50 randomly generated initial poses that vary in position by 32 pixels, a little less than one third of the width of the head, rotations of 28 degrees, and scalings of up to 20%, the "correct" alignment is obtained reliably. Final alignments were well within one pixel in position and within 0.5% of the identity matrix for rotation/scale. We report errors in percent here because of the use of affine transformation matrices.

The two MRI images are fairly similar. Good alignment could probably be obtained with a normalized correlation metric. Normalized correlation assumes, at least locally, that one signal is a scaled and offset version of the other. Our technique makes no such assumption. In fact, it will work across a wide variety of non-linear transformations. All that is required is that the intensity transformation preserve a significant amount of information. On the right in Fig. 8. we show the model image after a non-monotonic (quadratic) intensity transformation. Alignment performance is not significantly affected by this transformation.

This last experiment is an example that would defeat traditional correlation, since the signals (the second and last in Fig. 8) are more similar in value when they are badly mis-aligned (non-overlapping) than they are when properly aligned.

6.2. Alignment of 3D Objects

6.2.1. Skull Alignment Experiments. This section describes the alignment of a real three dimensional object of its video image. The signals that are compared are quite different in nature: one is the video brightness, while the other consists of two components of the normal vector at a point on the surface of the model.

We obtained an accurate 3D model, including normals, of a skull that was derived from a computed tomography (CT) scan. Cluttered video images of the skull were obtained (see Fig. 9). In these images the pose of the model is displayed by projecting 3D points from the model's surface into the image plane and highlighting them in white. In the upper left of Fig. 9 the model is displayed in a typical initial pose. The final alignment of the skull model is in the upper right. Notice that the boundaries of the skull model and skull image are in close agreement. We would like to emphasize that in none of these experiments have we presegmented the image. The initial poses often project the model into regions of the image that contain a significant amount of clutter. EMMA reliably settles on a pose where few if any of the model points project onto the background.

One difference between the method used to perform 3D alignment and that used for 2D alignment is a Z-buffering step that is used to prune hidden points

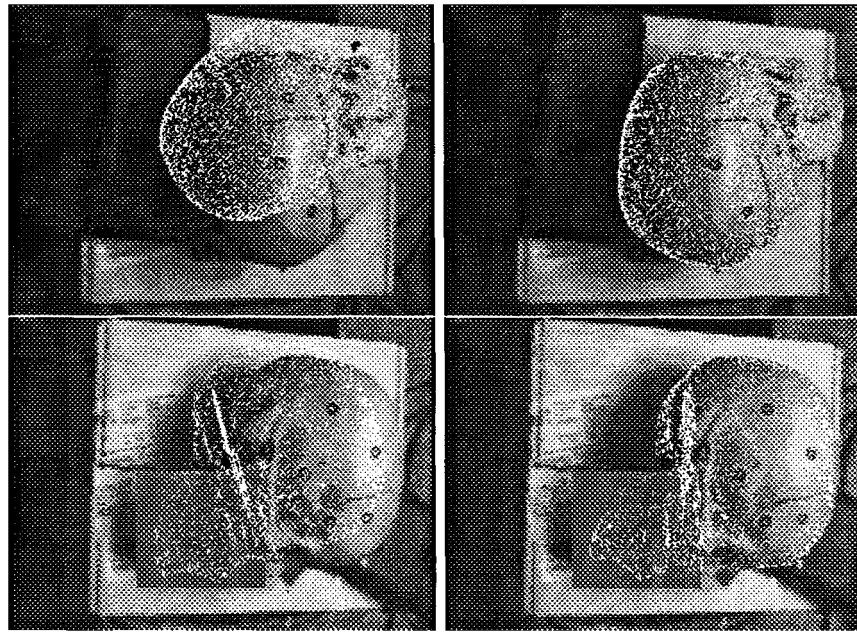

Figure 9. Skull alignment experiments: Initial alignment, final alignment, initial alignment with occlusion, final alignment with occlusion.

from the calculations. Since Z-buffer pruning is costly, and the pose does not change much between iterations, it proved sufficient to prune every 200 iterations. Another difference is that the model surface sampling was adjusted so that the sampling density in the image was corrected for foreshortening.

In this experiment, the camera has a viewing angle of 18 degrees. We represent T, the transformation from model to image coordinates, as a double quaternion followed by a perspective projection (Horn, 1986). Assuming diagonal covariance matrices four different variances are necessary, three for the joint entropy estimate and one for the image entropy estimate. The variance for the x component of the normal was 0.3, for the y component of the normal was 0.3, for the image intensity was 0.2 and for the image entropy was 0.15. The size of the random sample used is 50 points.

Since the units of rotation and translation are very different, two separate learning rates are necessary. For an object with a 100 millimeter radius, a rotation of 0.01 radians about its center can translate a model point up to 1 millimeter. On the other hand, a translation of 0.01 can at most translate a model point 0.01 millimeters. As a result, a small step in the direction of the derivative will move some model points up to 100 times further by rotation than translation. If there is only a single learning rate a compromise must be made between the rapid changes that arise from the rotation and the slow changes that arise from translation. Since the models used have a radius that is on the order of 100 millimeters, we have chosen rotation learning rates that are 100 times smaller than translation rates. In our experiments alignment proceeds in two stages. For the first 2000 iterations the rotation learning rate is 0.0005 and the translation learning rate is 0.05. The learning rates are then reduced to 0.0001 and 0.01 respectively for an additional 2000 iterations. Running time is about 30 seconds on a Sparc 10.

A number of randomized experiments were performed to determine the reliability, accuracy and repeatability of alignment. This data is reported in Table 1. An initial alignment to an image was performed to establish a base pose. From this base pose, a random uniformly distributed offset is added to each translational axis (labeled ΔT) and then the model is rotated about a randomly selected axis by a random uniformly selected angle ($\Delta\theta$). Table 1 describes four experiments each including 50 random initial poses. The distribution of the final and initial poses can be compared by examining the variance of the location of the centroid, computed separately in X, Y and Z. In addition, the average angular rotation from the true pose is reported (labeled $\overline{|\Delta\theta|}$). Finally, the number

Table 1. Skull alignments results table.

ΔT XYZ $\pm$ mm	$\Delta\theta$ °	Initial σ_X mm	Initial σ_Y mm	Initial σ_Z mm	Initial $\overline{\lvert\Delta\theta\rvert}$ °	Final σ_X mm	Final σ_Y mm	Final σ_Z mm	Final $\overline{\lvert\Delta\theta\rvert}$ °	Success %
10	10	5.94	5.56	6.11	5.11	.61	.53	5.49	3.22	100
30	10	16.53	18.00	16.82	5.88	1.80	.81	14.56	2.77	96
20	20	10.12	12.04	10.77	11.56	1.11	.41	9.18	3.31	96
$10 < \Delta < 20$	$20 < \Delta < 40$	14.83	15.46	14.666	28.70	1.87	2.22	14.19	3.05	78

of poses that successfully converged near the correct solution is reported. The final variance statistics are only computed over the "good" poses.

The lower images in Fig. 9 show the initial and final alignment from an experiment that includes an artificial occlusion that covers the chin area. The pose found is very close to the correct one despite the occlusion. In a number of experiments, we have found that alignment to occluded images can require more time for convergence. Our system works in the presence of occlusion because the measure of mutual information used is "robust" to outliers and noise (see (Viola, 1995) for further discussion).

These experiments demonstrate that the alignment procedure is reliable when the initial pose is close to the "correct" pose. Outside of this range gradient descent, by itself, is not capable of converging to the correct solution. The capture range is not unreasonably small however. Translations as large as half the diameter of the skull can be accommodated, as can rotations in the plane of up to 45 degrees. Empirically it seems that alignment is most sensitive to rotation in depth. This is not surprising since only the visible points play a role in the calculation of the derivative. As a result, when the chin is hidden the derivative gives you no information about how to move the chin out from behind the rest of the skull.

6.2.2. Head Tracking Experiment. This section summarizes recent results obtained using the methodology described above to track a moving human head in a video sequence. The results are shown in Fig. 10. The images on the left of each square have been digitized from video tape at 3 frames per second. A 3D model of the subject's head, along with surface normals, was derived from a Cyberware scan of the subject. It is rendered on the right to illustrate the poses determined by the alignment method. (Recall that alignment proceeds using video brightness and model surface normals.)

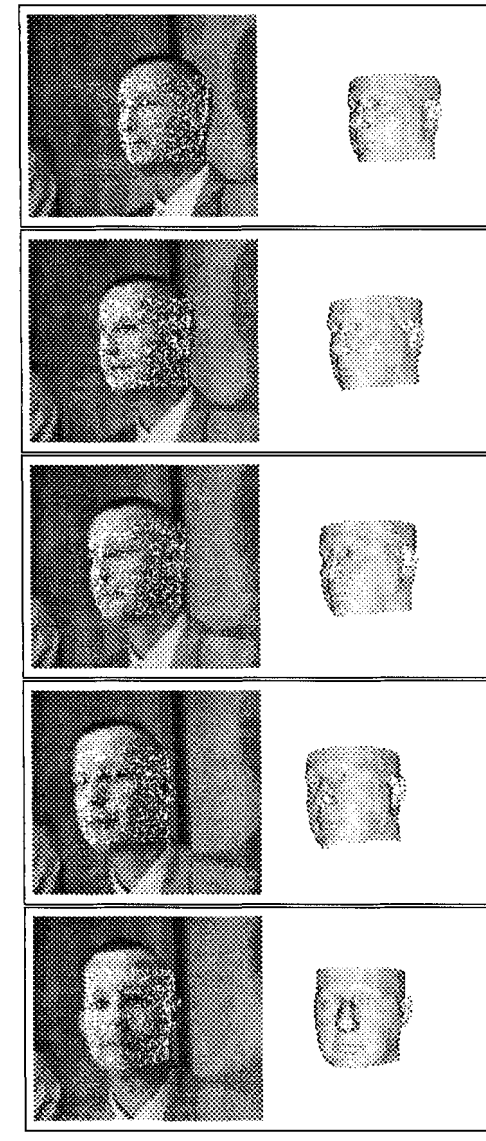

Figure 10. Video head tracking experiment.

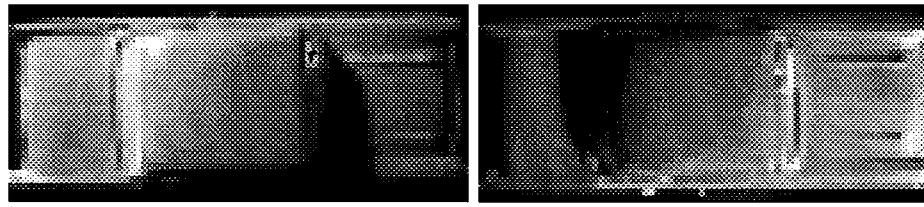

Figure 11. Car model images.

An initial alignment of the model to the first frame of the sequence was obtained using a manually-generated starting pose (this frame is not shown). In subsequent frames, the previous final pose was used as the initial pose for the next alignment. Each pose refinement took about 10 seconds on a Sparc 10.

How are the face experiments different from the skull experiments? Firstly, the face model is much smoother than the skull model. There really aren't any creases or points of high curvature. As a result it is much less likely that an edge-based system could construct a representation either of the image or the model that would be stable under changes in illumination. Secondly, the albedo of the actual object is not exactly constant. The face contains eyebrows, lips and other regions where the albedo is not the same. As a result this is a test of EMMA's ability to handle objects where the assumption of constant albedo is violated. Thirdly, not all of the occluding contours of the object are present in the model. The model is truncated both at the chin and the forehead. As a result experiments with this model demonstrate that EMMA can work even when the occluding contours of the image and model are not in agreement.

6.3. View Based Recognition Experiments

In the previous vision experiments we used knowledge of the physics of imaging to show that the surface normal of an object should be predictive of the intensity observed in an image. Unfortunately, in many experimental situations no three dimensional model is available. In these situations it is frequently the case that the only available information about an object is a collection of images taken under a variety conditions. One approach for solving problems like this is to use a collection of images as the model. This is often called a "view based" approach since the model is made up of a number of views of the model object. Given a novel image of some object, each model image is compared to it in turn. If some model image is "close enough" to the novel image, the model and novel image are considered aligned (or recognized). One can significantly reduce the number of model images required by adding an affine transformation to the comparison process. The novel image is then compared to each model image under a set of affine transformations. The most commonly used comparison metric is correlation. Correlation makes the assumption that the model and the image are identical (or possibly related by a linear function).

In general the set of images that can arise from a single object under varying illumination is very broad. Figure 11 shows two images of the same object in the same pose. These images are very different and are in fact anti-correlated: bright pixels in the left image correspond to dark pixels in the right image; dark pixels in the left image correspond to bright pixels in the right image. No variant of correlation could match these images together.

We have presented techniques based on entropy that can match both correlated and anti-correlated signals. These techniques require only that there is some consistent relationship between model and image. Discouragingly, it is not difficult to find two images of the same object for which there is no consistent relationship. Figure 12 shows a novel image which is aligned with the two model images. Figure 13 contains two scatter plots of the pixel values in the novel image versus the pixel values in the model images. Clearly, there is no simple consistent relationship displayed in either of these graphs. Neither correlation nor EMMA could be used to match this novel image to either model image.

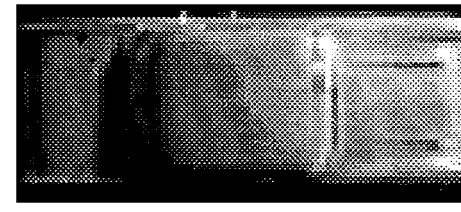

Figure 12. A novel image of the car model.

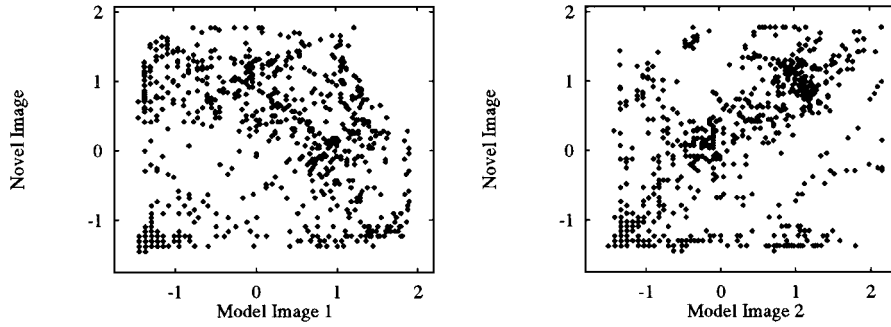

Figure 13. The relationship between pixels in the novel image and each of the model images.

6.3.1. Photometric Stereo. By itself each model image does not contain enough information to constrain the match between image and model. However, it is well known that taken together a collection of images can be used to determine the 3D shape of an object. As we've seen the 3D shape is sufficient to constrain the match between image and model.

When multiple images of an object are available a technique called *photometric stereo* can be used to estimate 3D shape (Horn, 1986). Photometric stereo works with images which are taken from the same location but under different illumination conditions. It is assumed that detailed information both about illumination and surface properties are available for each image. As a result a reflectance map can be computed for each image.

The reflectance map together with the intensity of a pixel acts as a constraint on the normal vector visible from that pixel. The allowable normals usually lie along a closed curve on the unit circle. From a second image, and its associated reflectance map, another set of allowable normals can be computed. By intersecting these constraints, two images are sufficient to determine the surface normal at each pixel. From the normals the shape can be obtained through integration.

Once the shape of the object is determined, the correct alignment could be found using the three dimensional version of EMMA alignment. The imaging function of this new two stage process is:

$$v(T(x_i)) = F(G(u_1(x_i),\ r_1, u_2(x_i), r_2), q)$$

where $G(\,)$ is the photometric stereo function that takes two images and two reflectance maps and returns the shape, and $F(\,)$ is our original imaging function which predicts image intensities from object normals.

In practice, however, performing photometric stereo requires the kind of detailed metric information about illumination that is only available under very controlled circumstances. One cannot use natural images where the lighting is unknown or difficult to determine. Luckily, we need not actually know $G(\,)$, r_1, r_2, $F(\,)$, or q. As long as they exist there will be high mutual information between any novel image and a *pair* of model images. This is the essence of view based EMMA alignment. We don't actually perform photometric stereo, we simply assume that it is possible. As a result a pair of images should give information about any third image.

To demonstrate this approach we have built a model using the two images in Fig. 11. Figure 14 shows the target image, the initial pose of the model, and the final pose obtained after alignment.

Technically this experiment is very similar to the MRI alignment experiment. The main difference is that the model is constructed from a pair of model images. A sample of the model $u(x) = [u_1(x), u_2(x)]^T$ is a two dimensional vector containing the intensity of the two images at location x. This is similar to the two component representation of normal used in the three dimensional alignment experiments. For this experiment σ is 0.1. The parameters were updated for 1000 iterations at a rate of 0.002. From a set of randomized experiments we have determined that the capture range of the alignment procedure is about 40% of the length and width of the car, and 35 degrees of rotation.

7. Discussion and Related Work

We have presented a metric for comparing objects and images that uses shading information, yet is explicitly insensitive to changes in illumination. This metric is unique in that it compares 3D object models directly to raw images. No pre-processing or edge detection is

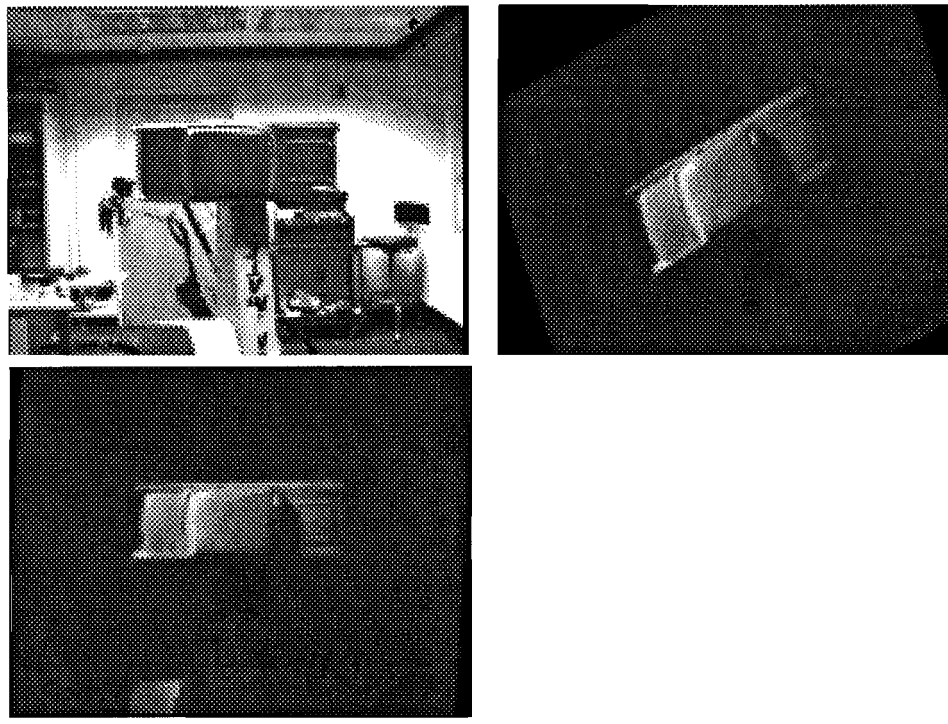

Figure 14. Top left: A novel image of the car. Top right: The initial pose of the car model. Though the model is made up of multiple images, only one is shown here. Bottom left: The aligned pose of the car model.

required. The metric has been rigorously derived from information theory.

In a typical vision application EMMA alignment is intensity-based, rather than feature based. While intensity based, it is more robust than traditional correlation—since it is insensitive to negating the image data, as well as a variety of non-linear transformations (e.g., Section 6.1), which would defeat conventional intensity-based correlation.

The sensitivity of intensity correlation may be corrected, to some extent, by performing correlations on the magnitude of the intensity gradient. This, as well as edge-based matching techniques, can perform well on objects having discontinuous surface properties, or useful silhouettes. These approaches work because the image counterparts of these discontinuities are reasonably stable with respect to illumination, however they typically make two very strong assumptions: the edges that arise are stable under changes in lighting, and the models are well described as a collection of edges.

There are many schemes that represent models and images by collections of edges and define a distance metric between them, Huttenlocher's use of the Hausdorff distance (Huttenlocher et al., 1991) is an example. Some methods use a metric that is proportional to the number of edges that coincide (see the excellent survey articles: (Besl and Jain, 1985; Chin and Dyer, 1986)). A smooth, optimizable version of such a metric can be defined by introducing a penalty both for unmatched edges and for the distance between those that are matched (Lowe, 1985; Wells III, 1992). This metric can then be used both for image/model comparison and for pose refinement. Additional technical details on the relationship between mutual information and other measures of alignment may be found in (Viola, 1995).

Alignment by extremizing properties of the joint signal has been used by Hill et al. (1994) to align MRI, CT, and other medical image modalities. They use third order moments of the joint histogram to characterize the clustering of the joint data. We believe that mutual information is perhaps a more direct measure of the salient property of the joint data at alignment, and demonstrate an efficient means of estimating and extremizing it. Recently, Collignon et al. (1995) described the use of joint entropy as a criterion for registration of CT and MRI data. They demonstrated a good minimum by probing the criterion, but no search techniques were described.

Image-based approaches to modeling have been previously explored by several authors. Objects need not have edges to be well represented in this way, but care must be taken to deal with changes in lighting and pose. Turk and Pentland have used a large collection of face images to train a system to construct representations that are invariant to some changes in lighting and pose (Turk and Pentland, 1991). These representations are a projection onto the largest eigenvectors of the distribution of images within the collection. Their system addresses the problem of recognition rather than alignment, and as a result much of the emphasis and many of the results are different. For instance, it is not clear how much variation in pose can be handled by their system. We do not see a straightforward extension of this or similar eigenspace work to the problem of pose refinement. In other related work, Shashua has shown that all of the images, under different lighting, of a Lambertian surface are a linear combination of any three of the images (Shashua, 1992). A procedure for image alignment could be derived from this theory. In contrast, our image alignment method does not assume that the object has a Lambertian surface.

Entropy is playing an ever increasing role within the field of neural networks. We know of no work on the alignment of models and images, but there has been work using entropy and information in vision problems. None of these techniques use a non-parametric scheme for density/entropy estimation as we do. In most cases the distributions are assumed to be either binomial or Gaussian. Entropy and mutual information plays a role in the work of (Linsker, 1986; Becker and Hinton, 1992; Bell and Sejnowski, 1995).

Acknowledgments

We were partially inspired by the work of Hill and Hawkes on registration of medical images. Sanjeev Kulkarni introduced Wells to the concept of relative entropy, and its use in image processing. Nicol Schraudolph and Viola began discussions of this concrete approach to evaluating entropy in an application of un-supervised learning.

We thank Ron Kikinis and Gil Ettinger for the 3D skull model and MRI data. J.P. Mellor provided the skull images and camera model. Viola would like to thank Terrence. J. Sejnowski for providing some of the facilities used during the preparation of this manuscript.

We thank for following sources for their support of this research: USAF ASSERT program, Parent Grant#: F49620-93-1-0263 (Viola), Howard Hughes Medical Institute (Viola), ARPA IU program via ONR#: N00014-94-01-0994 (Wells) and AFOSR # F49620-93-1-0604 (Wells).

Notes

1. EMMA is a random but pronounceable subset of the letters in the words “Empirical entropy Manipulation and Analysis”.
2. Log likelihood is computed by first finding the Gaussian distribution that fits the residual error, or noise, best. The log of (6) is then computed using the estimated distribution of the noise. For small amounts of noise, these estimates can be much larger than 1.
3. Correlation matching is one of many techniques that assumes a Gaussian conditional distribution of the image given the model.
4. Here we speak of the empirically estimated entropy of the conditional distribution.

References

Becker, S. and Hinton, G.E. 1992. Learning to make coherent predictions in domains with discontinuities. In *Advances in Neural Information Processing*, J.E. Moody, S.J. Hanson, and R.P. Lippmann, (Eds.), Denver 1991. Morgan Kaufmann: San Mateo, vol. 4.

Bell, A.J. and Sejnowski, T.J. 1995. An information-maximisation approach to blind separation. In *Advances in Neural Information Processing*, Denver 1994. Morgan Kaufmann: San Francisco, vol. 4.

Besl, P. and Jain, R. 1985. Three-dimensional object recognition. *Computing Surveys*, 17:75–145.

Bridle, J.S. 1989. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Advances in Neural Information Processing 2*, D.S. Touretzky (Ed.), Morgan Kaufman, pp. 211–217.

Chin, R. and Dyer, C. 1986. Model-based recognition in robot vision. *Computing Surveys*, 18:67–108.

Collignon, A., Vandermuelen, D., Suetens, P., and Marchal, G. 1995. 3D multi-modality medical image registration using feature space clustering. In *Computer Vision, Virtual Reality and Robotics in Medicine*, N. Ayache (Ed.), Springer Verlag, pp. 195–204.

Cover, T.M. and Thomas, J.A. 1991. *Elements of Information Theory*. John Wiley and Sons.

Duda, R. and Hart, P. 1973. *Pattern Classification and Scene Analysis*. John Wiley and Sons.

Haykin, S. 1994. *Neural Networks : A comprehensive foundation*. Macmillan College Publishing.

Hill, D.L., Studholme, C., and Hawkes, D.J. 1994. Voxel similarity measures for automated image registration. In *Proceedings of the Third Conference on Visualization in Biomedical Computing*, pp. 205–216, SPIE.

Horn, B. 1986. *Robot Vision*. McGraw-Hill: New York.

Huttenlocher, D., Kedem, K., Sharir, K., and Sharir, M. 1991. The upper envelope of Voronoi surfaces and its applications. In *Proceedings of the Seventh ACM Symposium on Computational Geometry*, pp. 194–293.

Linsker, R. 1986. From basic network principles to neural architecture. *Proceedings of the National Academy of Sciences*, USA, vol. 83, pp. 7508–7512, 8390–8394, 8779–8783.

Ljung, L. and Söderström, T. 1983. *Theory and Practice of Recursive Identification*. MIT Press.

Lowe, D. 1985. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers.

Shashua, A. 1992. Geometry and Photometry in 3D Visual Recognition. Ph.D. thesis, M.I.T Artificial Intelligence Laboratory, AI-TR-1401.

Turk, M. and Pentland, A. 1991. Face recognition using eigenfaces. In *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, Lahaina, Maui, Hawaii, pp. 586–591. IEEE.

Viola, P.A. 1995. Alignment by Maximization of Mutual Information. Ph.D. thesis, Massachusetts Institute of Technology.

Wells III, W. 1992. Statistical Object Recognition. Ph.D. thesis, MIT Department Electrical Engineering and Computer Science, Cambridge, Mass. MIT AI Laboratory TR 1398.

Widrow, B. and Hoff, M. 1960. Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, IRE, New York, 4:96–104.