AFRL-SR-BL-TR-98-

0107

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	11/3/97	Final 9/1/93 - 6/30/97

4. TITLE AND SUBTITLE

Learning Maneuvers Using Neural Network Models

5. FUNDING NUMBERS

Grant #:
F49620-93-I-0379

6. AUTHOR(S)

Tomas Lozano-Perez

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Marc Q. Jacobs
AFOSR/NM
110 Duncan Avenue, Room B115
Bolling AFB, DC 20332-8080

10. SPONSORING/MONITOR AGENCY REPORT NUMB

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

19980129 078

13. ABSTRACT *(Maximum 200 words)*

This grant covered the completion of the PhD thesis of Paul Viola and the initiation of the PhD work of Oded Maron. Viola's work was on alignment of 2- and 3-dimensional objects based on maximization of mutual information. The technique depends only on object shape and is robust to variations of illumination. The algorithms are quite general and can foreseeably be used in a wide variety of imaging situations. Maron's work has focused on a variation on supervised learning called multiple-instance learning, where the task is to learn a concept given positive and negative bags of instances. Each bag may contain many instances but a bag is labeled positive even if only one of the instances in it falls within the conept. A bag is labeled negative only if all the instances in it are negative. This framework has been applied to a variety of problem domains.

14. SUBJECT TERMS

Alignment
Mutual Information
Multiple-instance Learning

15. NUMBER OF PAGES

22

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

1 Introduction

This grant covered the completion of the PhD thesis of Paul Viola and the initiation of the PhD work of Oded Maron. This report summarizes the key results in both of these pieces of work.

Viola's work was on alignment of 2- and 3-dimensional objects based on maximization of mutual information. The technique depends only on object shape and is robust to variations of illumination. The algorithms are quite general and can foreseeably be used in a wide variety of imaging situations. Paul Viola's Ph.D. thesis is available as an MIT AI Laboratory Technical report and is reachable from his Web page (http://www.ai.mit.edu/people/viola). The summary here is from a paper that appeared in ICCV 95.

Maron's work has focused on a variation on supervised learning called multiple-instance learning, where the task is to learn a concept given positive and negative bags of instances. Each bag may contain many instances, but a bag is labeled positive even if only one of the instances in it falls within the concept. A bag is labeled negative only if all the instances in it are negative. This framework has been applied to a variety of problem domains. Oded Maron plans to finish his doctoral work by May 1997. The work summarized here is from a paper to appear in NIPS 97.

2 Alignment and Maximization of Mutual Information — Paul Viola and William Wells

In many different visual processing problems, including object recognition, 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. This is typically a difficult problem. If we had a good imaging model then deciding whether an image contained a particular model at a given pose is straightforward: compute the predicted image and compare it to the actual image directly. Given a perfect imaging model the two images will 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. In

the part of the scene containing an image of the object, we can formulate an imaging equation

$$v(T(x)) = F(u(x), P) \ , \tag{1}$$

where x are coordinates of a surface patch of the object model, $u(x)$ describes the properties of the surface of the model (e.g., surface normal, albedo, etc.) at position x , and P are parameters of the imaging process, such as the illumination conditions. F is the image formation function that generates the brightness of the surface patch in the image. Thus, $v(T(x))$ is the brightness image of the object placed in the scene by coordinate transformation $T(\cdot)$. If F and P were known in detail it would be feasible to make an accurate prediction of scene intensities, since the physics of image formation are well understood. But, because of the complexity of visible light imaging, it may be difficult to determine the particular F and P for a given scene.

One reason that it is, in principle, possible to find F is that the model does supply much information about the scene. 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 by dealing with this mutual information directly. Such a technique would attempt to find the alignment of the model in the scene by maximizing the information that the model provides about the scene. We will present an algorithm that does just this. 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.

3 Description of Method

3.1 Alignment by Maximization of Mutual Information

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,

$$\hat{T} = \arg\max_{T} I(u(x),\ v(T(x))) \ . \tag{2}$$

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.

Mutual information is defined in terms of entropy in the following way [1] :

$$\begin{aligned} &I(u(x),\ v(T(x))) \equiv \\ &\quad H(u(x)) + H(v(T(x))) - H(u(x), v(T(x))) \end{aligned} \tag{3}$$

. $H(\cdot)$ is the entropy of a random variable, and is defined as $H(x) \equiv -\int p(x) \ln p(x) dx$, while the joint entropy of two random variables x and y is $H(x,y) \equiv -\int p(x,y) \ln p(x,y) dx dy$. Entropy can be interpreted as a measure of uncertainty, variability, or complexity.

The mutual information defined in Equation 3 has three components. The first term on the right 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. This is the essence of mutual information.

3.2 Estimating Entropies and their 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 $G_\psi(x) \equiv (2\pi)^{\frac{-n}{2}} |\psi|^{\frac{-1}{2}} \exp(-\frac{1}{2} x^T \psi^{-1} x)$. This method of density estimation is widely known as the *Parzen Window* method. It is described in the textbook by Duda and Hart[2].

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,

$$H(z) \approx \frac{-1}{N_B} \sum_{z_i \in B} \ln \frac{1}{N_A} \sum_{z_j \in A} G_\psi(z_i - z_j) \ . \tag{4}$$

In order to find maxima of mutual information, we calculate the derivative of entropy with respect to the transformation T. After some manipulation, this may be written compactly as follows,

$$\frac{d}{dT} H(z(T)) \approx \tag{5}$$
$$\frac{1}{N_B} \sum_{z_i \in B} \sum_{z_j \in A} W_z(z_i, z_j)(z_i - z_j)^T \psi^{-1} \frac{d}{dT}(z_i - z_j)$$

, 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 [2]) $D_\psi(z) \equiv z^T \psi^{-1} z$. 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 [3] on the negative of the Mahalanobis distance to indicate correspondence between z_i and elements of A.

The summand in Equation 5 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.

3.3 Stochastic Maximization of Mutual Information

The entropy approximation described in Equation 4 may now be used to evaluate the mutual information of the model and image (Equation 3). In order to seek a maximum of the mutual information, we will calculate an approximation to its derivative,

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

Using Equation 5, and assuming that the covariance matrices of the component densities used in the approximation scheme for the joint density are block diagonal: $\psi_{uv}^{-1} = \mathrm{DIAG}(\psi_{uu}^{-1}, \psi_{vv}^{-1})$, we can obtain an estimate for the derivative of the mutual information as follows:

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

The weighting factors are defined as

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

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

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

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

If we are to increase the mutual information, then the first term in the brackets 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: $\frac{d}{dT} v(T(x_i)) = \nabla v(T(x_i)) x_i^T$.

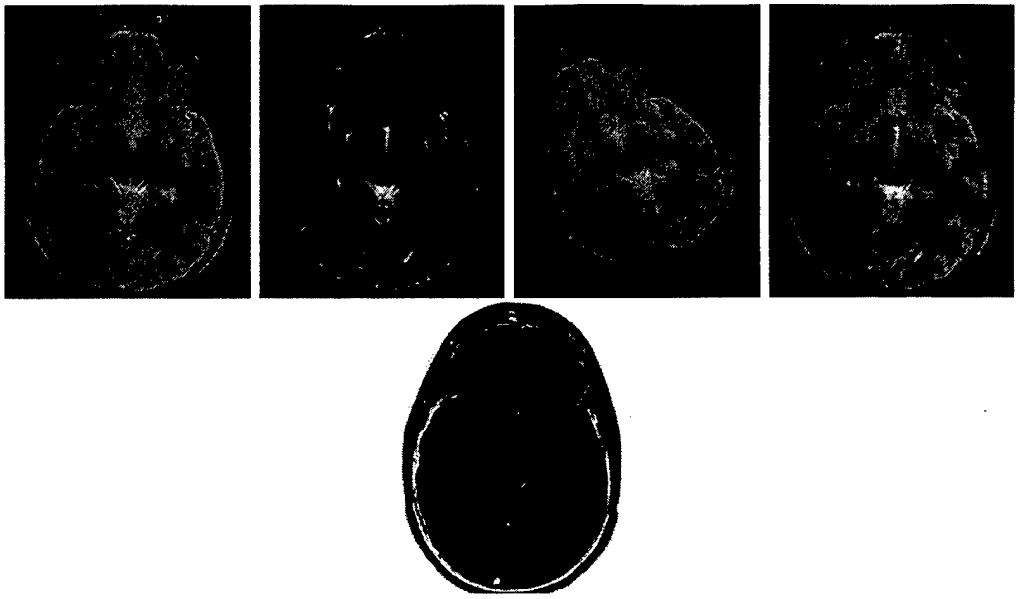


Figure 1: MRI Alignment: Original Proton-Density Image, Original T2-Weighted Image, Initial Alignment, Composite Display of Final Alignment, Intensity-Transformed Image

3.3.1 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:

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

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 [4], [5], and [6]). 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 [7].

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.

3.4 Estimating the Covariance

In addition to λ, the covariance matrices of the component densities in the approximation method of Section 3.2 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 [8]. 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.

4 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.

4.1 MRI Alignment

Our first and simplest experiment involves finding the correct alignment of two MR images (see Figure 1). 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. The pixel values in the two images are pre-scaled so that they vary from 0 to 1.

A typical initial alignment appears in the center of Figure 1. 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 20x20 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. The sample metric used is squared distance, the component densities have $\sigma = 0.1$, 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

ΔT	$\Delta\theta$	INITIAL				FINAL				SUCCESS
X Y Z		σ_X	σ_Y	σ_Z	$\overline{\lvert\Delta\theta\rvert}$	σ_X	σ_Y	σ_Z	$\overline{\lvert\Delta\theta\rvert}$	
$\pm$mm	°	mm			°	mm			°	%
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	14.466	28.70	1.87	2.22	14.19	3.05	78

Table 1: Skull Alignments Results Table

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 Figure 1 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 Figure 1) are more similar in value when they are badly mis-aligned (non-overlapping) than they are when properly aligned.

4.2 Alignment of 3D Objects

4.2.1 Skull Alignment Experiments

This section describes the alignment of a real three dimensional object to 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 Figure 2). On the left we see the 3D points from the model at an initial pose projected into the image plane and highlighted in white. A typical final alignment of the skull model into the image appears next. Notice that the boundaries of the skull model and skull image are in close agreement.

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

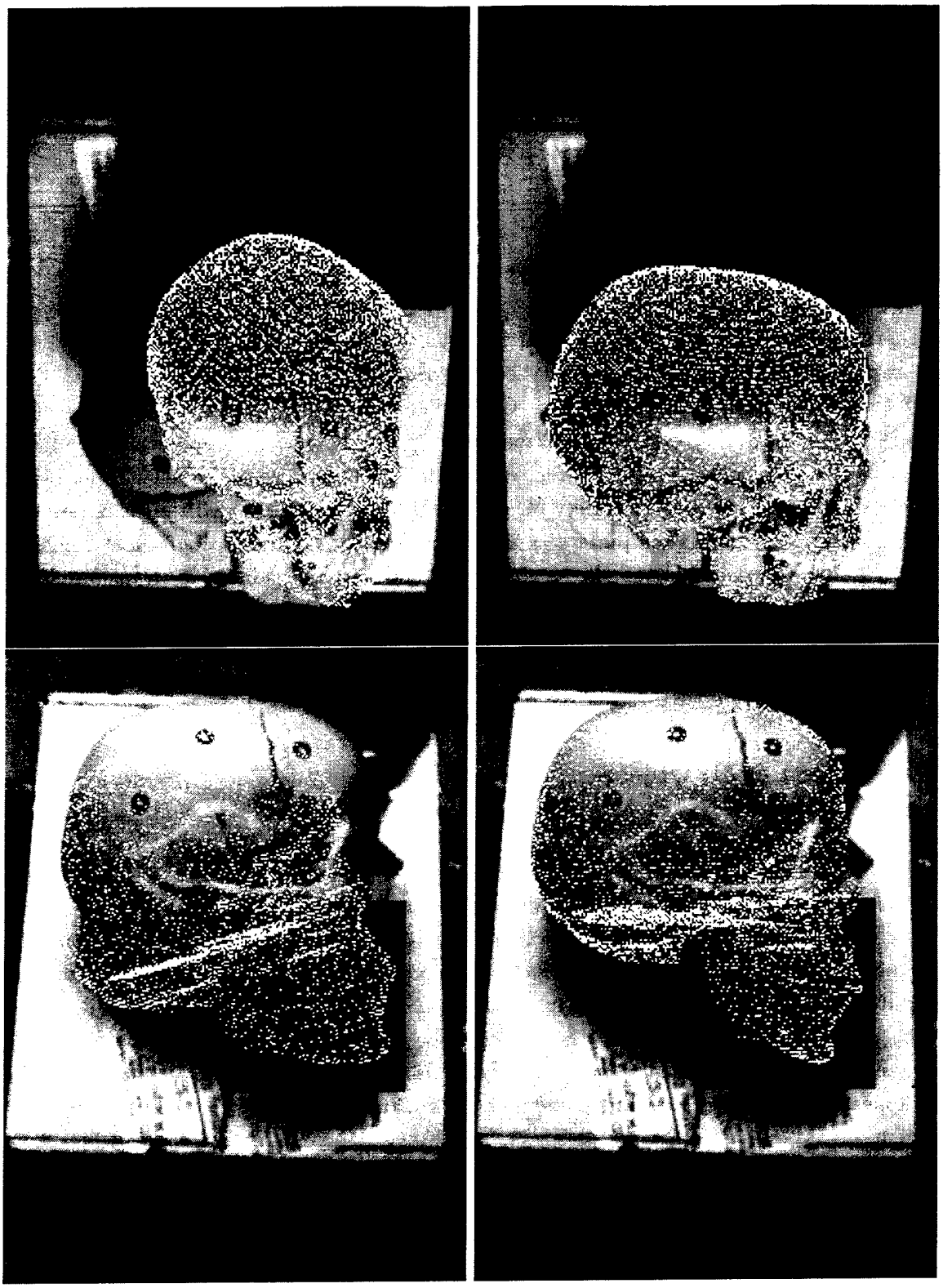


Figure 2: Skull Alignment Experiments: Initial Alignment, Final Alignment, Initial Alignment with Occlusion, Final Alignment with Occlusion

projection [9]. We used a vector difference metric for the normals. 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 a 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 includes 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 $\overline{|\triangle\theta|}$). Finally, the number of poses that successfully converged near the correct solution is reported. The final variance statistics are only computed over the "good" poses.

The third and fourth images in Figure 2 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 [7] for further discussion).

These experiments demonstrate that maximization of mutual information can align complex 3D objects to real images efficiently and reliably. Mutual information does have local maxima from which stochastic gradient ascent cannot escape. A complete object recognition system would require some mechanism for discarding local maxima.

4.2.2 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 Figure 3. The images on

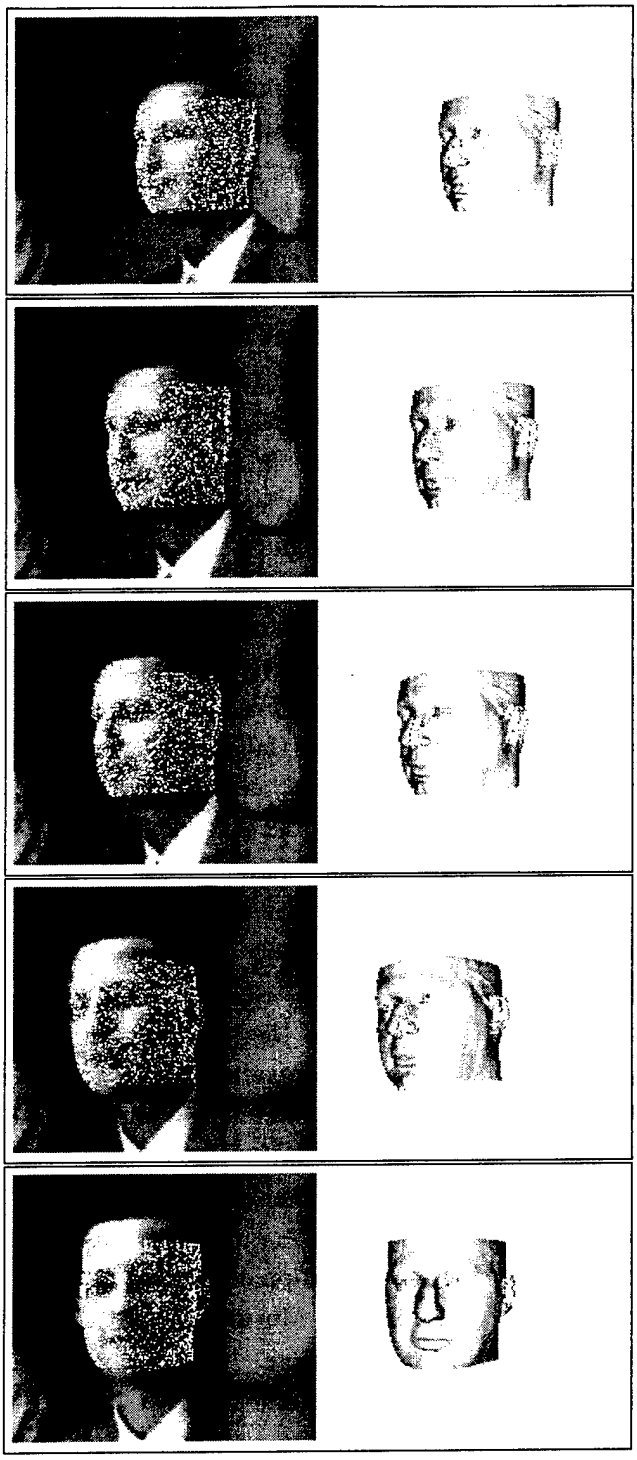


Figure 3: Video Head Tracking Experiment

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.)

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.

4.3 Image-Based Alignment

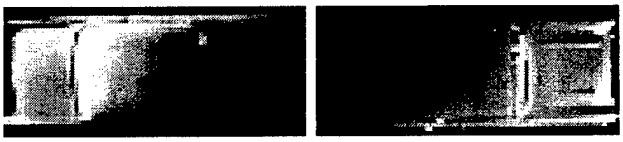


Figure 4: Car Model Images

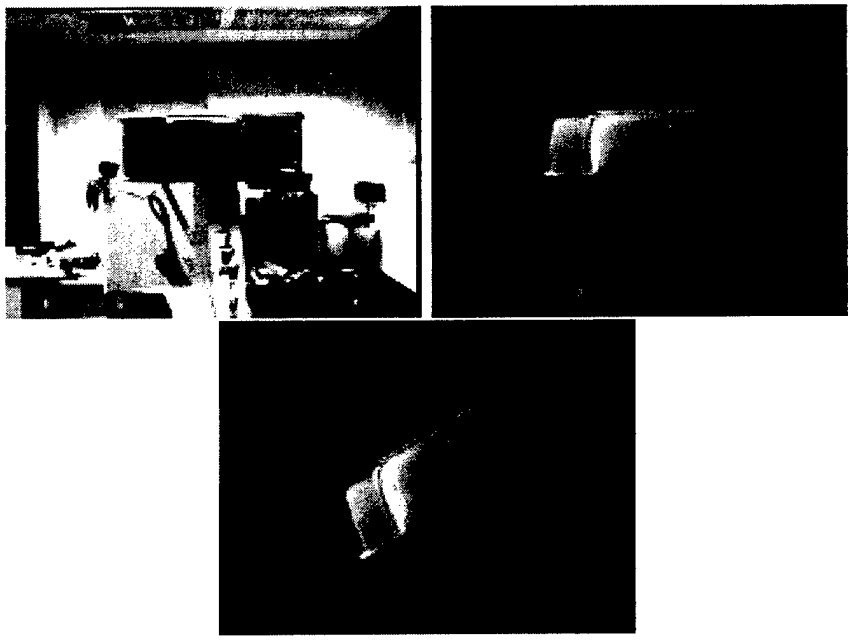


Figure 5: Car Image, Final Pose of Car Model, and Initial Pose

In our final experiment we align video images taken of an object under different lighting conditions. We were motivated by a commonly occurring situation: it is often difficult to obtain a good 3D model of an object. Here we construct a model from a pair of images that can be aligned to new target images taken under different lighting conditions. An example is shown in Figures 4 and 5.

It is well known from photometric stereo research [9] that three images under different illumination are sufficient to build a three dimensional model of a surface. The three images and knowl-

edge of the surface properties of the object are enough to constrain the missing parameters of the model: the normal and the albedo. Furthermore, for any surface patch the normal, the albedo and the lighting are sufficient to predict the intensity of a novel image. As before we can define a function that relates the model and a target image $v(T(x_i)) = F(G(u_1(x_i), u_2(x_i), u_3(x_i)), P)$. $G()$ is a function from the three model images to the normal and the albedo, and $F()$ predicts image intensities. Luckily, we need not actually know $G()$ or $F()$. If they exist and are informative there will be high mutual information between any novel image and the model.

If we knew, a priori, that the entire model had the same albedo we would need only two images to determine the remaining unknown parameter: the surface normal. Consequently, a model that comprised two images would have high mutual information with novel images. Interestingly, this can be true even when the model contains several discrete types of surface. If we could separate the points that came from each type of surface, each group would have a separate unknown function that predicted the target image from the model. Conditioned on being from a particular group, the model would have high information about the the target image. If there were a small number of groups there would be only a small number of values that the target image could take on at any point, one for each group. The resulting joint distribution retains high mutual information even when the group of the point is unknown.

To demonstrate this phenomena we built a model using the two images in Figure 4. Figure 5 shows the target image, the final pose obtained after alignment, and the initial pose of the model.

Technically this experiment is very similar to the MRI experiments, the main difference being that u had two dimensional values. We used a σ of 0.1 for all distances. The sample size was twenty. The learning rate was 0.002 for 1000 iterations. Experiments demonstrated a capture range of about 40% of the length and width of the car, and rotations of up to 35 degrees.

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

In a typical vision application it is an intensity-based, rather than feature based method. 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 4.1), 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 [10] is prominent among them. Some methods use a metric that is proportional to the number of edges that coincide (see the excellent survey articles: [11][12]). 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 [13] [14]. 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 [7].

Alignment by extremizing properties of the joint signal has been used by Hill and Hawkes [15] 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. [16] 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 [17]. 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 [18]. 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 technique uses 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 [19], Becker and Hinton [20] and Bell and Sejnowski [?].

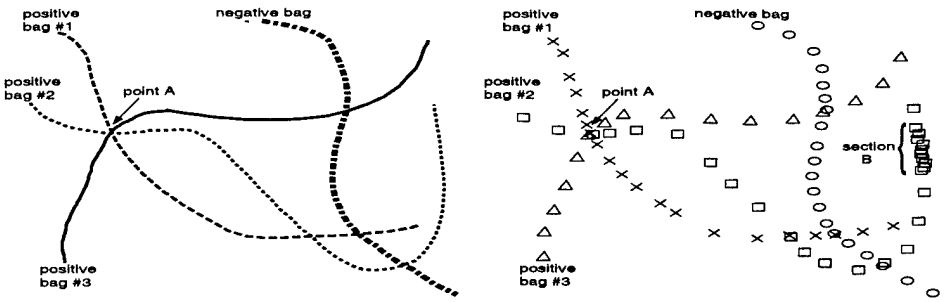
6 A Framework for Multiple Instance Learning — by Oded Maron

One of the drawbacks of applying the supervised learning model is that it is not always possible for a teacher to provide labeled examples for training. Multiple-instance learning provides a new way of modeling the teacher's weakness. Instead of receiving a set of instances which are labeled positive or negative, the learner receives a set of *bags* that are labeled positive or negative. Each bag contains many instances. A bag is labeled negative if all the instances in it are negative. On the other hand, a bag is labeled positive if there is at least one instance in it which is positive. From a collection of labeled bags, the learner tries to induce a concept that will label individual instances correctly. This problem is harder than even noisy supervised learning since the ratio of negative to positive instances in a positively-labeled bag (the noise ratio) can be arbitrarily high.

The first application of multiple-instance learning was to drug activity prediction. In the activity prediction application, one objective is to predict whether a candidate drug molecule will bind strongly to a target protein known to be involved in some disease state. Typically, one has examples of molecules that bind well to the target protein and also of molecules that do not bind well. Much as in a lock and key, shape is the most important factor in determining whether a drug molecule and the target protein will bind. However, drug molecules are flexible, so they can adopt a wide range of shapes. A positive example does not convey what shape the molecule took in order to bind – only that *one* of the shapes that the molecule can take was the right one. However, a negative example means that none of the shapes that the molecule can achieve was the right key.

The multiple-instance learning model was only recently formalized by [21]. They assume a hypothesis class of axis-parallel rectangles, and develop algorithms for dealing with the drug activity prediction problem described above. This work was followed by [22], where a high-degree polynomial PAC bound was given for the number of examples needed to learn in the multiple-instance learning model. [23] gives a more efficient algorithm, but makes very restrictive assumptions on the way the data is generated.

In this paper, we describe a framework called *Diverse Density* for solving multiple-instance problems. Diverse Density is a measure of how well a hypothesis performs with multiple-instance training examples. Maximizing Diverse Density, either within a feature space or across different feature subsets, is the goal of our algorithm. We show results of applying this algorithm to a difficult synthetic training set as well as the "musk" data set from [21]. We then use Diverse Density in two novel applications: one is to learn a simple description of a person from a series of images that are labeled positive if the person is somewhere in the image and negative otherwise. The other is to deal with a high amount of noise in a stock selection problem.



(a)

The different shapes that a molecule can take on are represented as a path. The intersection point of positive paths is where they took on the same shape.

(b)

Samples taken along the paths. Section B is a high density area, but point A is a high Diverse Density area.

Figure 6: A motivating example for Diverse Density

7 Diverse Density

We motivate the idea of Diverse Density through a molecular example. Suppose that the shape of a candidate molecule can be adequately described by a feature vector . One instance of the molecule is therefore represented as a point in n-dimensional feature space. As the molecule changes its shape (through both rigid and non-rigid transformations), it will trace out a manifold through this n-dimensional space[1]. Figure 6(a) shows the paths of four molecules through a 2-dimensional feature space.

If a candidate molecule is labeled positive, we know that in at least one place along the manifold, it took on the right shape for it to fit into the target protein. If the molecule is labeled negative, we know that none of the conformations along its manifold will allow binding with the target protein. What do the positive and negative manifolds tell us about the location of the correct shape in feature space? The answer: it is where all positive feature-manifolds intersect without intersecting any negative feature-manifolds. For example, in Figure 6(a) it is point A.

Unfortunately, a multiple-instance bag does not give us complete distribution information, but only some arbitrary sample from that distribution. Therefore, Figure 6(a) becomes Figure 6(b). The problem of trying to find an intersection changes to a problem of trying to find an area where there is both high density of positive points and low density of negative points. The problem with using positive density is illustrated in in Figure 6(b), Section B. We are not just looking for high density, but high "Diverse Density". We define Diverse Density at a point to be a measure of how many different positive bags have instances near that point, and how far the negative instances are from that point.

[1]In practice, one needs to restrict consideration to shapes of the molecule that have sufficiently low potential energy. But, we ignore this restriction in this simple illustration.

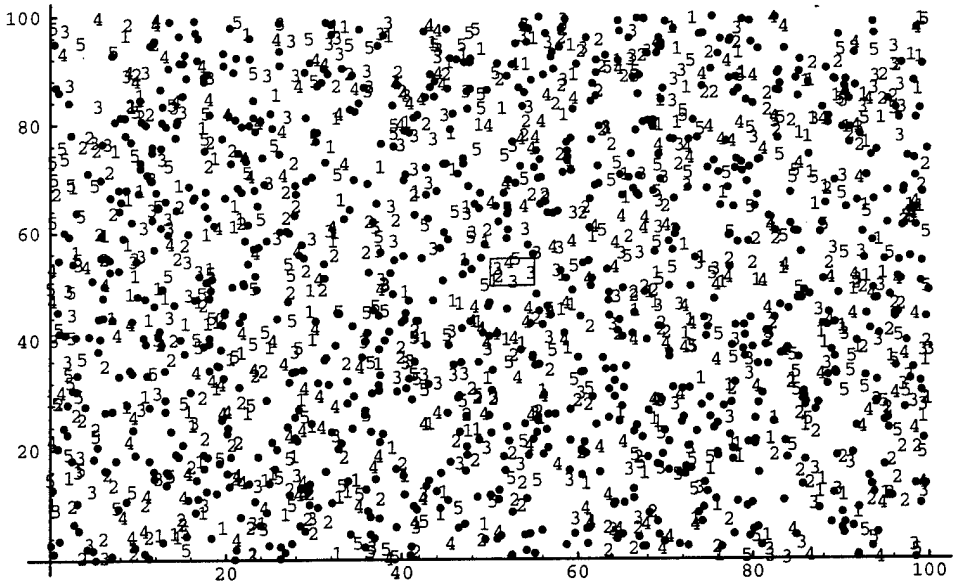


Figure 7: Negative and positive bags drawn from the same distribution, but labeled according to their intersection with the middle square. Negative instances are dots, positive are numbers. The square contains at least one instance from every positive bag and no negatives,

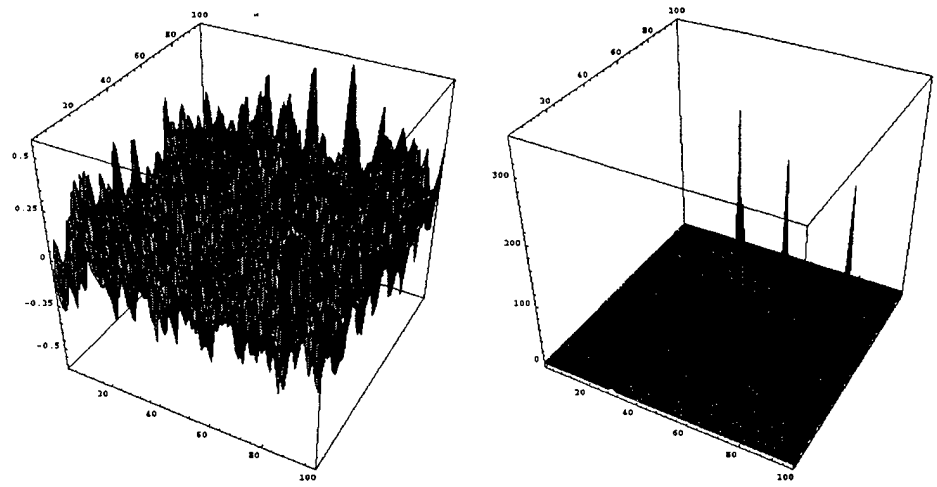
7.1 Algorithms for multiple-instance learning

In this section, we derive a probabilistic measure of Diverse Density, and test it on a difficult artificial data set. We denote positive bags as B_i^+, and the j^{th} point in that bag as B_{ij}^+. Likewise, B_{ij}^- represents a negative point. Assuming that the true concept is a single point t, we can find it by maximizing $\Pr(x = t \mid B_1^+, \cdots, B_n^+, B_1^-, \cdots, B_m^-)$ over all points x in feature space. If we use an uninformative prior over the concept location, this is equivalent to maximizing the likelihood $\Pr(B_1^+, \cdots, B_n^+, B_1^-, \cdots, B_m^- \mid x = t)$. By making the additional assumption that the bags are conditionally independent given the target concept t, the best hypothesis is

$$\arg\max_x \prod_i \Pr(B_i^+ \mid x = t) \prod_i \Pr(B_i^- \mid x = t) \tag{6}$$

This is a general definition of Diverse Density, but we need to define the terms in the products to instantiate it. One possible instantiation is a noisy-or model: the probability that not all points missed the target is $\Pr(B_i^+ \mid x = t) = 1 - \prod_j (1 - \Pr(B_{ij}^+ = x))$, and likewise $\Pr(B_i^- \mid x = t) = \prod_j (1 - \Pr(B_{ij}^- = x))$. If the instances within a bag are not independent, then we can use the instance from each bag which is closest to the target: $\Pr(B_i^+ \mid x = t) \approx \Pr(B_{ij}^+ = x)$, where $j = \arg\min_k \parallel B_{ik}^+ - x \parallel$. Finally, we assume that the data is noisy so we model $\Pr(B_{ij}^+ = x)$ with a Gaussian-like distribution of $\exp(- \parallel B_{ij}^+ - x \parallel)$. Diverse Density at an intersection of n bags is exponentially higher than it is at an intersection of $n - 1$ bags, yet all it takes is one well placed negative instance to drive the Diverse Density down. Note that we can perform feature weighting by maximizing Equation 1 with respect to the set of weights used in computing the distance ($\| \cdot \|$), as in [24].

To test the algorithm, we created the following artificial data set: n positive bags and m



(a) Surface using regular density (b) Surface using Diverse Density

Figure 8: Density surfaces over the example data of Figure 3

negative bags, each with k instances. Each instance was chosen randomly from a $[0, 100] \times [0, 100] \in \mathcal{R}^2$ domain, and the concept was a 5×5 square in the middle of the domain. A bag was labeled positive if at least one of its instances fell within the square, and negative if none did. An example (with $n = m = 5$, and $k = 200$) is shown in Figure 7. The square in the middle contains at least one instance from every positive bag and no negative instances. This is a difficult data set because both positive and negative bags are drawn from the same distribution. They only differ in a small area of the domain.

Using regular density (adding up the contribution of every positive bag and subtracting negative bags; this is roughly what a supervised learning algorithm such as nearest neighbor performs), we can plot the density surface across the domain. Figure 8(a) shows this surface for the data set in Figure 7, and it is clear that finding the peak (a candidate hypothesis) is difficult because of the abundance of local maxima and because many points have similar maxima. However, when we plot the Diverse Density surface (using the noisy-or model) in Figure 8(b), it is easy to pick out the global maximum which is within the desired concept.

The other major peaks in Figure 8(b) are the result of a chance concentration of instances from different bags in another part of the space. With a bit more bad luck, one of those peaks could have eclipsed the one in the middle. However, the chance of this decreases as the number of bags increases. This can be seen in Figure 9, where the probability of the top Diverse Density landing within the true concept increases as either the number of positive or negative bags increases. The number of points per bag was held at 200 and 100 randomly generated data sets were used to estimate the probability of a correct run at every point.

There are two critical factors involved in making a Diverse Density algorithm computationally efficient. One is to insure that the time to compute the Diverse Density at a point does not grow as fast as the total number of training instances. The second is to insure that the time to find the maximum Diverse Density does not grow exponentially with the number of features. One of the reasons for the use of Gaussians as weighting functions is that they drop off fairly quickly.

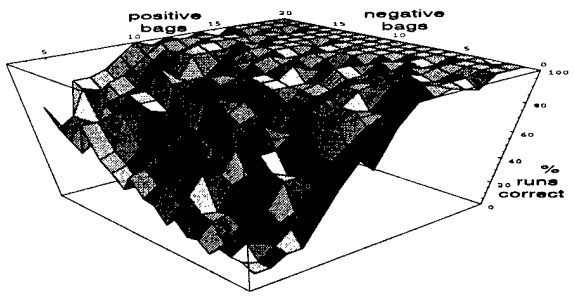


Figure 9: Success of Diverse Density vs. number of training bags

To compute the contribution of a bag at point p, we only need to look at instances that are close to p – all the other ones contribute nearly zero. By partitioning the space (much like [25]), we can achieve nearly constant time computation of Diverse Density at a point.

Finding the maximum Diverse Density is a more difficult issue. In general, we are searching an arbitrary density landscape and the number of local maxima and size of the search space could prohibit any efficient exploration. In this paper, we use gradient ascent with multiple starting points. This has worked succesfully in every test case because we know what starting points to use. The maximum Diverse Density point is made of contributions from some set of positive points. If we start an ascent from every positive point, one of them is likely to be closest to the maximum, contribute the most to it and have a climb directly to it.

8 Applications of Diverse Density

By way of benchmarking, we tested the Diverse Density approach on the "musk" data sets from [21], which were also used in [23]. We also have begun investigating two new applications of multiple-instance learning. We describe preliminary results on all of these below. The musk data sets contain feature vectors describing the surfaces of a variety of low-energy shapes from approximately 100 molecules. Each feature vector has 166 dimensions. Approximately half of these molecules are known to smell "musky," the remainder are very similar molecules that do not smell musky. There are two musk data sets; the Musk-1 data set is smaller, both in having fewer molecules and many fewer instances per molecule. Many (72) of the molecules are shared between the two data sets, but the second set includes more instances for the shared molecules.

We approached the problem as follows: for each run, we held out a randomly selected 1/10 of the data set as a test set. We computed the maximum Diverse Density on the training set by multiple gradient ascents, starting at each positive instance. This produces a maximum feature point as well as the best feature weights corresponding to that point. We note that typically less than half of the 166 features receive non-zero weighting. We then computed a distance threshold that optimized classification performance under leave-one-out cross validation within the training set. We used the feature weights and distance threshold to classify the examples of the test set; an example was deemed positive if the weighted distance from the maximum density point to any of its instances was below the threshold.

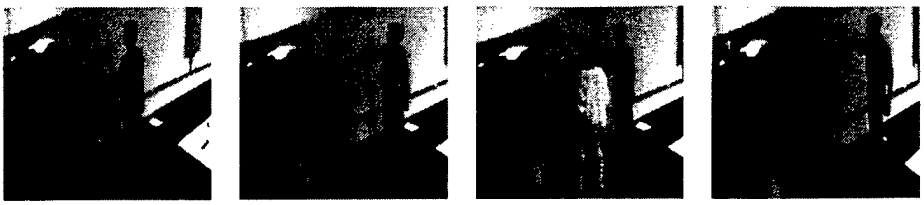


Figure 10: A training set of images with one person in common

The table below lists the average accuracy of twenty runs, compared with the performance of the two principal algorithms reported in [21] (`iterated-discrim APR` and `GFS elim-kde APR`), as well as the `MULTINST` algorithm from [23]. We note that the performances reported for `iterated-discrim APR` involves choosing parameters to maximize *test set* performance and so probably represents an upper bound for accuracy on this data set. The Diverse Density results, which required no tuning, are comparable or better than those of `GFS elim-kde APR` and `MULTINST`.

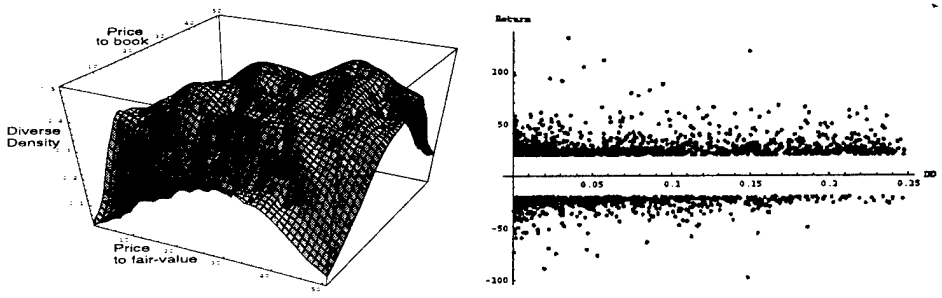
Musk Data Set 1	
algorithm	accuracy
iterated-discrim APR	92.4
GFS elim-kde APR	91.3
Diverse Density	88.9
MULTINST	76.7

Musk Data Set 2	
algorithm	accuracy
iterated-discrim APR	89.2
MULTINST	84.0
Diverse Density	82.5
GFS elim-kde APR	80.4

We also investigated two new applications of multiple-instance learning. The first of these is to learn a simple description of a person from a series of images that are labeled positive if they contain the person and negative otherwise. For a positively labeled image we only know that the person is somewhere in it, but we do not know where. We sample 54 subimages of varying centers and sizes and declare them to be instances in one positive bag since one of them contains the person. This is repeated for every positive and negative image.

We use a very simple representation for the instances. Each subimage is divided into three parts which roughly correspond to where the head, torso and legs of the person would be. The three dominant colors (one for each subsection) are used to represent the image. Figure 10 shows a training set where every bag included two people, yet the algorithm learned a description of the person who appears in all the images.

Another new application uses Diverse Density in the stock selection problem. Every month, there are stocks that perform well for fundamental reasons and stocks that perform well because of flukes; there are many more of the latter, but we are interested in the former For every month, we take the 20 stocks with the highest return and put them in a positive bag, hoping that at least one of them did well for fundamental reasons. Negative bags are created from the bottom 5 stocks in every month. A stock is described by two features: a ranking of its price to book ratio and a ranking of its price to fair-value ratio. Figure 11(a) shows the resulting Diverse Density landscape. The training data is taken from the 600 top stocks over the last 18 years. When tested on stocks that had wild fluctuation in their returns (more than $\pm 20\%$ change in a month), the stocks with high Diverse Density had an average monthly return of 21.88. Those with high



(a) Diverse Density surface (b) Return vs. Diverse Density

Figure 11: Applying Diverse Density to stock selection

price to book and price to fair-value had an average return of 5.27. The baseline average return over all wildly fluctuating stocks is 7.04. In Figure 11(b), the return of every wildly fluctuating stock is plotted against the Diverse Density of that stock. The higher the Diverse Density, the more likely the stock is to have positive return rather than negative return.

Acknowledgments

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

References

[1] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, Inc., third edition, 1991.

[2] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[3] John S. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In David S. Touretzky, editor, *Advances in Neural Information Processing 2*, pages 211–217. Morgan Kaufman, 1989.

[4] B. Widrow and M.E. Hoff. Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, volume 4, pages 96–104. IRE, New York, 1960.

[5] Lennart Ljung and Torsten Söderström. *Theory and Practice of Recursive Identification.* MIT Press, 1983.

[6] Simon Haykin. *Neural Networks: A comprehensive foundation.* Macmillan College Publishing, 1994.

[7] Paul A. Viola. *Alignment by Maximization of Mutual Information.* PhD thesis, Massachusetts Institute of Technology, 1995. MIT AI Laboratory TR 1548.

[8] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* John Wiley and Sons, 1991.

[9] B.K.P. Horn. *Robot Vision.* McGraw-Hill, New York, 1986.

[10] D.P. Huttenlocher, K. Kedem, K. Sharir, and M. Sharir. The Upper Envelope of Voronoi Surfaces and its Applications. In *Proceedings of the Seventh ACM Symposium on Computational Geometry*, pages 194–293, 1991.

[11] P.J. Besl and R.C. Jain. Three-Dimensional Object Recognition. *Computing Surveys*, 17:75–145, 1985.

[12] R.T. Chin and C.R. Dyer. Model-Based Recognition in Robot Vision. *Computing Surveys*, 18:67–108, 1986.

[13] D.G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, 1985.

[14] W.M. Wells III. *Statistical Object Recognition.* PhD thesis, MIT Department Electrical Engineering and Computer Science, Cambridge, Mass., 1992. MIT AI Laboratory TR 1398.

[15] Derek LG Hill, Colin Studholme, and David J. Hawkes. Voxel Similarity Measures for Automated Image Registration. In *Proceedings of the Third Conference on Visualization in Biomedical Computing*, pages 205 – 216. SPIE, 1994.

[16] A. Collignon, D. Vandermuelen, P. Suetens, and G. Marchal. 3D Multi-Modality Medical Image Registration Using Feature Space Clustering. In N. Ayache, editor, *Computer Vision, Virtual Reality and Robotics in Medicine*, pages 195 – 204. Springer Verlag, 1995.

[17] M.A. Turk and A.P. Pentland. Face Recognition using Eigenfaces. In *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, Lahaina, Maui, Hawaii, June 1991. IEEE.

[18] A. Shashua. *Geometry and Photometry in 3D Visual Recognition.* PhD thesis, M.I.T Artificial Intelligence Laboratory, AI-TR-1401, November 1992.

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

[20] Suzanna Becker and Geoffrey E. Hinton. Learning to make coherent predictions in domains with discontinuities.

[21] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the Multiple-Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence Journal*, 89, 1997.

[22] P. M. Long and L. Tan. PAC-learning axis alligned rectangles with respect to product distributions from multiple-instance examples. In *Proceedings of the 1996 Conference on Computational Learning Theory*, 1996.

[23] P. Auer. On Learning from Multi-Instance Examples: Empirical Evaluation of a theoretical Approach. NeuroCOLT Technical Report Series, NC-TR-97-025, March 1997.

[24] D. G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 7:72–85, 1995.

[25] S. M. Omohundro. Bumptrees for Efficient Function, Constraint, and Classification Learning. In Lippmann, Moody, and Touretzky, editors, *Advaces in Neural Information Processing Systems 3*, San Mateo, CA, 1991. Morgan Kaufmann.