Reconnaissance des Formes: Techniques Récentes et Applications

Shape Recognition: Recent Techniques and Applications

Martial Hébert

The Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213, U.S.A.
hebert@ri.cmu.edu

Résumé

Le développement de techniques pour la reconnaissance de formes, au sens large, est un pas important dans le développement de systèmes intelligents qui peuvent être utilisés dans des applications réelles. Dans cet article, nous nous proposons de définir des techniques de reconnaissance générales et robustes en utilisant des représentations intermédiaires des données qui ne requièrent pas l'extraction de représentations structurelles explicites. Ces techniques sont basées sur une définition précise des représentations des données, étant donnée une tâche de reconnaissance. Trois exemples sont utilisés pour illustrer cette approche. Les exemples sont choisis dans les domaines de reconnaissance d'objets, reconnaissance d'images, et reconnaissance d'évènements dans des séquences d'images. Des applications à des problèmes industriels et scientifiques sont également présentés.

Mots Clef

Reconnaissance d'objets, reconnaissance d'images, séquence d'images.

Abstract

The development of shape recognition techniques, in its broadest sense, is a key step toward developing intelligent systems that can operate in the real world. In this paper, we argue that general, efficient, shape recognition techniques that use intermediate representations of the data without requiring restrictive, higher-level representations can be designed. The key is proper engineering of the data representation, based on the target task, and of the comparison algorithm used for comparing observed and model data. Three examples are used to illustrate this approach. The examples are in the area of 3-D object recognition, image recognition, and recognition of events in sequences of images. Applications to industrial and scientific problems are also discussed.

Keywords

Object recognition, image recognition, image séquences.

1 Introduction

The development of object recognition techniques, in its broadest sense, is a key step toward developing intelligent systems that can operate in the real world. This endeavor will be successful only if techniques that are, at the same time, efficient, robust, and general, can be developed. The situation is complicated by the fact that massive amounts of data have to be handled, e.g., images or sequences of images. The traditional way of approaching object recognition problems is to define high-level representations that carry geometric or semantic information that is presumably easier to manipulate than the data itself. For example, features in images are compared between observed data and model representation. Supposedly, this approach leads to efficient algorithms because of the reduction in the volume of data to be manipulated for recognition.

Unfortunately, it is generally the case that extracting such structural or semantic representations from raw data is in itself a difficult problem which requires algorithms that are, unfortunately, neither efficient nor robust. At the same time, because of the drastic simplifications and data reduction that they entail, high-level representations can never accurately encompass the broad spectrum of entities that they are intended to represent; no representation can be as rich as the data itself. A similar observation was made by Brooks earlier in the context of mobile robots: "The world is its best model."

Based on those observations, it would seem that a better way to approach recognition problems would be to stay as close to the data as possible. More precisely, in this approach, the raw input data, either from the observed signals or from the stored models, is re-arranged so that different data sets can be directly compared. This type of approach has been used in many fields. For example, in speech recognition, moving from feature-based, structural algorithms to data-based algorithms based on MRF representations has been tremendously successful. In computer vision, the introduction of area-based matching for stereo, a technique that works directly with the raw data, in favor of feature-based techniques

has led to an explosion in the number of systems in practical use. In the area of object recognition in images, the use of image comparison has been successfully used in limited cases such as the eigen-images techniques for the recognition of isolated objects and some of the face recognition techniques.

For more complex recognition tasks, such as 3-D/3-D or 3-D/2-D recognition, however, the techniques still rely largely on high-level representation. In this paper, we argue that it is possible to design general, efficient, shape recognition techniques can be designed that use intermediate representations of the data without requiring restrictive, higher-level representations. The key is proper engineering of the data representation, based on the target task, and of the comparison algorithm used for comparing observed and model data. Since it is difficult to make general statements about such techniques, the approach is illustrated through three examples that address increasingly less constrained recognition problems.

The first example is in the most traditional area of shape recognition: 3-D to 3-D recognition. This example will show that, by projecting the raw 3-D data into the appropriate space, discriminating signatures in the form of 2-D signals can be created for every point on a surface, and that those signatures can be directly used for comparing and matching surfaces. The second example addresses a less constrained problem of recognizing 3-D objects from prior 2-D observations of the object in its natural environment. This example will show that recognition can be achieved by comparing image structures without feature matching, provided that the appropriate attributes are extracted from the images. This approach borrows heavily from, and benefits from, classical developments in the area of image registration and mosaicing. Finally, the third approach considers a much broader recognition problem in which the entities to be recognized include not only shape and appearance information, but also information about the change in shape and appearance over time. Although this problem uses weaker models, it can be shown that, with the proper data representation, effective recognition strategies can be designed.

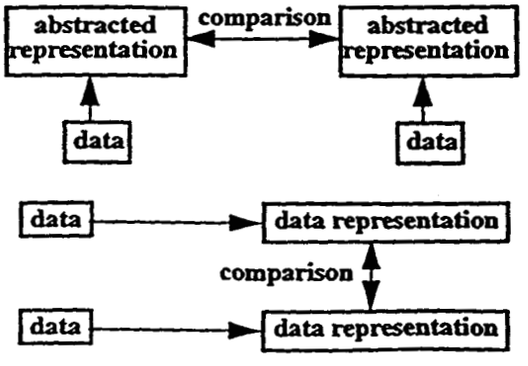

Figure 1: Two approaches to recognition.

Demonstrating those techniques in actual applications is critical in order to validate claims on efficiency and robustness. The first example was extensively applied to interior mapping applications in industry, and well as 3-D model building applications. In this case, objects were reliably recognized in hundreds of images of industrial environments in a system operable by untrained users. The second example was used in the context of mobile robot positioning using landmark recognition. Recognition was performed on outdoor scenery with little control over the illumination and viewpoints. Full-scale integration in robot systems is under way. The third example was developed in the context of a biological application in which the developmental history of embryos is automatically recovered from large sequences of microscope images. Recognizing events and sequences of events in such sequences leads to a better understanding of the morphogenesis of the development and, possibly, to the discovery and understanding of mutations.

2 Object recognition with strong geometric models and application to interior mapping

The first example addresses the most traditional aspect of shape recognition: recognizing three-dimensional objects in complex scenes. This problem is a traditional one that has been extensively studied. In typical approaches, higher-level representations are computed from the 3-D data in the form of surface features, parametric surface patches, or representations of parts, such as generalized cylinders and are then compared to similar representations for the shape models. Although great progress has been achieved in this area, the existing approaches require a substantial amount of algorithmic machinery to construct high-level representations from data sets in a manner that is robust to noise, clutter, and occlusion. Furthermore, most techniques impose severe limitations on the class of objects that they can manipulate because of strict constraints on the type of surface representations that they use.

The alternative explored here is to attempt to compare shapes by directly comparing point sets using the appropriate data structures and the appropriate comparison functions. The key to this approach is to transform the data from a format that is hard to manipulate, i.e., 3-D surfaces, to a 2-D image format for which an arsenal of processing and matching operators exists. Results obtained using this approach show that careful engineering of the data representation leads to effective recognition techniques suitable for a variety of applications.

This section is an brief overview of the data representation and the corresponding recognition algorithm. Detailed descriptions of the basic representation, algorithms, and theoretical analysis of clutter can be found in [16] and [17]; applications to surface registration can be found in [19]; sup-

porting algorithms for surface filtering and sampling are described in [18].

2.1 Data representation

Given an object, consider as a basic shape element an oriented point, defined by a point p on its surface and the normal vector n at p. The pair $O = (p,n)$ defines a local coordinate system using the tangent plane $\mathcal{P}$ through p oriented perpendicularly to n and the line $\mathcal{L}$ through p parallel to n. The two coordinates of any point q in this basis are α, the perpendicular distance to the line $\mathcal{L}$, and β the signed perpendicular distance to the plane $\mathcal{P}$ A straightforward mapping S_O that maps 3-D points x to the 2-D coordinates of a particular basis (p,n) corresponding to oriented point O can be defined. Each oriented point O on the surface of an object has a unique mapping S_O associated with it.

When S_O is applied to all of the other points on the surface of the object, a set of 2-D points is created which can be represented by an image I_O. I_O is a description of the shape of an object because it is the projection of the relative position of 3-D points that lie on the surface of an object to a 2-D space where some of the 3-D metric information is preserved. Since the images describe the shape of an object independently of its pose, they are object-centered shape descriptions.

Another way to view the image I_O is as a two-dimensional signature associated with O. Although this signature is not unique, it can be shown that, for a general curved object, points can be discriminated based on this signature alone. More precisely, correspondences are established between oriented points by comparing their corresponding images. Comparing the 2-D signature images is a fast operation which can be performed for a large number of points, thus making recognition possible. In other words, by selecting the appropriate data representation, matching becomes a simple image comparison operation.

Figure 2 shows some images for a CAD object. The image values are encoded so that the darker a pixel is, the more points(α,β) have fallen into the corresponding bin. A number of technical issues such as image resolution, smoothing and quantization have to be addressed in order to generate those images; those issues are not described in detail here.

The idea of encoding the relative position of many points on the surface of an object in an image or histogram is not new. Ikeuchi et. al. [14] propose invariant histograms for SAR target recognition. This work is view-based and requires feature extraction. Guéziec and Ayache [12] store parameters for all points along a curve in a hash table for efficient matching of 3-D curves. Their method requires the extraction of extremal curves from 3-D images.

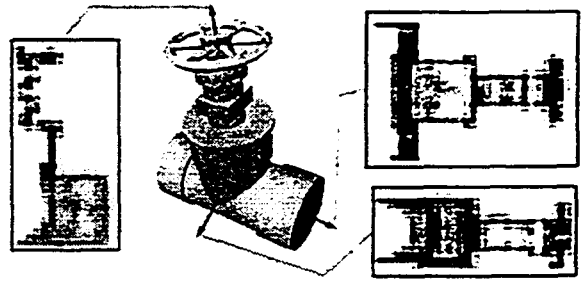

Figure 2: Some example images generated for three different oriented points on a CAD model of a valve.

Chua and Jarvis [6] present an algorithm for matching 3-D free-form surfaces by matching points based on principal curvatures. Similarly, Thirion [37] presents an algorithm for matching 3-D images based on the matching of extremal points using curvatures and Darboux frames. Pipitone and Adams [27] propose the tripod operator which, when placed on the surface of an object, generates a few parameters describing surface shape. Bergevin et. al. [2] propose a registration algorithm based on matching properties of triangles generated from a hierarchical tessellation of an object's surface. The approach presented here differs from these because the images computed at each point are much more discriminating than principal curvatures and angles used in other approaches.

2.2 Recognizing shapes

Images generated from the scene and the model will be similar because they are based on the shape of objects imaged. However, they will not be exactly the same due to variations in surface sampling and noise from different views. For example, in Figure 3 the vertex positions and connectivity of two models of a femur are different, yet the images from corresponding basis points are similar. The key in finding corresponding points on two surfaces is to define a suitable shape comparison function which can compare the images as defined above and be robust to noise, occlusion, and sampling.

The linear normalized correlation coefficient provides a simple way to compare two images that can be expected to be similar across the entire image. In practice, images generated from range data will have clutter (extra data) and occlusions (missing data). A first step in limiting the effect of clutter and occlusion is to compare these images only at the pixels where both of the images contain valid data. In other words, the data used to compute the linear correlation coefficient is taken only from the region of overlap between two images. In this case, knowledge of the image generation process is used to eliminate outliers in the correlation computation.

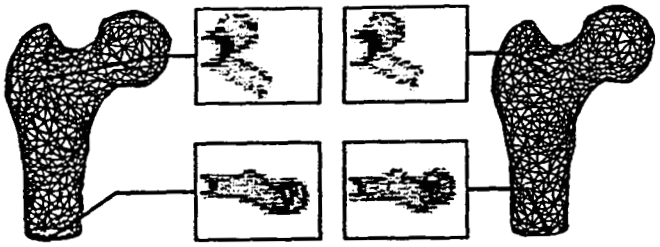

Figure 3:: Images generated from two different samplings of a model of a femur. Although the samplings are different, the images generated from corresponding points are similar.

Since the linear correlation coefficient is a function of the number of bins used to compute it, the amount of overlap will have an effect on the correlation coefficients obtained. The more bins used to compute the correlation coefficient, the more confidence there is in its value. The variance of the correlation coefficient is included in the calculations of the relative similarity between two images so that the similarity measure between pairs of images with differing amounts of overlap can be compared. The actual similarity function C used for comparing images P and Q where N is the number of overlapping bins is:

$$C(P,Q) = (\text{atanh}(R(P,Q)))^2 - \lambda\left(\frac{1}{N-3}\right)$$

The similarity function will return a high value for two images that are highly correlated and have a large number of overlapping bins. The change of variables, a standard statistical technique ([10] Chapter 12) performed by the hyperbolic arctangent function, transforms the correlation coefficient into a distribution where the variance is independent of the mean. The coefficient λ is used to weight the variance against the expected value of the correlation coefficient.

Before recognition (off-line), images are generated for all points on the model surface mesh and stored in an stack. At recognition time, a scene point is selected randomly from the scene surface mesh and its image is generated. The scene image is then compared to all of the images in the model image stack and the similarity value C for each image pair is calculated and inserted in a histogram. This procedure to establish point correspondences is repeated for a random sampling of scene points that adequately cover the scene surface.

Possible corresponding model points are chosen by finding the upper outliers in the histogram of the similarity values for each scene point. This method of choosing correspondences is reliable for two reasons. First, if no outliers exist, then the scene point has an image that is very similar to all of the model images, so definite correspondences with this scene point should not be established. Second, if multiple outliers exist, then multiple model points are similar to a single scene point, and thus should be considered in the matching process. For this, a standard outlier detection method is used ([10] Chapter 1). Figure 4 shows a similarity measure histogram with detected outliers.

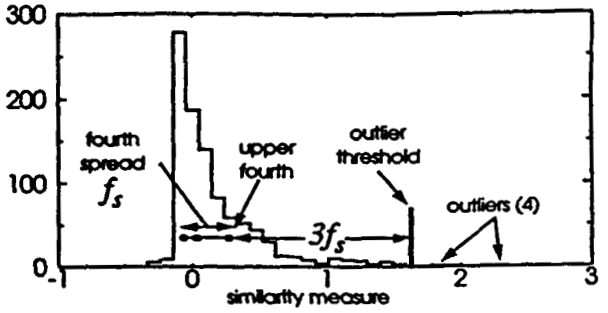

Figure 4:: Typical similarity measure histogram; the outliers correspond to potential matches.

During matching, a single point can be matched to more than one point for two reasons. First, symmetry in the data and in image generation may cause two points to have similar images. Second, spatially close points may have similar images. Furthermore, if an object appears multiple times in the scene, then a single model point will match multiple scene points.

The similarity measure provides a way to rank correspondences so that only reasonable correspondences are established. During matching, some points selected from scene clutter may be incorrectly matched to model points. However, given the numerous correspondences, it is possible to reason about which correspondences are actually on the model based on properties of the correspondences taken as a group. This *integral* approach is robust because it does not require reasoning about specific point matches to decide which correspondences are the best. This approach is in contrast to hypothesize and test and alignment paradigms of recognition where the minimal number of correspondences required to match model to scene are proposed and then verified through some other means.

First, the similarity measure is used to remove unlikely correspondences. All correspondences with similarity measures that are less than a given fraction of the maximum similarity measure of all of the correspondences are eliminated.

The second method for filtering out unlikely correspondences uses geometric consistency which is a measure of the likelihood that two correspondences can be grouped together to calculate a transformation of model to scene. If a correspondence is not geometrically consistent with other correspondences, then it cannot be grouped with other correspondences to calculate a transformation, and it should be eliminated.

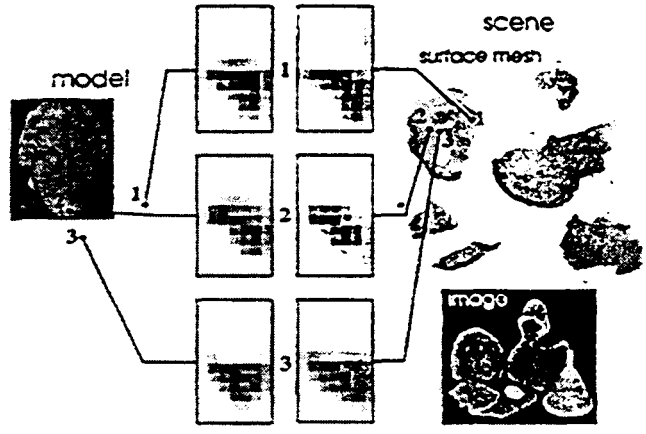

Figure 5: Three scene points and their best matching model points shown with associated best matching images for a simple scene.

Single correspondences cannot be used to compute a transformation from model to scene because an oriented point basis encodes only five of the six necessary degrees of freedom. At least two oriented point correspondences are needed to calculate a transformation if position and normals are used. To avoid combinatorial explosion, geometric consistency is used to cluster the correspondences into a few groups from which plausible transformations are computed. Since many correspondences are grouped together and used to compute a transformation, that resulting transformation is more robust than one computed from a few correspondences.

The verification algorithm is a formulation of the iterative closest point algorithm [38] that can handle partially overlapping point sets and arbitrary transformations because it is initialized with a transformation generated from correspondences determined by matching of images.

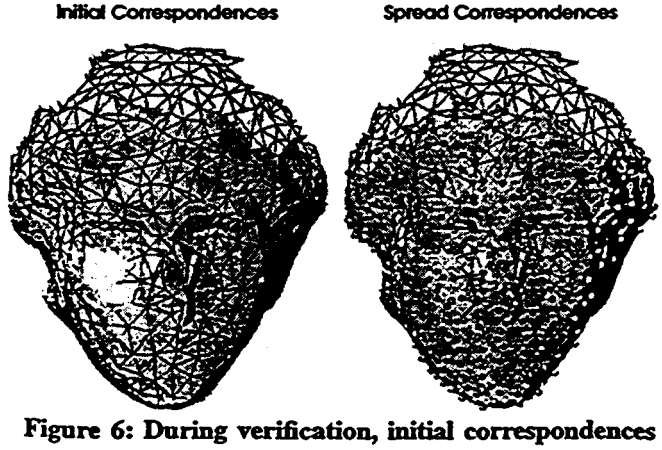

Figure 6: During verification, initial correspondences are spread over two views (one wireframe, the other shaded). Correspondences are prevented from being established outside the overlap of the views.

Verification starts with an initial set of point correspondences from which the transformation of model to scene is computed and then applied to the model points. For each correspondence, new correspondences are established between the nearest neighbors of the model point and nearest neighbors of the corresponding scene point if the distance between closest points is less than a threshold D_v. By finding scene points that are close to model points, this step grows the correspondences from those correspondences already established. The transformation based on the new correspondences is computed and then refined using traditional ICP. The growing process is repeated until no more correspondences can be established.

Figure 6 illustrates how initial correspondences, established by matching images, are spread over the surfaces of two range views of a plastic model of the head of the goddess Venus. The correspondences are established only in the regions where the two surface meshes overlap, thus preventing a poor registration caused by correspondences being established between non-overlapping regions.

The recognition algorithm can be easily extended to simultaneous recognition of multiple models. Recognition with multiple models is similar to recognition with one model except that each scene point is compared to the images stored for all the models. The rest of the algorithm is the same except that correspondences with model points from different models are prevented from being clustered.

2.3 Limiting the effect of clutter and occlusion

Because an image is a global encoding of the surface, it would seem that any disturbance such as clutter and occlusion would prevent matching. In fact, this representation is resistant to clutter and occlusion, assuming that some precautions are taken. This will be described in detail in Section 2.3.

In real scenes, clutter and occlusion are omnipresent. Any object recognition system designed for the real world must somehow deal with clutter and occlusion. Some systems perform segmentation before recognition in order to separate clutter from interesting object data. The effect of clutter is manifested as a corruption of the pixel values of images generated from the scene data. To some extent, the effect of clutter and occlusion can be limited by computing the images only locally around each basis point. This is done by limiting the maximum distance between the oriented point basis and a point in the mesh contributing to the image and by limiting the angle between the oriented point basis surface normal and the surface normal of other points on the surface.

In order to quantify the effect of clutter, a simple model can be built for elliptical objects. The clutter model combines the angular and distance thresholds explained above with the

fact that objects of non-zero thickness cannot intersect to show that clutter is limited to connected regions in images. The size of these connected regions can be evaluated based on the percentage of clutter in the scene. Finally, the decrease in similarity due to clutter can be estimated based on the size of the clutter regions.

Clutter and occlusion manifest themselves as extra and missing points in the scene where the number of these points is bounded. Therefore, it is reasonable to assume that the total change δ_i of any pixel in a scene image that is corrupted by clutter data is bounded $|\delta_i| \leq \delta$. Assuming that the number of corrupted pixels in the scene image is N_C and the total number of pixels is N and that the model and scene pixel values are normalized on [0,1], it can be shown that the lower bound on the correlation coefficient when comparing model and scene images is:

$$\rho_{LB} = \left(\sigma_m^2 - \frac{N_C}{N}\delta\right) \Big/ \left(\sigma_m \sqrt{\sigma_m^2 + \frac{N_C}{N}(\delta + \delta^2)}\right)$$

where σ_m^2 is the variance of the pixels in the model image. Hence, the worst case effect of clutter and occlusion grows sub-linearly with the area of corruption in the scene image. Since clutter and occlusion cannot corrupt an entire image and the effect of the corruption on the correlation coefficient is bounded, it can be concluded that matching of images is only moderately affected by clutter and occlusion. This result can be verified in practice by measuring the amount of clutter in a scene and comparing the predicted correlation value with the actual image correlation. Figure 7 shows an example with both occlusion and clutter.

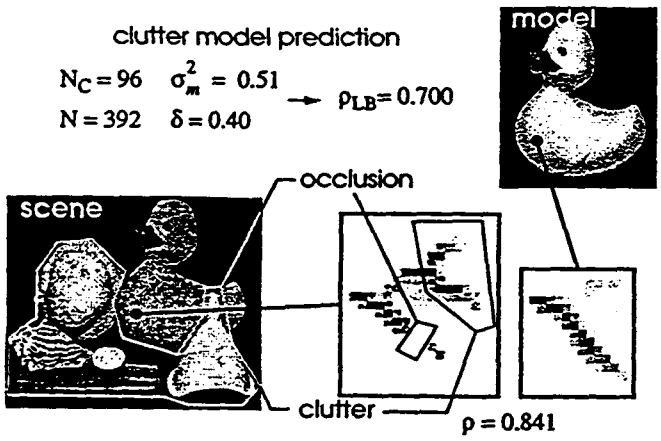

Figure 7: Experimental verification of clutter model. 96 of 392 pixels in the scene image are corrupted by an amount δ less than 0.40. The correlation coefficient for the two images (0.841) is well above the lower bound (0.700) predicted by the theoretical model.

2.4 Discussion

This shape recognition algorithm does not require the extraction of high-level primitives, or the generation of a structural representation. Instead, it uses a simple and carefully designed data representation in order to convert the input data sets, i.e., sets of points on 3-D surfaces, to another representation, the 2-D signature images, that are suitable for point comparison. The key here is that while 3-D surfaces are difficult to compare, the converted data representation is easy to compare. In particular, standard image comparison can be modified in order to construct an effective similarity measure. This example illustrates the basic premise in the case of classical 3-D shape recognition; careful selection of the data representation leads to simpler, more general, and more effective algorithms.

An important measure of the success of an approach is its applicability to real problems. The 3-D recognition technique described here was applied primarily to the problem of mapping industrial plants using sequences of range images. The goal is to identify as many of the objects in the environment as possible based on a library of models in order to build a virtual model of a plant. The virtual model can then be used to perform operations by using a robot [20].

The challenge in this application is to be able to reliably recognize objects over hundreds of images, and to compute their positions accurately enough for robotic manipulation. In addition, because this system is to be used by workers unfamiliar with the underlying technology, a high level of recognition accuracy over many hours of operation is necessary.

Figure 8 shows the result of recognizing four different industrial objects in cluttered industrial scenes. Before recognition, the scene data is processed to remove long edges and small surface patches, then smoothed and re-sampled. In all examples, the scene data is complex with a great deal of clutter. Furthermore, all the models exhibit symmetry, which makes the recognition more difficult, because a single scene point can match multiple model points. Figure 9 shows an example of a virtual environment constructed using the recognition algorithms and of the robot used for manipulation tasks in this environment. The system has been successfully used in actual demonstration scenarios.

In addition to this application, this representation has been used for multi-view merging and alignment of terrain maps, and for building precise models of 3-D structures [19].

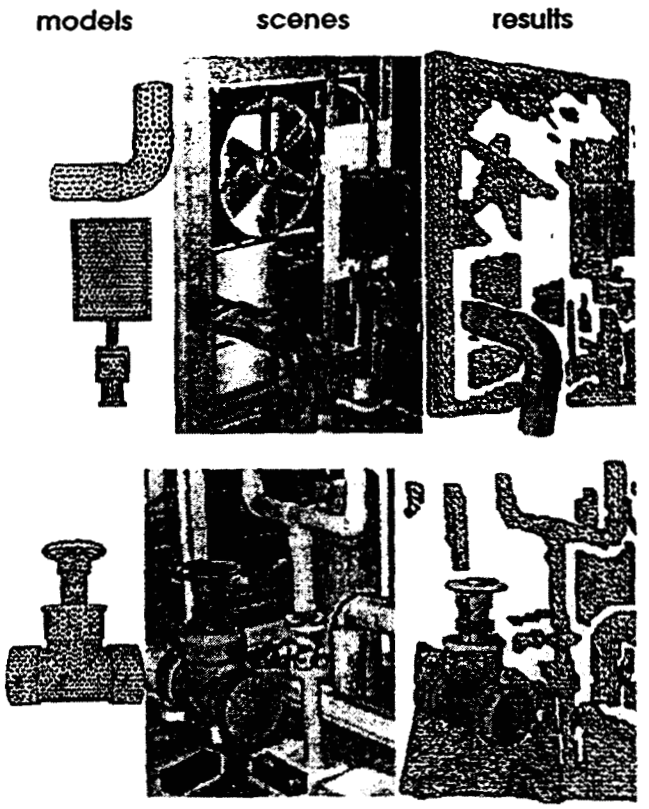

Figure 8: The recognition of shapes in complex scenes. These results demonstrate the recognition of complicated symmetric objects in 3-D scene data containing extreme clutter and occlusions. All of the scene data points are used in the recognition and no object/background segmentation is performed.

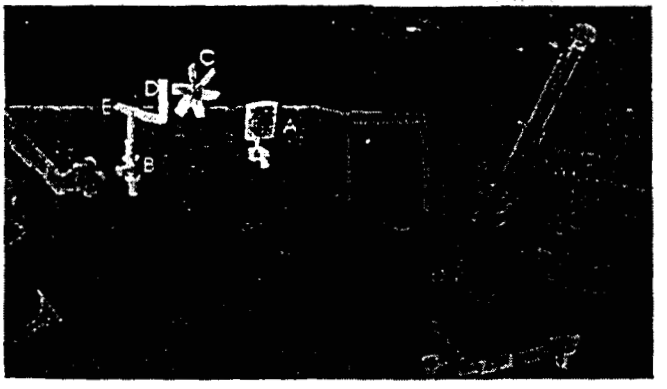

Figure 9: Robotic application of object recognition. (Left) Partial virtual environment built from images and model library; (Right) Robot used in the experiments.

3 Finding images in video sequences with application to landmark recognition

A more general recognition is the problem of recognizing shapes, or classes of shapes, directly from 2-D images. This problem is more challenging because of the substantial changes in appearance of any object due to changes in viewpoint and to changes in illumination. In addition, image variations due to clutter can also occur. Many approaches based on feature matching and other structural approaches have been proposed. All those approaches require an explicit geometric model of each shape to be recognized.

An alternative approach is to directly compare data sets, i.e., images or attribute images, without extracting a structural description first. In this type of approach, "models" are represented by collections of images which are supposed to capture the "typical" appearance of the objects. The information most relevant to recognition is extracted from the collection of raw images and used as the model for recognition. Although the situation is very different from that of the previous example, the philosophy is still the same: transforming the data into the appropriate representation in order to facilitate matching rather than extracting structural or semantic information.

Progress has been made recently in developing such approaches. For example, in object modeling [11], 2D or 3D models of objects are built for recognition applications. Extensions to generic object recognition are reported [13]. Other approaches use the images directly to extract a small set of characteristic images of the objects; these images are compared with observed views at recognition time, e.g, eigen-images techniques.

A similar problem, although in a different context, is encountered in image indexing, where the main problem is to store and organize images to facilitate their retrieval [1][26]. The emphasis in this case is on the kind of features used and the type of requests that can be made by the user.

As an example of a problem in which such an approach can be used, we consider here the problem of recognizing landmarks in sequences of images taken from a moving vehicle. Even with reasonable geometric constraints, such as the fact that the optical axis of the camera is generally at a small upward angle from the ground plane, this is a challenging problem for a number of reasons. First of all, the appearance of any given landmark varies substantially from one observation to the next. Changes in viewpoints, illumination, and external clutter all contribute to the variability of the observed landmarks. For those reasons, it is not possible to use many of the object recognition techniques based on strong geometric models.

The solution presented here in the context of landmark recognition uses image-based matching as a general approach without deriving higher-level semantic or geometric representation of the images and without explicit feature matching. In a training stage, the system is given a set of images in sequence. The aim of the training is to organize these images into groups based on similarity of image attributes. The basic image representation is based on distributions of different feature characteristics. A distance is defined to compare the distributions and to measure the similarity among imag-

es. This distance is then used to group the images. Each group is itself characterized by a set of attributes. When new images are given to the matching algorithm, it evaluates the distance between these images and the groups. The system determines to which group this image is the closest, and a set of thresholds is used to decide if the image belongs to this group.

3.1 Representing images

Two standard classes of attributes are used for describing the images: color and edge distributions. As was demonstrated in image retrieval work, color distribution can be a powerful attribute [34]. However, color information must be used with caution because large regions may have little color information and the effect of shadows may change the color distribution drastically. A technique similar to the one used in [28] for shadow reduction in outdoor imagery is used. In the remainder of this section, the term "color" refers to the single normalized red value computed at points of sufficeintly high saturation.

The color values are resampled by using a standard equal-size equalization. Specifically, the histogram of color values is divided into eight classes of roughly equal numbers of pixels. The color image is then coded on eight levels using these classes. This coarse quantization of color is necessary due to the potentially large color variations which make direct histogram comparison impossible. Figure 10 shows the color images, coded on eight levels, for two typical images from a training sequence. In the normalized images, only the pixels with sufficient saturation are shown; the discarded pixels are shown in white.

Because of the potentially large differences in viewpoint and illumination, color distribution cannot be directly compared in image space. Metrics have been proposed for comparing color histograms which can tolerate substantial variation in color distribution [29]. The approach chosen here uses a transition matrix rather than a direct histogram to represent the color distribution. Specifically, a color transition matrix *Cij* is created in which *Cij* is the number of pixels with value *i* and with at least one neighbor with value *j*. This transition matrix captures the global distribution, as in a histogram, and the spatial distribution of colors. The 8x8 transition matrix is used in the computation of the image distance metric described below.

Intensity edges constitute the second class of features. Figure 11 shows a typical edge image after linking and expansion of edge elements into segments. Several image attributes are computed using the image segments: a reduced, 120x160 binarized edge image E is used for image registration and distance comparison; the histogram, H_l, of segment lengths is computed and normalized by the total number of segments N_s. The histogram is taken over 20 buckets in the current implementation. N_s is also retained as an image attribute. Similarly, a histogram H_o of the orientation of the segments is computed over 18 buckets; pairs of intersecting segments are identified and a histogram of their relative orientations, H_i, is constructed over 18 buckets; pairs of parallel segments are also identified and histograms of their lengths and orientations are computed in H_{pl} and H_{po}, respectively. The histograms are normalized by the total number of parallel segments, N_p.

So far, we have described attributes computed over the entire image. In practice, global comparison of images does not perform well because of substantial variations in the image as the viewpoint changes. For example, features that are visible in one view may disappear in another view even though the feature distribution on the object of interest may be identical in the two images. In order to handle this problem, the images are divided into sub-images; and the attributes described above are computed within each sub-image. In the results presented below, sixteen sub-images were used to form a regular 4x4 subdivision of the original image.

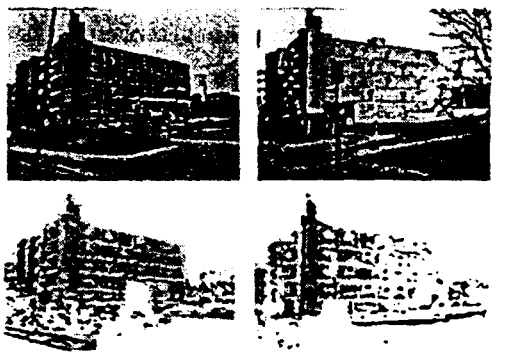

Figure 10: Color normalization; (top) Original images; (bottom) Normalized images.

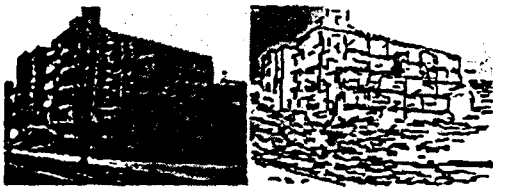

Figure 11: Segment distributions in a typical training image.

3.2 Comparing images

Because of the potentially large variation in viewpoint between images, the first step in comparing two images is to register them so that similar regions are in spatial correspondence. After registration, the attributes computed from the images can be directly compared and accumulated in a global image distance.

Given two images I_1 and I_2, the registration problem consists of finding a transformation H such that $H(I_1)$ is as close as possible to I_2. This registration problem can be made tractable through a few domain assumptions on the average camera position with respect to the objects, in particular, assuming that the object is far from the camera. Under those assumptions, the problem can be approximated by using an affine transformation H and by concentrating on the top half of the image, since the bottom part typically contains more of the ground plane and little information about the landmark of interest. A first estimate of H is computed from the contour at the top of the object of interest. This approach is similar to other registration algorithms based on skyline matching (see [7] for a review.) Starting with this initial estimate, H is iteratively refined by matching the rest of the image using an SSD criterion. This is similar to other registration algorithms using affine models, for example, for image mosaicing.

This algorithm converges as long as the initial estimate H_0 is close to the correct value. In particular, the algorithm performs well as long as large in-plane rotations do not contribute substantially to H. More general registration algorithms can be used in those situations.

Figure 12 shows an example of two images of a training sequence. As expected, the registration degrades from top to bottom in the image. As will be shown in the next sections, the registration is sufficient for correctly recognizing this building from a variety of viewpoints.

In the remainder of the discussion, whenever two images are compared, it is understood that this registration procedure has been applied and that the comparison is performed only on the overlapping part of the two images.

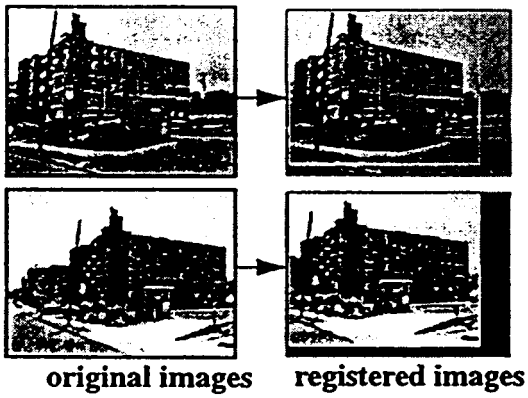

Figure 12: Approximate image registration; (top) reference image; (bottom) registered image.

Given a "model" image, I_M, and an observed image, I_O, a distance can be computed by comparing the image attributes. More precisely, I_O is first registered with I_M and the attributes are computed from the new, registered, image I_O'.

The global distance is defined as a sum of distances between the attributes of the sub-images of I_M and I_O' in the area in which they overlap. The distance between attributes is defined as follows. For the color transition matrix C, the distance is computed by computing the SSD of the entries of C from the two images. In computing this distance, it is natural to give more weight to regions in which there is more color variation rather than to uniform regions. This is implemented by giving more weight to the off-diagonal elements of C, which correspond to pixels with large variations of color, rather than to the elements close to the diagonal, which lie in uniform regions.

For single-values attributes, e.g., N_s, the distance is simply the squared-difference between the model and observed values. For histograms, e.g., H_l, the distance is the sum of squared difference between the elements of the histograms after a correction step. The correction step is used to compensate for mis-registration. Specifically, if the differences between the two histograms at x and $x+1$, $\Delta(x)$ and $\Delta(x+1)$ are of opposite sign and large magnitude, then, assuming $\Delta(x)$ is positive, then it is decreased by a small amount while $\Delta(x+1)$ is increased by the same amount. This procedure is repeated over the entire histogram until no further adjustment is needed. This procedure can be viewed as a coarse approximation of the earth-mover distance [29] in which a cost is assigned in moving an entry from one histogram to another entry and the set of such motions for which cost is minimal is computed. In practice, this approach effectively reduces the effect of mis-registration.

The distances between individual attributes are combined into a single distance $D(I_M^i, I_O^i)$ for each sub-image i by using a weighted sum. Finally the distances for all the sub-images are combined into a global distance, denoted by $D(I_M, I_O)$, which reflects the similarity of the images in appearance (color) and shape (edge distribution). In order to account for the particular geometry of our sequences, the weight of $D(I_M^i, I_O^i)$ decreases as i becomes closer to the bottom of the image.

3.3 Grouping images into models

The discussion above focused on comparing individual images. Because of the large variations in appearance, multiple images must be used for representing a given landmark. That is, groups of images that represent a single object of interest must be extracted from training sequences. In this section, we briefly describe the algorithm used for extracting a small number of discriminating groups of images from a training sequence, and how to use those groups as models for recognition. An overview of the grouping algorithms is given in the section since a more formal description of grouping algorithms was included in an earlier paper [35].

Grouping

Let us denote the training sequence by I_i, $i=1..N$. The mutual distance between images in the sequence can be computed as: $d_{ij} = D(I_i, I_j)$, where D is the image distance defined above. Iin particular, it is implicit in this definition that I_i and I_j are registered as part of the computation of D. A pictorial representation of the set of values d_{ij} is shown in Figure 13. In this representation, the d_{ij}'s are displayed as a two-dimensional image in which the dark pixels correspond to large values. The diagonal, i.e., the values d_{ii} is not shown since d_{ii} is always 0. Images for which the mutual distances are small are merged into a single group.

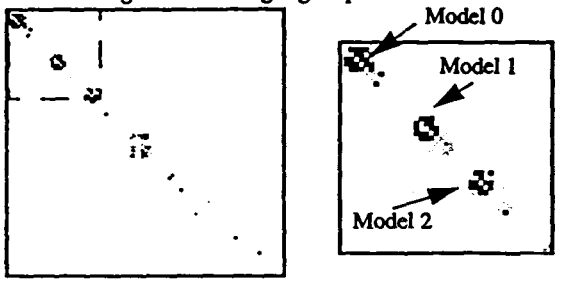

Figure 13: Distance matrix for a 145-images training sequence; darker points correspond to lower distances; the right image shows the distance matrix for the first 50 images.

Many groups can be found by using standard clustering approaches. For a recognition system to be useful, however, only a small number of groups is relevant. More precisely, we are interested in the groups that have enough information, i.e., a large number of images with sufficient variation, and that are discriminating with respect to the other images. The second criterion is important because it is often the case that many groups look alike in a typical urban sequence. This problem is addressed by comparing the initial groups with each other and discarding those with high similarity to other groups.

The right side of Figure 13 shows a magnified version of the distance graph in the neighborhood of the three main models extracted from the training sequence. Example images from each of the three models are shown in Figure 14.

Each of the groups extracted from the training sequence corresponds to a distinguishable landmark. Before being used for recognition, each group must be collapsed into a model suitable for comparison with test images. There are two aspects to this. First, a reference image must be created as the core image representation of the model. Second, image attributes from all the images in the group must be collapsed into a single set of attributes.

Given a group $\{I_i\}$, $i_{min} < i < i_{max}$, the first part is addressed by selecting a reference image I_o in the group -- usually the median image in the sequence. All the other images are registered to I_o using the approximate registration procedure described above, yielding new images I_i'. The second part is addressed by computing the attributes of each I_i'. For each attribute, the average value over all the images in the group is computed. In order to capture the variation of the attributes within the group, the variation of each attribute within the group is computed, also over all the images in the group.

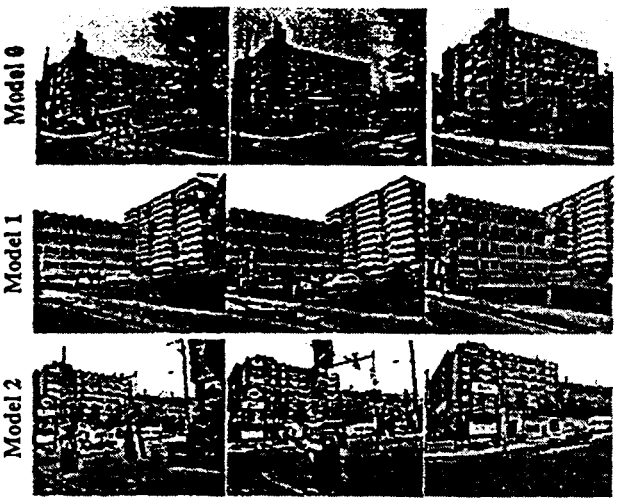

Figure 14: Three models extracted from training sequence Craig3; three example images are shown for each model.

3.4 Comparing images to models

Run-time recognition involves comparing a new image I with all the models in order to determine the best match. The first step in comparing I to a model M is to register I with I_M and to compute the attributes of the registered image I' over the overlapping region between I' and I_M. The attributes P^I_i from I' are compared with the attributes P_i of M by using a sum of distance weighted by the variation of the parameters in the model: $D(I, M) = \Sigma_i \alpha_i (P^I_i - \overline{P}_i)^t C_i^{-1} (P^I_i - \overline{P}_i)$. In this definition, the coefficients α_i are fixed and represent the relative importance of the different types of attributes. Those weights are computed using a principal component analysis technique on the set of attributes from the training sequence, as described in an earlier paper. The sum in $D(I, M)$ is evaluated over all the attributes and all the sub-images in common between I' and I_M. It implements a maximum likelihood estimate of the distance between image and models.

The definition of distance given above does not take into account the fact that there might be little overlap between I' and I_M. In fact, $D(I, M)$ could be small and could force a match if the registration is poor, i.e., the overlap is small.

This situation is addressed in two ways. First, I is not matched with M if the registration area $R(I,M)$ is below a threshold T_R. This threshold is computed automatically from the training set by computing the mean registration area between images of the same model and adding $3\sigma_R$, where σ_R is the variation of registration area over all the images of the model. Second, the actual distance used for matching is modified to: $D'(I, M) = D(I, M)/R(I,M)$. This weighting penalizes images with low overlap with the model.

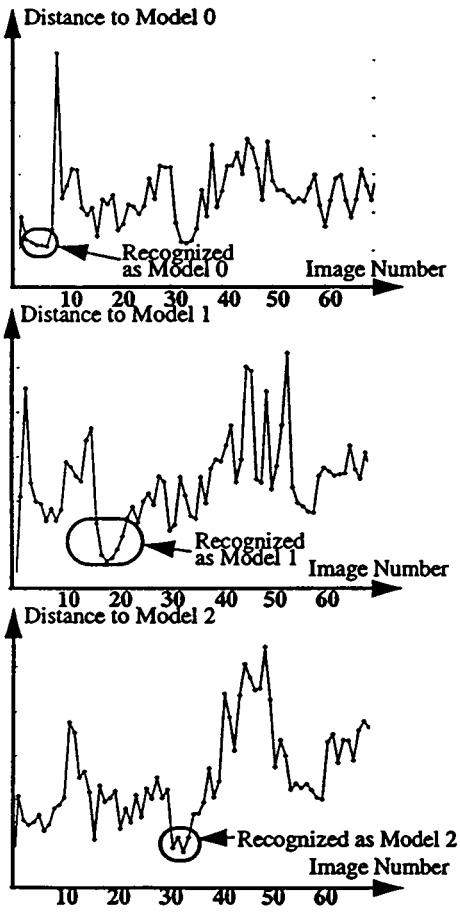

Figure 15: Graph of the distance to three models for a test sequence.

Figure 15 shows the distances between the images of a test sequence and three models from a training sequence. The same scale is used in all three graphs. The recognized models are indicated in the graphs. For reasons of space, results on all the test images cannot be included here. The graphs show that the images are recognized by a substantial margin.

Figure 16: Example images from three different test sequences recognized as Model 0 under different illumination and viewpoints.

The model that minimizes $D'(I, M)$ over all the models is taken as the best match to the image. Simply using the minimum would lead to a high rate of false positives in cases in which $D'(I, M)$ is low for all the models, i.e., the image matches poorly all the models, and in cases in which the distances to two models are of similar magnitude, i.e., the image is ambiguous. Both cases are addressed in standard ways. The first case is addressed by rejecting the image if $D'(I, M)$ is below a threshold T_D. The second case is addressed by rejecting the image if the margin between D' for the best model and for the next best model is lower than a threshold T_m. Both T_D and T_m are computed automatically from the training sequences [36]

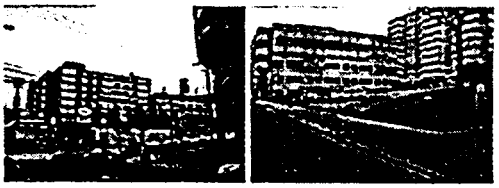

Figure 17: Two examples of images not recognized because of large variations in aspect.

Figure 18: Two examples of images not recognized because of extreme illumination conditions.

3.5 Discussion

Experiments were conducted with sequences of images taken in urban environments. In particular, the three models shown in Figure 14 were tested with ten different sequences containing about 300 images of the models and 300 images of other areas. The sequences were taken from different locations and under different illumination conditions from those of the training sequence. Under those conditions, the aggregate recognition rate is 78% on images of the models, while the rejection rate is 99.7% on non-model images. It is important to note that the algorithm is tuned to favor rejection of images in favor of mis-classification. As a result, no

model images are mis-classified and only one non-model image is mis-classified. The rejection rate for model images is 22.6%. The main reason for rejection is extreme variation in viewpoint (Figure 17) or in illumination conditions (Figure 18.) An application of this technique for automatic determination of the location of a vehicle based on a collection of views is described in [].

This example illustrates one possible approach to image-based recognition in which the images themselves are used for representing the models. Based on attributes computed from the image data, the images are grouped into classes corresponding to individual objects. The classes are represented by prototypical images that are compared to observed images at run-time. Although this approach does have several limitations, most importantly with respect to viewpoint variability, it does perform well within the specific domain constraints. The key is the proper engineering of the appropriate data representation leading to simple algorithms for comparison.

4 Recognition of space-time features and application to biology

The two examples above addressed static recognition problems in the sense that changes over time in the scene were not of particular interest. However, In many areas of computer vision, one is interested in recognizing patterns that are defined in both space an time. For example, in gesture recognition [8], one is interested in detecting transitions over time between different shape configurations. Similar situations occur in the area of expression recognition, and in many areas of medical image understanding. In all those examples, the entity to be recognized is a composed not only of a of a geometric shape in the traditional sense, of appearance parameters, such as brightness, but also of a prototypical history. Furthermore, the underlying structures in those problems are deformable shapes as opposed to the previous examples in which rigid shapes or projections of rigid shapes were used. In order to capture the importance of time, such entities will be called "events" in the remainder of this section.

Typically, a precise mathematical description of an event is difficult because of the complex, non-rigid nature of the shape component, and because of the introduction of the time dimension. Similarly, traditional feature matching may not apply because of the lack of well-defined features. The situation is complicated by the fact that, because of the time component, the algorithms may have to deal with massive amounts of data.

An alternative approach is to manipulate the data to organize it in a way that incorporates both its spatial and temporal aspects and that facilitatfse the comparison with event models. If properly designed, such a space-time representation would alleviate the problems mentioned above.

The remainder of this section describes such an approach in the context of a biological application. The goal of this work is to automatically describe the developmental history of an embryo from large sequences of microscope images. Although this is a problem in a specific application domain, it has all the ingredients described above. The entities to be recognized are events described both by changes in brightness and shape, and by patterns of motion. Results show that, not only can individual events be recognized, but collections of events representing the history of the development can be identified as well.

The section below provides the minimum biological background necessary for the understanding of the problem. The remainder of the discussion is a brief summary of the approach used to represent the data and to recognize the events.

4.1 Illustrative example: automatic analysis of embryonic development

As an example of the recognition of space-time events for biological applications, we concentrate on the problem of analyzing embryonic development from microscope imagery. Developmental biologists routinely make movies of the embryonic development of *Drosophila Melanogaster* (the fruit fly), a standard subject for research in embryo development [4]. The embryogenesis of *Drosophila* offers an interesting example of nonrigid deformation. The changes in the object are quite drastic, yet non-random and stereotypical (different embryos of the same species all deform in the same characteristic way). This research is about understanding these complex changes.

Since living tissues are both transparent and able to tolerate the transmission of light, optical microscopy (in particular, optical section microscopy) is commonly used to study them. An optical section microscope is essentially a normal light microscope with a narrow depth of field. As the focal place of the microscope is adjusted, different planes within the specimen come sharply into focus. By this means, a organism may be studied live, in cross-section.

Prior to study, fluorescent dye is commonly applied to highlight structures of interest. Subsequent to injection into an organism, a dye will become active (begin to fluoresce) by selectively binding to some tissue or reacting with some biochemical. When pumped with light, active dye will fluoresce and highlight the structure of interest. The injection of vgal dye into the intervitelline space (space between the vitelline containing membrane and the embryo surface) of the fruit fly embryo produces a clear negative image of the embryo surface, permitting the observation of the morphological changes that the surface undergoes in the course of the development of the embryo into a larva. During this develop-

ment, furrows, invaginations and ridges are observed to form, change in shape, and die. These surface changes have been used by developmental biologists to divide *Drosophila* embryogenesis into characteristic stages [4].

Figure 19 shows a typical sequence of images obtained of the *Drosophila* embryo, with intervitelline injection of vgal dye by using a 20X optical microscope. In these images, the vitelline membrane appears as an oval slightly deforming shape, with the embryo inside the membrane being also roughly oval, but possessing numerous convolutions and invaginations that change with time. Wherever there is an indentation of the embryo surface, there is some space between the embryo and the vitelline membrane, where bright vgal dye collects and is observed. So, in these images, the intervitelline space appears as a sort of bright belt of varying (or vanishing) brightness and thickness just inside the oval boundary of the vitelline membrane.

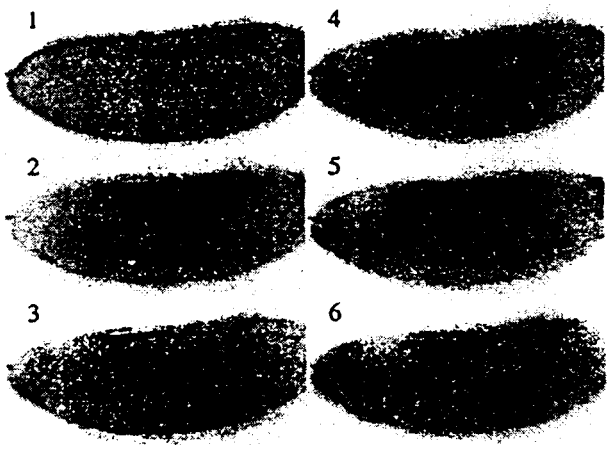

Figure 19: Typical sequence of frames from an intervitelline movie; the frames are equally spaced in time, and normalized by their maximum intensity. They span about 9 hours in real time.

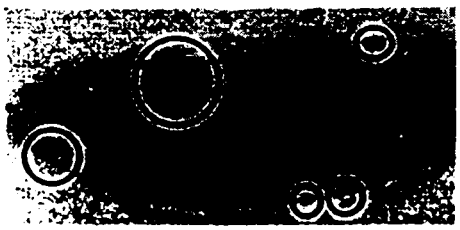

Figure 20: Typical vgal fluorescence image; dark regions, some circled, indicate presence of dye between the outer vitelline membrane and the inner embryo surface; circled areas indicate shapes of potential interest.

Figure 20 shows a typical image in which a few features of interest are marked. As indicated above, those features are places where the outer surface of the embryo is subject to substantial deformations. Features in individual images are not of particular interest to the biologist; more interesting is the evolution of the features over time. Specifically, from a computer vision perspective, we need to detect specific patterns of changes over both time and space. Instead of dealing exclusively with spatial shapes and intensity distributions, we are also observing them over time, i.e., we need to recognize space-time events.

4.2 Space-time representation

The massive amount of data contained in a typical movie of several hundred images must be reduced to include only the data that is needed for extracting the events of interest, and for combining the temporal and spatial location of the feature in a single representation. Since the events occur at the boundary of the embryo, the first step in reducing the data is to extract contours in each image. The contours are extracted using a snake computed from the gradient image. The snake is computed on the first image of the sequence and tracks the outer shape of the embryo over subsequent images.

The contours are resampled so that they can be parameterized by arc length in a consistent manner between different time steps. The number of samples on each contour is computed from the image resolution; 400 sample points are used in the examples presented here.

After detection and resampling, the contours detected at consecutive time steps are aligned so that they are all in registration. This is possible in this application because two features, the anterior and posterior tip of the embryo, can be identified reliably based on the shape of the embryo; they essentially occur at the points of highest curvature near the two extremal points of an ellipse fit to the embryo. The contours are aligned based on the positions of those two registration points. After alignment, the contours are cut at one of the registration points and unfolded as shown in Figure 21. The structure obtained after stacking the unfolded contours is a two-dimensional representation of the contour in space and time (Figure 21.) After the unfolding of the contours, the space-time structure is similar to the one proposed in [3] for computing shape from motion. That representation was introduced for the same reasons, to avoid explicit feature detection and tracking by using the appropriate data representation. Space-time representations are also used in other areas of medical applications [31].

Different types of data computed on the contour can be stored in this space-time map, each one corresponding to a different "facet" of the data representation (Figure 21.) First, the contour curvature is stored in the representation. Second, an indication of the brightness of the points along the contour is also stored. Because the absolute intensity of the original images along the contour is not meaningful, the brightness is defined as the response of a directional second derivative operator applied in the direction of the normal to

the contour at each point. The size of the operator is variable and is computed from the local distribution of intensity.

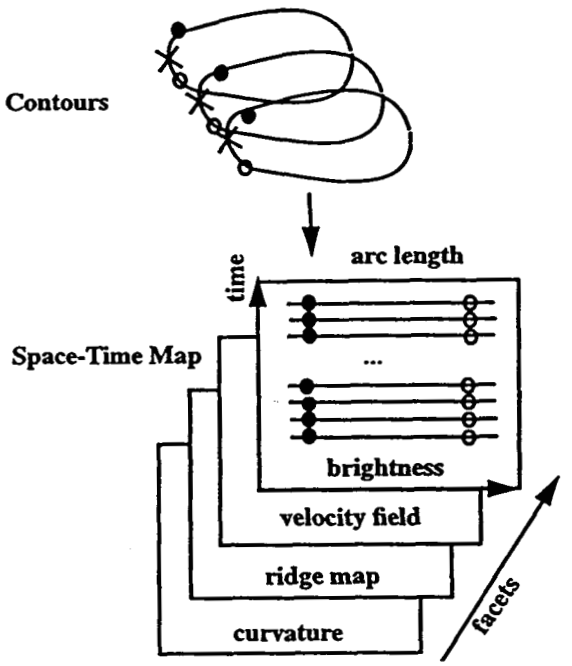

Figure 21: The representation of the image sequence is built by unfolding the contours after registration.

Another important facet of the representation is the velocity field on the contour. This velocity field approximates the flow of dye around the contour as it changes shape over time. Therefore, a high magnitude of the velocity field indicates a fast change in that part of the contour. The velocity flow is computed from the intensity distribution by correlating the intensity distribution on consecutive contours in order to find the optimal displacement vector. Because the contrast and the intensity distribution can vary substantially along the contour, a coarse-to-fine approach is used in which multiple windows with logarithmically spaced widths are used for computing the best velocity vector at each point.

A second facet of the data representation is a feature map. Features in the space-time map are ridges in the velocity field which are detected by a center-surround operator. Different shape of the operator will be used for extracting different types of features. Here again, variable window sizes are used for detecting features of different sizes [25]. Accurate localization of ridge endings can be achieved by using the ridge termination algorithm proposed in [30]. Ridge features are also tracked across time steps. Other features can be computed as facets such as ridges in the space-time surface formed by the contour by using techniques similar to [21]

After this processing, the basic data representation consists of several facets of the 2-D space-time map: intensity, contour curvature, velocity field, and ridge features.

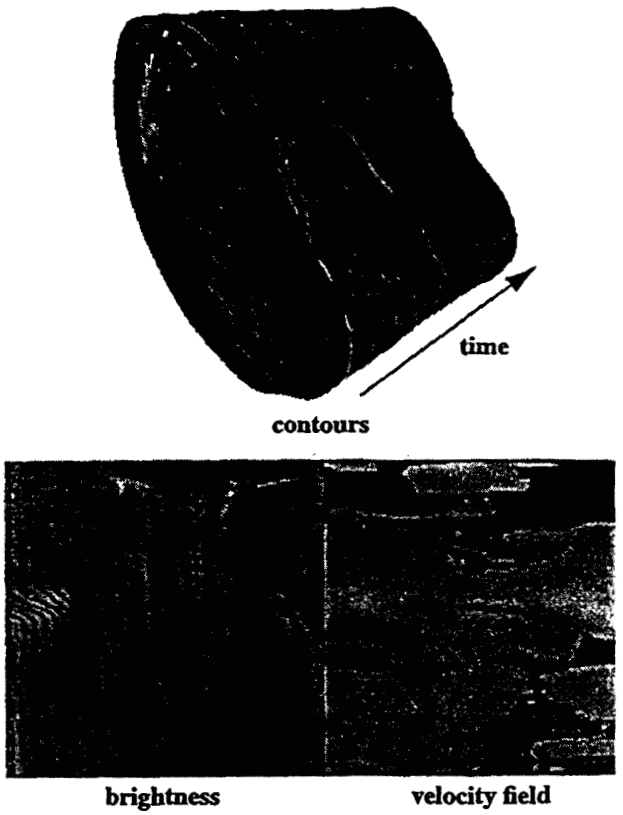

Figure 22: A set of contours extracted from a sequence and the facets of the corresponding space-time representation.

4.3 Event recognition

Each event is recognized in the space-time map using the appropriate operator on one or several facets of the data. Let us consider first one event of interest called "germ band extension." The germ band is a strip of tissue on the ventral side of the embryo that undergoes dramatic elongation and contraction during the course of embryogenesis. The effect of the elongation is to cause the tip of the germ band to extend from its initial position at the posterior tip, towards the anterior tip along the dorsal side of the embryo, and then retract to its original position during the contraction. Figure 23 shows the location and direction of motion of the germ band extension on a typical image. Candidate location of germ band extension events can be represented by a large connected region of high values in the velocity field. The velocity range in which this event can be detected is determined by the knowledge of the speed of actual germ band extension in prior experiments.

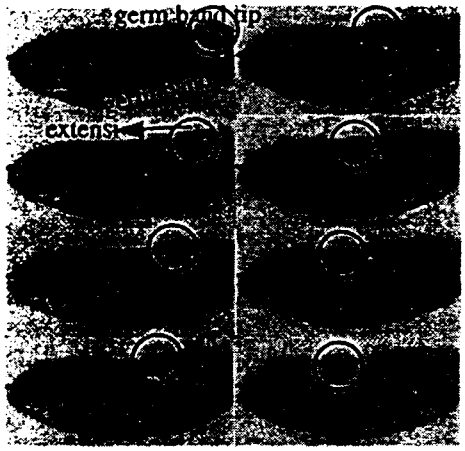

Figure 23: Typical shape and motion of the "germ retraction" feature in a sequence of images; a corresponding spatio-temporal event is constructed for recognition.

Figure 24 shows the result of recognizing this event in the image sequence partially shown in Figure 19. The area in which the event is detected is indicated by a rectangle. The velocity field facet of the space-time map is used in this example.

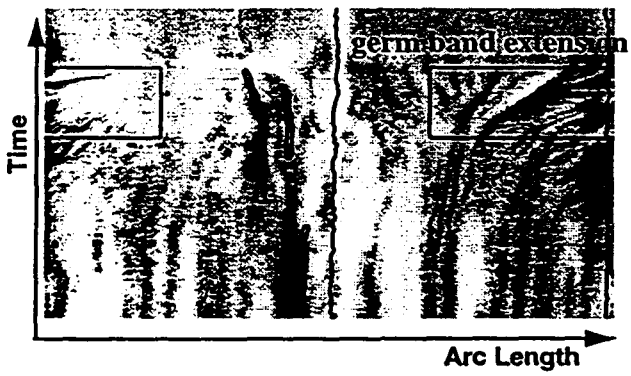

Figure 24: Example of recognition of the "germ band extension event" shown on a fragment of a complete image sequence; the rectangle shows the are in which the event is found, overlaid on the brightness map.

4.4 Reasoning about space-time events

A library of events can be recognized in a way similar to the example above. With each event is stored the following information: the facet(s) to be used for recognition, the filter used for enhancing the event, e.g., a directional filter on the velocity field in the case of the germ band extension, and the parameters used for building the filters and for detecting the feature based on the output of the filters. Those parameters are currently computed from known properties of the events, e.g., average velocity of the feature. Current work involves learning those parameters from training sequences.

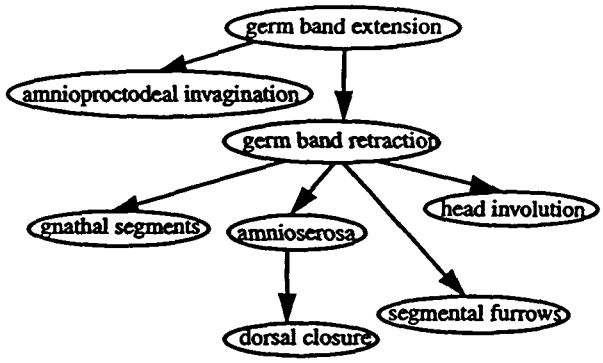

Figure 25: Partial view of the knowledge representation structure of the developmental model; each node is represented by a spatio-temporal event; transitions between nodes represent expected relative positions of the events.

Recognizing each event individually would inevitably lead to a significant number of false positives. Furthermore, the ultimate goal is to automatically characterize the entire developmental sequence of the embryo; such knowledge cannot be derived from recognizing individual events. For those reasons, the relationships between events need to be considered. This is achieved by using a knowledge base (Figure 25) which encodes the relationships of the events both in time and space. Each node of the tree is an event represented as described above. The transitions in the tree indicate relations between events such as the expected interval of time between events, the expected overlap between events, and the relative location in space of those events.

Conceptually, the recognition program walks through this tree, starting with the dominant event, i.e., the event that occurs in all the developmental sequences, and activates the recognition of its children in the tree. The recognition parameters for the children events are set based on the characteristics of the detected parent event. For example, the expected location of an event, which defines a window in the space-time map in which the event is searched for, is computed from the location of the parent and the relative position information provided by the transition between the events in the tree.

This approach, currently being implemented, will enable the entire history of the development to be explained rather than a collection of developmental events. Furthermore, by using known constraints on transitions between events, the number of false positives is drastically reduced.

Figure 25 illustrates the recognition of multiple events. Each event is indicated by a rectangle that surrounds the area of the space-time map in which the event is recognized.

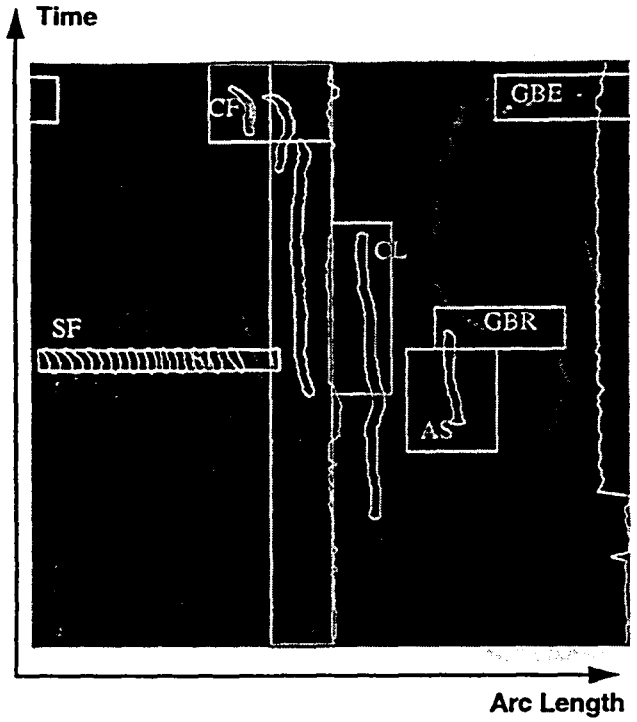

Figure 26: Recognition of multiple events; the space-time image is encoded based on the brightness map.

4.5 Discussion

This example shows how structures that are defined by shape, brightness, and motion can be recognized using the appropriate data representation. Such structures do not fit the traditional definition of "shape," but the general framework of selecting the appropriate data representation and the appropriate comparison operators still applies. Specifically, the techniques used in this example operate directly on the data, without using an intermediate structural representation. The first step is to extract the data relevant to the problem, in this case brightness values along contours; second, to arrange the data in a structure that can represent variation in both space and time, and to compute different facets of the data representation that facilitate recognition.

Preliminary results on automatic recovery of developmental history show that it is possible to use this type of data-level recognition approach in conjunction with symbolic knowledge on the relationships between events. In particular, work is underway to detect mutations by comparing the recovered developmental history with the expected history. For example, missing events or events occurring in unexpected configurations indicate potential mutations.

5 Conclusion

The examples presented in this paper show that it is possible to use low-level intermediate data representations, even for complex recognition tasks. This approach has led to substantial improvement in performance as demonstrated by its successful application to real-world problem. For example, the 3-D recognition algorithm was used in more complex and cluttered environments than has ever been shown before. The space-time representation for event detection is being used by biologists as part of an automated associate for assisting in the interpretation of large data sets.

A key feature of the data-driven approaches is the simplicity of the representations. In all cases, the intermediate representations rely on simple structures for which an arsenal of tools is available. This simplicity of the representations contributes both to the robustness of the resulting systems and to their generality. The price to pay is that a higher volume of data must be manipulated for recognition. However, as noted earlier, this increase in computation is more than offset by savings in the computation of complex, high-level representations inherent to traditional approaches. Although, the use of intermediate data representations is by no means a universal solution, those examples suggest that it can make complex recognition problems vastly more tractable in many cases.

Acknowledgments

The work described in this paper was conducted with Andrew Johnson (Section 2), Yutaka Takeuchi and Patrick Gros (Section 3), and Prem Janrdhan (Section 4.)

References

[1] Bach et al. The Virage image search engine:An open framework for image management. *SPIE Proc. Image Storage and Retrieval.* 1996.

[2] R. Bergevin, D. Laurendeau and D. Poussart. Registering range views of multipart objects. *Computer and Vision Image Understanding*, 61(1):1-16, 1995.

[3] R.C. Bolles, H.H. Baker and D.H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1), 1987, pp. 7-55.

[4] J.A. Campos-Ortega and V. Hartenstein. *The Embryonic development of Drosophila Melanogaster.*

[5] S. Carlsson. Combinatorial geometry for shape indexing. *Proc. Workshop on Object Representation for Comp.Vision.* Cambridge. 1996.

[6] C. Chua and R. Jarvis. 3-D free-form surface registration and object recognition. *Int'l J. Computer Vision*, 17(1):77-99, 1996.

[7] F. Cozman. Position estimation from outdoor visual landmarks. *Proc. WACV'96.* 1996.

[8] T. Darrell and A. Pentland. Space-time gestures. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 1993*, pp. 335-340.

[9] H. Delingette, M. Hebert and K. Ikeuchi, Shape representation and image segmentation using deformable

Surfaces, *Image and Vision Computing*, 10(3), April 1992, pp. 132-144.

[10] J. Devore. *Probability and Statistics for Engineering and Sciences*. Brooks/Cole, Belmont, CA, 1987.

[11] P. Gros, O. Bournez and E. Boyer. Using local planar geometric invariants to match and model images of line segments. *Int. Journal of Computer Vision and Image Understanding*. 1997.

[12] A. Guéziec and N. Ayache. Smoothing and matching of 3-D space curves. *International Journal of Computer Vision*, 12(1):79-104, 1994.

[13] R. Horaud, T. Skordas and F. Veillon. Finding geometric and relational structures in an image *Proc. of the 1st ECCV*. Antibes, France pages 374--384. April 1990

[14] K. Ikeuchi, T. Shakunaga, M. Wheeler and T. Yamazaki. Invariant histograms and deformable template matching for SAR target recognition. *Proc. Computer Vision and Pattern Recognition (CVPR 1996)*, pp. 100-105, 1996.

[15] A.K. Jain. *Fundamentals of Digital Image Processing*, Section 8.10, pp. 304-307.

[16] A. Johnson and M. Hebert. Recognizing objects by matching oriented points. *CMU Robotics Institute TR, CMU-RI-TR-96-4*, May 1996.

[17] A. Johnson and M. Hebert. Recognizing objects by matching oriented points. *Proc. Computer Vision and Pattern Recognition*. 1997.

[18] A. Johnson and M. Hebert. Control of mesh resolution for 3-D object recognition. *CMU Robotics Institute TR, CMU-RI-TR-96-20*, December 1996.

[19] A. Johnson and M. Hebert. Surface registration by matching oriented points. *Proc. Int'l Conf. Recent Advances in 3-D Digital Imaging and Modeling (3DIM)*. 1997.

[20] A. Johnson, P. Leger, R. Hoffman, M. Hebert, J. Osborn. 3-D object modeling and recognition for telerobotic manipulation. *Proc. Intelligent Robots and Systems 1995 (IROS '95)*, pp. 103-110, August 1995.

[21] J.J. Koenderink and A.J. Van Doorn. Local features of smooth shapes: ridges and courses. *Proceedings, SPIE Conference on Geometric Methods in Computer Vision II*, July 1993, pp. 2-13.

[22] Y. Lamdan and H. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. *Proc. Second Int'l Conf. Computer Vision (ICCV '88)*, pp. 238-249, 1988.

[23] B.Lamiroy and P.Gros. Rapid object indexing and recognition using enhanced geometric hashing. *Proc. of the 5th ECCV*, Cambridge, England, pages 59--70, vol. 1, April 1996.

[24] L-J. Lin and J.S. Judd. A robust landmark-based system for vehicle location using low-bandwidth vision. Siemens Corporate Research Inc. Technical Report SCR-95-TR-535, 1995.

[25] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition 1996*, pp. 465-470.

[26] R.W. Picard. A society of models for video and image libraries. *IBM Systems Journal*, 35(3-4):292-312. 1996.

[27] F. Pipitone and W. Adams. Tripod operators for recognizing objects in range images; rapid rejection of library objects. *1992 IEEE Robotics and Automation (R&A 1992)*, pp. 1596-1601, 1992.

[28] D.A. Pomerleau. Neural network-based vision processing for autonomous robot guidance. *Proc. Appl. of Neural Networks II*. 1991.

[29] Y. Rubner, L. Guibas, C. Tomasi. The Earth mover's distance, multi-dimensional scaling, and color-based image retrieval. *Proc. IU Workshop*. 1997.

[30] P.T. Sander and S.W. Zucker. Inferring surface trace and differential structure from 3D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9), September 1990, pp. 833-854.

[31] Y. Satol, J. Chen, S. Yamamoto, S. Tamura, N. Harada, T. Shiga, S. Harino and Y. Oshima. Measuring microcirculation using spatiotemporal image analysis. *Proceedings, First International Conference, Computer Vision, Virtual Reality and Robotics in Medicine, 1995*, pp. 302-8.

[32] C. Schmid and R. Mohr. Combining greyvalue Invariants with local constraints for object recognition. *Proc. CVPR*. San Francisco, California, USA. pages 872--877, June 1996.

[33] F. Stein and G. Medioni. Structural indexing: efficient 3-D object recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2): 125-145, 1992.

[34] M.J. Swain, D.H. Ballard. Color indexing. Int. J. of Comp.Vision, 7(1):11-32.1991.

[35] Y. Takeuchi, P. Gros, M. Hebert, K. Ikeuchi. Visual learning for landmark recognition. P*roc. Image Understanding Workshop*. New Orleans. 1997.

[36] Y. Takeuchi, P. Gros, M. Hebert. Finding images of landmarks in video sequences. *Tech. Report CMU Robotics Institute*. November 1997.

[37] J. Thirion. New feature points based on geometric invariants for 3D image registration. *Int'l J. Computer Vision*, 18(2):121-137, 1996.

[38] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int'l J. Computer Vision*, 13(2):119-152, 1994.