PHYSICALLY-BASED ACTIVE SHAPE MODELS: INITIALIZATION AND OPTIMIZATION

A. GARRIDO* and N. PEREZ DE LA BLANCA

Departamento de Ciencias de la Computación e I.A., ETS Ingeniería Informática, Universidad de Granada, 18071 Granada, Spain

(Received 4 October 1997)

Abstract—In this paper we describe a new approach for 2-D object segmentation using an automatic method applied on images with problems as partial information, overlapping objects, many objects in a single scene, severe noise conditions and locating objects with a very high degree of deformation. We use a physically-based shape model to obtain a deformable template, which is defined on a canonical orthogonal coordinate system. The proposed methodology works starting from the output of an edge detector, which is processed to automatically obtain an approximation of the shape. The final estimation of the shapes is obtained fitting a deformable template model, which is defined on a learned surface of deformation. Results from biological images are presented.

Deformable template Modal analysis Hough transform Physically-based modeling
Finite element method Active shape model

I. INTRODUCTION

One of the most interesting problems in image analysis is automatic detection, segmentation and extraction of biological objects. There are many cases in which it is impossible to improve the image-capturing process, and in consequence we obtain images with problems such as non-uniform illumination or noise. Moreover, biological objects present a very difficult problem because of how the objects appear in the image is out of our control. We then find images in which objects are overlapping or only partial information is available. To solve the problem of recognizing and locating biological objects (whose appearance can vary), a review of existing methods suggests the use of flexible templates. They can be divided into two classes:

- Methods with local shape constraints.
- Methods with global shape constraints.

In methods with local constraints, the template can be deformed to match salient image features. However, they are not possible to preserve global structure when the objects can present high variability (these methods sacrifice model specificity). In methods with global constraints, the template can be deformed to accommodate global deformations. However, when the objects present local deformations (due to natural variability or noise) the problem becomes more difficult and, in some applications, we have to assume low variability or to make a very specific template model.

We address the problem of locating biological objects, which are able to present natural deformations, in the presence of noise, clutter and occlusion. Without a contour model, the contour extraction in the presence of noise, clutter and occlusion is an ill-posed problem. Therefore, in this paper we propose a supervised method based on a deformable template model to detect and extract shapes. The object model that we propose is defined in terms of a closed polygonal approximation of the contour and a coordinate system, learned from a set of samples, to handle deformations.

The segmentation method that we present is composed by 3 steps:

1. A stage to extract external information from the image. We have to select the type of information to be extracted and to design an appropriate algorithm.
2. A learning stage. This stage starts from a set of samples. The goal is to obtain an object model from it. The object model that we propose is determined by a polygonal approximation of the object boundary together with a set of deformations. To do this, it is necessary firstly to align the shapes and secondly to process the obtained deformations.
3. A module which joins both external and internal information to determine the object position. This module is divided into two parts: initialization and optimization. In both, we have to use the object
model together with the external information in order to segment the image.

In this paper, we study the main components of the proposed methodology and present the results of segmentation on biomedical images that demonstrate the merits of the algorithms. In Section 2, we propose the method to build external information from the image. In Section 3, we discuss the method developed to deal with deformations and to obtain an object model (internal information). In Sections 4 and 5, we give the algorithms to initialize and optimize the position of the deformable template, respectively. Finally, in Sections 6 and 7, we present experimental results from some examples, discussion and conclusions.

2. INFORMATION FROM THE IMAGE

To determine what is the external information, there are two questions to solve: (1) What is the type of information to be extracted? (2) What is the algorithm to be used? The use of different cues has been suggested in the literature (colours, textures, etc.) but, in general, we can classify them into two categories:

- **Edge-based information.** This uses the gradient to fit the contour of the object. Traditionally, edge-based segmentation has been divided into two independent stages: edge detection and edge linking (supervised or not). Using a deformable template, we can evaluate the goodness of a solution from the gradient vectors determined by the contour [see references (1, 2, 9, 10)]. Then, postprocessing the gradient image (thresholding and linking) to select edge points is not necessary. However, if we consider the output from an edge detector, we can make information to allow recovering another one.*

- **Region-based information.** This uses the homogeneity property of the regions. By measuring the homogeneity level (colours, textures, etc.) inside the object, we can decide whether it is a good candidate to be the solution. This information has been used in several works to define the external energy function [see references (5, 11)].

In general, selecting a region-based or an edge-based strategy depends on the problem to be solved. Anyway, a more powerful solution may be built from both strategies, that is, mixing both types of information in a single energy function. We will mainly use edges to locate the object on the image, although it is interesting to show that different sources (several edge maps) may be used jointly.³

To use edge information, we have to preprocess the output from an edge detector according to our algorithm. From a semantic standpoint, we can distinguish several levels of information. A possible division follows:

- **Pixel.** Because it is the most basic information, it is necessary to use a stronger a priori information (for example, the Hough transform¹²,¹³) to achieve good segmentation.

- **Pixel and its neighbourhood.** The use of higher level of information implies better results from the algorithms. A way to improve the edge map is to use methods which work at this level (like relaxation methods¹⁴).

- **Chains.** This is the highest level including a wide set of possibilities. It could be divided into, for example:

  - **Straight-line segment.** Because of the noise and the discretization, we have to define a criterion which can be used to decide whether consecutive points lie on a straight-line segment or not.

  - **Multiline.** In this case, it is interesting to emphasize that the dominant points information [see references (15, 16)] is related to multiline information (establishing a relation between a dominant point and the link point connecting two segments).

  - **Polygon** (it defines, approximately, the solution).

Of course, algorithms can be implemented [see reference (9)], which generate new information as required, from the pixel to a solution. We have to select an appropriate level of information as the input to the proposed algorithm. We might select the most basic information, however, because of the noise in the image, pixel information is very unstable, and it is preferable to use a higher level to avoid the problems that can otherwise result. In this work, we have selected the straight segments computed from the edges as the input information (as we will see further on) because they are stable enough, easy to obtain, and they are consistent with the proposed object model. Anyway, the method may be generalized to other possibilities (using other level or using several levels, that is several sources as in Yuille, Hallinan and Cohen¹⁵).

Since we use edges to localize the object, we have to select an edge detector. Once the edges have been obtained, we have to postprocess them. We have selected the Canny filter¹⁶ because:

- we are working on noisy images. Therefore, smoothing is essential to get good results.

- Gaussian filter is optimal in terms of smoothing and localization in both the spatial and frequency domains.

- If we use a single threshold, the noise usually causes the breaking up of the edge contour. This problem may be eliminated by thresholding with hysteresis.

*In fact, to achieve better results, we can implement a first stage to obtain good inputs for our algorithms.

³In this sense, we are working on using several sources derived from cells images where different parts of the contour are detected at different scales. If we obtain a set of relevant scales, we are able to use them to find the cells by merging the diverse sources of information jointly.
Before starting the localization process, the edges have to be detected. This postprocessing stage consists of:

- Edge chain extraction (this stage consists of analysing multiple points and the chain orientation).
- To determine where straight segments are located.

Multiple points are formed by different chains crossing each other (up to three chains joined at the same point). Hence, when we visit these chains to search for their extremities, when a multiple point is reached, it is not clear how to proceed. Therefore, first we delete the multiple points. To delete these points, we develop an algorithm that basically studies the length of the three chains, cutting the branch which is not long enough (probably caused by noise). If the chains are long, we delete the point and its neighborhood ending up with different chains at this stage of the process. Once these points are suppressed, we can code the chains but all chains must be oriented in a run direction according to the described model. In our experiments, we use the counterclockwise direction. The correct run direction of the chains is determined from the sign of the vectorial product between the tangent line to the chain and the gradient vector at this point. Both vectors form a right angle (subject to precision or noise problems). The run direction can be calculated without the exact estimation of the tangent vector.

Secondly, once we have obtained the chains, we have to decide when a chain corresponds to a straight segment. We have selected the simplest method. Accordingly, the maximum distance between each of the points along the chain and the straight segment must be less than a given threshold. Of course, this threshold depends on both, the noise of the image and the object that is being looked for. Experimental results with several distinct objects show that the threshold value may be selected easily in a robust way.

3. DEFORMATION MODEL

Let us consider a shape (with null deformation) described by a vector

\[ X = (x_1, \ldots, x_n, y_1, \ldots, y_n)^T. \]  

where \((x_i, y_i)\) are the nodal points along the contour (i.e., may be thought of as a polygonal approximation of the shape). Then, a deformed shape may be calculated from

\[ X_\theta = X + \Phi \theta, \]  

where \(\theta\) is a \(2n \times 1\) vector and \(\Phi\) is a \(2n \times 2n\) matrix, that is, composed by \(2n\) vectors

\[ \phi_1 \cdots \phi_{2n}, \]  

each of them corresponding to an axis of deformation. So, the problem is to determine a good set of axes.

As we mentioned earlier, we propose a supervised method to detect and extract shapes based on a deformable template model. Thus, we need to determine the way in which the template will become deformed. The desired characteristics of the system to handle deformations are:

- The method should be general, i.e., it should not be related to a specific shape [see, for example, reference (6)].
- It should provide a global-to-local ordering of shape deformation. If we want to use the main deformations, they have to be classified according to their importance.
- It should facilitate learning, i.e., the design of algorithms to learn the object model.

We use the deformation system proposed by Pentland and Sclaroff based on modal analysis [references (8, 18)]. In modal analysis, the standard finite element method (FEM) computations are simplified by posing the dynamic equations in terms of the equations' eigenvalues. The eigenmodes (known as the object's free vibration or deformation modes) of this physical shape model are used to obtain a canonical, frequency-ordered orthogonal coordinate system. Moreover, these generalized symmetries allow us to separate the rigid body modes (displacements and rotations) from the non-rigid modes and they may be used to align a set of samples in order to process its deformations.

3.1. Finite element method

The mathematical formulation of this model is based on the standard engineering technique for simulating the dynamic behaviour of an object [reference (19)], i.e., on the FEM. To use the modal analysis, the object is represented from a set of nodal points as the finite element nodes. Then, stiffness and mass matrices are built and shape functions are used to relate the displacement (or other properties) of a single point to the relative displacements of all the other nodes of an object. These interpolation functions allow modelling continuous material properties.

The polynomial shape function for each element may be written in vector form as

\[ u(x, y) = H(x, y)U, \]  

where \((x, y)\) is the point where we want to know the displacement, \(H\) is the interpolation matrix and \(U\) is the vector composed by the displacements of each element node.

Once we have the polynomial shape functions, they can be used to calculate the strains (the ratio of the change in length) and the stresses corresponding to them as

\[ \varepsilon(x, y) = B(x, y)U, \]  

\[ \tau = C \varepsilon + \tau', \]
where $\varepsilon$ is the strain, $\tau$ is the stress, $B$ is the strain displacement matrix (the rows of $B$ are obtained by appropriately differentiating and combining rows of the element interpolation matrix $H$). Vector $t'$ denotes given initial stresses and $C$ is the stress-strain material matrix. This may be written for isotropic materials and for a plane strain element as

$$C = \frac{E(1 - \nu)}{(1 + \nu)(1 - 2\nu)} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 - \nu & 0 \\ 0 & 0 & \frac{1 - 2\nu}{2(1 - \nu)} \end{pmatrix},$$

where $E$ is the Young's modulus and $\nu$ the Poisson's ratio.

Then, to establish the dynamic equilibrium equation

$$M\ddot{U} + D\dot{U} + KU = R,$$

where $R$ is the load vector, we have to calculate the element mass, damping, and stiffness matrices ($M$, $D$ and $K$ respectively) as

$$M = \int_V \rho H^T H \, dV,$$

$$K = \int_V B^T C B \, dV,$$

$$D = \int_V \kappa H^T H \, dV.$$ 

Note that $D$ is very difficult to calculate, and it is necessary to assume Rayleigh damping [see reference (19)]

$$D = \alpha M + \beta K,$$

where $\alpha$ and $\beta$ are constants, that is, $D$ is a linear combination of the mass and stiffness matrices.

### 3.2. Modal analysis

An FEM technique to improve the method that solves the system equations is to change the basis to modal generalized displacements, i.e. to transform the equilibrium equations into a more effective form for direct integration. This transformation may be written as

$$U(t) = PX(t),$$

where $P$ is a square orthogonal transformation matrix and $X(t)$ is a vector of generalized displacements. Then, we can change the basis as follows:

$$P^T M \ddot{X} + P^T D \dot{X} + P^T K X = P^T R,$$

$$\ddot{X} + \ddot{X} + \xi X = \ddot{R},$$

where

$$\ddot{M} = P^T M P, \quad \ddot{D} = P^T D P, \quad \ddot{K} = P^T K P, \quad \ddot{R} = P^T R.$$ 

The optimal transformation matrix $P$ is derived from the free vibration modes of the equilibrium equation, i.e. we assume the solution has the form

$$U = \phi \sin \omega(t - t_0)$$

and solve the equation

$$\ddot{M} \ddot{U} + \ddot{K} U = 0.$$ 

Then, an eigenvalue problem can be derived:

$$K\phi = \omega^2 M\phi,$$

which will determine an optimal transformation basis set. There are $2n$ eigensolutions

$$(w_1^1, \phi_1^1), (w_2^1, \phi_2^1) \cdots (w_{2n}^1, \phi_{2n}^1),$$

where all the eigenvectors are $M$-orthonormalized, i.e.

$$\phi_i^T M \phi_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

and

$$0 \leq w_i^2 \leq \cdots \leq w_{2n}^2.$$ 

The eigenvector $\phi_i$ is called the $i$th mode shape vector and $w_i$ is the corresponding frequency of vibration. These vectors establish a very good basis to represent the deformations, separating the rigid body modes (displacements and rotations) from the non-rigid modes and providing a global-to-local ordering of shape deformations.

Finally, in Appendix A the expressions proposed in Sclaroff and Pentland and used in this work to calculate the matrices, and to solve the eigenvalue problem, are presented. In Fig. 1 the vibrations modes of a square are shown.

### 3.3. Principal components analysis

Once the deformations have been expressed using the free vibration modes, we have to select the principal components in order to obtain a coordinate system as simple as possible. Directly, we are able to select the main components from this basis, i.e., we can take out the last modes (the highest frequency modes), which correspond to local shape variations (the ones most sensitive to noise) and so, we avoid the excessive number of dimensions caused by an oversampled contour. In spite of this, an easier way to select the principal components may be used, i.e. via the Karhunen–Loève transform. The main advantages from this transform are:

- We can determine the number of high-frequency modes to eliminate.
- If we use a training set, we are able to order the low-frequency modes and even to delete some modes although these ones correspond to low frequencies.
- We will find the correlations between different axes. Then a new low-dimensional coordinate system to integrate all the main deformations may be built.
Let us suppose we have available a set of samples \( E_i \) \( (1 \leq i \leq m) \) which correspond to \( m \) different possibilities of deformation of a template \( T \) (obviously, of null deformation). The process to select principal components from them is as follows:

(1) Align each shape with the template \( T \), processing both the rigid transformation and non-rigid deformation to obtain \( m \) aligned samples \( X_i \) \( (1 \leq i \leq m) \). A very efficient method based on modal analysis is described in Sclaroff and Pentland.\(^8\)

(2) Once the samples are aligned, each of them with \( n \) modal points [see equation (11)], we are able to calculate the \( m \) vectors \( U_i \) as

\[
U_i = X_i - T
\]

and considering that

\[
U_i = \Phi \bar{U}_i, \quad \Phi'\Phi = I,
\]

where \( \bar{U}_i \) correspond to the modal generalized displacements, the transformation to modal coordinates is

\[
\bar{U}_i = \Phi^{-1} U_i = \Phi' M U_i .
\]

Each sample may then be written as

\[
X_i = \Phi \bar{U}_i + T .
\]

where \( \bar{U}_i \), \( \bar{U}_i \), and \( \bar{U}_i \) are null because of the previous alignment.

(3) From the \( m \) samples we are able to calculate the mean and covariance matrix as:

\[
m_eff = \frac{1}{m} \sum_{i=1}^{m} \bar{V}_i,
\]

\[
C_T = \frac{1}{m} \sum_{i=1}^{m} \bar{V}_i \bar{V}_i' - m_eff m_eff'.
\]

where \( \bar{V}_i = (\bar{U}_{i,1}, \ldots, \bar{U}_{i,n})' \). Then we can obtain \( 2n - 3 \) eigenvalues \( \lambda_i \leq \lambda_{i+1} \) and \( 2n - 3 \) eigenvectors from

\[
C_T \bar{v}_i = \lambda_i \bar{v}_i
\]

and using the matrix

\[
A = (c_1, c_2, \ldots, c_n)' ,
\]

the Karhunen–Loeve transform is given by

\[
V_i = A (\bar{V}_i - m_eff).
\]

(4) Selecting the largest eigenvalues, we select the most significant modes of variation in the variables used to derive the covariance matrix, i.e. in the modal axis. Obviously, each \( \bar{V}_i \) may be obtained using

\[
\bar{V}_i = A' V_i + m_eff
\]

and thus, each sample \( X_i \) may be written as

\[
X_i = \Phi \cdot ((0,0,0),(A' V_i + m_eff)' + T .
\]

4. Initialization

In the literature we can find many works using deformable models but not an automatic initialization stage.\(^4\)\(^,\)\(^10\) As our goal is to obtain a completely automatic segmentation process to extract biological objects, we cannot use solutions taking into account only small deformations\(^2\)\(^,\)\(^6\) or solutions to be applied in a specific application.\(^10\) In this section we address the problem of initializing deformable templates of known objects in images which may contain many objects in a single scene, overlapping objects, missing information and noise.
Consider the curve
\[ C(t) = (f_x(t), f_y(t)), \] (32)
where \(0 \leq t \leq L\). Let us suppose that a set of deformation axes
\[ D = \{e_x, e_y, c_x, c_y, \ldots, c_n\}, \] (33)
is given where \((e_x, e_y, c_x)\) are the axes of translation and rotation. The next axes correspond to non-rigid deformations. Then the set of possible location of the curve \(C\) is determined by \((3 + n)\)-tuples. Thus, our goal is to find a set of parameters defining the location of the objects, that is, to obtain a tuple that determines the object location
\[ (\Delta x, \Delta y, \Delta r, \Delta d_1, \ldots, \Delta d_n). \]

A powerful technique to solve this problem is to use the Hough transform.\(^{12,21}\) The Hough transform uses a parameter space of dimension \(3 + n\), sampling each axis to build a discrete parameter space \(E\) where the evidential content of the image will be accumulated. The location of an instance of the hypothesized object will be identified by a local maximum in this accumulator. However, it is impossible to apply this method directly because:

- The edges may be displaced due to a smoothing operation, noise or local deformations. The calculated position to update the accumulator will be not correct.
- The algorithm requires a lot of time and space to perform the localization. Excessive memory is needed to store the accumulator because the parameter space size is too large. In addition, each detected instance in the image will take a lot of time (we will have to update a subset of positions in the accumulator).

Thus, we have to reformulate the generalized Hough transform (GHT). The solution that we propose is as follows:

- The results from GHT may be erroneous because the input information (for example, point along the contour from an edge detector) is very noisy. The adopted solution is to use more stable information available at a higher semantic level (points, corners, segments, regions, etc.). In this work straight segments have been used.
- If we want to use a method like GHT, we have to decrease the parameter space size. We propose to compromise on the accuracy of the solution (i.e., accurate location) in order to be able to apply the reformulated GHT.

4.1. Reformulating the Hough transform

Let us consider a contour describing an object and a 2-D parameter space (only translation). When a piece of evidence is detected, the Hough transform updates every possible location of the shape according to the position of the evidence in the image. If the evidence is displaced, we cannot know the exact location. Therefore, given an evidence an uncertainty region may be inferred.

In Fig. 2A we can distinguish the point \(C\) (reference point of the Hough transform) and the point \(p\) (an evidence). If the point \(p\) is displaced to another position, \(p'\), the Hough transform, without considering any deformations, will localize point \(C'\) as the reference point. Considering every possible displacement of the point \(p\), we are able to describe the uncertainty region corresponding to the reference point (see Fig. 2B).

The GHT may be modified to include this capability, i.e., the cells in the parameter space are no longer updated at a single position but for the whole uncertainty region. It is necessary to emphasize some aspects of this approximation:

- It is not necessary to determine the uncertainty region exactly. We can use a larger region including it. The motivation for this is that it may be easier to

\[ \begin{align*}
\text{(A)} & \quad \text{(B)}
\end{align*} \]

Fig. 2. Uncertainty region.
design an algorithm to calculate this region (for example, approximating an ellipsoidal region which contains all the samples). In this work, a circular region (i.e. the easiest option) has been chosen.

- We can consider the possibility of a weighted voting. This allows us to include more a priori information (for example, a probability distribution on the deformation space).

- The uncertainty region may change according to each position along the contour, i.e. this region may be defined as a function of parts of the shape.

Of course, a large region implies a deterioration of the precision. This problem will be solved considering an optimization stage as we will see further on. Moreover, the results may be erroneous because the input information (for example, point along the contour from an edge detector) is very noisy. As we have commented, a solution is to use more stable information such as points, corners, segments, regions, etc. In this work straight segments have been used.

Another problem to solve is how to accumulate the evidence. Let us consider a shape defined by \( n \) segments \( r_i \) of length \( l_i \) (1 \( \leq \) i \( \leq \) n). If we detect \( m \) segments of length \( L_i \) (1 \( \leq \) i \( \leq \) m) corresponding to the tendency \( r_i \) and referencing the position \( p \) in the accumulator, we can calculate the value \( E(p) \) as follows:

\[
E(p) = \sum_{i=1}^{n} a_i \max_{1 \leq j \leq m} \left\{ \min \left( \frac{L_i, l_j}{l_j} \right) \right\}
\]

where

\[
\sum_{i=1}^{n} a_i = 1.
\]

Thus, \( a_i \) weighs each tendency, i.e. it weighs each part of the shape. The value \( E(p) \) is between 0 and 1 and corresponds to the proportion of the contour that has been detected.

When we are locating complex objects, the number of axes to perform the Hough transform may be too large even if we take into account only the principal components. Let us consider, for example, segmenting nematodes (see figures in section Experimental results, where we have added a cell image as non-random noise to confirm the stability of the method). If we select a good set of axes using the principal components analysis, we can reduce the number of axes to seven (translations, rotations, and four non-rigid deformations). Even though the number of axes has been reduced, the parameter space size may still be too large. The solution is to sample each axis, i.e. to create a dictionary of shapes. Obviously, it is possible to find an object in the image which is not included in the dictionary but there will be a similar contour, i.e. an approximation. Therefore, when a big shape variability exists the axes will be sampled to make a dictionary and the non-registered shapes will be taken into account by using the uncertainty region. In fact, we might take out several axes but the uncertainty region would grow very much. In the problem of segmenting nematodes we have sampled the four non-rigid axes using (7,3,3,3) samples, respectively; i.e. we have made a dictionary with only 189 entries to register the wide variability of this shape. Obviously, when we decrease the number of samples, we lose accuracy, however, we have to remember that, in this stage, we need only an approximation. The algorithm to obtain the axes by sampling has been the simplest one, aiming to reduce the number of samples while the distance between each contour (from the training set) and its representation in the reduced accumulator is under a threshold (this threshold is selected according to the size of the uncertainty region).

5. Optimization

Once we have an approximation of the solution, we have to optimize it. This stage may be accomplished by using a proper energy function and a proper algorithm to get the optimum. Obviously, selecting a simple and efficient algorithm, if the initial position of the template is far from the solution this algorithm might fail, so a good initialization is recommended.

The obtained results from the initialization stage are quite good, therefore, there are many deformation models that we can use because the approximation is near enough to the desired solution. However, we will use a new one according to the exposed theory and taking into account that we have an automatic initialization stage which locates a good initial position of the template in the literature obviate it and they initialize the template by hand; see, for example, references (2-4, 11), etc.

To design a new deformable model, we have to define an energy function and to select an optimization method. This global energy function is composed by an external energy function (which contributes with information from the image) and an internal energy function (which contributes with information from the object model). Next, we study the functions (which will be used in a steepest descent method) to evaluate the goodness of a location.

The functions must to be defined on the set of possible locations. An object location is described by a closed curve:

\[
C = \{ f_s(s), f_y(s) \} \quad 0 \leq s \leq L.
\]

Then, the functions may be defined as

- The external energy function. We can use a potential function \( P(x, y) \) to define the following expression:

\[
E_{ext}(C) = \frac{1}{L} \int_0^L P(f_x(s), f_y(s)) ds.
\]

The potential \( P \) is computed as a function of the image data according to the desired goal. For
example, it could depend on the gradient of the image. Obviously, we can use several sources. The external energy function would be defined as

$$E_{\text{ext}}(C) = H(E_{\text{ext}}^1(C), \ldots, E_{\text{ext}}^n(C)),$$

(38)

where $H$ represents a merging function of $n$ different channels of information.

- **The internal energy function** is simply a measure of distance between the curve $C$ and our object model. In the literature, we can find very different definitions, from local constraints determining the elasticity and rigidity of the contour [see references (2, 10) etc.] to complex solutions defining a probabilistic distribution on the space of deformations. We have selected the simplest method, i.e. we have delimited the admissible space of deformations from the learning samples (using the axes calculated from the given principal component analysis), although we can use more complex methods to include other a priori knowledge.

### 5.1. External energy

Because we are using edge information to get the initial location, the first option to define this energy is to use the gradient as follows:

$$P(x, y) = -\Vert \nabla g(x, y) \Vert.$$  

(39)

The main problems in this solution are:

- It is very sensitive to noise. If we are working with a very noisy image, there are a lot of local minima. This is a good option but only if we are very near to the solution. In this sense, it is interesting to use more stable information (for example, only the points from an edge detector) and to use this gradient function in the last step.

- We are not using all the information. The gradient has two components and without the direction the results will be poorer.

- We ignore known information. It is possible to know which edges are involved in a given location, so we are able to refine the position using firstly this more stable information.

To use the edge points, we have several possibilities. An easy and efficient solution is to calculate the distances to the nearest edge point:

$$d(x, y) = \min_{(r, s) \in \mathcal{E}} \{ \Vert (x, y) - (r, s) \Vert \},$$

(40)

where $\mathcal{E}$ is the set of edge points. Obviously, we are able to define the potential function $P$ as:

$$P(x, y) = f(d(x, y))$$

(41)

or even use a measure of significance of the edge points:

$$P(x, y) = \min_{(r, s) \in \mathcal{E}} \{ I(r, s) + f(\Vert (x, y) - (r, s) \Vert) \},$$

(42)

where $I$ weights the edge points. For example, we can define

$$I(r, s) = -\Vert \nabla g(r, s) \Vert$$

(43)

or include other criteria of stability (a small edge is probably a noisy edge, an edge appearing in several scales is a good edge, etc.).

In addition, to improve the results, we can use the gradient direction. A way to accomplish this is to change the definition of the potential function $P$:

$$P(x, y, \theta) = \min_{(r, s)} \{ I(r, s) + h(\Vert (x, y) - (r, s) \Vert, \theta) \}$$

(44)

so the external energy function will have to be changed to

$$E_{\text{ext}}(C) = \frac{1}{L} \int_0^L P(X(s), Y(s), \theta(s)) ds,$$

(45)

where $\theta(t)$ is defined as the angle between the vectors

$$\nabla g(f_s(t), f_t(t)) \quad \text{and} \quad \left( -\frac{df_s}{ds}(t), \frac{df_t}{ds}(t) \right).$$

(46)

A simple definition of $h$ is

$$h(x, \theta) = \begin{cases} f(x) & \text{if } -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}, \\ +\infty & \text{otherwise} \end{cases}.$$  

(47)

and an example using $f(x) = x$ and $I(r, s) = 0$ is shown in Fig. 3.

In our experiments, we have used equations (44) and (45) to define the external energy function ($f(x) = x$ and $I(r, s) = 0$). Note that the energy

![Fig. 3. Representation of the potential function $p(x, y, 0)$ and $p(x, y, \pi/2)$ from a circle.](image-url)
function is defined using the simplified coordinate system (deformation model) from the given principal component analysis. For each iteration, the algorithm calculates $E_{\text{rms}}(C)$ in the neighbourhood of the actual point, then it changes the actual point using the largest decrease. This algorithm iterates until a local minimum has been reached or the obtained point is out of the admissible space of deformations delimited by the learning samples. Obviously, this algorithm converges to the optimal solution if the initialization point is near enough to the desired solution. The experimental results show that the method proposed to initialize the template obtains a good point for the optimization stage.

6. EXPERIMENTAL RESULTS AND DISCUSSION

In this section we present the results on several images. We have considered two types of images: nematodes and hands. We show the results on six images (we have applied the algorithms on other images with similar results). The initial location of the objects has been improved by using a simple algorithm (steepest descendent) with edge information (an energy function as in Fig. 3) in the optimization step.

The uncertainty region is a circle (radius = 10 pixels). The parameter space sizes (non-rigid deformations) have been reduced to (7, 3, 3) (nematodes) and (5, 4, 3) (hands). Moreover, the rotation axis has been reduced to 36, i.e. the result is equivalent to add an axis with size 18.9 and 6 (respectively) to handle non-rigid deformations.

The description of the shown results follows:

- Test images (Fig. 4). (a) and (d): an image of cells has been used to introduce non-random noise to the original one. (b) and (e): several hands have been used to create a scene with overlapping problems; (c) and (f): a component of salt-and-pepper-noise has been introduced ($\text{SNR} = 0.5$).
- Edges (Fig. 5). These images are the results using Canny algorithm with $\sigma = 1.5$ and standard hysteresis thresholding (low threshold = 0.5, high threshold = 0.9).
- Parameter space (Fig. 6). These images represent the parameter space $E$. Because it is very difficult to show it (due to its high dimension), in these images we present the function

$$
\gamma(x, y) \propto \max_{\Delta r, \Delta d_{1}, \ldots, \Delta d_{6}} E(x, y; \Delta r, \Delta d_{1}, \ldots, \Delta d_{6})
$$

As we can see, the objects in the image may be easily segmented from these accumulators.
- Initialization (Fig. 7). In this figure, we show initialization obtained using the proposed algorithm. We have used 0.75 as the threshold applied to each parameter space, i.e. we locate an object when 75% (at least) of the contour has been detected (after thresholding, we select the maximum from each obtained region).
- Final results (Fig. 8). Between 30 and 70 iterations were needed to reach the optimum. In this figure, we draw only the most interesting case for each one of the images, but the results for all the others are similar.

Fig. 4. Images.
6.1. Discussion

It is interesting to study the computational and space complexity of the proposed method in order to emphasize the goodness of our algorithm in comparison with other methods in the literature.

Consider the curve $C(t)$ in equation (32) and a set of deformation axes $\{a_1, \ldots, a_n\}$ which define the set of shapes to locate $C(t) = f(a_1, \ldots, a_n, t)$ ($0 \leq t \leq 2\pi$). In order to use Hough transform, we have to define the $n$-dimensional discretized Hough space $H(A_1, \ldots, A_n)$, which is composed by $[\pi]^n \times n$ cells. Given an evidence
\[(x, y), \text{ we are able to use the equation:}\]

\[A_n = f_1(A_1, \ldots, A_{n-2}, x, y, t)\]  \hspace{1cm} (49)

\[A_{n-1} = f_2(A_1, \ldots, A_{n-2}, x, y, t) \quad 0 \leq t \leq 2\pi\]  \hspace{1cm} (50)

\[\text{to determine a parametric curve in the 2-D plane}\]

\[\text{(}A_n, A_{n-1}\text{) (without loss of generality, we can suppose that these axes correspond to translation).}\]

\[\text{Given } (x, y, t), \text{ standard Hough transform (SHT) computes the set of tuples } (A_1, \ldots, A_n) \text{ according to the}\]
equations (49) and (50) and the nearest cells $H(A_1, \ldots, A_n)$ of the discretized space are incremented by one. Therefore, for each tuple $(x, y, t)$, computational and space complexity of the SHT is given by $\prod_{i=1}^{n_t} n_t$ and $\prod_{i=1}^{n_s} n_s$, respectively.

In our algorithm, the set of axes is simplified and a region of uncertainty is used. Let $R$ be the size of this region and $H(B_1, \ldots, B_m)$ the simplified Hough space composed by $\prod_{i=1}^{n_t} m_t$ cells. For each tuple $(x, y, t)$, computational and space complexity of our reformed Hough transform is given by $R \cdot \prod_{i=1}^{n_t} m_t$ and $\prod_{i=1}^{n_s} m_i$, respectively. First, if we obtain a simplified set of axes and a coarse resolution to define the simplified Hough space, computational complexity is much better (note that it is interesting to simplify the number of axes and to obtain an uncertainty region as small as possible). Second, space complexity is clearly improved because of the smaller accumulator array.

Other efficient implementations of the Hough transform, such as adaptive Hough transform (AHT)\cite{125} or fast Hough transform (FHT),\cite{13,26} have been proposed in order to minimize the above mentioned requirements. In these methods, a small accumulator array is used. They analyse the accumulator array at a coarse resolution and then zoom down into the regions where a peak has been detected. Although these algorithms are more efficient than SHT because of the multiresolution accumulator array, the algorithm we propose is better because: (1) FHT and AHT decrease computational cost because of the coarse resolution (the number of parameter remains the same), therefore they cannot be applied in problems with high dimensionality. However, the proposed algorithm removes a large number of axes by using the uncertainty region and, in consequence the dimensionality of the accumulator is much smaller, which allows us to apply Hough transform in those problems. For example, let us consider the problem of the hands by using a simple polygonal approximation with 14 points (actually, our template is composed by 35 points!). If we applied HT at a coarse resolution using only $n_t = 3 \forall t$, we would need $3^{28}$ cells to store the accumulator! (2) Our algorithm takes into account local deformations, thus it may be applied to locate biological objects. (3) AHT extracts curves one-by-one and the whole process has to be repeated to find a new curve. (4) The structure of the accumulator in AHT may cause problems if a peak is close to the borders of cells because of peak spread. However, in our algorithm, we have defined an uncertainty region which guarantees we obtain all the peaks. Moreover, we have defined a simple method to determine the threshold used to locate peaks in the accumulator space; thus the algorithm determines automatically the number of objects in the image. Unfortunately, if we use a large uncertainty region to locate objects from complex scenes the probability of false peaks can be very high. In order to solve this problem, we propose the use of qualitative features (for example, straight segments) which are effective in reducing the number of spurious interpretations\cite{127}.

On the other hand, the method we have presented in this paper can be classified as an active shape model\cite{28,29} because the shape can only deform to fit the data in ways consistent with the training set. However, this paper presents a more robust and efficient method because: (1) We use a physically-based shape model which allows us to start from a canonical, frequency-ordered orthogonal coordinate system. In fact, if we use the coordinate system from the Coote’s model, our experiments show that the number of needed axes to accommodate shape variability is greater than from our model. (We obtain a coordinate system in which the first modes are large-scale variations of shape and the higher order modes are more localized.) (2) We have designed a new method to obtain the initial guess. This method takes advantage of both, the reduced number of deformation axes and the Hough transform. For practical applications, we cannot suppose that the objects to segment will have strong edges,\cite{29} therefore we need a good initialisation stage. (3) To cope with the noise, we have defined a more robust external energy function which allows us to use a simple optimization algorithm.

7. CONCLUSIONS

In this paper we have described a new approach to 2-D object segmentation using an automatic method applied to images with problems such as partial information, overlapping objects, many objects in a single scene, severe noise conditions and a very high degree of deformation. We illustrate our approach on many different examples, which demonstrates its applicability to a broad range of image interpretation problems.

With the aim to develop an automatic method, we have described how to learn the object model and how to locate it from external information. Because the object has a high variability, we have introduced a new deformation model. It has been obtained from modal analysis, allowing us to use a robust deformation surface, which reduces the number of deformation axes and provides a global-to-local ordering of shape deformation. Because of the usual problems encountered in images, the Hough transform is an excellent tool to segment such objects from the image background and other clutter. However, using its original formulation, the Hough transform cannot be applied to locate deformable objects. The main contribution of our paper is that the Hough transform now may be used to fit the initial position of a template. This has been achieved by reformulating the original method, thus producing a new algorithm, in which the precision has been decreased to keep the required time and space within practical limits. The located position of the shape from this algorithm is not the desired solution. However it is near enough to allow
the application of a simple optimization method to refine it. The initialization and optimization method, jointly make up a completely automatic algorithm independent of the object to be located. In conclusion, this approach holds considerable promise not only as a technique to segment images but also, as a first work to design new and better algorithms.

8. SUMMARY

This paper proposes a new methodology to automatically segment images. There is a large collection of publications on shape matching using either rigid or deformable templates. In this work, our goal is to propose a general (not limited to a specific application) biological object localization and retrieval scheme using deformable template in the presence of noise, clutter and occlusion. Because of these problems, a robust algorithm is required.

The Hough transform (HT) method is relatively insensitive to noise and occlusion, i.e. it is robust. However, its application is limited because of its excessive requirement of memory and computation especially as the number of parameters increases. Besides, the contour model has to be a parametric description of the shape such that it may be used to characterize other objects in different applications. In order to solve these problems, we propose:

- A new deformable template model. The method starts from both, a polygonal approximation of the shape and a set of samples, to obtain a deformation system. Once the object model has been learned, we are able to use it to detect and extract the objects model from every image. On the one hand, the algorithm to approximate the contour by a polygon is out of this work, although this problem is studied in a separate paper. On the other hand, we describe in detail how to transform the deformation system based on modal analysis (proposed by Pentland and Sclaroff for matching contour) in order to get a good coordinate system, that is, a good parametric description of the shape.

- A method to obtain a first approximation and to optimize it. The method to initialize the template is based on HT. We have reformulated HT to describe a robust algorithm which can be used in spite of the number of parameters already reduced at learning stage. Finally, the algorithm optimizes the first approximation making use of the learned coordinate system and an energy function based on edge information.

Obviously, this work is not completely closed, i.e. we think this is a first paper in which implemented algorithms and obtained results indicate a new and promising way not only to improve this method but to make new algorithms in accordance with the proposed methodology.

Acknowledgements—We are grateful to the anonymous reviewers of the paper for their comments and suggestions. Special thanks to Josef Kittler for his comments when the first author was visiting the University of Surrey. This research has been partially supported by grant TIC97-1134-c02-01.

REFERENCES


**APPENDIX A. FORMULATING THE EIGENVALUE PROBLEM**

Let us consider $n$ nodal points $(x_i, y_i)$. The 2-D interpolation matrix from the shape functions is defined by

$$H(x, y) = \begin{pmatrix} h_1(x, y) & \cdots & h_n(x, y) \\ 0 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & h_1(x, y) \end{pmatrix}$$

We build the interpolation function $h_i$ as the sum of $n$ basis functions, one per nodal point $(x_i, y_i)$

$$h_i(x, y) = \sum_{k=1}^{n} a_k g_k(x, y)$$

where $g_k(x, y)$ is a Gaussian function with centre $(x_k, y_k)$. The coefficients $a_k$ may be obtained by inverting the matrix

$$G = \begin{pmatrix} g_1(x_1, y_1) & \cdots & g_1(x_n, y_n) \\ \vdots & \cdots & \vdots \\ g_n(x_1, y_1) & \cdots & g_n(x_n, y_n) \end{pmatrix}$$

The mass matrix is calculated using

$$M = \begin{pmatrix} M_{aa} & 0 \\ 0 & M_{bb} \end{pmatrix},$$

$$M_{aa} = M_{bb} = \rho \pi \sigma^2 G^{-1} R G^{-1},$$

where the elements of $R$ are the square roots of the elements of the $G$ matrix.

To obtain a 2-D stiffness matrix $K$ we need to compute a stress-strain interpolation matrix $B$:

$$B(x, y) = \begin{pmatrix} \frac{\partial}{\partial x} h_1(x, y) & \cdots & \frac{\partial}{\partial x} h_n(x, y) \\ 0 & \cdots & 0 \\ \frac{\partial}{\partial y} h_1(x, y) & \cdots & \frac{\partial}{\partial y} h_n(x, y) \end{pmatrix}.$$

From this interpolation matrix, we can obtain

$$K = \begin{pmatrix} K_{aa} & K_{ab} \\ K_{ba} & K_{bb} \end{pmatrix},$$

where

$$k_{aa} = \frac{\pi \beta \sum_{k,l} a_k a_l \left[ 1 + \frac{\xi}{2 (x_k - x_l)^2} + \frac{\eta}{2 (y_k - y_l)^2} \right] r_{ij}}{4\sigma^2},$$

$$k_{bb} = \frac{\pi \beta \sum_{k,l} a_k a_l \left[ 1 + \frac{\xi}{2 (y_k - y_l)^2} + \frac{\eta}{2 (x_k - x_l)^2} \right] r_{ij}}{4\sigma^2},$$

$$k_{ab} = k_{ba} = \frac{\pi \beta (\chi + \xi)}{4\sigma^2} \sum_{k,l} a_k a_l (x_k - x_l)(y_k - y_l) r_{ij}.$$

and

$$\chi = \frac{v}{1 - v}, \quad \beta = \frac{E(1 - v)}{(1 + v)(1 - 2v)}, \quad \xi = \frac{1 - 2v}{2(1 - v)}.$$
About the Author—A. GARRIDO was born in Granada, Spain, in 1969. He received the B.S. and Ph.D. degrees both in Computer Science from the University of Granada in 1992 and 1996, respectively. Since 1993 he has been with the Computer Science Department (DECSAI) at Granada University. His current interest includes pattern recognition, multiresolution method, deformable templates, image registration, and biomedical applications. Dr. A. Garrido is a member of the IAPR Association.

About the Author—N. PÉREZ DE LA BLANCA was born in Granada, Spain. He received the B.S. and Ph.D. degrees both in Mathematics from the University of Granada in 1975 and 1979, respectively. He was at the Statistical Department of the University of Granada from 1976 to 1986. In 1986 he moved to the Computer Science and Artificial Intelligence Department (DECSAI). Now he is Professor of Artificial Vision. His current interest includes pattern recognition, image registration, deformable templates, 3-D medical applications. Prof. Pérez de la Blanca is a member of the IAPR, SPIE. He is the Vice-president of the IAPR’s Spanish chapter.