Rigid Object Recognition Invariable to Rotation and Translation based on Contour Detection and Least-Squares Minimization of a Multi-Layer Perceptron Neural Network

R. ROSAS-ROMERO, J. RODRIGUEZ-ASOMOZA, D. BAEZ LOPEZ
Departamento de Ingeniería Electrónica
Universidad de las Américas, Puebla.
Exhacienda Sta. Catarina Mártir, Cholula, Puebla.
México.

Abstract:-- This work begins describing a technique for representation of an object by collecting points over its surface and then using this set of points to feed a supervised adaptation algorithm for a multi-layer perceptron neural network that serves as a distance function. The scheme for extracting the surface in objects supports semi-automatic segmentation of sets of images. A collection of distance-function perceptrons representing different rigid objects is generated, and these perceptrons are used for rigid object recognition by measuring the mismatch between an object to be identified and an object described by its distance-function perceptron. Furthermore, our object identification approach is invariant to rigid transformations.

Keywords:-- Object Recognition, Object Identification, Multi-Layer Perceptron, Distance Function.

1 Introduction
The problem of object recognition consists of two stages: Object Representation (or Modeling) and Object Identification. In the first stage, modeling of an object is accomplished first by reconstructing the object shape from points collected from its physical surface extracted from sets of images, and then a supervised multi-layer perceptron neural network is adjusted to find the optimum distance function which gives the distance from any point in space to the object under modeling. Object surfaces are extracted from images by using an Active Contour Model. Once the representation for different objects is established, a “testing” object is identified (“labeled”) by comparing it with each “model”. The distance function of each “reference” object gives the mismatch between the “testing” object and the “reference” object, and the shortest mismatch indicates the “label” to be assigned to the “testing” object as it is shown in Figure 1. One problem in comparing two objects is that they can be misaligned so that rigid transformations (translation and rotation) are systematically applied to the “testing” object until it gets as aligned as possible to the “model” object to be compared with so that object identification is invariable to translation and rotation.

2 Extraction of an Object Surface from a Set of Images based on an Active Contour Model
Extraction of the surface of an object from an image set consists in reconstructing its shape from points collected from its physical surface. There is a set of images that shows different views of an object, and the intersection of each imaging plane with the object gives a contour. Tracing of contour points on different imaging planes and joining them generates a 3-D surface. We use active contour models to extract contour points from images ([1], [2], [6]). Consider the problem of detecting the borders of a 3-D
reference object. If there are \( m \) planes per 3-D object, and \( n \) points per plane; then there are \( N = mn \) contour points to be searched. By using an Active Contour Model, a single imaging plane is used to detect \( n \) contour points on that specific plane, and this process is repeated for each of \( m \) imaging planes in the object. Instead of attempting to perform contour detection for each imaging plane in isolation, we directly approach it as a 3-D problem; so that the \( mn \) contour points corresponding to the object surface are detected at once from multiple imaging planes ([2], [3], [4]). An active contour model is a deformable curve which approaches contours on images by optimizing the placement of the points that form the curve. In the 3-D model, each point \( v(r,s) = \{x(r,s), y(t,s), z(t,s)\}^T \) is a function of two parameters \( r \) (spatial index), \( s \) (imaging plane index); so that the function to be optimized is defined as \( f(v) = \alpha_i \|v_r\| + \alpha_2 \|v_t\| + \beta_i \|v_{rt}\| + \beta_2 \|v_{rr}\| + E \), where \( \{\alpha_i\} \) are constants imposing a bending constraint and \( \{\beta_i\} \) are constants imposing a tension constraint, and \( E \) is some sort of image gradient function.

### 3 Multi-Layer Perceptron Neural Network as a Distance Function for Object Representation

Since a “reference” object \( R \) is fixed in space, the mismatch between \( R \) and a “testing” object \( T \) is computed by generating a distance function corresponding to the set of surface points on \( R \). The distance function gives the minimum distance from \( R \) to any point in the space. The model that we use for a distance function is an extension of the chamfer distance transform ([3], [4], [5]) to three dimensions by adjusting the coefficients of a \( f \) multi-layer perceptron neural network with a least-mean squares adaptive algorithm. The input signals to the perceptron are the three coordinates of a spatial point \( P_A(x_A, y_A, z_A) \) and the desired response is the distance from this point to \( R \); \( d(R, P_A) \), so that the task of the perceptron is to assign each received set of three signals \( x_A, y_A, z_A \) to one distance value ([5], [7]). An example of the set of data used for the supervised adaptation algorithm of the perceptron that represents a 2-D object is shown in Figure 2. Black pixels correspond to points on the contour of a 2-D objects and the distance at these points is set to zero. To have a non-trivial output signal, points out of the surface and their corresponding distances are computed by using a gradient function.

During supervised adaptation, for each surface point \( P_A(x_A, y_A, z_A) \), the perceptron knows in advance the desired response \( d(R, P_A) \), computes the difference between the desired and actual response, evaluates the criterion of performance, and uses it to adjust its coefficients as it is shown if Figure 3. After completion of the supervised adaptation algorithm the discrete distance map used as a priori information is transformed into a continuous distance function \( d_R() \). Once the distance function is established, it can be used to measure the mismatch between the surface points of a “sample” or “testing” object \( S \) (represented by a set of surface points \( S = \{P_1, P_2, \ldots, P_n\} \) and the “reference” or “model” object \( R \) represented by its distance function \( d_R() \). Each point \( P_i \) on the “sample” surface \( S \) is projected onto the

<table>
<thead>
<tr>
<th>( (0,0) )</th>
<th>( (0,1) )</th>
<th>( (0,2) )</th>
<th>( (0,3) )</th>
<th>( (0,4) )</th>
<th>( (0,5) )</th>
<th>( (0,6) )</th>
<th>( (0,7) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (1,0) )</td>
<td>( (1,1) )</td>
<td>( (1,2) )</td>
<td>( (1,3) )</td>
<td>( (1,4) )</td>
<td>( (1,5) )</td>
<td>( (1,6) )</td>
<td>( (1,7) )</td>
</tr>
<tr>
<td>( (2,0) )</td>
<td>( (2,1) )</td>
<td>( (2,2) )</td>
<td>( (2,3) )</td>
<td>( (2,4) )</td>
<td>( (2,5) )</td>
<td>( (2,6) )</td>
<td>( (2,7) )</td>
</tr>
<tr>
<td>( (3,0) )</td>
<td>( (3,1) )</td>
<td>( (3,2) )</td>
<td>( (3,3) )</td>
<td>( (3,4) )</td>
<td>( (3,5) )</td>
<td>( (3,6) )</td>
<td>( (3,7) )</td>
</tr>
<tr>
<td>( (4,0) )</td>
<td>( (4,1) )</td>
<td>( (4,2) )</td>
<td>( (4,3) )</td>
<td>( (4,4) )</td>
<td>( (4,5) )</td>
<td>( (4,6) )</td>
<td>( (4,7) )</td>
</tr>
<tr>
<td>( (5,0) )</td>
<td>( (5,1) )</td>
<td>( (5,2) )</td>
<td>( (5,3) )</td>
<td>( (5,4) )</td>
<td>( (5,5) )</td>
<td>( (5,6) )</td>
<td>( (5,7) )</td>
</tr>
<tr>
<td>( (6,0) )</td>
<td>( (6,1) )</td>
<td>( (6,2) )</td>
<td>( (6,3) )</td>
<td>( (6,4) )</td>
<td>( (6,5) )</td>
<td>( (6,6) )</td>
<td>( (6,7) )</td>
</tr>
<tr>
<td>( (7,0) )</td>
<td>( (7,1) )</td>
<td>( (7,2) )</td>
<td>( (7,3) )</td>
<td>( (7,4) )</td>
<td>( (7,5) )</td>
<td>( (7,6) )</td>
<td>( (7,7) )</td>
</tr>
</tbody>
</table>

Figure 2. A priori information about input signals and the desired response signal to the adaptive filter that serves as a distance function for a 2-D object under modeling. Points on the object contour are assigned a zero distance and points out of the surface have their distance computed by means of the gradient.
distance function $d_i = d_R(P_i)$ to give a value that is its distance to the nearest point on the “reference” surface $R$. The mismatching between two objects is computed as the average of the distances of all the surface points of the “sample” object to the “reference” ($\{R\}$, $\{S\}$); so that mismatching between $R$ and $S$ is given by

$$\sum_{i=1}^{N} d_R(P_i).$$

The perceptron structure is shown in Figure 4. The estimate of $y(n, \zeta)$ is computed by using the linear combiner $\hat{y}(n) = \sum_{k=1}^{3} c_k(n)x_k(n, \zeta)$, where $c(n) = [c_1(n), c_2(n), c_3(n)]^T$ is the coefficient vector and $x(n) = [x_1(n, \zeta), x_2(n, \zeta), x_3(n, \zeta)]^T$ is the input data vector. Error formation is given by $e(n, \zeta) = y(n, \zeta) - \hat{y}(n, \zeta)$, and the updating of the coefficient vector to a new estimate is given by $c(n, \zeta) = c(n - 1, \zeta) + \Delta c[x(n, \zeta), e(n, \zeta)]$ so that the output $\hat{y}(n)$ is a better estimate of the desired response $y(n)$. The mean square deviation $||e(\zeta)||^2$ is used to evaluate the perceptron performance and by minimizing with the aid of the Levenverg-Marquadt Least-Mean Squares method the perceptron gets close to the optimum distance function.

4 3-D Object Identification Invariable to Rigid Transformations

Once a set of surface points in an object to identify $T$ (“testing” object) is collected from a set of images, we need to find the rigid transformation function $T()$ that matches $T$ to a “reference” object $R$ modeled by its distance function $d_R()$ so that $T(R, P) = T'$ gets closer to $R$. The search of the rotation and translation parameters $P = \{\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma\}$ is an optimization process that minimizes the differences between two objects $T'$ and $R$ (Levenverg-Marquardt Least Squares Minimization). During this optimization process, rigid transformations are systematically applied to $T$, by adjusting the set of parameters $P$, until the corresponding sets of transformed points $T'$ get as close as possible to the reference $R$, until the distance $d_R(T)$ between both objects is minimized ([8], [10], [11]). Thus, estimation of the similarity between two objects invariant to rigid transformations can be referred to as the minimization of the cost function, $C(p) = d_R(T(\{T\}))$.

Figure 3. Supervised Multi-Layer Perceptron Neural Network that implements the distance function for modeling of an object.

5 Experimental Results

Experiments were conducted to identify automobile nameplates, with the system trained with 27 alphabet characters and 10 number characters from 15 random images of nameplates as the one shown in Figure 5. After training, 50 automobile plates were tested by recognizing each of its 7 characters. The performance of the identification process for each of the plates is measured as $(100x/7)\%$, where $x$ is the number of identified characters. The performance of the plate identification process was of 97.6% and it was measured by computing $(100y/50)\%$, where $y$ is the total number of correctly identified plates. Other experiments were conducted to compare 58 different rigid objects with 18 models (airplane, car, cow, chair, glass, dog, etc.) used to train the system. During these experiments, sets of 9 imaging planes were obtained for each object as the one shown in Figure 6. Results reported a performance of 86.5 % for our object recognition method.
Conclusions
This paper presents a new technique for object recognition invariant to rigid transformations based on a multi-layer perceptron neural network. The extraction of the object surface supports semi-automatic segmentation of sets of images. The proposed surface extraction technique employs gradients in sequences of images to attract a deformable surface model by using imaging planes that correspond to multiple locations in space. The effectiveness and accuracy of the modeling of an object by generating its distance function depends on the number of surface points extracted from sets of images. An issue concerning this relation that is difficult and deserves attention is to develop a methodology to find the optimal number of points that gives the best estimates and does not sacrifice speed of computation. Experiment results report a performance of 86% for our object recognition method.

References
Engineering in Medicine and Biology Society;
Cancún, México; 2003.

