Fuzzy Associative Databases for Visual Recognition of 2D and 3D Objects

Aaron Mavrinac, Xiang Chen, and Ahmad Shawky

Abstract

It is desirable for automated object recognition using computer vision systems to emulate the human capacity for recognition of shapes invariant to various transformations. We present an algorithm, based on a Fuzzy Associative Database approach, which uses appropriately invariant metrics and a neuro-fuzzy inference method to accurately classify both two- and three-dimensional objects (using different metrics for each). The system is trained using a small number of images of each object class under varying degrees of the transformations, and as we show experimentally, is then able to identify objects under other non-explicitly-trained degrees of the transformations.

Keywords: computer vision, fuzzy associative database, object recognition.

1. Introduction

Humans are generally able to recognize two-dimensional shapes, regardless of changes in orientation, scale, or skew, after having seen the shape in one such configuration. This shape recognition has a very wide range of applications, and accordingly, much work has gone into automating it with computers. The basic theory is that shapes can be extracted from otherwise cluttered and cumbersome images, from which some set of quantifiers efficiently describing the shapes can be obtained and compared to known values through some algorithm for classification. The nature of these quantifiers and the classification algorithm are a subject of much research; most use quantifiers invariant to the aforementioned transformations (rotation, scale, skew, etc.) such as Fourier descriptors, moment invariants, and Hough transformations, and most use machine learning methods for classification.

Humans are also generally able to recognize three-dimensional objects, regardless of their orientation, after having seen a sufficient number of different views (depending, of course, on the nature of the object itself).

To generalize from the two-dimensional case, it is possible to automate this process in a similar manner by obtaining quantifiers describing the three-dimensional surface rather than the two-dimensional shape. Such quantifiers can be extracted from range images. However, a single such image gives information only from a certain perspective. To approach complete three-dimensional information, range images must be taken from different perspectives around the object. For classification to continue to work as generalized from the two-dimensional case, the sets of quantifiers from each perspective must be combined to fully describe the object, and the classification algorithm must be designed to operate on this type of information.

In this paper, we present results in using invariant image descriptors and a Fuzzy Associative Database (FAD) to recognize both two-dimensional and three-dimensional shapes in binary and range images, respectively. The FAD consists of a fuzzy database (FD), which contains the trained information about the object classes and is analogous to human memory of previously-seen views of an object, and a fuzzy search engine (FSE), which processes new incoming information against the FD and is analogous to human reasoning. The FD contains two tables; the FSE uses the information in the first table to construct a Bank of Fuzzy Associative Memory Matrix (BFAMM) to conduct a search over the second table. The FSE thus executes recognition by establishing a correspondence between an input view of an object and one trained class in the second table.

The remainder of this paper is organized as follows. In Section 2, we outline some representative prior work in the areas of two- and three-dimensional object recognition, as well as methods using neuro-fuzzy systems in particular, to establish the state of the art. In Section 3 we give some theoretical background necessary for a self-contained understanding of the three-dimensional case. We detail the invariant metrics chosen for classification in Section 4. The Fuzzy Associative Database approach and our algorithm, which are the main contributions of this work, are fully described in Section 5. Empirical evaluation using real image data, for both two- and three-dimensional applications, is presented in Section 6. Finally, some concluding remarks are given in Section 7.
2. Related Work

A. Two-Dimensional Recognition

Two-dimensional shape recognition is one of the earliest applications of image processing and pattern recognition. Methods for shape analysis—the extraction of the salient information needed for recognition—are numerous and well-studied; Loncaric [1] presents a fairly comprehensive survey. A variety of approaches to the recognition stage have been proposed as well; examples include the template matching approach (see Penz et al. [2]), the correlation classification approach (see Blais et al. [3]), and the directional flow change approach (see Huang et al. [4]).

B. Three-Dimensional Recognition

There are several cases where two-dimensional moment invariants have been used for recognition of three-dimensional objects. Mashor et al. [5] and Nian et al. [6] compute moment invariants on a series of intensity images of the object taken from a variety of positions around it; it is demonstrated that with a sufficient number of images and proper handling of the multi-image input in an artificial neural network scheme, two-dimensional moments are applicable to three-dimensional recognition. However, these methods do not examine three-dimensional information about the object directly, and require a large number of explicitly-ordered views to operate. In addition to the cost of capturing these views, objects are not identified from an arbitrary unknown pose.

Methods have also been proposed which operate on invariants of three-dimensional range data. Campbell and Flynn [7] extend the concept of computing characteristic vectors of multiple images to range images, allowing for object recognition in arbitrary pose unaffected by illumination. Hetzel et al. [8] compute local feature histograms, invariant to translations and rotations as well as being robust to partial occlusions, directly on range images; recognition is then performed using histogram matching or probabilistic recognition.

There are a number of alternate possibilities which employ other descriptors entirely. One example is the work of Lin and Lee [9], in which chromaticity distributions from a variety of images of the object are used to identify the object; this method of recognition, while pose-invariant, is adversely affected by variations in illumination, though the work attempts to alleviate these problems.

C. Equations

Neuro-fuzzy classifiers [10] are used to solve a wide range of recognition problems. In particular, a number of fuzzy LVQ schemes have been proposed for prototype-based classification and recognition. Methods such as those described by Karayiannis and Pai [11], Kusumoputro et al. [12], and Wu and Yang [13] employ a fuzzy neighbourhood function on training data with specific classes, whereas others, such as Thiel et al. [14], attach fuzzy labels to the training data themselves.

Fuzzy associative memory models, introduced by Kong and Kosko [15], [16], have been employed to store rules for classifications based on fuzzy LVQ. This is the approach we have taken in our previous work on the problem of 2D and 3D object recognition [17]-[19].

3. Preliminary Theory: Depth Maps

In order to quantify three-dimensional shapes in a manner useful for recognition, some representation of the shape must be generated by the sensor. A stereo vision system provides data which can be analyzed in a variety of ways to obtain three-dimensional information, but the crucial point, in this case, is for the representation to lend itself to some analog of the two-dimensional work in [17]. Fortunately, a representation exists to which a similar recognition scheme may be applied, and it is in fact relatively easy to obtain.

For the purpose of this description and throughout this work, the following convention is used for the world and image coordinate systems: lowercase $x$ and $y$ represent image coordinates with origin at the upper left corner of the image and positive axes right and down respectively, and uppercase $X$, $Y$, and $Z$ represent world coordinates (which, unless otherwise specified, are mutually orthogonal with $Z$ perpendicular to the rectified image planes and have their origin at the optical center of the left camera). Figure 1 illustrates their relationship.

![Figure 1. Coordinate System Convention.](image-url)
Given a pixel of coordinates \((x_1, y_1)\) in one image of an epipolar-rectified stereo pair, and a corresponding pixel \((x_2, y_2)\) in the other (where \(y_1 = y_2\)), their disparity \(d\) is defined as \(x_2 - x_1\) [21]. This can be used to triangulate the depth to the original 3D point in the environment (from the optical center of one camera) in the world coordinate system according to the following relation:

\[
Z = \frac{bf\lambda}{d}
\]

where \(b\) is the baseline (distance between the two optical centers), \(f\) is the focal length, and \(\lambda\) is a scale factor relating the pixel width to real-world measurements.

A depth map is a 2D matrix \(D\) containing the disparity of each pixel in one image with respect to the corresponding pixel, if any, in the other. Thus, if pixel \((i, j)\) in the first image corresponds to pixel \((k, j)\) in the second, \(D_{ij} = k - i\). The depth map essentially results in a range image when its values are normalized and/or quantized to a range of grayscale values which can be displayed and manipulated as such. This provides an important visualization tool and allows existing image-invariant computation algorithms to function unmodified on the data.

With a calibrated stereo vision system, the parameters \(b, f, \) and \(\lambda\) are known and (1) may be used to calculate the actual depth (\(Z\) coordinate) of the real points represented by pixels in the depth map. However, for the purposes of 3D object recognition this is not necessary. Instead, the invariant descriptors (see Section 3B) are computed from the depth map, or more specifically, from its associated range image.

In order to construct a depth map for the first image in a stereo pair, it is necessary to establish correspondences in the second image for each pixel in the first. Correlation-based methods such as the sum of square difference (SSD) and normalized cross-correlation (NCC) criteria may be used for this purpose.

Correlation-based correspondence consists of maximizing, for each left-image pixel \(p_i\), a similarity criterion \(C\) on the displacement \(d = [d_1, d_2]^T\), selecting \(d + p_i\) as the corresponding right-image pixel.

\[
C(d) = \sum_{k=-w}^{w} \sum_{j=-w}^{w} \psi(l_i(i + k, j + i), l_r(i + k - d_r, j + 1 - d_j))
\]

(2)

In this case, since the images \(l_i\) and \(l_r\) are rectified and correspondences are therefore found on the same horizontal line, \(d_1\) can be constrained to zero [25]. We use here the SSD criterion for \(\psi\), that is, for two pixel values \(u\) and \(v\), \(\psi(u, v) = -(u-v)^2\).

4. Invariant Descriptors

We examined a variety of invariant descriptors calculated from 2D images, evaluating their usefulness in describing different range views of an object qualitatively and quantitatively. Two in particular were selected to describe two-dimensional images, and three were selected to describe three-dimensional range images.

A. Compactness

The first useful descriptor is the compactness, a Fourier descriptor which describes a distribution of intensity values in an enclosed region. When applied to a depth map, it describes the range distribution invariant to translation and rotation. The compactness of a greyscale image can be calculated as follows, adapted from [26]:

\[
C = \frac{\left(\sum_{y=1}^{n} \sum_{x=1}^{w} f_{\text{boundary}}(x, y)\right)^2}{\sum_{y=1}^{n} \sum_{x=1}^{w} f(x, y)}
\]

(3)

where \(f(x, y)\) is the value of the image at pixel \((x, y)\) and \(f_{\text{boundary}}(x, y)\) defines pixels on the perimeter of a region (object).

B. Image Moments

The second descriptor is the image moment, a weighted average of pixel intensities which describes the shape in some way. The “raw” moment of order \((i+j)\) is computed for an image as follows:

\[
M_{ij} = \sum_{x} \sum_{y} x^i y^j f(x,y)
\]

(4)

where, as before, \(f(x,y)\) is the value of the image at pixel \((x, y)\). For two-dimensional object recognition, the second-order moment is used, calculated as \(M = \mu_{20} + \mu_{02}\), where \(\mu_{20} = M_{20} - xM_{10}\) and \(\mu_{02} = M_{02} - yM_{01}\).

Hu [27] describes seven moment invariants which are invariant to translation, rotation, and scale. Only the lowest-order moment is applied to the depth maps, as it is robust against the inherent noise from imperfect correspondences and occlusions. It is calculated as follows:

\[
I_1 = \frac{M_{20} - \bar{x}M_{10} + M_{02} - \bar{y}M_{01}}{M_{00}}
\]

(5)

C. Histogram

The final descriptor is the histogram, a Fourier descriptor which describes the overall distribution of intensities in an image. When applied to a depth map, it describes rather the range distribution. The histogram is not a scalar value like the previous two descriptors, but may be compared for two different images as follows [8]:

\[
X^2(f_1, f_2) = \sum_{i=0}^{n} \frac{(h_{1i} - h_{2i})^2}{h_{1i} + h_{2i}}
\]

(6)

where \(f_1\) and \(f_2\) are the images, \(h_{1i}\) and \(h_{2i}\) are the \(i\)th elements of the first and second histogram, respectively, and \(n\) is the final element in the histogram, which is normally 255 as the upper limit of the normalized range for a depth map.

The histogram is used in classifying three-dimensional objects only, as it is sensitive to perturbations in the ob-
ject boundary in the two-dimensional case.

5. Fuzzy Associative Database Algorithm

Fuzzy set theory lends itself particularly well to the problem of recognition based on a set of imprecise descriptors with much variation and overlap. However, it is generally impractical to develop a rule set for classification directly, since it is not immediately obvious what each descriptor represents about the object and how they combine. In such cases, one may train and optimize the parameters of the fuzzy system using a neural network, in a configuration known as a neuro-fuzzy system [10].

We describe here a fuzzy associative database originally described in [17], here generalized for the invariant values of both planar objects and depth maps, and for multiple training images expected to differ as a result of the viewpoint change. The basic approach is to store a table of fuzzy sets associated with the corresponding membership functions, where each class (type of object to be recognized) has one fuzzy set for each invariant value, which are constructed from fuzzified invariant values extracted from the depth maps of the object from several different viewpoints (the training set). Recognition can then be accomplished by comparing input invariant values to the fuzzy sets in each class and determining which matches best.

A. Original FAD Algorithm

An FAD algorithm is designed for multiple planar object recognition. With this algorithm, one first trains an FAD, and then the algorithm can be applied to recognition of multiple planar objects automatically by associating the incoming object to one of the trained classes. Both phases, training and classification phases, require the invariant values of the incoming object. The following three subsections present the details of this FAD algorithm.

Using the invariant values, a fuzzy knowledge database is constructed. This knowledge base consists of two tables (Figure 2) and holds the information about the known classes while serving the memory function for recognizing different objects.

Given the moment factor \( M \) and the compactness factor \( C \) of an image, two fuzzy sets \( F(z, a, k) \) induced by \( a = M \) and \( a = C \) separately are called the fuzzified moment factor and the fuzzified compactness factor of the image as defined by

\[
F(z, a, k) = \max \left( \min \left( \frac{z - (1 - k)a}{2ka}, \frac{(1 + k)a - z}{2ka} \right), 0 \right) \tag{7}
\]

where \( z \) is a self-variable and \( k \) is a support factor coefficient determined by the object thickness. For example, for the objects with the thickness less than one centimeter, \( k \) is 0.05.

If an object is under training, the fuzzified invariant values of the trainee are kept in the first table in the fuzzy database. Therefore, in general, the first table contains \( m \) records corresponding to \( m \) trained classes with two fields containing the fuzzified moment factor and the compactness factor respectively for the corresponding trainee. The second table is then indexed in the trained classes of objects, corresponding to the record in the first table.

In order to conduct fuzzy searching for recognition, a Bank of Fuzzy Associative Memory Matrix (BFAMM) [16] is first built based on the information held on Table 1 of the trained fuzzy database. The BFAMM associates incoming object to the one of trained classes recorded on the Table 2 of the Fuzzy Database. The BFAMM is constructed as follows:

\[
BFAMM = \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{m1} \\ f_{12} & f_{22} & \cdots & f_{m2} \end{bmatrix} \tag{8}
\]

where \( f_{ij}, i \in \{1, 2, \ldots, m\} \) and \( j \in \{1, 2\} \), are the fuzzified invariant values for the \( i \)th trained class. Next a composition operation [28], [29] is applied to associate input to output. This operation is defined as follows [16]:

Let \( A \) be a \( 2 \times 1 \) fuzzy vector and \( B \) be a \( 2 \times m \) fuzzy matrix. Let \( a_i, j \in \{1, 2\} \) be the fuzzy entries of \( A \) and \( b_{ij}, i \in \{1, 2, \ldots, m\}, j \in \{1, 2\} \) be the fuzzy entries of \( B \). Then the composition operation \( \Theta \) returns a new \( 1 \times m \) fuzzy vector \( D = A^T \Theta B \) with the fuzzy entry \( d_i \) of \( D \) given by

\[
d_i = \prod_{j=1}^{2} \min (a_i, b_{ij}) \tag{9}
\]

for \( i \in \{1, 2, \ldots, m\} \).

When an incoming object image is captured, a \( 2 \times 1 \) fuzzy vector \( A \) can be obtained from the fuzzified moment factor \( M \) and the compactness factor \( C \) of this image. A fuzzy search through the fuzzy database can then be described by applying the composition operation \( \Theta \) to \( A^T \) and BFAMM to result in a fuzzy vector \( D = A^T \Theta \) BFAMM, of which the fuzzy entries \( d_i \) are given by

\[
d_i = \prod_{j=1}^{2} \min (a_i, f_{ij}) \tag{10}
\]

for \( i \in \{1, 2, \ldots, m\} \).

Note that the elements of the vector \( D \) show the extent of the similarity between the incoming object and the trained classes in the second table of the fuzzy database. The index \( i^* \) of the closest trained class in the second table to the incoming object is the one which maximizes \( d_i \), that is

\[
d_{i^*} = \max_{i \in \{1,2,\ldots,m\}} d_i \tag{11}
\]

Therefore the \( i^* \)th record in the second table of the fuzzy database determines the class of the incoming object. A sample FAD network with four invariant input
values and three trained classes is shown in Figure 2.

Figure 2. FAD Network with 4 Invariants and 3 Classes.

B. Extension to Three-Dimensional Objects

The key difference between the two-dimensional and three-dimensional recognition problems is the use of multiple images (different views) for training and recognition. In the 2D case, the invariant values are sufficient to characterize all possible planar views of the object, and therefore result in relatively compact membership functions of the corresponding fuzzy sets. In the 3D case, multiple views are necessary to capture the full structure of the object, and although the values are invariant to certain planar transformations of the object from a given view, across different views the resulting membership functions may be quite different. This may result in large areas of overlap among the input membership functions and these must therefore be scaled relative to themselves and one another to better describe the object characteristics.

It is possible and beneficial to emphasize the more unique and descriptive portions of the fuzzy sets before they are used for training or recognition. The fuzzy adaptive database is modified for the multi-image case as described in the following sections.

C. Supervised Training

During the supervised training stage, the invariant descriptors are computed from a depth map of an object of known class. These are first fuzzified into a fuzzy set with a Gaussian membership function:

\[ F(x, m, \sigma) = e^{-\frac{(x-m)^2}{2\sigma^2}} \]  (12)

where \( x \) is the universe of discourse, \( m \) (the mean) is the input crisp value and \( \sigma \) is the standard deviation of the Gaussian, which is determined by trial and error.

Data from multiple views is thus entered, and the fuzzy sets are joined via a union operator. This results in a joint fuzzy set in each invariant value describing the object in an unbiased fashion from multiple viewpoints. In other words, the fuzzy set describes the entire range of acceptable invariant data associated with the object class. The value \( \sigma \) is chosen so that this statement is as true as possible without any more overlap with other classes than is necessary.

The net result so far, assuming a good training set and a good value of \( \sigma \), is that the fuzzy system comprised of the fuzzy sets for each invariant, for a given class, should return a strong response to input invariants generated by a depth map of any viewpoint of an object of the correct class. However, it is also highly likely at this point that there is much overlap among the different classes for certain invariants, and there is no practical way to directly account for such ambiguities.

In order to correct for this, once the fuzzy sets (and the corresponding membership functions) have been constructed for all training examples, they are adaptively scaled, essentially competing for the ranges of each invariant which best describe their classes. To accomplish this, the crisp invariants from the training set are first clustered according to the following algorithm [30]:

1. Taking values of the network inputs as the initial values to form the weight vector;
2. Determining the winner unit based on the minimum distance;
3. Updating the weight vectors of the winner as follows:

\[ w_i(N + 1) = w_i(N) + \alpha(\rho - w_i(N)) \]  (13)

where \( N \) is the number of training epochs (iterations), \( \rho \) is the network inputs (crisp invariant values in our case), and \( \alpha \) is the learning rate (for example \( \alpha = e^{-0.13q - 0.69} \) where \( q \) is the number of trainees in a specific class).

After the cluster centers are found, each fuzzy input is scaled by a measure of the distance from the crisp input data to the associated cluster center as shown below:

\[ A_{ij} \leftarrow A_{ij}e^{\frac{|w_i - \rho_{ij}|}{|w_i + \rho_{ij}|}} \]  (14)

where \( w_i \) is the location of the cluster center in the \( i \)th class, \( A_{ij} \) is the \( j \)th fuzzy input data of the \( i \)th class, and \( \rho_{ij} \) is the \( j \)th crisp input data in the \( i \)th class. As the distance between the cluster center \( w_i \) and input \( \rho_{ij} \) increases, \( A_{ij} \) approaches zero, thus reducing the contribution of data that is far from the cluster center of the class.

Figure 3. Fuzzy Membership Function Before Scaling.
Figure 4. Fuzzy Membership Function After Scaling.

Figures 3 and 4 show an example of scaling on a simple fuzzy membership function.

D. Recognition

Once the fuzzy associative database has been constructed, recognition is a relatively simple process. The system takes crisp invariant values computed from a depth map of the object to be recognized (in any allowable orientation).

The crisp invariants are compared exhaustively to the FAD fuzzy set for each class, returning the total of the responses from each fuzzy set. The inference method found to best quantify the similarity for individual invariant values is a simple crisp value response, according to a standard inference equation:

$$
\mu_a = \bigvee \left[ \mu_j(x) \land I(x) \right]
$$

where $$\mu_j(x) \land I(x)$$ represents the fuzzy intersection between the trained fuzzy set for invariant $$j$$ and the fuzzified invariant from input image $$I$$, and the leading $$\bigvee$$ (union) indicates the fuzzy union over all invariant values. The class with the highest overall degree of membership $$\mu_a$$ is returned as the probable object class.

E. System Overview

The operation of the system is summarized in flowchart diagrams. The basic process of capturing the images used as input for training and recognition in the 2D and 3D cases is shown in Figures 5 and 6, respectively. The last flowchart (Figure 7) shows the actual recognition network, including training, as described in subsections 5C and 5D.

Note in Figure 7 that the invariant fuzzy set scaling and clustering process takes place after all views have been captured by the vision system (with the fuzzified invariant membership functions stored unscaled), so that the resulting database incorporates descriptive characteristics of the object from all of the views.
6. Experimental Results

A. Apparatus

Testing for two-dimensional object recognition was conducted using a monocular vision platform consisting of one high-resolution CCD camera mounted on a 2D positioning rig. The platform is shown in Figure 8.

Testing for three-dimensional object recognition was conducted using a vision platform consisting of two high-resolution CCD cameras, mounted on a robotic arm and calibrated for stereo triangulation. No particular constraints were applied to camera or object positioning other than generally placing the objects reasonably within the field of view of the system. The platform is shown in Figure 9.

B. Computing Invariant Values

In a practical system, conditions may not be ideal for generating proper invariant descriptors without some prior processing of the images or depth maps. Since we want to recognize objects under different transformations (or from different viewpoints), it must also be assumed that the objects might be found in different places in the field of view of the system, and with a background scene present this has a serious effect on the resultant images or depth maps, and by extension, on the invariant descriptors.

Fortunately, given a static background, it is a relatively simple task to compare each pixel to a stored image of the background itself and segment out everything but the object. Many methods exist in the computer vision and image processing literature, some more complex than others; we have employed a simple thresholding technique, with experimentally-tuned parameters $t$, $F$, and $B$, outlined below:

1. For each pixel $p_{ij}$ and stored background pixel $s_{ij}$, if $|p_{ij} - s_{ij}| > t$, mark as foreground.
2. Mark as background all foreground pixels in regions with contiguous area less than $F$.
3. Mark as foreground all background pixels in regions with contiguous area less than $B$.

The descriptors we use for recognition are invariant to translation, among other things, so once background subtraction has been performed it is of no concern where in the image the object lies, so long as it is fully within the image.

C. Results for 2D Recognition

Two-dimensional object recognition was tested using four object classes derived from a foam barrier part being inspected. The objective is to identify whether the barrier has been squeezed, stretched, or rotated, as shown in Figure 10.

After training, 40 test objects (10 of each class) are tested. Figure 11 shows the classification results for each
of these objects. All 40 objects are correctly identified to the appropriate classes.

![Figure 11. 2D Classification Results.](image)

**D. Results for 3D Recognition**

Three-dimensional object recognition was tested using the training set of Figure 12 on a set of 200 depth maps taken from different viewpoints of 3 different objects.

![Figure 12. Training Set of Range Images.](image)

The recognition rates of the experiment using Gaussian fuzzification, three training views, and the simple crisp-value inference method are shown in Table 1.

![Table 1. Recognition Results.](image)

<table>
<thead>
<tr>
<th>Test</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>94.00%</td>
<td>93.81%</td>
<td>86.67%</td>
</tr>
<tr>
<td>B</td>
<td>98.00%</td>
<td>98.97%</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

Test A used no data scaling whereas Test B employed the LVQ self-scaling method. A very high recognition rate was achieved in all three classes, despite noise in the generated depth maps and ambiguity in the shapes of the objects.

**7. Conclusions**

After examining a variety of possible invariant descriptors for recognition of two-dimensional and three-dimensional objects, we have found particular combinations which yield the most salient classification data and thus the best recognition results. In the case of two-dimensional shapes, we identify compactness and the second-order image moment. In the case of three-dimensional range images, we identify compactness, the first Hu moment, and the histogram difference.

The recognition method used a neural network to optimize fuzzy membership functions for the invariant descriptors against one another, which successfully mitigated misclassification introduced by ambiguities in the individual functions. After training the recognition system with just a few views of an object, as described in Section 5, a very high recognition rate was achieved on images and depth maps generated from arbitrary views.

**Acknowledgment**

This research was funded in part by the Natural Sciences and Engineering Research Council of Canada.

**References**


Aaron Mavrinac received the B.A.Sc. and M.A.Sc. degrees in Electrical and Computer Engineering in 2005 and 2008, respectively, from the University of Windsor. He is currently working toward the Ph.D. degree there as well. His research interests include 3D computer vision, camera networks, camera calibration, vision-based control, and robotics.

Xiang Chen received his M. Sc. and Ph. D. degrees in 1996 and 1998 from Louisiana State University, all in System and Control. He is currently a Professor in Department of Electrical and Computer Engineering, University of Windsor, Canada. His research interests include: robust control, control of complex systems, vision sensor network and vision based control, industrial applications of control theory.