

# GROUPING OBJECTS BASED ON THEIR APPEARANCE

Altamirano Robles Luis Carlos<sup>1</sup>, Zacarías Flores Fernando<sup>2</sup>, De León Gómez Oscar Antonio<sup>3</sup> and Sánchez López Abraham<sup>4</sup>

<sup>1,2,3,4</sup>Computer Science, Autonomous University of Puebla (BUAP), Puebla, México

fzflores@yahoo.com.mx, altamirano@cs.buap.mx, oleongomez@gmail.com  
asanchez@cs.buap.mx

## ABSTRACT

*The use of clustering algorithms for partition to establish a hierarchical structure in a library of object models based on appearance is deployed. The main contribution corresponds to a novel and intuitive algorithm for clustering of models based on their appearance, closer to “human behavior”. This divides the complete set into subclasses. Immediately, divides each of these in a number of predefined groups to complete the levels of hierarchy that the user wants. Whose main purpose is to obtain a competitive classification compared to what a human would perform.*

## KEYWORDS

*Vision, Appearance, Models, Clustering and Hierarchy*

## 1. INTRODUCTION

At the beginning of object recognition through digital images, was used the well-known geometric paradigm. This was mainly to the wide availability of algorithms for detecting edges and corners. Despite the good results obtained, is shown that using only the way to recognize objects has its limitations. For instance, if we have two equal cubes in shape and size but one red and the other blue, all systems based on geometric characteristics will say that it is the same object because both objects have the same shape. This happens because there is a substantial information loss, inherent to the object in question.

Other side, appearance-based approach has been proposed as an alternative to geometrical approaches. However, this approach also has advantages (increases recognition rates, principally) and disadvantages (depends on illumination conditions, pose and other parameters, y of course, requires a lot of images in order to build models). But, since appearance models offer a promises alternative to traditional models, they are been studying widely [10, 13, 14].

Because it is not easy to establish a hierarchy among models learnt by the machine, most authors prefer to works in different directions, such that: improve recognition rate, improve lost of information, compression methods (PCA, for example); one exception are works made by Nelson and others [11, 12]. However, they use just contour of objects for establish any kind of hierarchy no clear and depends of primitives previously selected.

In this paper, we present a novel and intuitive approach to establish a hierarchy which does not depend of any primitive or uses just contour of objects. Instead, we use all visual information about object appearance: of course contour, but also, color, texture, pose, etc. Our approach is intuitive because is closer to “human behavior”: our proposal uses a not supervised approach in

order to build several different clusters and each one has an explanation in terms of human behavior. They establish a hierarchy beginning with objects and finalizing with 1, 2, or the number of trees that we give a priori, like we want to divide a group between men and women: we establish the only parameter for our algorithm equal to 2 and we can obtain this grouping with 2 classes, but which classes? Answer: several different 2 classes, one pair for each possible duality: men/women, black/white, short/tall, etc., like human does.

So, we developed a more natural object classifier that considers the following aspects:

- Appropriate ways to differentiate between objects modeled by sets of digital images (measures of similarity/dissimilarity).
- Clustering algorithms able to put objects models in class,
- Classification based not only on points, as the traditional algorithms.

Different experiments (shown below) shows that our approach exhibits results close to “human results”, in the sense above explained.

## 2. PRELIMINARIES

The following subsections present basic concepts about processing objects.

### 2.1. Models based on appearance

The object appearance is given by: the combination of shape, its reflectance properties, and its position in front the camera in scene and the illumination present in this [6]. Should be noted that the first two are inherent to the object, however, in the third, the object’s position in front of the sensor and illumination may vary from scene to scene[6]. Likewise, is clear that this definition implies that the appearance of an object at a particular time can be derived from a digital image in which that element is captured in a specific scene. Similarly, then may be used a set of images to generate a model of the object under different circumstances. In figure 1, you can see a graphical representation of an object's appearance[6].

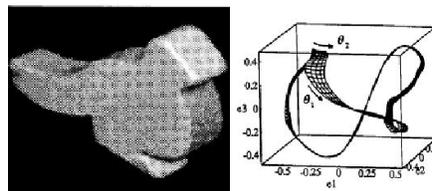


Figure 1. An object and its model

Usually, appearance-based approach works as follows: are taken a number of images of the object you want to build the model, an image for each of the possible ways in which the object is intended that may appear in an image and combined with the various lighting conditions that impinge upon it. Next, these images are stored in a format chosen (gif, tiff, bmp, etc.), for later use; these images themselves can form the model of the object or, they can extract some features that you think describe the object accurately, thus, these constitute the model. Finally, the recognition process itself, is to compare the model with a new image that is likely to contain the object (of course, this latter process requires some extra processing on the new image, such as object segmentation of interest in the image, interpolating between images, etc.).

### 2.2. Creating models using Hotelling transform

Due to the need for a large number of images to a single object model and in practice it is common to have large quantities of different object, we need to find a compressed representation

of all the images. Normally, It is possible to use a common technique in pattern recognition named “Hotelling transform” or commonly known as principal component analysis [6].



Figure1. Eigen-image and a reconstructed image

The general procedure is as follows: [6], [4]:

1. Each image is stacked by columns
2. Each image is normalized by dividing by their respective norm

$$i'_j = \frac{i_j}{\|i_j\|}$$

3. We obtain the average vector of all images

$$i_m = \frac{\sum_{j=1}^r i'_j}{r}$$

4. Each image is subtracted from the average
5. Matrix P is obtained, where each column is one of the vectors previously calculated
6. We obtain the covariance matrix implied.

$$\bar{Q} = P^T * P$$

7. Are obtained the eigenvalues and eigenvectors of which are obtained corresponding to the covariance matrix,

$$\lambda_i = \bar{\lambda}_i$$

$$e_i = \bar{\lambda}_i^{-1/2} P e_i$$

Next, we obtain the set of eigenvectors or eigenimages, these vectors and eigenvalues form a low-dimensional eigenspace. The dimension of the eigenspace corresponding to the number of principal components you want to use. 20 are normally sufficient as maximum. We consider that this should include all the images of all objects. Immediately, we solve the following system of equations

$$eA = P$$

We obtain the matrix A containing necessary coefficients to reconstruct any image of any pattern. This compression process shows insignificant losses of information for problem that concerns us. In figure 3 the result of the manifolds extraction is shown for two objects and, in figure 2 some of the eigenimages can be observed obtained by means of this method.

### 2.3. Similarity measures

Typically when you have a set of models and a test object but with different views that contain some overlap in these, global data can differ greatly the strategy is to find the common view and measure the similarity of those parts of the data. Then, we define the distance between manifolds as follows:

$$d(M_1, M_2) = \min d(C_i, M_2) \quad (1)$$

where  $i = 1 \dots r$ , is the number of patches that make up the first manifold. Thus, the distance between manifolds is calculated as the best patch similarity adjusted.

For this calculation it is necessary to obtain other measures presented below.

### 2.3.1. Measure distance between subspaces

The measure of distance between subspaces is denoted by  $d(S_1, S_2)$  and currently there is not a unified definition. In this paper we used the concept of principal angles for good performance [1]. With these concepts are defined the following measures distance. Main idea is that the nonlinear manifolds are composed of linear subspaces and the distance of these components defines the similarity between two manifolds. Formally denoted as:  $M = C_1, C_2, \dots, C_m$  where  $M$  is a manifold consisting of linear subspaces.

Once partitioned manifolds, it is only necessary to calculate the distance among subspaces to proceed. A way to make it is to calculate the main angles that reflect the variations among the subspaces without taking the data into account [1].



Figure 3. Examples calculated

Other methods take into account the similarity through the averages of each local pattern in a similarity measure among exemplary. It is easy to understand that to use the coalition of both measures is better for the problem than we want to solve.

### 2.3.2. Measure distance between subspaces

The distance from a subspace to a manifold is denoted as  $d(S, M)$  and is defined in the following way

$$d(S, M) = \min_{C_i \in M} d(S, C_i)$$

This means to find the minimum distance between a subspace and the lineal components of manifold.

### 2.3.3. Distance from a manifold to a manifold

The distance of a manifold to a manifold is denoted as  $d(M_1, M_2)$  and is defined in the following way:

$$\begin{aligned} d(M_1, M_2) &= \min_{C_i \in M_1} d(C_i, M_2) \\ &= \min_{C_i \in M_1} \min_{C'_j \in M_2} d(C_i, C'_j) \end{aligned}$$

Clearly, the similarity between  $M_1$  and  $M_2$  is calculated as similarity of local models that better they adapt. In [Wang2008] this distance is used in recognition of faces through sets of images. In this paper we will use it to get grouping of models of objects based apparently.

## 2.4. Distance among manifolds

As it was already defined previously, Matrix A contains manifolds of the objects in the library. These define a trajectory parameterized in  $R^k$  where k is the number of selected main components. Main problem of grouping models of objects based on appearance is that it should be treated with this compact representation. It is essential to find a measure or set of measures of similarity are necessary to make a classification of objects similar to what a human would. In the literature it is common to find similarity measures between objects represented as a vector (column arrow) with  $d$  characteristics, stored in database of N objects (dataset) [2]. The strength of the relationship between two points of a dataset is indicated by a *similarity coefficient*, the larger this coefficient, said objects are related more.

Next, we define the distance between manifolds as a measure of similarity between models based on appearance.

### 2.4.1. Maximal Lineal Patch

After obtaining the Manifolds, as mentioned in previous section, is necessary that each manifold is divided into a series of subspaces called maximal lineal patches. The concept of maximal lineal patche (MLP) is defined as a set of points defining a maximal linear subspace, where the linear perturbation of these points is given naturally as the deflection between the Euclidean distance and geodesic distance.

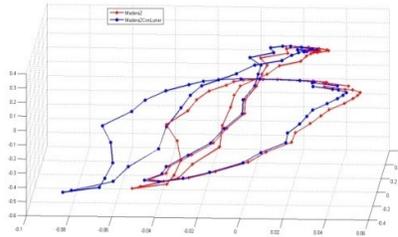


Figure 4. Examples of Manifolds obtained from COIL-20

The main idea is to split the data set into several MLPs such that sets of images containing very similar to each other. These patches are calculated from a random point of dataset until the condition of linearity is lost. This procedure guarantees the local linearity of models and adaptively controls the number of models calculated. Formally, [7] be  $X = x_1, \dots, x_N$  a data set, in our case are points that make up manifolds, where  $x_i \in R^D$  and  $N$  is the number of points in  $X$ . we divide  $X$  into a disjoint collection of MLPs  $C_i$  such that:

$$X = \bigcup_{i=1}^m C_i,$$

$$C_i \cap C_j = \emptyset (i \neq j, i, j = 1, 2, \dots, m)$$

$$C_i = x_1^{(i)}, x_2^{(i)}, \dots, x_{N_i}^{(i)} \left( \sum_{i=1}^m N_i = N \right) n$$

where  $m$  is the number of patches and  $N_i$  is the number of points in the patch.

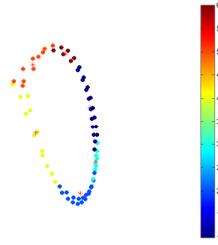


Figure 5. Partitioning of a model based on appearance

For partitioning is necessary to calculate the Euclidean distance matrix  $D_E$  and the matrix of geodesic distances  $D_G$ . Calculation of the MLPs is used as an intermediate step to find the distance between two manifolds. This distance has been used in face recognition through image sets [7]. As mentioned above, the idea is to split the set of images in several patches which are embedded in a linear subspace. The following shows the algorithm developed.

**Initially**

$i = 1, C_i = \emptyset, X_T = \emptyset, X_T = \emptyset, X_R = X$ .

1. **While**  $X_T \neq \emptyset$
2. Randomly select a point in  $X_R$  as seed  $x_1^{(i)}$ ,
  - 2.1.  $C_i = \{x_1^{(i)}\}, X_R = X_R - \{x_1^{(i)}\}$
  - 2.2. **For**  $\forall x_m^i \in C_i$ 
    - 2.2.1. Identify each of k nearest neighbors  $x_c$  as candidate
    - 2.2.2. **If**  $x_c \in X_R$  and  $(D_G(x_c, x_n^{(i)}) / D_E(x_c, x_n^{(i)})) < \theta$
    - 2.2.3.  $C_i = C_i \cup \{x_c\}$
    - 2.2.4.  $X_R = X_R - \{x_c\}$
- EndIf**
- EndFor** 3  $X_T = \bigcup_{j=1}^i C_j, X_R = X - X_T; i \leftarrow i + 1, C_i = \emptyset$

**EndWhile**

After calculating a linear patch, its average is calculated. This is taken as a representative of set and summarizes information of selected patch that is formed by numerically similar images. The importance of these representatives is reflected in the reduction of the number of images required to represent the object appearance and is taken into account for the calculation of the similarity of a complete model with other.

**2.4.2. Measure distance between subspaces by variations**

For two subspaces  $C_i$  and  $C_j$  we define “a distance measure by variations” as their average canonical correlations as follows:

$$d_V(C_i, C_j) = r / \sum_{k=1}^r \sigma_k$$

Where  $\sigma_1 \dots \sigma_r$  are the canonical correlations and are obtained by the method described in [1] and  $r$  is the minimum sizes of linear subspaces.

### 2.4.3. Measure distance between copies

To complement the above measure is defined as measured between copies of the duplicate correlation  $e_i$  and  $e_j$ :

$$d_E(e_i, e_j) = \frac{\begin{matrix} \vdots & e_i & \vdots & \dots & \vdots & e_j & \vdots \\ & & & & & & \end{matrix}}{e_i^T e_j}$$

### 2.4.4. Similarity between models based on appearance

Finally, the distance between subspaces is obtained as the weighted average of the two previous measures as follows:

$$d(C_i, C_j) = (1-\alpha)d_E(C_i, C_j) + \alpha d_V(C_i, C_j)$$

When comparing two sets of images the first shows how similar the appearance of objects, while the other is so close that common variation trends of the two sets. Thus, we can easily define the distance between manifolds as in equation (1).

## 2.5. Clustering algorithms

The K-means algorithm is one of the most used due to its simplicity and speed. The core idea is to group the data repeatedly around the centroids calculated from an initial allocation of the data in a fixed number of  $k$  groups. The class membership changes according to the evaluation of an function error until this error does not change significantly or data belonging either. The error function is defined as follows [5]:

Let  $D$  be a data set with  $n$  instances, and be  $C_1, C_2, \dots, C_k$   $k$  cluster disjoint from  $D$ . The error function is calculated as:

$$E = \sum_{i=1}^k \sum_{x \in C_i} d(x, \mu(C_i))$$

where  $\mu(C_i)$  is the centroid of cluster  $C_i$  and  $d(.,.)$  is the Euclidean distance typically.

The Keywords section begins with the word, “Keywords” in 13 pt. Times New Roman, bold italics, “Small Caps” font with a 6pt. spacing following. There may be up to five keywords (or short phrases) separated by commas and six spaces, in 10 pt. Times New Roman italics. An 18 pt. line spacing follows.

### 2.5.1. Spectral clustering

Graph theory can be used to partition data. In these techniques we use a measure of similarity between the data to form a weighted adjacency matrix. Intuitively, we want find a partition of the graph where the edges of the nodes that are in different groups have low weight and have high weights when they are in the same group [3], [8].

Let  $G = (V, E)$  be an undirected graph with a set of vertices  $V = v_1, \dots, v_n$ . We assume that each edge has an associated positive weight. The heavy adjacency matrix  $W = w_{ij}, i, j = 1, \dots, n$ , where if  $w_{ij} = 0$ , vertices  $i$  and  $j$  are not connected. Since  $G$  is not addressed must be  $w_{ij} = w_{ji}$ . Furthermore, *degree of vertex* is defined as follow:

$$d_i = \sum_{j=1}^n w_{ij}$$

The diagonal matrix having the degrees of the vertices is called *degree matrix* and is denoted as  $D$ . Likewise define heavy adjacency matrix and degree matrix. Next, we define the un-normalized Laplacian matrix of a graph as:

$$L = D - W$$

This matrix has several interesting properties and is positive semi-definite and the number of eigenvectors of this matrix is directly related to the number of connected components of the graph [3]. Besides, this method is closely related to the method kmeans kernelizado and cuts of graphs normalized [9].

### 2.5.2. Normalized Laplacian Matrices

In the literature there are two standard Laplacian matrices. Each one is closely related to each other and is defined as:

$$L_{sym} = D^{1/2} L D^{1/2}$$

$$L_{rw} = D^{-1} L$$

Matrix  $L_{sym}$  is denoted so because it is symmetrical and matrix  $L_{rw}$  is intimately related to random walks.

As in the case of the non-normalized matrix, the multiplicity of the first Laplacian eigenvalue of the matrix is related to the number of connected components of the graph of heavy adjacencies [8]. In this study we used the non-standard matrix and matrix  $L_{sym}$  for the calculation of the groupings.

## 3. Our proposal

The intention of using different clustering methods for partitioning is to achieve an initial division of the set of models and then apply the same algorithm again, dividing the classes obtained in the previous step in another hierarchical level. Besides using the similarity measure manifolds also be transformed so that they can be manipulated as a single object and not as a set of points. This was achieved by calculating the centroid of each manifold and calculating the standard deviation. This defines a sphere that contains almost all points of the manifold. This defines then a sphere that contains almost all points of the manifold.

Immediately show the method generates a hierarchy of a set of object models.

The proposed method can be summarized as follows:

1. It creates an initial partition of the original set.
2. Each class obtained is divided k into groups, the parameter k is user defined for each level of the hierarchy
3. Results are stored
4. Repeat from step 2 to complete the required levels

Finally, we obtained a tree structure that shows how have been divided every class obtained, in each of the iterations calculated. The rates of each object displayed in the tree nodes.

## 4. TEST AND RESULTS

Tests were conducted on the classic-looking set of models supplied by Columbia University, **COIL-20**. The results obtained were compared with those validated by a user visually, this in

order to check that the classification obtained is similar or not. The results obtained are presented in a tree structure where nodes tags are interpreted according to the following scheme:

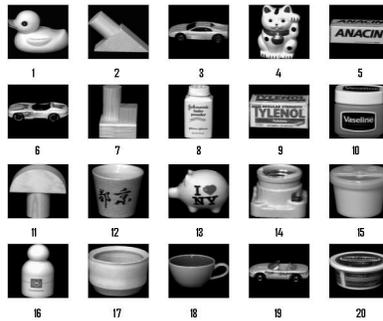


Figure 6. scheme of object index

The system was executed several times for each of the algorithms and with different numbers of levels and Partitions by level, found empirically that in the particular case of the library used, three levels and no more than three partitions per group are sufficient to achieve good rankings. In figure 7, you can see one of the groups calculated by the algorithm. Note that the classification of objects 4.8 and 16 places in the same class and this will not change using other methods proposed, change only the objects that are grouped into higher levels of the hierarchy This is because the calculation of the distance between manifolds gives greater weight to objects appearance so those in which the dominant gray tone is very similar are grouped in the same class as evidenced in the case of the wood pieces. This measure can also reflect some special characteristics as how objects in the case of figure 7 the majority shares (objects are higher than wide, almost the same gray and have almost the same kind of symmetry) which may be an indication that the results can be improved somewhat by mixing these results to those obtained using only borders.

Make a change of models representation calculating the centroids allows us to use unmodified existing algorithms but at the same time preserves all the innate drawbacks of these same. To avoid the particular algorithms derived using k-means, we will use the spectral clustering on the centroids.

#### 4.1. Spectral clustering by manifolds centroids

Below are shown some of the results obtained in testing algorithm. For this particular test we used three levels and a vector  $v = (3, 3, 2)$ .



Figure 7. One group obtained with the proposal.

## 5. CONCLUSIONS AND FUTURE WORK

Although algorithm is the type of divide and conquer, its complexity is polynomial since the algorithms used (kmeans, spectral) assure us that the groups formed are very similar to each other and very different from the other groups thereby avoiding combinatorial check what is the best partition. However, when an object has been misclassified, is impossible to assign another class and in the best case can be segregated from the group to the next level calculated. On the other hand, using a measure of similarity between objects, results were very similar to those obtained by other methods since the model library allows. In addition, it was found that the system is able to match the results obtained by the user and also is able to detect similarities not evident at a glance. In figure 8 we show some results obtained with our proposal.

The results have demonstrated the usefulness of similarity measures alternative to classic and the data processing as a means to obtain better results. However, much work remains to be done in this field in which this research aims to pioneer. Some things to incorporate into future research are:

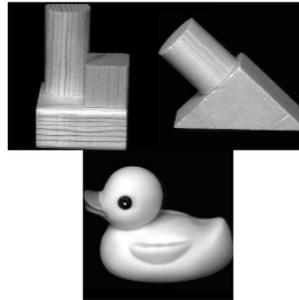


Figure 8. An unexpected group?.

- **Use models based on appearance only for objects borders.** With the aim of using as a complementary method.
- **Select different algorithms to apply to each level.** The results may be improved if different algorithms are used at each level of the hierarchy.
- **Testing with other models libraries and compare results.** It is essential to experiment with different libraries to delimit the scope of the proposal developed in this work and and modify it to make it applicable to more general problems.
- **Automatically analyze the results.** To quantitatively compare the classifications obtained by the system with those made by a human.
- **Improving the representation of hierarchical structure.**
  - Explore the possibility of calculating the tree dendrogram.
  - Investigate possible applications of this proposal.

All these points can be taken as a basis for further work in order to find new applications, detect problems and generate other proposals that serve to extend knowledge currently has in the areas of computer vision and clustering algorithms.

## ACKNOWLEDGEMENTS

The authors would like to thank to Autonomous University of Puebla for their financial support. Óscar De León Gómez would like to thank to CONACyT by the scholarship No. 368458.

## REFERENCES

- [1] Å. Björck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27:579594, 1973.

- [2] Clustering. IEEE Press, 2009.
- [3] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.
- [4] Taylor C.J. Cootes, T.F. Statistical model of appearance for computer vision. Technical report, Imaging Science Biomedical Engineering, University of Manchester, March, 2004.
- [5] Jianhong Wu Guojun Gan, Chaoqun Ma. *Data Clustering: Theory, Algorithms, and applications*. ASA-SIAM Series on Statistics and Applied Probability, SIAM, 2007.
- [6] Nene, Nayar, Murase. Parametric appearance representation. Nayar, S. K., Poggio, T., (Eds.) *Early Visual Learning*, Oxford University Press. New York, 1:131–160, 1996.
- [7] Xilin Chen WenGao Ruiping Wang, Shiguang Shan. Manifold-manifold distance with application to face recognition based on image set. *IEEE*, 2008.
- [8] Francis R. Bach y Michael I. Jordan. Learning spectral clustering. Technical Report UCB/CSD-03-1249, EECS Department, University of California, Berkeley, Jun 2003.
- [9] Inderjit S. Dhillon. Kernel k-means, spectral clustering and normalized cuts. In pp. 551–556. ACM Press, 2004.
- [10] Mundy, J., Liu, A., Pillow, N., et. al., “An experimental comparison of appearance and geometric model based recognition”, *Object Representation in Computer Vision II*, in Proceedings of ECCV96 International Workshop, Cambridge U.K., pp. 247-269. April, 1996.
- [11] Andrea Selinger and Randal C. Nelson, “A Perceptual Grouping Hierarchy for Appearance-Based 3D Object Recognition”, *Computer Vision and Image Understanding*, vol. 76, no. 1, October 1999, pp.83-92.
- [12] Randal C. Nelson and Andrea Selinger “A Cubist Approach to Object Recognition”, *International Conference on Computer Vision (ICCV98)*, Bombay, India, January 1998, 614-621.
- [13] Zaman Khan and Adnan Ibraheem. Hand Gesture recognition: A literature review. *International journal of artificial Intelligence & Applications*, Vol 3, No. 4, July 2012.
- [14] Bimi Jain, M.K. Gupta and Jyoti Bharti. Efficient iris recognition algorithm using method of moment. *International journal of artificial Intelligence & Applications*, Vol 3, No. 5, September 2012.

### Authors Biography

Dr. Fernando Zacarias is full time professor in the Computer Science Department at Autonomous University of Puebla. He has directed and participated in research projects and development in this area from 1995, with results that they have been reported in more than 50 publications of international level. And one of the most important is “Answer set programming and applications”. Conacyt project, Reference number: 37837-A. He has been a member of several international committees such as: The IEEE Latin America Transaction, IAENG – International journal of Computer Science, International conference on Advances in Mobile Computing and Multimedia, etc.



Dr. Luis Carlos Altamirano received his Ph.D. degree at National Polytechnic Institute (México) in 2002. His interest areas include: Computer Vision, Image Processing, Remote Sensing and Artificial Intelligence. He has participated in several different projects including: Oil projects and Computer Vision Projects. Currently, he is full time professor in the Computer Science Department at Autonomous University of Puebla and head of Postgraduate Department at the same Institution.



Dr. Abraham Sánchez López is full time professor in the Computer Science Department at Autonomous University of Puebla. He was member of SNI (National System of Researchers) from 2003 until 2012 (Level 1). His main areas include: Artificial Intelligence, Robot planning and simulation. He received his Ph. D. degree in 2003.

M.C. Óscar Antonio De León Gomez received his Master’s degree at Computer Science Department at Autonomous University of Puebla (2012). He has been developing several different projects on Computer Vision at INAOE (National Institute of Astrophysics, Optical and Electronics), where currently he is working