Object Classification Based on Associative Memories and Midpoint Operator

Roberto A. Vázquez, Humberto Sossa and Ricardo Barrón
Centro de Investigación en Computación – IPN
Av. Juan de Dios Batíz, esquina con Miguel Otón de Mendizábal
Ciudad de México, 07738, México.
Contact: robertov@sagitario.cic.ipn.mx , hsossa@cic.ipn.mx, rbarron@cic.ipn.mx

Abstract. In this paper we describe a way to build an associative memory for object classification. The operation of the new architecture is based on the functioning of the well-know mid-point operator widely used in signal processing. The proposal is an alternative to the one described in [H. Sossa, R. Barrón, R. A. Vázquez. Real-Valued Pattern Classification based on Extended Associative Memory. In Proc. Fifth Mexican Conference on Computer Science (ENC2004), 213-219 (2004). The proposal is tested with image of realistic objects.

1 Introduction

One important problem in computer vision is object classification. The solution to this problem would strongly influence the functionality of many systems such us: content-based image retrieval systems, video indexing systems, automatic robot guidance systems, object tracking systems, object manipulation systems, and so on. Many approaches to solve this problem have been proposed in the literature: the well-known statistical approach, the structural approach and the neural approach. The idea of using associative memories to solve the object classification problem is relative new. Refer for example to [1-7].

In this paper we describe an associative model by which we can get the class index of an object given a description of it terms of some its features. We propose a new way to build an associative memory combining well-known set operations of min and max and midpoint operator well-used in signal processing. We show several examples with numerical a real patterns where the effectiveness of the proposal is tested.

Rest of paper is organized as follows. In section 2, the proposal is described in detail. In section 3, a numerical example to better follow the functioning of the proposal is given. In section 4, experimental results with images of realistic objects are provides, while in section 5, conclusions and directions for further research are given.

© A. Gelbukh, R. Monroy. (Eds.)
Advances in Artificial Intelligence Theory
Research on Computing Science 16, 2005, pp. 131-140

2 The Proposal

Let $\left(x^{\xi}, i\right)_{\xi=1}^{p}, x^{\xi} \in \Re^{n}, i = 1, \ldots, m$ a set of p fundamental couples (SFC), composed by a pattern and its corresponding class-index. The problem is to build an operator **M**, using this SFC, that allows to classify the patterns into their classes, i.e. $\mathrm{M} \otimes x^{\xi} = i$ for ξ=1,…, p and that even in the presence of distortions it classifies them adequately, i.e. $\mathrm{M} \otimes \tilde{x}^{\xi} = i$, where $\tilde{x}^{\xi}$ is an altered version of x^{ξ}. A first approach in this direction was presented in [7]. Operator $\otimes$ is chosen such that when operating vector x^{ξ} with matrix **M**, produces as result the corresponding index class of pattern x^{ξ}.

Matrix **M** is build in terms of a function ϕ as follows:

$$\mathbf{M} = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_m \end{bmatrix} \tag{1}$$

where each ϕ_i represents the i-th row of a matrix **M** and this function is a codification of all patterns belonging to the class i.

Function ϕ can take several forms. In this paper we propose well-known mid-point to build function ϕ.

2.1 Mid-point operator

Arithmetic averaging is widely used in pattern recognition to perform Euclidean distance classification. Arithmetic averaging allows obtaining representative vectors of different patterns to be classified. Representative vectors computed this way are useful only when standard deviation of all patterns belonging to the class i, is low.

Mid-point operator usually used in signal filtering could be an alternative to get also representative vectors for each class. We preferred to use mid-point operator because as we will next see it allows better classification results than arithmetic average operator and other known operators.

Mid-point operation works as follows: Given a set of p values: $f_1 \le f_2 \le \cdots \le f_p$:

$$f_{mid} = \frac{f_1 + f_p}{2}. \tag{2}$$

With respect to average operator, mid-point value f_{mid} is always between values f_1 and f_p, while f_{ave}'s position will depend on the distribution of values: $f_1,\cdots f_p$, defined as $f_{ave}=\frac{1}{m}\sum_{i=1}^{m} f_i$.

For the case of vectors, mid-point operator takes the form:

$$\phi_i^j=\frac{\gamma_i^j+\lambda_i^j}{2} \tag{3}$$

where

$$\gamma_i^j=\overset{p}{\underset{\xi=1}{\vee}}\left(x_i^{\xi,j}\right) \tag{4}$$

and

$$\lambda_i^j=\overset{p}{\underset{\xi=1}{\wedge}}\left(x_i^{\xi,j}\right) \tag{5}$$

i stands for the object's class and j goes from 0 to n, the size of the pattern. As you can appreciate, the idea is to build a hyper-box enclosing patterns belonging to class *i*, by means of max "$\vee$" and min "$\wedge$" set operators.

Example 1. Suppose we want to build matrix $\mathbf{M}_{mid}$ from the following set of associations:

pattern	class	pattern	class	pattern	class
(1.0, 1.0, 1.0)	1	(4.0, 4.0, 4.0)	2	(10.0, 9.0, 10.0)	3
(1.0, 2.0, 1.0)	1	(4.0, 4.0, 5.0)	2	(9.0, 9.0, 10.0)	3
(2.0, 1.0, 1.0)	1	(4.0, 5.0, 5.0)	2	(10.0, 10.0, 10.0)	3
(1.0, 1.0, 2.0)	1	(5.0, 4.0, 4.0)	2	(10.0, 11.0, 11.0)	3
(2.0, 2.0, 2.0)	1	(5.0, 4.0, 5.0)	2	(10.0, 9.0, 11.0)	3

According to equations (4) and (5): $\gamma_1=(2,2,2)$, $\gamma_2=(5,5,5)$ and $\gamma_3=(10,11,11)$. Also, $\lambda_1=(1,1,1)$, $\lambda_2=(4,4,4)$ and $\lambda_3=(9,9,10)$. Thus $\phi_1=(1.5,1.5,1.5)$, $\phi_2=(4.5,4.5,4.5)$ and $\phi_3=(9.5,10.0,10.5)$. Finally:

$$\mathbf{M}_{mid} = \begin{bmatrix} 1.5 & 1.5 & 1.5 \\ 4.5 & 4.5 & 4.5 \\ 9.5 & 10.0 & 10.5 \end{bmatrix}.$$

An advantage of mid-point operator over other operators to build matrix **M** is that the distance of representative to farthest class elements is always the same as can be appreciated in Figure 1 (a). For other operators such the well-known arithmetic average operator, representative vector is not always at the center (Figure 1(b)).

Other adavantages of mid-point operator over other operators are the following:

1. It is less expensive to compute a **min** an a **max** than adding up all vectors associatedd to a class as for example with arithmetic average operator.
2. It is less expensive to compute a **min** an a **max** than to order a vector for the case of median operator.
3. It is less expensive to compute a **min** an a **max** than to compute inverse matrices and probabilities as for example with Bayesssian classifier.

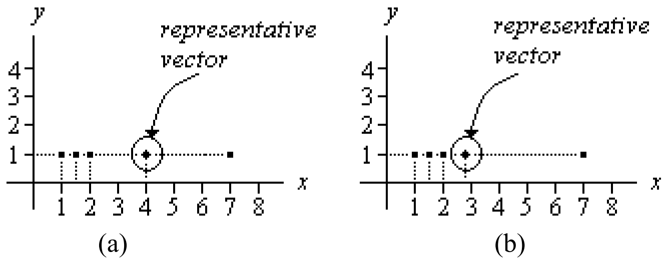


Figure 1. (a) Representative pattern obtained by means of mid-point operator is always at the center between farthest class elements. (b) Representative vector obtained by means of arithmetic average operator is not always at central position. This depends on distribution values of class elements.

2.2 Pattern classification

Pattern classification is performed as follows. Given a pattern $x^{\xi} \in \Re^n$, not necessarily one of the already used to build matrix $\mathbf{M}_{mid}$, class to which x is assigned is given by

$$i = \mathrm{M} \otimes x^{\xi} = \arg_l \left[\bigwedge_{l=1}^{m} \bigvee_{j=1}^{n} \left| m_{lj} - x_j \right| \right] \quad \textbf{(6)}$$

Operators $\vee \equiv \max$ and $\wedge \equiv \min$ execute morphological operations on the difference of the absolute values of the element m_{lj} of $\mathbf{M}_{mid}$ and the components x_j of pattern x^{ξ} to be classified. Thus $\bigvee_{j=1}^{n} \left| m_{lj} - x_j \right|$ is the metric of the max between row *l* of $\mathbf{M}_{mid}$ and pattern x^{ξ}, thus it can be written as $d(x, m_l) \equiv \bigvee_{j=1}^{n} \left| m_{lj} - x_j \right|$, m_l row of $\mathbf{M}_{mid}$.

From the point of view of this metric, pattern classification consists on assigning pattern x^{ξ} to the class which index of row of $\mathbf{M}_{mid}$ is the nearest.

Conditions for correct recall of either a pattern of the FS or from an altered version of one its patterns are given as:

Theorem 1 [7]. Let $\mathrm{d_i} = \bigvee_{\mathrm{x\ class\ i}} \mathrm{d}(\mathbf{x}, \phi_{\mathrm{i}})$ and $\mathrm{R_i} = \left\{ \mathbf{x} : \mathrm{d}(\mathbf{x}, \phi_{\mathrm{i}}) \le \mathrm{d_i} \right\}$ hyper-boxes centered at ϕ_i and semi-side $d_i, i = 1,...,m$. If $d(\phi_i, \phi_j) > 2\max\{d_i, d_j\}$, then:

i) $R_i \cap R_j = \varnothing, 1 \le \mathrm{i}, \mathrm{j} \le \mathrm{m}, \mathrm{i} \ne \mathrm{j}$.

ii) $\mathbf{x} \in \mathrm{R_i}$ implies $\mathrm{d}(\mathbf{x}, \phi_{\mathrm{i}}) \le \mathrm{d}(\mathbf{x}, \phi_{\mathrm{j}})$.

iii) $\mathbf{x} \in \mathrm{R_j}$ implies $\mathrm{d}(\mathbf{x}, \phi_{\mathrm{j}}) \le \mathrm{d}(\mathbf{x}, \phi_{\mathrm{i}})$.

3 Numerical Example

To better understand the idea of the functioning of the proposal, let us study the following numerical example.

3.1 Classification of a pattern belonging to the training set

From example 1, let us take pattern $(10.0, 9.0, 11.0)$ that we know it belong to class 3, and let us verify that it is correctly classified. By applying equation 6, we have:

$$l = 1 : \max\left[\left|1.5 - 10.0\right|, \left|1.5 - 9.0\right|, \left|1.5 - 11.0\right|\right] = \max\left[8.5, 7.5, 9.5\right] = 9.5$$
$$l = 2 : \max\left[\left|4.5 - 10.0\right|, \left|4.5 - 9.0\right|, \left|4.5 - 11.0\right|\right] = \max\left[5.5, 4.5, 6.5\right] = 6.5$$
$$l = 3 : \max\left[\left|9.5 - 10.0\right|, \left|10.0 - 9.0\right|, \left|10.5 - 11.0\right|\right] = \max\left[0.5, 1.0, 0.5\right] = 1.0$$

Thus $i = \arg_l \left[\overset{3}{\underset{l=1}{\wedge}} (9.5, 6.5, 1.0) \right] = \arg[1.0] = 3$.

Then the pattern $(10.0, 9.0, 11.0)$ is assigned to class 3.

3.2 Classification of a noisy pattern

From example 1, let us now take distorted version $(9.3, 10.5, 11.5)$ of pattern $(10.0, 9.0, 11.0)$ belonging to class 3. Let us verify that in presence of noise, it is assigned to class 3.

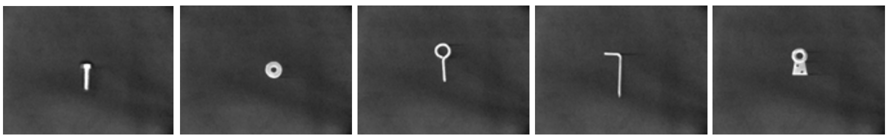
$$l = 1: \max\left[|1.5 - 9.3|, |1.5 - 10.5|, |1.5 - 11.5|\right] = \max[7.8, 9.0, 10.0] = 10.0$$
$$l = 2: \max\left[|4.5 - 9.3|, |4.5 - 10.5|, |4.5 - 11.5|\right] = \max[4.8, 6.0, 7.0] = 7.0$$
$$l = 3: \max\left[|9.5 - 9.3|, |10.0 - 10.5|, |10.5 - 11.5|\right] = \max[0.2, 0.5, 1.0] = 1.0$$

Thus $i = \arg_l \left[\overset{3}{\underset{l=1}{\wedge}} (10.0, 7.0, 1.0) \right] = \arg[1.0] = 3$.

Then the pattern $(9.3, 10.5, 11.5)$ is assigned to class 3.

4 Experimental Results

In this section, the proposal is tested with the set of realistic objects shown in Figure 2. Objects were not directly recognized by their images but instead from invariant descriptions of them. With these invariant descriptions matrix $\mathbf{M}_{mid}$ is built. Twenty images of each object in different positions, translations and scaled changes were used to get the invariant descriptions.



(a) (b) (c) (d) (e)

Figure 2. The five objects used in the experiments. (a) A bolt. (b) A washer. (c) An eyebolt. (d) A hook. (e) A dovetail.

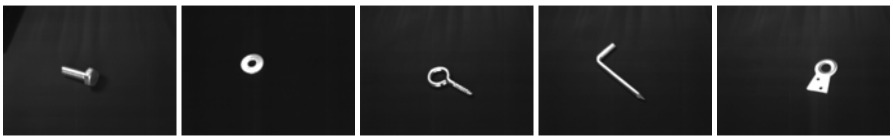
4.1 Training phase

To each image of the 20 images of each object a standard thresholder [8] was applied to get its binary version. Small spurious regions from each image were eliminated by means of standard size filter [9]. Next, to each of the 20 images of each object (class) seven well-known Hu geometric moments invariant to translations, rotations and scale changes were computed [10]. After applying methodology described in Section 2, matrix $\mathbf{M}_{mid}$ is:

$$\mathbf{M}_{mid} = \begin{bmatrix} 0.4394 & 0.1598 & 0.0071 & 0.0028 & 1.96E-5 & 0.0011 & -8.47E-6 \\ 0.1900 & 8.72E-5 & 7.47E-6 & 1.28E-14 & 7.23E-14 & -2.93E-10 & -1.6E-14 \\ 0.7092 & 0.2895 & 0.1847 & 0.0730 & 0.0088 & 0.0394 & -0.0015 \\ 1.4309 & 1.6009 & 0.7944 & 0.2097 & 0.0831 & 0.1565 & 0.0118 \\ 0.2475 & 0.0190 & 2.5E-5 & 8.66E-5 & 4.82E-9 & 1.20E-5 & -1.4E-9 \end{bmatrix}$$

4.2 Classification

Three sets of images were used to test the efficiency of proposal. A comparison with others proposals was also performed. First set of consisted of 100 images (20 for each of the five objects) different from those used to get matrix $\mathbf{M}_{mid}$. Set number two consisted on other 100 images (20 for each five objects) but projectively deformed. One image of each object is shown in Figure 3, where you can easily appreciate the deformation introduced to the objects. Finally, set number three consisted on other 100 images of five objects (20 for each object) different of those used to get matrix $\mathbf{M}_{mid}$. Figure 4 shows one image of each object. The idea is to verify to which class the object assigned by the classifier.



(a) (b) (c) (d) (e)

Figure 3. First image of each object projectively deformed to test performance of proposal. (a) A bolt. (b) A washer. (c) An eyebolt. (d) A hook. (e) A dovetail.

With the first set of images the associative memory built by means of mid-point operator provided 100% of classification. All objects were put in their corresponding class. Thus, performance of proposal was of 100%.

With second set of images, consisting of deformed objects, proposal provided also 100% of efficiency. Again, all objects were correctly sent to their corresponding classes.

Figure 4. First image of each object not belonging to the classes of objects to test proposal. (a) Wood bolt. (b) Hook with thread. (c) Open eyebolt. (d) Key. (e) Open S.

	Bolt	Washer	Eyebolt	Hook	Dovetail
Wood bolt	-	-	-	100%	-
Hook with thread	-	-	-	100%	-
Open eyebolt	-	-	45%	55%	-
Key	25%	-	-	75%	-
Open S	-	-	100%	-	-

Table 1. Percentage of classification for set number three when proposal is applied.

	$\mathbf{M}_{prom}$	$\mathbf{M}_{median}$	Euclidean	Bayesian	$\mathbf{M}_{mid}$
Bolt	100%	90%	100%	100%	100%
Washer	100%	100%	100%	100%	100%
Eyebolt	100%	100%	100%	100%	100%
Hook	100%	95%	100%	100%	100%
Dovetail	100%	75%	100%	100%	100%

Table 2. Comparative classification percentages with respect to other classification schemes when first group of objects is used.

	$\mathbf{M}_{prom}$	$\mathbf{M}_{median}$	Euclidean	Bayesian	$\mathbf{M}_{mid}$
Bolt	100%	100%	100%	100%	100%
Washer	100%	100%	100%	80%	100%
Eyebolt	100%	70%	90%	90%	100%
Hook	100%	50%	100%	100%	100%
Dovetail	100%	90%	100%	70%	100%

Table 3. Comparative classification percentages with respect to other classification schemes when second group of objects is used.

For third set of images, Table 1 summarizes the classification results. From this table you can appreciate that in general, the objects were associated to the classes of more similar objects already learned. This experiment was only performed to verify that the proposal sends unlearned objects to their most similar object classes.

Compared to other recently published approaches [7] and classical approaches (Euclidean and Bayesian approach), as can be appreciated in Tables 2 and 3, proposal offers better or competitive classification results, with the advantages already mentioned in Section 2.1.

5 Conclusions and Ongoing Research

In this paper, we have proposed a very simple way to build an associative memory based on mid-point operator. It uses **min** and **max** set operations to build memory. Proposal has been tested in different scenarios with images of real objects represented by their moment invariants. Results obtained with proposal are comparable and in some cases better than other proposals as shown in Section 4.

One thing that can be done to probably improve the performance of the proposal is to normalize the values of the invariants so that each feature has the same range of values.

One main drawback of mid-point operator is the presence of outliers in the data. This question will be faced in future works.

Nowadays, we are testing others ways to build function ϕ, especially when the values of ϕ are so close and do not satisfy Theorem 1.

Acknowledgements. This work was economically supported by CGPI-IPN under grants 20050156 and CONACYT by means of grant 46805. H. Sossa specially thanks COTEPABE-IPN, CONACYT (Dirección de Asuntos Internacionales) and DAAD (Deutscher Akademischer Austauschdienst) for the economical support granted during research stay at Friedrich-Schiller University, Jena, Germany.

References

1. K. Steinbuch (1961). Die Lernmatrix. Kibernetik, 1(1):26-45.
2. T. Kohonen (1972). Correlation matrix memories. IEEE Transactions on Computes, 21(4):353-359.
3. J. Hopfield (1982). Neural networks and physical system with emergent collective computational abilities. Proceedings of tha National Academy of Sciences, 79:2554-2558.
4. S. Bandyopadhyay and A. K. Datta (1996). A novel neural hetero-associative memory model for pattern recognition. Pattern Recognition, 29(5):789-795.
5. G. X. Ritter et al. (1998). Morphological associative memories. IEEE Transaction on Neural Networks, 9:281-293.
6. F. Galindo (1998). Matrices Asociativas. Científica, 2(8):17-21.

7. H. Sossa, R. Barrón, R. A. Vázquez (2004). Real-Valued Pattern Classification based on Extended Associative Memory. In proceedings of Fifth Mexican Conference on Computer Science (ENC2004), IEEE Computer Society 213-219.
8. N. Otsu. A threshold selection method from gray-level histograms. IEEE Transactions on SMC, 9(1):62-66 (1979).
9. R. Jain et al. Machine Vision. McGraw-Hill. 1995, pp. 47-48.
10. M. K. Hu. Visual pattern recognition by moment invariants. IRE Transactions on Information Theory, 8:179-187 (1962).