SENSOR INTEGRATION
IN ASSOCIATIVE VISUAL STRUCTURES

Fabio Ancona, Giancarlo Parodi, and Rodolfo Zunino
D.I.B.E. - Dept. of Biophysical and Electronic Engineering, University of Genova
Via all'Opera Pia 11a, 16145 Genova, Italy
Tel: +39 10 3532269; fax: +39 10 3532175
e-mail: {ancona,gian,zunino}@dibe.unige.it

ABSTRACT

The paper describes the use of associative models for integrating different sensors. Integrated associative structures are outlined and related to previous approaches; the enhanced robustness resulting from the integration of Associative Memories (AMs) and Neural Networks (NNs) is shown. Discussion then focuses on how different information sources can cooperate on associative visual recognition. Experimental results on real-image testbeds are reported, which confirm theoretical expectations.

1 INTRODUCTION

The paper describes an integrated system that combines an Associative Memory (AM) [1] with a Neural Network (NN) [2] to exploit the benefits of both approaches. AMs are used for high-dimensional data processing but exhibit the drawback of a limited storage capacity (i.e., enhancing the number of stored items increases crosstalk); NNs can overcome this limitation, thanks to their generalization properties, although training complexity and theoretical bounds often restrict the dimensionality of the application domain. The present research follows an integrated AM+NN approach by including the use of multiple sensors to enhance a system's robustness. The overall goal is image classification, that is, the system is expected to associate an unknown visual pattern to one of a set of image classes. A visual class includes a set of labelled reference images ("prototypes") which constitute a training set.

Sensor integration combines information from different sources and drives AM addressing without affecting generality. An important aspect of this mechanism is that data fusion occurs at the lowest level of data representation and no preprocessing or trasform is required a priori.

2 THE INTEGRATED VISION STRUCTURE

The integrated structure (Fig.1) covers two domains which differ in the dimensionality of processed data. The system's principle of operation is the following: 1) images generate keys for memory addressing, whereas general bit strings ("messages") convey the information associated with images; 2) The AM stores and retrieves message-key associations; 3) the NN processes memory-recalled messages to remove crosstalk. The gererality of message contents makes this approach compatible with many applications, including classification, control, etc.

The Noise-like Coding Model [3] of AM has been chosen for this research because it involves simple conditions on keys and is not subject to any constraint on stored images. In this model, memory writing and reading use convolution and correlation, respectively, between images and keys [4-6]. The module building a key from a supplied image will be dentoed as K-GEN in the following. A key is a 2-D pattern satisfying noise-like conditions [3].

Memory-stored information is represented by monodimensional bit strings to reduce data dimensionality for proper NN processing. To

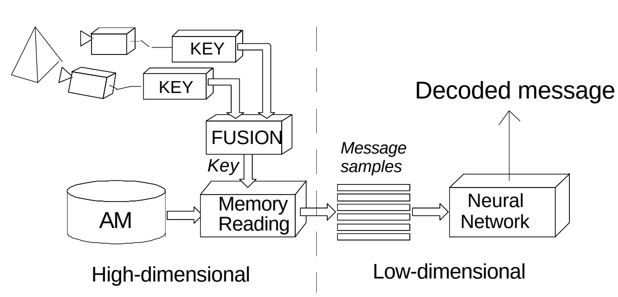

Fig.1 - The integrated associative schema

simplify models, a message is converted by a module (MAPPER) into a 2-D pattern ("Message Virtual Prototype - MVP"), whose result has the same dimensionality of keys and facilitates the AM implementation. An MVP is the result of replicating K copies of a message within a 2-D matrix [4]; its inverse process (INVERSE MAPPER) operates on memory recalls and maps a recall matrix into K message samples.

The system training process involves two separate phases, for the AM and the NN, respectively. First, each stored image is (arbitrarily) associated with a message. For each such couple, K-GEN generates a pseudo-random key [5] from the image, whereas MAPPER generates an MVP from the corresponding message. Pattern couples are pairwise convolved, and the summing all convolutions builds the actual memory:

$$\mathbf{M}_{\mathrm{TOT}} = \sum_{p=1}^{P} \mathscr{M}(\mathbf{m}_p) * \mathscr{K}\left(\mathbf{I}_p\right) \qquad (1)$$

where $\mathscr{M}$ and $\mathscr{K}$ denote the MAPPER and K-GEN processes, whereas $\mathbf{I}_p$ and $\mathbf{m}_p$ represent a stored image and message, respectively.

The second phase trains the NN to filter out memory crosstalk from recalled infomration. The network involves a classical feedforward model [1], with two layers of weights and sigmoidal units; the input and the output layer have as many neurons as a message dimensionality, binary units in the output layer represent decoded message bits (Fig.2). Message length was set to 64 bits, whereas 16 hidden units were used.

For the NN training, a memory-recall cycle re-uses the keys associated with images to retrieve the corresponding MVPs; applying INVERSE MAPPER yields K samples of each original message. The set of all such samples constitutes the training set for the neural network, which undergoes standard back-propagation [2] optimization of parameters. Therefore, the NN operates as a nonlinear adapter filter tuned to remove crosstalk noise [4].

As a result of the overall training process, the AM holds image-message associations. Recalls of these associations are in practice affected by crosstalk due to a non perfect uncorrelation among keys, hence the NN is taught to perform robust decoding accordingly.

At run time (Fig.3), the input image generates a key, which is convolved with the AM for information retrieval; the result represents a (noisy) version of a stored MVP. INVERSE MAPPER extracts message samples, which are processed by the NN for filtering. In order to compress K sample estimates into one filtering decision, an average operation on the sample set is performed. The formulation of the system's run-time output, $\mathscr{C}$, can be expressed as:

$$\mathscr{C}(\mathbf{I}_x) = \mathscr{N}\left\{\mathscr{E}\left[\mathscr{M}^{-1}\left(\mathscr{K}(\mathbf{I}_x) \otimes \mathbf{M}_{TOT}\right)\right]\right\} \qquad (2)$$

where $\mathscr{E}$ and $\mathscr{N}$ indicate the average operator and the neural network transfer function, respectively, and $\mathbf{I}_x$ is the processed image.

3 MULTIPLE SENSOR INTEGRATION

The basic framework described in [7] distinguishes the data-fusion process, in which data are merged at the same representation level, from the sensor-integration process, in which the system actually takes advantage from multiple information sources.

This paper shows that the advantages of integrated information can be exploited directly at the fusion stage. The key idea in sensor integration is to have all inputs contributing to memory writing and reading. In practice, each sensor contributes to building up a portion of the memory-addressing key

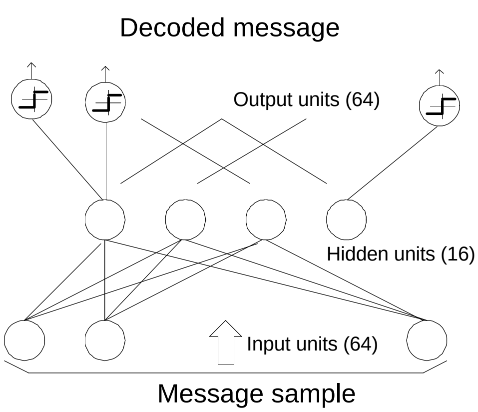

Fig.2 - The neural network filter

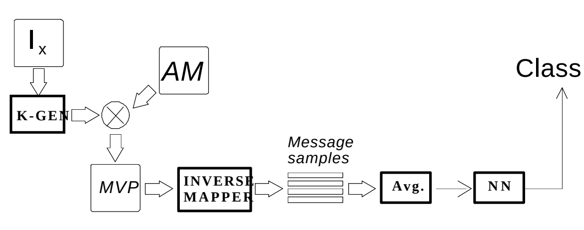

Fig.3 - Run-time operation of the integrated system

(Fig.4). This approach aims at exploiting the memory's pattern completion properties: missing or noisy key elements (e.g., due to sensor fault or malfunctioning) will be compensated for by information from other sources.

Integrating data at the lowest level preserves (and exploits) mutual data cross-correlations, without necessarily rendering them explicit. The AM operates as a nonlinear, flexible and robust cross-correlation storage device. As keys and not data are fused, the integration method is independent of the specific output format provided by each sensor. In practice, a new information source to be included in the system will just require a single "driver" module, which is actually a K-GEN module.

Robustness can be improved by a duplication mechanism. Thanks to the chaotic nature of K-GEN, one input can generate different keys if the initial K-GEN parameters are changed. Assembling results from different sensors and from multiple keys provides redundancy in memory addressing and increases robustness. On the other hand, the inherent parallelism makes it possible to include multiple sensors without affecting timing performances.

In the following, $\mathscr{K}$ will denote the whole multisensor key generation; if, for simplicity, two visual sensors are used, the following is the formalism for the integrated system functioning:

$$\mathscr{C}(\mathbf{I}'_x,\mathbf{I}''_x)=\mathscr{N}\left\{\mathscr{E}\left[\mathscr{M}^{-1}\left(\mathscr{K}(\mathbf{I}'_x,\mathbf{I}''_x)\otimes\mathbf{M}_{TOT}\right)\right]\right\} \quad (3)$$

showing that the increased complexity from multisensor integration is marginal. This is a result of decoupling the high-dimensional from the low-dimensional domain representations.

To assess robustness at image-message mapping quantitatively, increasing percentages of corrupting noise are added to test images, and correctness in message decoding is checked. Noise injection consists in randomly cancelling percentages of non-zero image pixels; this noise model intentionally stresses retrieval accuracy [11]. The procedure stops when even only one message bit is decoded incorrectly; the tolerated noise percentage is indicative of the system's insensitivity.

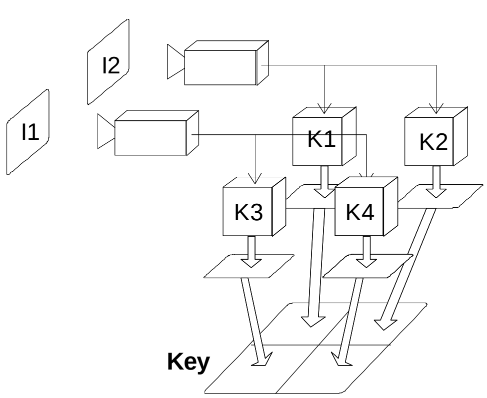

Fig.4 - Sensor integration in key generation

Message-decoding robustness is measured when: a) both sensors are operating; b) the first sensor is obscured; and c) the second sensor is obscured. These conditions correspond to :

$$\text{a)}\ \mathscr{C}\left(\mathbf{I}'_p+\mathbf{n}',\mathbf{0}\right);\qquad \text{b)}\ \mathscr{C}\left(\mathbf{0},\mathbf{I}''_p+\mathbf{n}''\right);$$

$$\text{c)}\ \mathscr{C}\left(\mathbf{I}'_p+\mathbf{n}',\mathbf{I}''_p+\mathbf{n}''\right);\qquad p=1,...,P \qquad (4)$$

where **n'** and **n"** are random variables representing noise. The visual testbed includes different domains (geometrical shapes, human faces, cell nuclei, etc.) to verify generality, and the methodology can be tested under different conditions:

I) one sensor is operating - tests (4.a) and (4.b);

II) both sensors receive the same data: $\mathbf{I}'_p \cong \mathbf{I}''_p$;

III) sensors receive different information: $\mathbf{I}'_p \neq \mathbf{I}''_p$.

To stress condition I), each sensor processes a set of images often very similar to one another, which make single-sensor discrimination hard. As expected, robustness results using one active sensor agree with those from previous experiences [4,6].

Under condition II, sensors perceive the same image. If noise sources are independent, robustness increases considerably; instead, if noise sources are correlated, it does not improve very much. This can be explained in terms of associative integration. In the former case, noise sources compensate for each other, and the memory's pattern completion increases insensitivity; in the latter case, correlated noise in the two images inhibits mutual compensation, and the system behaves like a single-sensor one.

Sensor-integration benefits become evident under condition III), when the two sensors contain images quite different from one another — a particular situation that simulates the general case of two independent sensors. In such condition, robustness improves significantly and independently of mutual noise-correlation. This stresses the importance of associative cross-correlation between different sources, and confirms another intuitive expectation: heterogenous sources strongly contribute to enhancing effectiveness and discrimination in associative information processing.

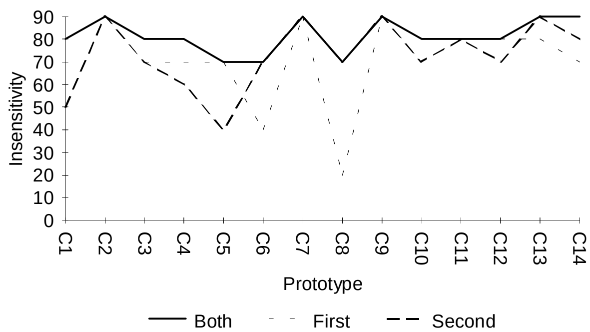

Fig. 5 - Robustness of the integrated structure

4 RESULTS AND DISCUSSION

Fig.5 gives detected noise insensitivity at message decoding with independent information sources om a testbed of 14 couples, some examples of which are presented in Fig.6. Sensor integration increases average robustness and smooths peculiarities of the actual visual set. The system's early failure on couple C8, when using only the first sensor, is caused by the similarity of image C8' to other images in the testbed. Likewise, this explains errors on image C5" when using only the second sensor. The graph shows that using both sources makes up for these oddities .

When both sensors process equal information, insensitivity follows the single-sensor curves if noise sources are correlated; otherwise, robustness increases significantly up to an almost constant value equal to 90%, which makes the associative decoder appear as a sort of "voting structure" where members compensate for each other's faults.

The overall represented research confirms the validity of using associative devices for multisensor integration; future lines include the use of different sensors and the inclusion of application-specific preprocessing to further increase effectiveness.

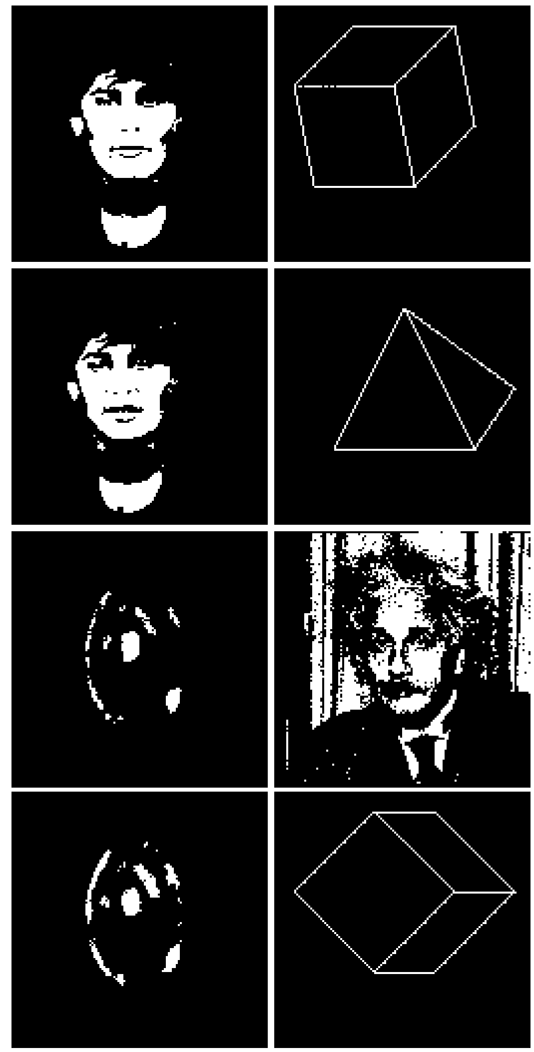

Fig.6 - Sample couples used in the testbed

5 REFERENCES

[1] Hinton GE, Anderson JA (Eds) *Parallel Models of Associative Memory - (Updated edition)* Lawrence Erlbaum, New York, 1989.

[2] Rumelhart DE, Hinton GH, Williams RJ "Learning internal representation by error propagation" in Rumelhart, McClelland (Eds) *Parallel Distributed Processing - Explorations in the Microstructure of Cognition*, MIT Press, 1986.

[3] Bottini S "An algebraic model of an associative noise-like coding memory" *Biological Cybernetics*, No. 36, 1980, pp.221-228.

[4] Anguita D, Parodi GC, Zunino R "Associative structures for Vision" *Multidimensional Systems and Signal Processing*, No. 5, 1994, pp.75-96.

[5] Parodi GC, Ridella S, Zunino R "Using Chaos to Generate Keys for Associative Noise-Like Coding Memories" *Neural Networks*, Vol.. 6, 1993, pp. 559-572.

[6] Pagano F, Parodi GC, Zunino R "Parallel implementation of associative memories for image classification" *Parallel Computing*, Vol. 19, No. 6, 1993, pp. 667-684.

[7] Luo RC, Kay MG "Multisensor integration and fusion in intelligent systems" *IEEE Trans. on Systems, Man, and Cybern.*, vol. SMC 19-5, Sept/Oct 1989, pp.901-931.