Designing and Development of a Robotic Puzzle Solver Using Morphological Associative Memories

A. Arvanitidis, S. A. Chatzichristofis, A. Ch. Kapoutsis, A. Amanatiadis and Y. Boutalis
School of Engineering, Democritus University of Thrace, Xanthi, Greece
{antoarva,schatzic,akapouts,aamanat,ybout}@ee.duth.gr

***Index Terms*—Two-dimensional puzzles, IR52c, Morphological Associative Memories**

***Abstract*—In this paper a novel approach for solving two-dimensional puzzles and assembling the puzzle using the IR52c robotic arm is presented. When compared to many classical methods for resolving puzzles, the proposed method is characterized by low computational cost and ease of development. In order to achieve shape recognition, classical techniques from the field of image processing are combined with pattern association techniques. A camera captures an image of the whole disassembled puzzle, whose pieces are scattered in the working area of the robotic arm. In the first step, this image is converted to a binary form using Otsus method. Then, the connected components are identified and their inclination is corrected. In order to achieve recognition of every single shape, Morphological Associative Memories (MAMs) that have been previously trained are used. After object recognition, the system communicates with the robotic arm to assemble the pieces of the puzzle. Experimental results demonstrate the effectiveness of the proposed method.**

I. INTRODUCTION

Sensorimotor architectures for automatic puzzle solvers try to address the challenges by making significant progress in several layers of abstraction [1]. While different architectures and systems have been proposed, the main core systems are common; Automatic puzzle solvers strongly depend on object recognition. Over the last decade, a number of techniques have been proposed in the literature which has led to the understanding that algorithms for object recognition must be scale, rotation, and transformation invariant. The vast majority of the state of the art methods for object recognition is based on shape descriptors. However, the challenging task of shape descriptors is the accurate extraction and representation of shape information.

Among the most commonly used shape detectors are the image moments [2], [3]. The MPEG-7 [4] also includes shape descriptors [5]. The overall performance of shape descriptors can be divided into qualitative and quantitative performance. The qualitative characteristics involve their retrieval performance based on the captured shape details for representation. Their quantitative performance includes the amount of data needed to be indexed in terms of number of descriptors, in order to meet certain qualitative standards, as well as their retrieval computational cost [6].

In order to achieve shape recognition, local features may also be used. A very popular feature extraction algorithm is the scale-invariant feature transform (SIFT) proposed in [7]. SIFT extracts feature points from training objects and stores them in a database. On new images, a feature vector is extracted and is compared using Euclidean distance to the feature vectors in the database. SIFT is invariant to scale, orientation and affine distortion and is partially invariant to illumination changes.

Bay et al. [8] proposed a feature extraction algorithm SURF (Speeded Up Robust Features). SURF is claimed to be faster than SIFT and uses an approximation of the Haar Wavelet of the determinant of Hessian blob detector. GLOH (Gradient Location and Orientation Histogram) [9] extends SIFT by changing the location grid and using principal components analysis (PCA) to reduce the high dimensionality of the descriptor. GLOH is designed to increase the robustness and distinctiveness of the SIFT descriptor. Local Energy based Shape Histogram (LESH) is a feature that uses a local energy model of feature perception [10]. The underlying shape is encoded with the accumulating local energy of the underlying signal after several filter orientations. In this way many local histograms are extracted and then are used to form a 128-dimensional compact spatial histogram. LESH is scale invariant.

In the past, several content based image retrieval (CBIR) techniques have been adopted in robot grasping systems to facilitate object recognition [1],[11],[12]. In this paper, a new CBIR method which adopts Morphological Associative Memories (MAMs) is proposed. In Section 2 MAMs are described in detail. The proposed system implements classic digital image processing techniques in order to solve the puzzle using the IR52c robotic arm. The puzzle consists of n wooden pieces of which only m ($m \leq n$) are considered beneficial. The n parts are located in a random position but within the working space of the arm. Actually the center of gravity of each object should be inside the working space. The image is taken from a web camera which is located at height of 1.30 m. After the puzzle pieces have been identified, they are placed at specified positions. The entire system is described in detail in Section III and the conclusions are presented in Section IV.

II. MORPHOLOGICAL ASSOCIATIVE MEMORIES

Morphological Neural Networks (MNNs) represent artificial neural networks whose neurons perform an elementary operation of mathematical morphology [13], [14]. Unlike Hopfield networks [15], MNNs provide the result in a single pass through the network, without any significant amount of training. The underlying algebraic system used in these models is the set of real numbers $\mathbb{R}$ together with the operations of addition and multiplication and the laws governing these operations. The basic computations occurring in morphological networks are based on the algebraic lattice structure $(R, +, \wedge, \vee)$, where $\wedge, \vee$ denote the binary operations for minimum and maximum, respectively.

Morphological associative memories (MAMs) are based on the algebraic lattice structure [16] $(R, +, \wedge, \vee)$. They were initially proposed for associating binary pattern vectors [17] and are far more robust to noise than the conventional linear associative memories.

There are two basic approaches to record k vector pairs $(x_1, y_1), \ldots, (x_k, y_k)$ using a morphological associative memory. The first approach consists of structuring an $m \times n$ matrix $\mathbf{W_{xy}}$ with elements computed by:

$$w_{ij} = \bigwedge_{k}^{r=1} (y_i^r - x_j^r) \;\; i = 1, \ldots, m \;\; j = 1, \ldots, n. \tag{1}$$

The dual approach consists of constructing an $m \times n$ matrix $\mathbf{M_{xy}}$ with elements:

$$m_{ij} = \bigvee_{k}^{r=1} (y_i^r - x_j^r) \;\; i = 1, \ldots, m \;\; j = 1, \ldots, n. \tag{2}$$

If matrix $\mathbf{W_{xy}}$ receives a vector x^r as input, the product $y^r = \mathbf{W_{xy}} \otimes x^r$ is formed. The product is called max product and each element of the resulting vector y^r is computed by the formula:

$$y_i^r = \bigvee_{j=1}^{n} (w_{ij} + x_j^r) \;\; i = 1, \ldots, m \;\; j = 1, \ldots, n. \tag{3}$$

Likewise, if matrix $\mathbf{M_{xy}}$ receives x^r as input the so-called min product $y^r = \mathbf{M_{xy}} \otimes x^r$ is formed, where each element of the resulting vector y^r is computed by the formula:

$$y_i^r = \bigwedge_{j=1}^{n} (m_{ij} + x_j^r) \;\; i = 1, \ldots, m \;\; j = 1, \ldots, n. \tag{4}$$

Matrices $\mathbf{W_{xy}}$ and $\mathbf{M_{xy}}$, computed by Eqs. 1 and 2 respectively, constitute the memory of the MAM. The only required training of the network is simply the computation of either of the two matrices $\mathbf{W}$ or $\mathbf{M}$. The difference between the two memories arises when noisy patterns appear at their input. Memory $\mathbf{M_{xy}}$ is able to retrieve y^r in case a noisy vector x^r corrupted by dilative noise appears at its input, while memory $\mathbf{W_{xy}}$ is able to retrieve y^r in case an eroded version of the input pattern appears at its input.

Ritter et.al. proposed the idea of two-step MAMs and the production of the so-called kernel vectors, as an elegant representation of the associations (x^r, y^r). The conditions for a vector $\mathbf{z}$ to be a kernel vector are given in [17]. These vectors, in conjunction with matrices $\mathbf{M}$ and $\mathbf{W}$, are suitable for recalling y^r when an arbitrarily corrupted (both eroded and dilated) input pattern appears at the input of the MAM. A binary vector $\mathbf{z}$ is said to be a kernel of the association (x, y) between the vectors $\mathbf{x}$ and $\mathbf{y}$ if it is a subset of $\mathbf{x}$ and satisfies the following conditions:

$$\mathbf{M_{zz}} \oplus \mathbf{x} = \mathbf{z} \tag{5}$$

$$\mathbf{W_{zy}} \otimes \mathbf{z} = \mathbf{y} \tag{6}$$

Where matrices $\mathbf{M_{zz}}$ and $\mathbf{W_{zy}}$ are computed according to Eqs. 1 and 2. In case of autoassociation Eqs. 5 and 6 are written as:

$$\mathbf{M_{zz}} \oplus \mathbf{x} = \mathbf{z} \tag{7}$$

$$\mathbf{W_{zx}} \otimes \mathbf{z} = \mathbf{x} \tag{8}$$

When an arbitrarily computed input pattern appears at the input of a morphological auto associative network, the original uncorrupted input pattern x is recalled by the following equation:

$$\mathbf{W_{zx}} \otimes (\mathbf{M_{zz}} \oplus \tilde{\mathbf{x}}) = \mathbf{x} \tag{9}$$

III. System Overview

The proposed system is designed to solve a puzzle whose pieces are randomly distributed around the workspace of a robotic arm. The puzzle consists of n pieces of which only m $(m \leq n)$ are considered beneficial.

An image which includes all the puzzle pieces is taken from a still camera located 1.3 m above the puzzle plane. In the initial stage, the image is converted from RGB to YIQ color space. The auto brightness correction method proposed in [18] is applied to the Y channel. This method is partially inspired by the HVS (Human Vision System). In particular, the method adopts some of the shunting characteristics of the on-center off-surround networks, in order to define the response function for a new artificial center-surround network. This network compares every pixel to its local average and assigns a new value in order to lighten the dark image regions, while minimally affecting the light regions. The aim of this filter is to cover alterations in brightness that might result from the settings of the instrument used to capture the image. The same auto brightness correction approach has also been adopted as preprocessing step in several image retrieval systems (e.g. [19]).

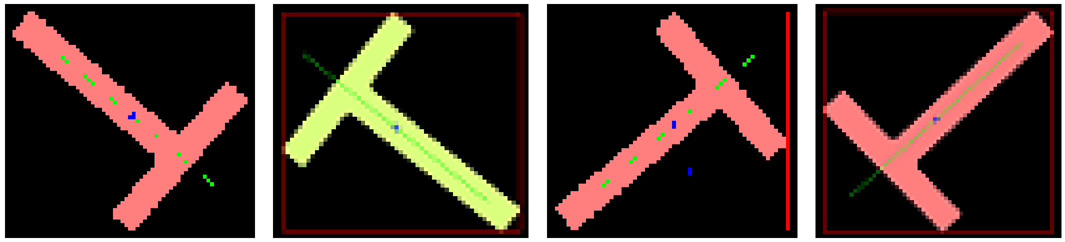
Then, the image is converted to binary using the Otsu method. This step is necessary prior to extracting and labeling all connected components. The algorithm assumes that the image to be thresholded contains two classes of pixels (i.e., foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal.

The goal then is to select the threshold that minimizes the combined spread. We can define the within-class variance as the weighted sum of the variances of each cluster:

$$\sigma^2_{\text{Within}}(T) = n_B(T)\sigma^2_B(T) + n_O(T)\sigma^2_O(T), \tag{10}$$

where

$$n_B(T) = \sum_{i=0}^{T-1} p(i) \tag{11}$$



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

Fig. 1. Possible Rotation Angles. Angle of blob (a): $40°$; Angle of blob (b): $40°$; Angle of blob (c): $-40°$; Angle of blob (d): $-40°$.

$$n_O(T) = \sum_{i=t}^{N-1} p(i) \tag{12}$$

and $\sigma_B^2(T)$ the variance of pixels in the background (below threshold), $\sigma_O^2(T)$ the variance of pixels in the foreground (above threshold) and $[0, N-1]$ the range of intensity levels.

If one subtracts the within-class variance from the total variance of the combined distribution, you get something called the between-class variance:

$$\begin{aligned}\sigma_{\text{Between}}(T) &= \sigma^2 - \sigma^2_{\text{Within}}(T) \\ &= n_B(T)[\mu_B(T) - \mu]^2 + n_O(T)[\mu_O(T) - \mu]^2\end{aligned} \tag{13}$$

where σ^2 is the combined variance and μ is the combined mean. Notice that the between-class variance is simply the weighted variance of the cluster means themselves around the overall mean. Substituting $\mu = n_B(T)\mu_B(T) + n_O(T)\mu_O(T)$ and simplifying results in:

$$\sigma_{\text{Between}}(T) = n_B(T)n_O(T)[\mu_B(T) - \mu_O(T)]^2 \tag{14}$$

For each potential threshold T:

1) Separate the pixels into two clusters according to the threshold.
2) Find the mean of each cluster.
3) Square the difference between the means.
4) Multiply by the number of pixels in one cluster times the number in the other.

This depends only on the difference between the means of the two clusters, thus avoiding having to calculate differences between individual intensities and the cluster means. The optimal threshold is the one that maximizes the between-class variance (or, conversely, minimizes the within-class variance). The implementation of the Otsu method is based on the img(Rummager) [20] implementation.

In order to separate the image of each puzzle piece for the subsequent identification, we used a proprietary implementation of the OPENCV library. This implementation is a linear-time component-labeling algorithm using a contour tracing technique to detect objects in an image. This algorithm not only labels components but also extracts component contours and sequential orders of contour points, which can be useful for many applications. The main step of this algorithm is to use a contour tracing technique to detect the external contour and possible internal contours of each component, and also to identify and label the interior area of each component.

This method labels each component using a contour tracing technique. This method is based on the principle that a component is fully determined by its contours, just as a polygon is fully determined by its vertices. This method also provides a procedure for finding all component pixels.

This algorithm has 4 key points:

- Algorithm visits each pixel a constant number of times.
- Algorithm runs in linear time.
- All pixels in the same component are assigned the same label.
- Pixels in different components are assigned different labels.

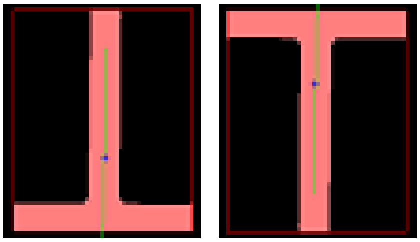
The algorithm has several advantages. First, it requires only one pass over the image. Contour points are visited more than once due to the aforementioned contour tracing procedure, but no more than a constant number of times. Second, it does not require any re-labeling mechanism. Once a labeling index is assigned to a pixel, its value is unchanged. Third, we obtain as by-products all contours and sequential orders of contour pixels. Fourth, experimental results show that our algorithm is faster than traditional component-labeling algorithms and improves performance dramatically.

After this image separation process, we extract and label all the connected components forming the blobs. Each blob is the binary image of each puzzle piece. To be able to recognize each item, a rotation correction algorithm must be applied to each blob since MAMs are not rotation independent.

The algorithm we used to extract and label connected components gives us some characteristics such as the location of the blob, rotation angle and center of gravity. The value of the rotation angle is between $[-90, 90]$. For example, the rotation angle of a piece may be in one of the 4 different positions as illustrated in Fig. 1.

Correcting the rotation angle to 90 degrees will have results shown in either Fig. 2(a) or Fig. 2(b).

Matrices $\mathbf{M}$ and $\mathbf{W}$ that were described in Section II have been computed offline. Binary images of the letters that are included at the puzzle, were used as inputs of the system, as they were captured from the systems camera. Artificially generated binary images were used as associate images. For every kernel that was extracted from the input images, a



(a) (b)

Fig. 2. Correcting the rotation angle. (a): Rotation angle 90° for the blobs of Fig. 1(a) and 1(c), (b): Rotation angle 90° for the blobs of Fig. 1(b) and 1(d).

different associated image was used.

These associated images differed by as little as a single pixel. As a result of the process mentioned in Section II, when a noisy is used as input to the system, MAMs will return to the user the associate images that were used during the training process of the system. After scanning this image pixel-by-pixel, a descriptor with binary values is extracted. This new descriptor is called Morphological Associative Descriptor (MAD).

After the puzzle pieces have been identified, they are placed at specified positions. The specified positions are stored in an XML database with entries for each piece that include the following properties:

1) Letter ID
2) Morphological Associative Descriptor (MAD)
3) Final Position X
4) Final Position Y
5) Boolean (0 or 1) value that defines if the specific letter is part of the puzzle

For every blob that is used as input to the system, a MAD descriptor is extracted and then compared to the MAD descriptors stored in the XML file. Because MAD descriptors are binary vectors, in order to calculate the distance between two descriptors, a single XOR logic gate is used. If a blobs MAD descriptor distance to a MAD descriptor in the XML file is zero, the system can identify whether that blob is part of the puzzle from the stored Boolean value and, if so, retrieve the final position X and Y. In order to avoid an error produced during the correction of the rotation angle, the XML database contains information concerning every piece of the puzzle and the symmetric of this piece.

After each shape is recognized, the coordinates of the current and final locations are sent to the robotic arm. The rotation angle of each object is found from the method used to calculate how many degrees each blob should rotate in order to avoid the problem of improperly functioning MAMs. Note that when one of the normalized blobs is identified as a known figure, the rotation angle of the blob is ($90° -$ blob angle). If a blob corresponds to the symmetric form of a known shape, the rotation angle of the object is ($90° -$ blob angle) + $180°$.

IV. Experimental Results

Initially, to evaluate the performance of the MAD descriptor, a number of simulated experiments were carried out. In order to have a meaningfully sized data set, 26 uncorrupted binary patterns were initially created. The set contains the 35×34 binary images of the capital letters of the Latin alphabet and is shown in Fig. 4. Please note that the same dataset has also been used in [21]. The MAD descriptor of these pattern has been extracted and stored in the XML database of the system.

Next, we evaluated the ability of the presented descriptor to accurately retrieve the correct pattern. The Morphological Associative Descriptor of all the 26 patterns was re-calculated and each one of them compared (by employing a simple XOR gate) with the entire database of the system. As expected, the system performed a 100% success in recalling in right pattern in all the cases. Next, mixed type (both dilative and erosive) of noise was randomly applied on each binary image. The noise percentage is computed by counting the number of the altered (noisy) pixels and divide them with the total number of image pixels. For each pattern, we applied 5 different levels of random noise (10, 30, 40, 50 and 60 %). Each experiment was repeated 100 times. In other words, for each Latin character, we reproduced 100 noisy images for each noise level. Overall, we evaluated the performance of MAD descriptor using 13000 (26 (letters) $\times$ 5 (noise levels) $\times$ 100 (samples)) query images. As illustrated in Table 1, the MAD descriptor is capable to accurately recall by 100% the correct pattern even in cases with 30% noise. Moreover, it is worth noting that even in cases with appearance of 60% of noise, the presented approach achieves a 33% accuracy.

TABLE I
MAD Evaluation Results in Case of Noisy Patterns

Noise Levels	10%	30%	40%	50%	60%
No Rotation	**100%**	**100%**	85%	72%	33%
Rotated Images	**92%**	**87%**	64%	51%	27%

In the sequel, a preprocessing step was performed to determine the robustness of the proposed approach. Each one of the query image was rotated by three different randomly generated angles. The set of 3×13000 images was used to evaluate the ability of the proposed approach to automatically correct the rotation of the input samples. As depicted in Table 1, the rotation correction approach, in combination with the MAD descriptor, manage to achieve an impressive 87% accuracy even in cases of 30% appearance of noise.

Finally, the proposed system has been tested as a whole with an IR52c robotic arm. The experimental setup is illustrated in Fig. 5. The robot's task is to recognize the letters D,U,T and H, to locate their position and of course, to move them into the desirable location. It is worth noting that the desirable location of each letter, the correct orientation as well as the appropriate grasping position are known to the robot. Its main objective is to recognize, based on their shape, the letters.

A low cost web-camera (mounted 1.3 meter above the robot) captures an image of the whole disassembled puzzle, whose

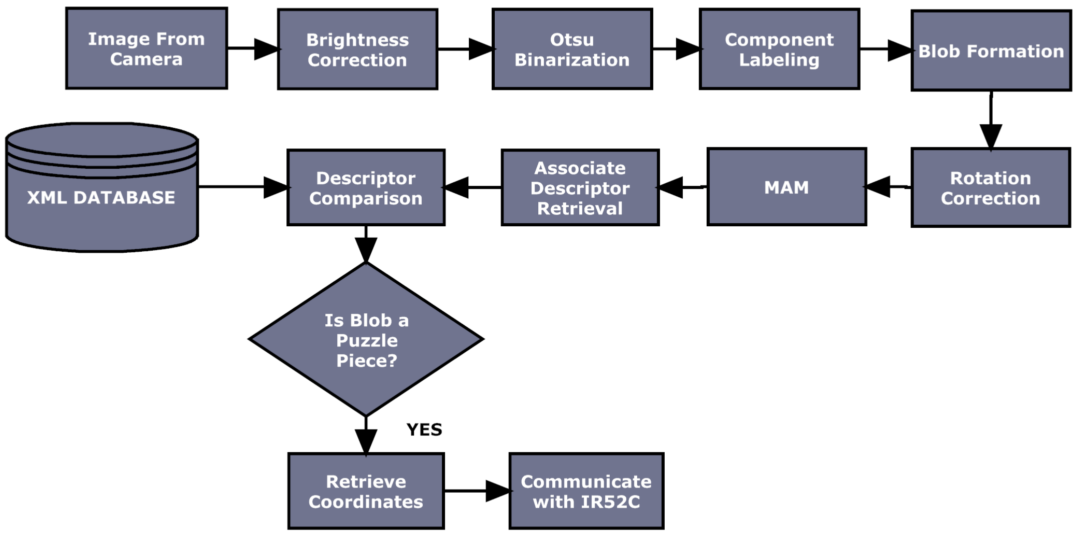


Fig. 3. System Pipeline Overview.

Fig. 4. The complete evaluation set consisting of 26 binary images of the capital letters of Latin alphabet

pieces are scattered in the working area of the robotic arm. The captured image is analyzed by the system (Fig. 6). Initially, the auto brightness correction method enhances the quality of the image and the Otsu binarization algorithm convert the improved capture into a binary one. The component labeling module, together with the blob formation one, identifies the letters' area and their location. In the sequel, the rotation correction module automatically rotates each letter to the desirable orientation. Next, the MAD descriptor is extracted from each blob and based on the stored information, the letter is recognized. Finally, the robotic arm retrieves details from the database about the desirable location and the orientation of each letter.

The experimental results have shown that the system is capable of successfully recognizing and solving the puzzle. The experiment was repeated several times with the same results, under different lighting conditions

V. Conclusions

In this paper a low cost method for solving 2D puzzles using Morphological Associative Memories is proposed. The proposed system implements classic digital image processing techniques in order to solve the puzzle using the IR52c robotic arm. The puzzle consists of n wooden pieces of which only m $(m \leq n)$ are considered beneficial. The n parts are located in random positions but within the working space of the arm. Actually the center of gravity of each object should be inside the working space. Puzzle pieces are placed at specified positions. Experimental results showed that the system is capable of working properly and can recognize puzzle pieces and construct the puzzle successfully.

References

[1] R. Kouskouridas, A. Amanatiadis, S. A. Chatzichristofis, and A. Gasteratos, "What, where and how? introducing pose manifolds for industrial object manipulation," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 8123–8133, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2015.06.039

[2] K. E. Van De Sande, T. Gevers, and C. G. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[3] E. Karakasis, A. Amanatiadis, A. Gasteratos, and S. A. Chatzichristofis, "Image moment invariants as local features for content based image retrieval using the bag-of-visual-words model," *Pattern Recognition Letters*, vol. 55, pp. 22–27, 2015.

[4] S.-F. Chang, T. Sikora, and A. Purl, "Overview of the mpeg-7 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 688–695, 2001.

[5] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7: multimedia content description interface*. John Wiley & Sons, 2002, vol. 1.

[6] A. Amanatiadis, V. G. Kaburlasos, A. Gasteratos, and S. E. Papadakis, "Evaluation of shape descriptors for shape-based image retrieval," *Image Processing, IET*, vol. 5, no. 5, pp. 493–499, 2011.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*, 2006, pp. 404–417.

[9] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[10] M. S. Sarfraz and O. Hellwich, "Head pose estimation in face recognition across pose scenarios." in *International Conference on Computer Vision Theory and Applications*, vol. 8, 2008, pp. 235–242.

[11] L. Hammouda, K. Kaaniche, H. Mekki, and M. Chtourou, "Robust visual servoing using global features based on random process,"

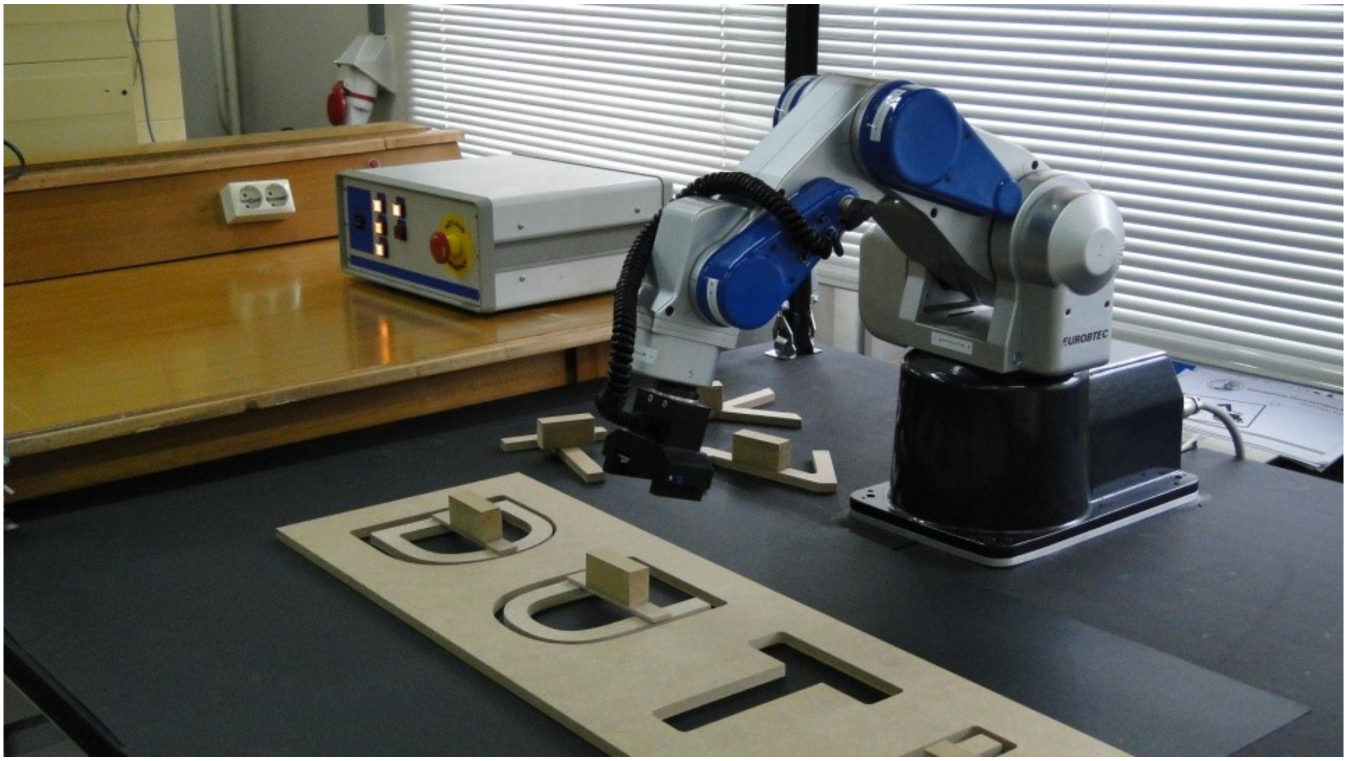


Fig. 5. The experimental setup in action: IR52c Robotic Arm

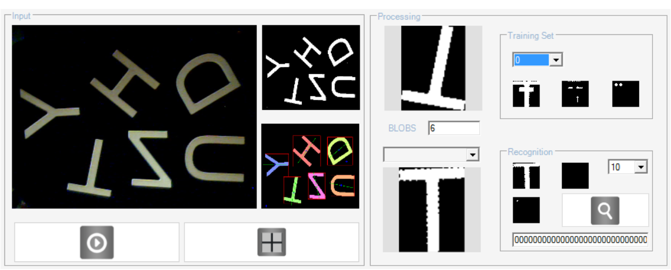


Fig. 6. The experimental setup in action: Software application screenshot

IJCVR, vol. 5, no. 2, pp. 138–154, 2015. [Online]. Available: http://dx.doi.org/10.1504/IJCVR.2015.068803

[12] J. J. Steil, F. Röthling, R. Haschke, and H. J. Ritter, "Situated robot learning for multi-modal instruction and imitation of grasping," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 129–141, 2004. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2004.03.007

[13] P. Sussner and M. E. Valle, "Morphological and certain fuzzy morphological associative memories for classification and prediction," in *Computational intelligence based on lattice theory*. Springer, 2007, pp. 149–171.

[14] J. L. Davidson and G. X. Ritter, "Theory of morphological neural networks," in *Digital Optical Computing*, vol. 1215, 1990, pp. 378–388.

[15] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[16] V. G. Kaburlasos, S. E. Papadakis, and A. Amanatiadis, "Binary image 2d shape learning and recognition based on lattice-computing (lc) techniques," *Journal of Mathematical Imaging and Vision*, vol. 42, no. 2-3, pp. 118–133, 2012.

[17] G. X. Ritter and P. Sussner, "An introduction to morphological neural networks," in *International Conference on Pattern Recognition*, vol. 4, 1996, pp. 709–717.

[18] V. Vonikakis, I. Andreadis, and A. Gasteratos, "Fast centre-surround contrast modification," *Image Processing, IET*, vol. 2, no. 1, pp. 19–34, 2008.

[19] S. A. Chatzichristofis and Y. S. Boutalis, "Content based radiology image retrieval using a fuzzy rule based scalable composite descriptor," *Multimedia Tools Appl.*, vol. 46, no. 2-3, pp. 493–519, 2010. [Online]. Available: http://dx.doi.org/10.1007/s11042-009-0349-x

[20] S. A. Chatzichristofis, Y. S. Boutalis, and M. Lux, "Img (rummager): An interactive content based image retrieval system," in *International Workshop on Similarity Search and Applications*, 2009, pp. 151–153.

[21] Y. S. Boutalis, "A new method for constructing kernel vectors in morphological associative memories of binary patterns," *Comput. Sci. Inf. Syst.*, vol. 8, no. 1, pp. 141–166, 2011. [Online]. Available: http://dx.doi.org/10.2298/CSIS091114026B