Neurocomputing 120 (2013) 90–100

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Efficient representation and feature extraction for neural network-based 3D object pose estimation

Rigas Kouskouridas [a,*], Antonios Gasteratos [a], Christos Emmanouilidis [b]

[a] Democritus University of Thrace, School of Engineering, Building I, Vas. Sofias 12, Xanthi 671 00, Greece
[b] CETI/ATHENA Research & Innovation Centre, Xanthi, Greece

ARTICLE INFO

Article history:
Received 4 March 2012
Received in revised form
1 November 2012
Accepted 4 November 2012
Available online 26 March 2013

Keywords:
Feature extraction
Feature representation
3D pose estimation
Supervised learning
Unsupervised learning
Neural networks

ABSTRACT

This paper introduces an efficient representation and feature extraction technique for 3D pose estimation of objects, incorporating a novel mechanism for the exploitation of the extracted visual cues. A combination of a fuzzy clustering technique for the input space, with supervised learning, results in a problem of reduced dimensionality and an efficient mapping of the input–output space. While other neural network-based approaches for 3D pose estimation focus on reducing dimensionality based on input space characteristics, such as with PCA-based approaches, the proposed scheme directly targets the input–output mapping, based on the available visual data. Evaluation results provide evidence of low generalization error when estimating the 3D pose of objects, with the best performance achieved when employing Radial Basis Functions. The proposed system can be adopted in several computer vision applications requiring object localization, pose estimation and target tracking.

1. Introduction

The task of estimating the 3D pose of an object is among the most challenging ones in computer vision due to its practical significance in a plethora of diverse approaches. In recent years, a number of applications have primarily focused on detection and estimation of objects' pose from either a single or multiple instances for a given template [1–3]. The ultimate goal is to diffuse this technology to deliver efficient accomplishment of complex tasks, such as object manipulation, robotic navigation, etc. [4–6]. Despite the substantial endeavors and certain achievements made so far, no advanced computer vision system characterized with sufficient trade offs between computational burden and performance, has yet been built.

To achieve adequate performance, the complexity of a regressor or a classifier needs to match the complexity of the modeling tasks and in most cases is influenced by the input space dimensionality, d, and the size of the sample data, N. Dimensionality reduction is sought to bring down costs associated with gathering, storing and processing data, limit modeling overfit and reduce the computational burden of the training task. Additionally, simpler input models are more tolerant to noise, outliers and other disturbances, while the process is controlled in a more efficient way when information is represented with fewer features. In computer vision applications, input data is usually represented as $(x,y)^{\kappa}$, where (x,y) are the image coordinates and κ is the dimension of the feature vector extracted by the detector and descriptor, respectively [7]. In [8–10] state of the art dimensionality reduction frameworks attempt to address issues that emerge in face recognition problems. Among the most popular feature extraction dimensionality reduction techniques are the Principal Component Analysis (PCA), the Independent Component Analysis (ICA), the Linear Discriminant Analysis (LDA) or the Isomap.

Although humans exhibit remarkable skills in estimating the relative pose of rigid objects given an initial hypothesis, such an ability is limited in contemporary computer vision systems. In this paper we attempt to address this issue, by introducing a neural network-based framework that is not only able to estimate the 3D pose of any object contained in the database, but also to generalize to unknown ones. The network is trained with numerous targets contained in several available datasets [11,12]. Furthermore, the training process is guided by a fuzzy evulsion of the centers of the extracted features by applying the Fuzzy c-means algorithm [13]. The 3D pose of an object in an unknown training instance can be considered as represented by the distances of the fuzzy centers from one particular center, as shown in Fig. 1. The high-level intuitive idea underlying the proposed method is that for a given pose and for any object, there exist similar topological patterns characterizing this pose. Should these patterns be able to be

* Corresponding author. Tel.: +30 25410 79359.
E-mail addresses: rkouskou@pme.duth.gr (R. Kouskouridas), agaster@pme.duth.gr (A. Gasteratos), christosem@ieee.org (C. Emmanouilidis).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.11.047

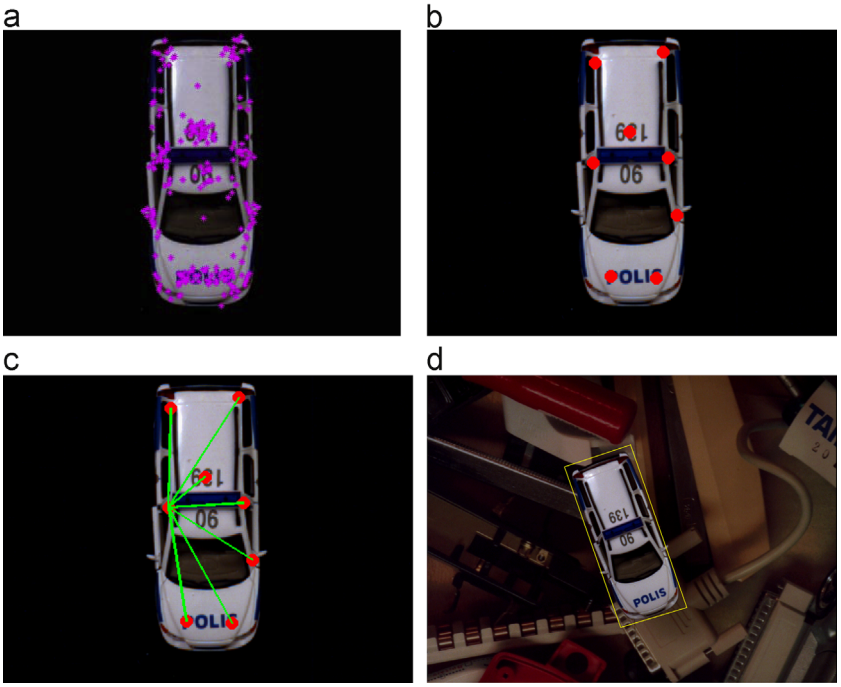

Fig. 1. In this figure the main idea underlying the proposed method is shown. After the extraction of keypoints (a) and in order to find the representative centers we perform a clustering procedure (b). The distance of each of these clusters with a given one stands for the model of the object-target (c). As a final step, those distances are considered as input to a neural network-based framework for estimating the 3D pose of the object (d).

extracted and recognized, the respective pose can be computed. The way the network of the centers of the clusters changes from one image shot to the other comprises such a pattern and, in order to quantify it, the distances between the nodes of the network are measured and kept. The proposed method involves a new input–output mapping that reduces the dimensionality of the input vectors with good performance. A number of experiments were executed in order, firstly, to demonstrate the performance of the present technique and, secondly, to evaluate several network architectures in known datasets.

The contribution of this paper entails the formalization of this new input–output method that outperforms the conventional dimensionality reduction techniques widely used in image processing applications. In learning a representative description of a 3D object pose, our algorithm requires limited supervision during the learning process and modest pre-processing of the input vectors. Moreover, experimental evaluation provided evidence of non-linear mapping between input and output space, justifying our choice to adopt a neural network-based strategy. In addition, we analyze the input–output mapping process and discuss generalization issues on unknown objects. To the best of our knowledge, in the field of neural network-based computer vision, such an attribute of efficient data handling constitutes a novel approach. Lastly, we present experimental evaluation on multiple network architectures, which are parameterized by the number of the extracted fuzzy clusters. The proposed method was compared to other contemporary or well-established techniques for 3D pose estimation and/or dimensionality reduction and was proved to excel in efficiency, while exhibiting robustness against occlusions.

The remainder of the paper is structured as follows. Section 2 discusses related work in neural network-based solutions for 3D object pose estimation and recent trends in feature representation and extraction processes. The proposed method is described in depth in Section 3. In Section 4 extensive quantitative and qualitative experimental results are presented. Finally, concluding remarks are presented in Section 5 along with some final notes and an outlook to future work.

2. Related work

Although neural networks are common place in several imaging applications, for the particular task of estimating the 3D pose of an object limited only activity is reported. Both early and more recent studies [14–18] showed that the adoption of neural networks in computer vision is recommended in cases where the task in hand encompasses great physical complexity. In more detail, a modification of Kohonen's self-organizing feature map (SOM) is trained with computer generated object views corresponding to one or more object orientation parameters. However, the methods presented in those papers reported significant gains in performance as demonstrated only over objects employed in training. Furthermore, in [19] SOM theory is combined with a three layer feed-forward network trained with dynamic learning vector quantization (DVLQ). Objects used for training are sampled from a limited database containing targets with minimal pose variations, whilst only two DoF's are efficiently accumulated.

Several methods in this area adopt dimensionality reduction schemes with the PCA and its variations being the most popular one. For instance, in [20] an appearance-based method for the efficient estimation of the pose of 3D objects, where the PCA is utilized for dimensionality reduction, is presented. The neural network is trained with the resilient backpropagation method and, as far as the rotation parameters of the pose are concerned, only two DoF's are estimated, corresponding to in and out of image plane orbits. An extension of the aforementioned technique is found in [21] where input feature vectors are derived by nonlinear PCA. Both methods fail to interpolate between two known pose configurations, since they utilize object views with a sampling interval of 3° and emphasize in distinguishing the input patterns into the corresponding classes. From the relevant literature, several other methods [22–24] could also be identified which, however, have vague architecture descriptions and inadequate robustness in performance.

Our work takes advantage of previous research conducted in the area of feature representation and extraction for 3D object

recognition and pose estimation [25–27,3]. Picking up the most prominent patterns represents a challenging task with a significant effect on the 2D–3D point correspondence [28]. In [29], a new method for efficient feature selection based on a Fuzzy Functional Criterion, for the evaluation of the linkage between the input features and the output score, is presented. However, this technique is dedicated, specifically, to the head pose problem, for which it can report remarkable efficiency when trained with numerous datasets. On the other hand, in [30] and its extension [31] it was shown that a compact model of an object can be portrayed by linking together diagnostic "parts" of the objects from different viewpoints. Such "parts" are defined as large and distinguishable regions of the objects that are composed of many local invariant features. Despite the efficiency of this architecture, the method is mainly devoted to 3D object categorization. To the best of our knowledge, the closest work to our paper, as far as feature representation and extraction is concerned, is presented in [32], where the authors employ a method for extracting 4 or 5 close features point, called Natural 3D Markers (N3Ms), which enjoy distinctive photometric properties and equal distribution over the object's visible surface. While the two approaches have similarities we believe our model provides a more compact and abstract representation of the 3D object.

3. Methodology

The proposed work is highly motivated by the remarkable skills of humans in the particular task of finding the relative pose of unknown objects given an initial hypothesis. Towards this end, we attempted to build an advanced scheme that incorporates sophisticated feature extraction aiming at simulating the non-linear relation between train and target models. In this section we present the input–output mapping process upon which the training of the neural network-based approach is performed. The overall system can be viewed as a mapping from a set of input variables $x=x_1,\ldots,x_d$, belonging to a feature-space $\mathcal{X}$, to a modeled output variable $y=y(x;w)\in\mathcal{Y}$, with w denoting the vector of the adjustable parameters. The ultimate goal of our system is to learn a regressor $g:\mathcal{X}\rightarrow\mathcal{Y}$ from an a priori training dataset $\{x^n,y^n\}$, in order to efficiently approximate the output $\mathcal{Y}_t$, when an unknown example $\mathcal{X}_t$ is provided (viz. Fig. 4). The proposed framework concerns the relative object pose estimation. It is apparent that, for such a procedure a pair of images is required, the first image being the reference, whilst the second being the measured one. During the testing process, none of the two images belong to the training set, whilst our feature extraction method abstracts key-features whose inner combination, when projected onto new established sub-spaces offers great generalization capacities. Experimental results provide evidence that the proposed technique enables the accurate estimation of the 3 DoF of any testing object given that the latter lays in certain distances from the sensor. Regardless of the objects used for training, through the proposed feature representation and input–output mapping our system is capable of generalizing to totally unknown objects. Fig. 2 illustrates the basic components of our system that are next discussed in the remainder of this section.

3.1. Input–output mapping

The labeled training dataset contains m training examples, i.e. images, of k objects–targets along with the corresponding pose groundtruth. The construction of the training set $\{x^n,y^n\}$ is based on an iterative process over m images of k objects. For the facilitation of the nomenclature and with a view to reader's better understanding, the remainder of this section presents the

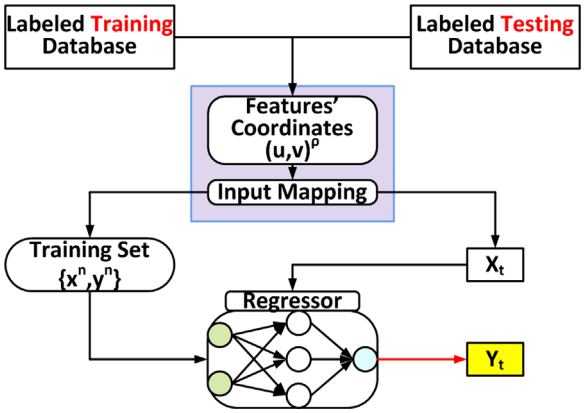

Fig. 2. Initially, labeled databases are divided into training and testing subsets, whilst for every object of the first set features' coordinates $(u,v)^\rho$ are extracted. As a follow-up step, the proposed input–output mapping technique takes over the construction of the set $\{x^n,y^n\}$ that is used for training the regressor. The ultimate purpose of our system is to provide an efficient approximation $\mathcal{Y}_t$ when an example $\mathcal{X}_t$, belonging to the testing subset, is presented to the network.

aforementioned iterative process for the specific object k^*. Initially, the image feature coordinates $(u,v)^\rho$ are calculated, with ρ denoting the number of the extracted keypoints. The next step is to employ the Fuzzy c-means clustering algorithm, in order to appoint feature vector $\mathbf{e}=(u^*,v^*)$, with the set of all feature vectors being $\mathcal{E}=\{e^c : c \text{ is the number of clusters organized as vectors}\}$.

Let $\mathbf{e}^*\in\mathcal{E}^*$ be a randomly selected example vector of clusters drawn from $\mathcal{E}^*\subseteq\mathcal{E}$. The proposed input–output mapping method proceeds by computing the Euclidean distance between vector $\mathbf{e}_i\in\mathcal{E}$ and anchor point $\mathbf{e}^*$:

$$\mathbf{x}^i=\|\mathbf{e}^i-\mathbf{e}^*\|^2=\sum_{i=1}^{c}\{\mathbf{e}^i-\mathbf{e}^*\}^2 \tag{1}$$

The most common approach for input normalization is the linear transformation of given vectors so that input variables are independent. Basically, such a kind of information transformation is generally based on the mean removal method and results in sets of input vectors that have zero mean and unit standard deviation. However, this linear rescaling treats input variables as independent, while in most of the cases they are not [7]. With a view to achieve an efficient solution to this problem we adopted a strategy which allows correlations amongst variables. Therefore, the input variable x^i is organized as a vector $\mathbf{x}=(x_1,\ldots,x_c)^\mathbf{T}$, while the sample mean vector and the covariance matrix with respect to the $\mathcal{L}$ data points of the training set are

$$\overline{\mathbf{x}}=\frac{1}{\mathcal{L}}\sum_{n=1}^{\mathcal{L}}x^n$$

$$\Sigma=\frac{1}{\mathcal{L}-1}\sum_{n=1}^{\mathcal{L}}(\mathbf{x}^n-\overline{\mathbf{x}})(\mathbf{x}^n-\overline{\mathbf{x}})^\mathbf{T} \tag{2}$$

This normalization results in vectors with the input variables given by the following formula:

$$\tilde{\mathbf{x}}^n=\Lambda^{-1/2}\mathbf{U}^\mathbf{T}(\mathbf{x}^n-\overline{\mathbf{x}}) \tag{3}$$

where $\mathbf{U}=(u_1,\ldots,u_c)$ and $\mathbf{\Lambda}=(\lambda_1,\ldots,\lambda_c)$ correspond to the eigenvectors and eigenvalues, respectively, which are calculated from the covariance matrix $\Sigma u_\zeta=\lambda_\zeta u_\zeta$.

The proposed input output mapping can be summarized as follows:

*Step*1: For each training image extract ρ SIFT features, denote them as $(u,v)^\rho$ and employ homography-based RANSAC for outliers' removal (Fig. 1(a)).

*Step*2: Given the locations of the ρ extracted SIFT keypoints $(u,v)^{\rho}$ apply the Fuzzy c-means algorithm to determine vector $\mathbf{e}=(u^*,v^*)^c$ (Fig. 1(b), where $c=8$). We would like to state that we do not cluster the key points in the training set into a fixed codebook.

*Step*3: Select the cluster $\mathbf{e}^*$ arbitrarily and build the input training example i by calculating $\mathbf{x}^i=\sum_{i=1}^{c}\{\mathbf{e}^i-\mathbf{e}^*\}^2$ (Fig. 1(c)). We should emphasize that the dimensionality of the input vector $\mathbf{x}^i$ equals to the number of the extracted clusters c.

*Step*4: Each *input vector* $\mathbf{x}^i$ is accompanied by the respective *output vector* $\mathbf{y}^3\in\mathcal{R}^3$ corresponding to the 3 DoF groundtruth rotational parameters.

3.2. Building the training set and simulating the network

The process presented in Section 3.1 iterates over m images of k objects holding information about the pose of the target. Since the employed databases contain numerous combinations of geometrical orientations, the most challenging task consists of finding features that repeat when matching one object's image with others depicting the same target under different viewpoints. More specifically, training datasets consist of images of objects shot every 5° and correspond to known poses y^n; the objects are placed on a turntable able to perform rotations around all the three axes X, Y and Z. In order to clarify the feature extraction process, the building of the training set and the simulation of the network we illustrate the entire procedure in Fig. 3. The training phase incorporates both the building of the training set $\{x^n,y^n\}$ and the training of the regressor. Δm as shown in Fig. 3(a), stands for the *tracking sensitivity* of our system and its span, e.g. $[-30°,+30°]$, constrains the output of the regressor to the same range, without affecting the efficiency of the tracking process though. The regressor g, as shown in the particular example of Fig. 3(a), is a RBF-based one dedicated to the estimation of the pose of the test object y^{test} as $g(\{x^n,y^n\};y^{test})$. In order to evaluate the potential of the architecture of each regressor we have examined the corresponding mean squared errors. It was shown that the performance of a neural network can be bootstrapped by adding noise to the input vectors during the training process. Practically, according to [7], training with noise approximates Tikhonov regularization. In cases where the inputs do not contain noise and the size $\mathcal{L}$ of the training dataset tends to infinity, the error function containing the joint distributions $p(t_\lambda,x)$ (of the desired values for the network output y_λ) obtains the form

$$\begin{aligned}E&=\lim_{\mathcal{L}\to\infty}\frac{1}{2\mathcal{L}}\sum_{n=1}^{\mathcal{L}}\sum_{\lambda}\{y_\lambda(x^n;w)-t_\lambda^n\}^2\\&=\frac{1}{2}\sum_{k}\iint\{y_\lambda(x^n;w)-t_\lambda^n\}^2p(t_\lambda,x)\,dt_\lambda\,dx\\&=\frac{1}{2}\sum_{k}\iint\{y_\lambda(x^n;w)-t_\lambda^n\}^2p(t_\lambda|x)p(x)\,dt_\lambda\,dx\end{aligned}$$

Let $\boldsymbol{\delta}$ be a random vector describing the input data with probability distribution $p(\delta)$. In most of the cases, noise distribution is chosen to have zero mean ($\int\delta_i p(\delta)\,d\delta=0$) and to be uncorrelated ($\int\delta_i\delta_j p(\delta)\,d\delta=variance\sigma_{ij}$). In cases where each input data point contains additional noise and is repeated infinite times, the error function over the expanded data can be written as

$$\tilde{E}=\frac{1}{2}\sum_{k}\iint\int\{y_\lambda((x^n;w)+\delta)-t_\lambda^n\}^2p(t_\lambda|x)p(x)p(\delta)\,dt_\lambda\,dx\,d\delta$$

Expanding the network function as a Taylor series in powers of δ produces

$$y_\lambda((x^n;w)+\delta)=y_\lambda(x^n;w)+\sum_i\delta_i\frac{\partial y_\lambda}{\partial x_i}\bigg|_{\delta=0}+\frac{1}{2}\sum_i\sum_j\delta_i\delta_j\frac{\partial^2 y_\lambda}{\partial x_i\partial x_j}\bigg|_{\delta=0}+\mathcal{O}(\delta^3)$$

By substituting the Taylor series expansion into the error function we obtain the following form of regularization term that governs the Tikhonov regularization:

$$\tilde{E}=E+variance\times\Omega$$

with

$$\Omega=\frac{1}{2}\sum_k\sum_j\iint\left\{\left(\frac{\partial y_\lambda}{\partial x_i}\right)^2+\frac{1}{2}\{y_\lambda(x)-t_\lambda\}\frac{\partial^2 y_\lambda}{\partial x_i^2}\right\}p(t_\lambda|x)p(x)\,dx\,dt_\lambda$$

The final stage of the proposed framework encompasses the training of the neural network-based regressor using the set $\{x^n,y^n\}$ and the simulation of its output. The efficacy of the proposed method trained with labeled training examples was tested by means of two different experimental setups. During the first experiment, the performance levels of our work were assessed through several labeled testing images that do not belong to the training set (Fig. 8). In order to further evaluate the generalization capacities of the proposed framework, we utilized totally unregistered objects (Fig. 9). The 3 DoF pose estimations acquired are considered as successful in cases where the sum of error for the 3 rotational angles is less than 5°. In-plane and out-of-plane translations result in higher estimation errors that are due to the limited size of the training set. The testing procedure of our method may be summarized as follows:

*Step*1: For each testing image of the same object, extract ρ SIFT features that are post-processed by the homography-based RANSAC method for outliers removal.

*Step*2: Perform a clustering procedure over the locations of the extracted SIFT features, via the Fuzzy c-means algorithm.

*Step*3: Randomly select a previously extracted cluster in order to build the input testing example $\mathcal{X}_t$.

*Step*4: Present the appointed testing vector $\mathcal{X}_t$ to the built regressor (Section 3.1) in order to obtain an accurate estimation about the 3 DoF rotational parameters of the testing object.

4. Experimental results

At this point it is very important to stress the serious lack of databases devoted to 3D object pose estimation, contrasting to those existing for recognition and classification purposes. For the experimental evaluation of the proposed method we made use of the only available datasets [11,12] for object configuration approximation. In addition, the feature extraction process is accomplished using the SIFT algorithm [33] followed by homography-based RANSAC [34] for outliers' removal. The proposed system is compared with (a) a common neural network-based technique for object pose estimation [20], (b) the standard baseline work proposed by Lowe in [3] and (c) a recently presented method for efficient point selection [32]. Moreover, as it is laborious to decide the architecture of the network and no method ensures adequate generalization, network configuration is mainly based on experience, even though, certain heuristics have claimed sufficient generalization. Finally, regarding the feature extraction process, the proposed system can be easily adjusted in order to employ any combination comprising a detector and a descriptor and it is not limited by the selection of SIFT. Likewise, concerning the clustering procedure, there are no limitations, while FCM was selected in order to credit a vague classification among the extracted visual cues. According to the literature there are two ways to choose the correct number of clusters [7]: (a) the “elbow” method, where the right number of clusters is considered producing the maximum

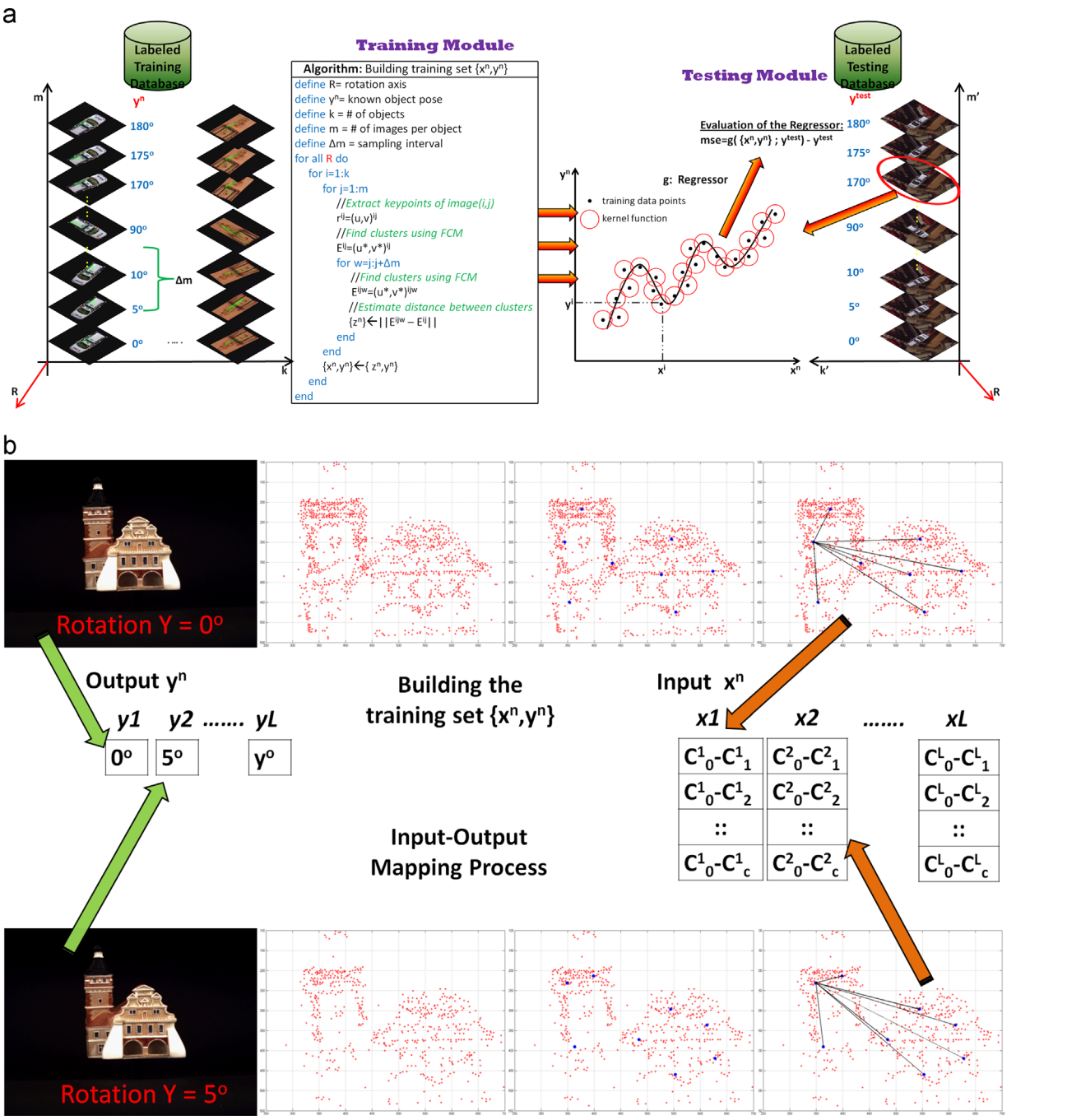

Fig. 3. The process of building the training dataset of the proposed system: (a) The training module encompasses the process of building the training set $\{x^n,y^n\}$ and that of training the neural network-based regressor. In the first instance, images of objects belonging to labeled datasets dedicated to training are drawn with the view to construct the training set to be fed to the regressor. As a final step, images of targets associated with the testing databases are further processed in order to provide an estimation of the pose of the test objects. (b) This figure demonstrates in a more illustrative way how a training example of the set $\{x^n,y^n\}$ is built. It, initially, entails the fuzzy clustering (blue spots) on the extracted SIFT features (red dots) followed by the process of estimating the distance of clusters from a known one. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

discrepancy between two neighboring values of the cost function; (b) by evaluating the clustering process based on a downstream purpose (in our case the 3 DoF object pose estimation). We adopted the second approach since we emphasize in solving the pose estimation problem and not the efficient clustering one. SIFT features for all the test images are clustered into the same number, whilst the neural network is fixed, i.e. no changes can be done on its nodes. Future work entails testing the performance of the system under several mixtures of detectors and descriptors and clustering techniques.

4.1. Training the network and convergence

We have tested the efficiency of our system for both Back-Propagation (*BP-based*) and RBF (*RBF-based*) neural networks. With regards to the Back-Propagation training schemes, several heuristics that significantly improve the algorithm's performance are presented in [35]. Among others, the tasks of normalizing inputs, stochastic versus batch update and the selection of the activation function affect directly the ability of the trained network to generalize. No special input data normalization process is

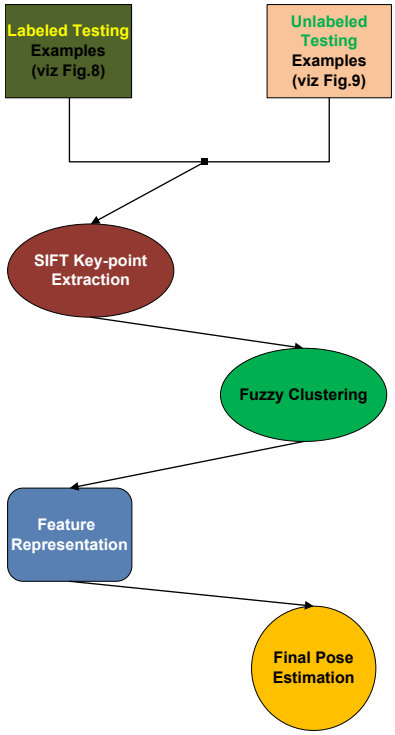

Fig. 4. A schematic of the proposed methodology.

needed in the proposed system, as it incorporates efficient input–output mapping where the inputs have zero mean value and unit standard deviation. We have tested our system's performance with training frameworks that are either batch-based or exhibit pattern-by-pattern updating. Batch-based networks are trained with the classic Levenberg–Marquardt algorithm [36] whereas stochastic nets utilize the One-Step Secant [37] method.

Training networks with large number of training examples constitute an interesting convergence procedure, since calculating the local minimum instead of the global one is quite common. In order to overcome such problems, we have adopted a regularization process which also improves the generalization ability of the network. Such a generalization procedure, in its general form, involves modifying the performance function by a new penalty term that incorporates the weights and biases of the networks. The proposed system is further tested when the training function is the Bayesian regularization framework as presented in [38]. Moreover, it is apparent that the size of the training set $\{x^n, y^n\}$ of our system depends on the number of clusters c extracted and the *tracking sensitivity* Δm.

4.2. Dealing with occlusions and training with noise

In computer vision applications, a very interesting facet comes to light in cases where the testing object is disturbed by partial occlusions. Unlike humans who are able to simultaneously recognize an object and estimate its pose in such conditions, vision algorithms fall considerably short to achieving any similar results. We have aimed at providing a solution to this problem by enhancing our training set with images of partially occluded objects. Partial occlusions are introduced artificially in the existing databases with the percentage of obstruction lying in the range [0–95]. Moreover, it was shown in Section 3.2 that a network's performance, in cases of a large training example, is bootstrapped by adding noise to the input vectors during the training process. The training examples of our system include both partially occluded objects and new vectors characterized as "noisy". By adopting the evaluation criterion presented in [32], we have tested our system's variance against occlusions and compared it with other highly related work producing the results depicted in Fig. 5 (a). In this figure, the condensed results from simulated networks with over 200 testing examples are presented. According to the adopted evaluation criterion, an estimation of the pose is considered successful, if the error of the estimated rotation parameters is less than 5°. The RBF-based version of our system presented invariance against partial occlusions compared to the Hinterstoiser et al. [32], Lowe [3] and Yuan et al. [20] methods. We have further evaluated all methods for a permissible error of 3° and the results are illustrated in Fig. 5(b). Compared to the results depicted in Fig. 5(a), RBF-based nets still constitute the most effective solution also for this problem, while BP-based ones proved to be more tolerant to partial occlusions than Hinterstoisser et al. [32]. In turn, the PCA-based approach [20] documented high variance to partial occlusions. We would like to state that for the comparison of the proposed method and the PCA-based one identical network architectures are utilized. Regarding the results provided by the Lowe [3], they were somehow expected since in [3], train and target models are assumed to be linearly related when it turns out they are not. Fig. 8 illustrates the representative visual results of the proposed method under varying percentages of partial occlusion.[1]

4.3. Tracking sensitivity Δm and comparison of network architectures

In Section 3.2 we defined Δm as the *tracking sensitivity* of our method, which in turn, represents the range of the output of the regressor, i.e. if $\Delta m = 1 \rightarrow [-5°, +5°]$ or if $\Delta m = 6 \rightarrow [-30°, +30°]$. Given a standard vision sensor that is able to capture 30 frames per second, the latter corresponds to a tracking ability of either 300 deg./s or to 1800 deg./s. In other words, the proposed system is able to efficiently track even unregistered objects and estimate their pose regardless of movement velocity. Moreover, in order to illustrate the influence of different network architectures on our system's performance, we present relevant comparative results as shown in Fig. 6. Training functions are parameterized with the number of clusters extracted by the FCM algorithm, whilst RBF family networks proved to be more stable along several variations of Δm. Concerning the Back-propagation-based networks the best generalization error was achieved when training for 1000 epochs with 2 hidden layers of 40 and 20 neurons respectively, using the mean squared error as the performance evaluation function. Concretely, we present the effect of the network structures for nets trained with Bayesian regularization (Fig. 7(a)), the One-Step Secant algorithm (Fig. 7(b)), the classic Levenberg–Marquardt algorithm (Fig. 7(c)) and with Radial Basis Functions (Fig. 7(d)). The visual outcome of the proposed framework whilst tracking unregistered objects is depicted in Fig. 9 where an unknown test example was presented to the system.

The size of the training set, without the addition of noise or artificially generated occlusions, depends on the number of clusters c and the *tracking sensitivity* Δm. For $\Delta m = 6$, i.e. a tracking range $[-30°, +30°]$, the size of training set is $[c \times 24\,304]$ that may rise to $[c \times 100\,000]$ by adding noise and occlusion parameters. This results in large computational burden for a standard computer. Practically, stochastic update-based back-propagation networks would require a 12 h training process on an average Windows-based PC with 8 GB RAM. Generally, BP-based architectures were less demanding than RBF-based ones since the training

[1] The effect of partial occlusions on the performance of the proposed work is presented in the following video: http://www.youtube.com/watch?v=SS-ZH1Jllr8.

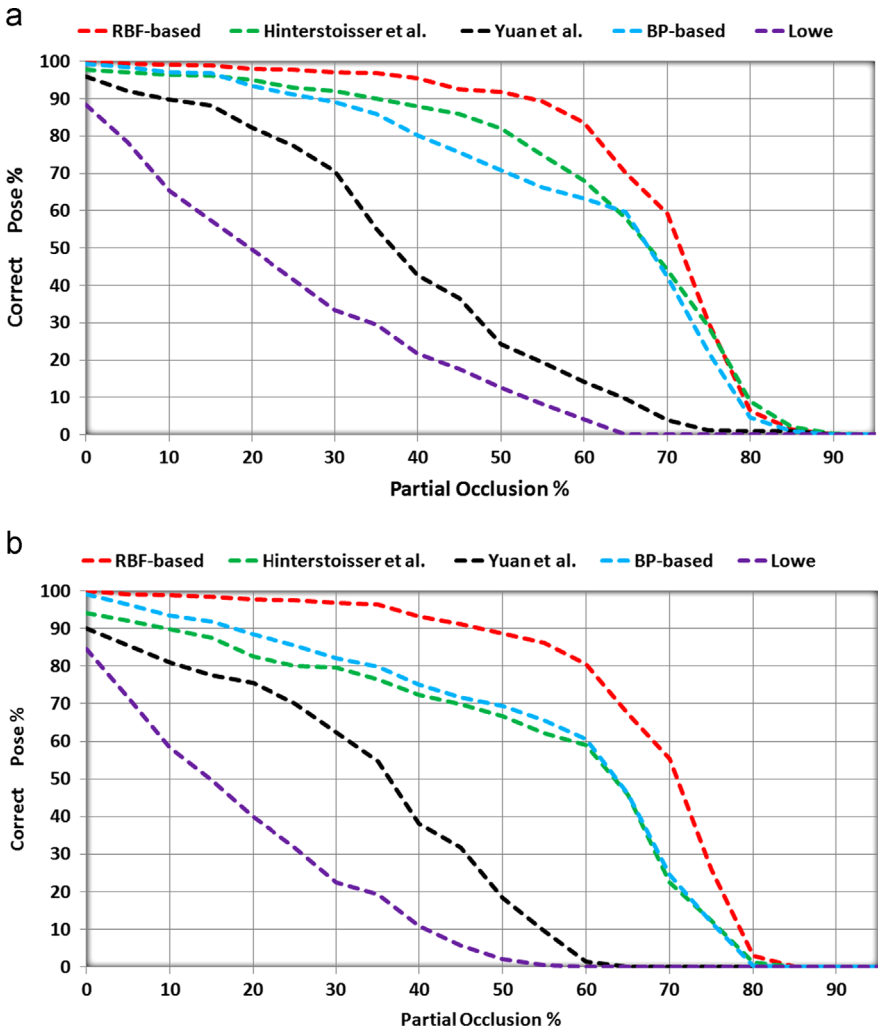

Fig. 5. Comparative evaluation of the proposed framework in cases where an estimation of the pose is considered successful if the error of the estimated rotation parameters is less than 5 (a) and 3 (b) degrees, respectively. (a) The RBF-based version of the proposed framework is seen to be more tolerant to partial occlusions, as compared to the one based on BP-based. It can also be seen that both the PCA-based technique [20] and the one presented in [3] are highly affected by partial obstruction. The results presented in this paper are contrary to the satisfactory performance levels reported in [32]. (b) This figure presents comparative results in cases where an estimation of the pose is considered successful if the error of the estimated rotation parameters is less than 3°. RBF-based nets proved to be more tolerant to partial occlusions than BP-based ones, whilst both methods outperformed those presented in [32,20,3].

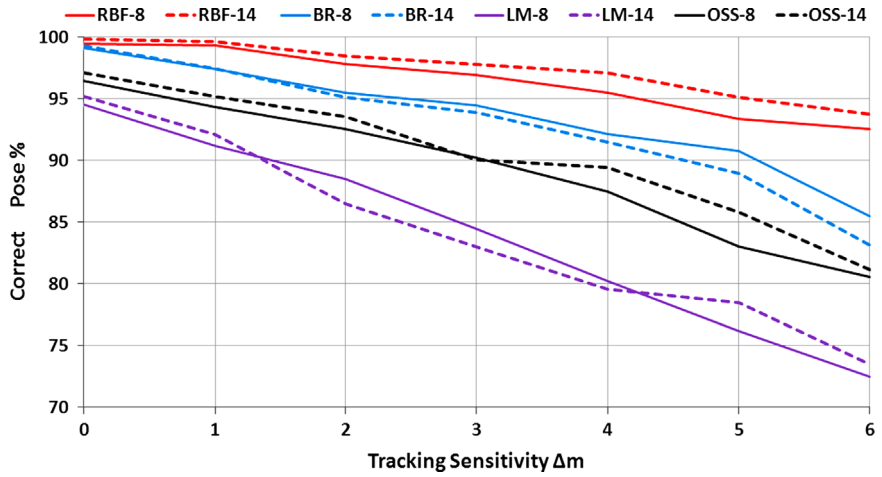

Fig. 6. During tests the training functions of the networks are parameterized on the number of extracted clusters basis. Dashed lines represent networks based on 14 clusters while solid lines correspond to nets based on 8 clusters with the selection of both of these figures based on numerous series of tests. Networks utilizing the RBF training functions have shown the best tracking efficiency, followed by nets based on Bayesian Regularization, One-Step Secant and Levenberg–Marquardt, in that order.

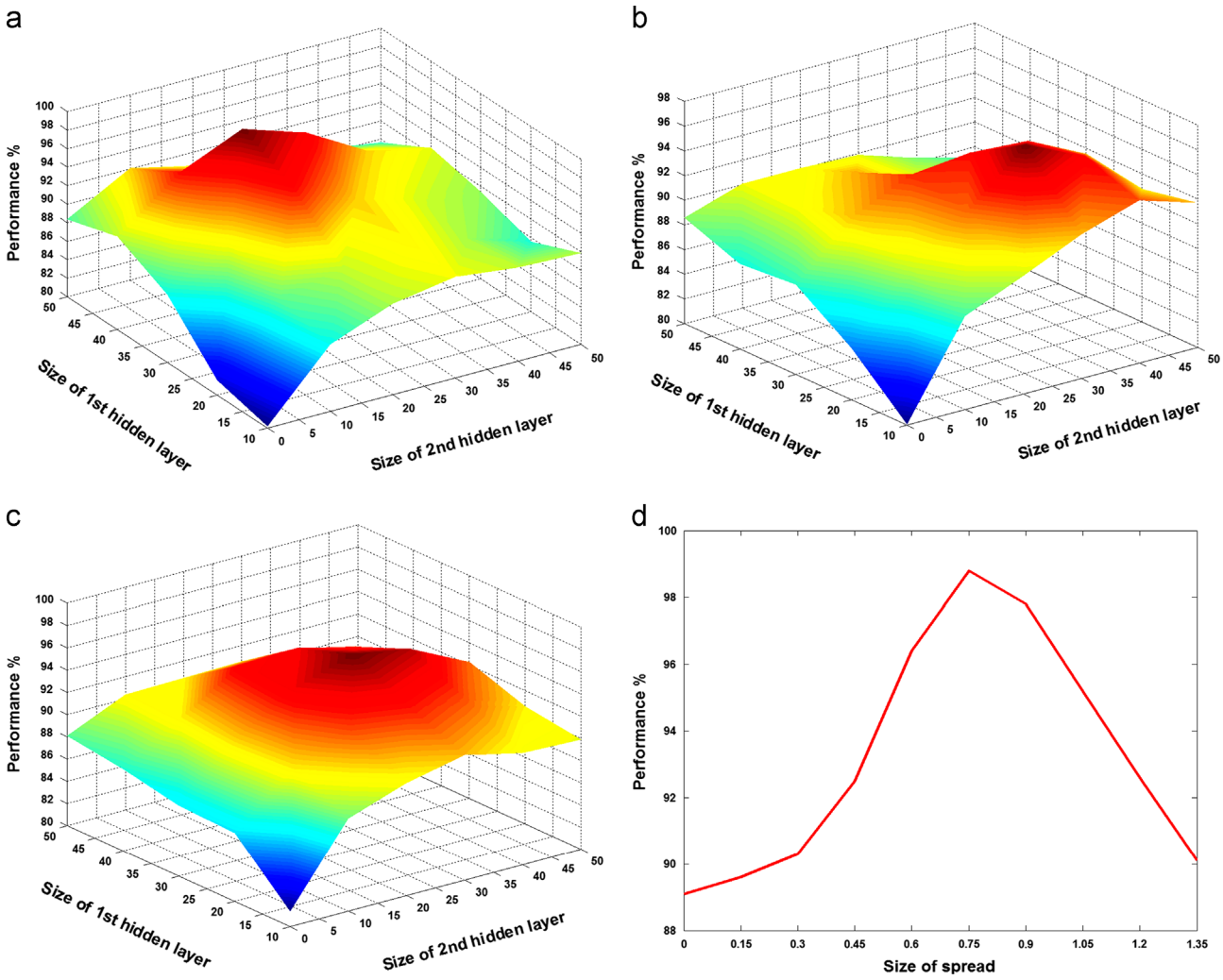

Fig. 7. This figure demonstrates the effect of network structures on the performance of our algorithm. The accuracy of the pose estimation method when training with only one hidden layer was significantly lower than with two ones. However, in the case that the number of layers is larger than two, the calculated mean squared error on the testing dataset rises dramatically. The architecture of the net plays also a significant role in the proposed method: the effect of the network size in the pose estimation accuracy when training with (a) Bayesian regularization; (b) the One-Step Secant; (c) the Levenberg–Marquart algorithm; (d) regarding the RBF-based networks, higher performance levels are exhibited for spread = 0.75.

of the latter may last 220 h (on the expanded version of training dataset). Fig. 10 illustrates additional visual results of the proposed system with varying percentages of partial occlusions.

5. Discussion

In this paper a new neural-based solution to the 3D object pose estimation problem was presented. The proposed system is based on a novel input–output mapping with the learning process being guided by a fuzzy clustering of the extracted visual cues. This new input–output mapping comprises a key contribution of this paper, by moving away from conservative dimensionality reduction schemes, such as the PCA, which inevitably incurs information loss. Concretely, our method builds an input vector of maximum 14 dimensions where in PCA-based methods, e.g. in [20], the dimensionality of the input space is around 65 000. A series of experimental setups were studied and evaluated proving, thus, the validity of our approach when compared to other works in this area. After training with the available databases for 3D object configuration, the proposed system has been shown to be able to estimate the 3D pose of any object, up to a scale, with remarkable accuracy. Additionally, unregistered objects are efficiently tracked in cases they lay in image planes that are proportional to those used for training. Furthermore, evidence has been provided that our system is more tolerant to partial occlusions, compared to other related projects. This is achieved with the introduction of *tracking sensitivity*, augmenting the tracking performance of the system. As far as the neural network part of the framework is concerned, after several experiments, RBF-based training functions are shown to achieve the least generalization error as opposed to the Back-Propagation-based ones. Moreover, experimental results revealed two major issues: (a) The relation between the input space, which is defined over the trained objects, and the output one, characterizing the target models, is a non-linear one; (b) our choice to adopt a neural network-based strategy, which efficiently captured this non-linear binding, was totally justified. We believe that the efficacy of the proposed method is primary due to its input–output mapping and feature extraction and, secondary, due to the attributes of the utilized network, i.e. training with noise, RBF kernel. Looking ahead to future work, the authors plan to construct a new database for 3D object pose estimation, with translation parameters included. Moreover, a global regressor based on committees of neural networks, to be

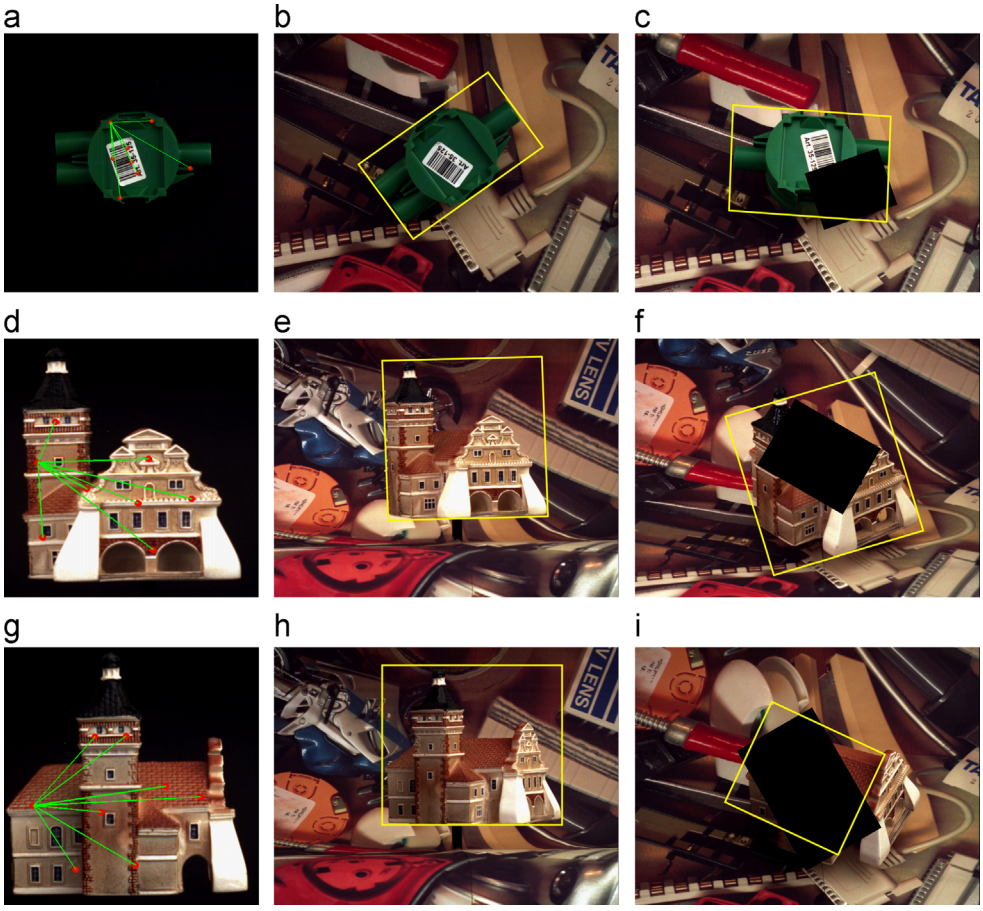

Fig. 8. This figure presents the outcome of the proposed framework for the test objects of (a), (d) and (g) under varying percentage of artificially generated (due to database shortages) partial occlusions. In the first two examples (b)–(f), our system is able to estimate the pose of the target remarkably well whilst, in the last case (i), partial occlusions lead to a slight deterioration of our system's performance.

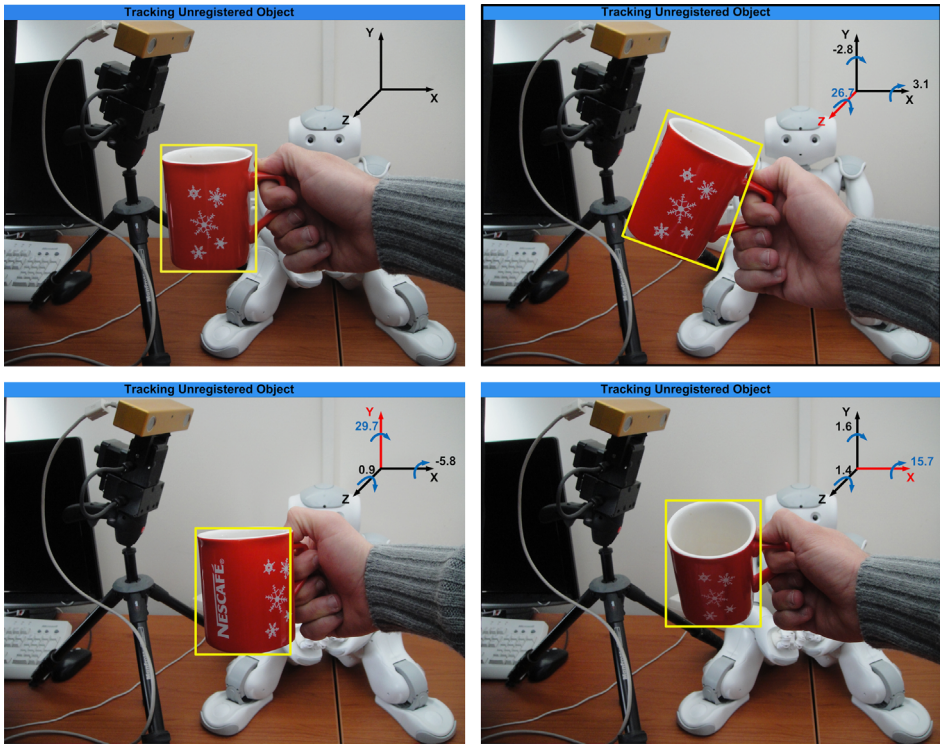

Fig. 9. Tracking an unregistered object in cluttered environment under different rotations over the three axes *X*, *Y* and *Z*.

Fig. 10. Visual outcome of the presented neural network-based framework for five testing objects under varying percentage of artificially generated partial occlusions.

able to integrate information derived from numerous pattern selection frameworks, is also considered.

References

[1] R. Detry, J. Piater, Continuous surface-point distributions for 3d object pose estimation and recognition, in: Asian Conference on Computer Vision, vol. 1, 2011, pp. 572–585.

[2] J. Ma, T. Chung, J. Burdick, A probabilistic framework for object search with 6-dof pose estimation, Int. J. Robotics Res. 30 (2011) 1209–1228.

[3] D. Lowe, Object recognition from local scale-invariant features, in: International Conference on Computer Vision, vol. 2, 1999, pp. 1150–1157.

[4] N. Cornelis, B. Leibe, K. Cornelis, L. Van Gool, 3d urban scene modeling integrating recognition and reconstruction, Int. J. Comput. Vision 78 (2008) 121–141.

[5] B. Rasolzadeh, M. Björkman, K. Huebner, D. Kragic, An active vision system for detecting, fixating and manipulating objects in the real world, Int. J. Robotics Res. 29 (2010) 133.

[6] R. Kouskouridas, A. Amanatiadis, A. Gasteratos, Guiding a robotic gripper by visual feedback for object manipulation tasks, in: International Conference on Mechatronics, 2011, pp. 433–438.

[7] C. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.

[8] Y. Pang, X. Li, Y. Yuan, D. Tao, J. Pan, Fast haar transform based feature extraction for face representation and recognition, IEEE Trans. Inf. Forensics Secur. 4 (2009) 441–450.

[9] Y. Pang, Y. Yuan, X. Li, Iterative subspace analysis based on feature line distance, IEEE Trans. Image Process. 18 (2009) 903–907.

[10] Y. Pang, X. Li, Y. Yuan, Robust tensor analysis with l1-norm, IEEE Trans. Circuits Syst. Video Technol. 20 (2010) 172–178.

[11] F. Viksten, P.-E. Forssén, B. Johansson, A. Moe, Comparison of local image descriptors for full 6 degree-of-freedom pose estimation, in: International Conference on Robotics and Automation, 2009.

[12] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (COIL-100), Technical Report, Columbia University, 1996.

[13] J. Bezdek, R. Ehrlich, et al., Fcm: The fuzzy *c*-means clustering algorithm, Comput. Geosci. 10 (1984) 191–203.

[14] S. Winkler, Model-based Pose Estimation of 3-d Objects From Camera Images by Using Neural Networks, Technical Report 515-96-12, 1996.
[15] P. Wunsch, S. Winkler, G. Hirzinger, Real-time pose estimation of 3d objects from camera images using neural networks, in: International Conference on Robotics and Automation, vol. 4, 1997, pp. 3232–3237.
[16] E. Chinellato, A. del Pobil, Distance and orientation estimation of graspable objects in natural and artificial systems, Neurocomputing 72 (2009) 879–886.
[17] W. Li, T. Lee, Projective invariant object recognition by a Hopfield network, Neurocomputing 62 (2004) 1–18.
[18] R. Nian, G. Ji, W. Zhao, C. Feng, Probabilistic 3d object recognition from 2d invariant view sequence based on similarity, Neurocomputing 70 (2007) 785–793.
[19] C. Yuan, H. Niemann, Object localization in 2d images based on Kohonen's self-organization feature maps, in: International Joint Conference on Neural Networks, vol. 5, 1999, pp. 3134–3137.
[20] C. Yuan, H. Niemann, Neural networks for the recognition and pose estimation of 3d objects from a single 2d perspective view, Image Vision Comput. 19 (2001) 585–592.
[21] C. Yuan, H. Niemann, Neural networks for appearance-based 3-d object recognition, Neurocomputing 51 (2003) 249–264.
[22] A. Saalbach, G. Heidemann, H. Ritter, Parametrized soms for object recognition and pose estimation, Artif. Neural Networks 2 (2002) 140.
[23] A. Castro, Y. Frauel, E. Tepichín, B. Javidi, Pose estimation from a two-dimensional view by use of composite correlation filters and neural networks, Appl. Opt. 42 (2003) 5882–5890.
[24] S. Hati, S. Sengupta, Pose estimation in automated visual inspection using ANN, Syst. Man Cybern. 2 (1998) 1732–1737.
[25] A. Berg, T. Berg, J. Malik, Shape matching and object recognition using low distortion correspondences, in: International Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 26–33.
[26] R. Fergus, P. Perona, A. Zisserman, Weakly supervised scale-invariant learning of models for visual recognition, Int. J. Comput. Vision 71 (2007) 273–303.
[27] B. Leibe, A. Leonardis, B. Schiele, Combined object categorization and segmentation with an implicit shape model, in: Workshop on Statistical Learning in Computer Vision. European Conference on Computer Vision, vol. 1, 2004, pp. 17–32.
[28] Y. Pang, W. Li, Y. Yuan, J. Pan, Fully affine invariant surf for image matching, Neurocomputing 85 (2012) 6–10.
[29] K. Bailly, M. Milgram, Boosting feature selection for neural network based regression, Neural Networks 22 (2009) 748–756.
[30] S. Savarese, L. Fei-Fei, 3d generic object categorization, localization and pose estimation, in: International Conference on Computer Vision, vol. 1, 2007, pp. 1–8.
[31] L. Mei, J. Liu, A. Hero, S. Savarese, Robust object pose estimation via statistical manifold modeling, in: International Conference on Computer Vision, 2011.
[32] S. Hinterstoisser, S. Benhimane, N. Navab, N3m: natural 3d markers for real-time object detection and pose estimation, in: International Conference on Computer Vision, vol. 1, 2007, pp. 1–7.
[33] D. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2004) 91–110.
[34] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (1981) 381–395.
[35] S. Haykin, Neural Networks and Learning Machines, 3rd ed., Prentice Hall, 2009.
[36] J. More, The Levenberg–Marquardt algorithm: implementation and theory, Numer. Anal. 1 (1978) 105–116.
[37] M. Hagan, M. Menhaj, Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Networks 5 (1994) 989–993.
[38] D. MacKay, Bayesian interpolation, Neural Comput. 4 (1992) 415–447.

Rigas Kouskouridas received the Diploma degree from the Department of Production and Management Engineering at the Democritus University of Thrace, Xanthi, Greece, in 2006. He is currently working towards the Ph.D. degree with the Laboratory of Robotics and Automation, Department of Production and Management Engineering, Democritus University of Thrace. His areas of interest include pattern recognition, machine learning, multicamera systems and robotics. He is involved in several national (Greek) and international (European) research projects in the field of machine vision systems. Mr. Kouskouridas is a member of the IEEE, euCognition II, the Technical Chamber of Greece (TEE), and the National Union of Production and Management Engineers.

Antonios Gasteratos is an Assistant Professor of "Mechatronics and Artificial Vision" at the DPME. He teaches the courses of "Robotics", "Automatic Control Systems", "Measurements Technology" and "Electronics". He holds a Diploma and a Ph.D. from the Department of Electrical and Computer Engineering, DUTH, Greece, 1994 and 1999, respectively. During 1999–2000 he was a Post-Doc Fellow at the Laboratory of Integrated Advanced Robotics (LIRA-Lab), DIST, University of Genoa, Italy. He has served as a reviewer to numerous of scientific journals and international conferences. He is the Greek Associate High Level Group (HLG) Delegate at EUREKA initiative. His research interests are mainly in mechatronics and in robot vision. He has published one textbook, three book chapters and more than 90 scientific papers. He is a member of the IEEE, IAPR, ECCAI, EURASIP and the Technical Chamber of Greece (TEE). Dr. Gasteratos is a member of EURON, euCognition and I*PROMS European networks. He organized the International Conference on Computer Vision Systems (ICVS 2008).

Christos Emmanouilidis is a Senior Researcher & Head of the Computational Systems & Applications Department at the CETI Institute of the ATHENA Research & Innovation Centre in Information, Knowledge and Communication Technologies. He holds a Diploma in Electrical Engineering from the Aristotle University of Thessaloniki, Greece (1992), an M.Sc. from the School of Engineering, University of Durham, UK (1998) and a Ph.D. from the School of Computing and Engineering Technology, University of Sunderland, UK (2002). He has acted as a Visiting Professor at the Department of Industrial Engineering, Politecnico di Milano, as adjunct Assistant Professor at the Department of Production and Management Engineering & the Department of Electrical & Computer Engineering, Democritus University of Thrace, having taught Integrated Industrial Information Systems, Asset Lifecycle Management, Operating Systems, Pattern Recognition and Decision Analysis & Game Theory. He is a member of the IEEE, member of the Technical Chamber of Greece, founding, board member and vice-chair of the Hellenic Maintenance Society, Founding Fellow of the International Society of Engineering Asset Management and a member of the Engineering Asset Management Committee of the European Federation of National Maintenance Societies.