Robotics and Computer-Integrated Manufacturing 29 (2013) 12–22

Contents lists available at SciVerse ScienceDirect

Robotics and Computer-Integrated Manufacturing

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

Object recognition using laser range finder and machine learning techniques

Andry Maykol Pinto*, Luís F. Rocha, A. Paulo Moreira

Faculty of Engineering of University of Porto Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal

ARTICLE INFO

Article history:
Received 13 September 2011
Received in revised form
2 April 2012
Accepted 13 June 2012
Available online 26 July 2012

Keywords:
Eye-in-hand
Laser ranger finder
Machine learning
Manufacturing robotic manipulator
Object recognition

ABSTRACT

In recent years, computer vision has been widely used on industrial environments, allowing robots to perform important tasks like quality control, inspection and recognition. Vision systems are typically used to determine the position and orientation of objects in the workstation, enabling them to be transported and assembled by a robotic cell (e.g. industrial manipulator). These systems commonly resort to CCD (Charge-Coupled Device) Cameras fixed and located in a particular work area or attached directly to the robotic arm (eye-in-hand vision system). Although it is a valid approach, the performance of these vision systems is directly influenced by the industrial environment lighting. Taking all these into consideration, a new approach is proposed for eye-on-hand systems, where the use of cameras will be replaced by the 2D Laser Range Finder (LRF). The LRF will be attached to a robotic manipulator, which executes a pre-defined path to produce grayscale images of the workstation. With this technique the environment lighting interference is minimized resulting in a more reliable and robust computer vision system. After the grayscale image is created, this work focuses on the recognition and classification of different objects using inherent features (based on the invariant moments of Hu) with the most well-known machine learning models: *k*-Nearest Neighbor (kNN), Neural Networks (NNs) and Support Vector Machines (SVMs). In order to achieve a good performance for each classification model, a wrapper method is used to select one good subset of features, as well as an assessment model technique called K-fold cross-validation to adjust the parameters of the classifiers. The performance of the models is also compared, achieving performances of 83.5% for kNN, 95.5% for the NN and 98.9% for the SVM (generalized accuracy). These high performances are related with the feature selection algorithm based on the simulated annealing heuristic, and the model assessment (*k*-fold cross-validation). It makes possible to identify the most important features in the recognition process, as well as the adjustment of the best parameters for the machine learning models, increasing the classification ratio of the work objects present in the robot's environment.

1. Introduction

Taking into consideration the current state of the world, mass production industries are searching for ways of improving their efficiency and reducing operating costs through automation, having as purpose to achieve more competitiveness. Therefore increasing the reliability of industrial procedures and techniques. Despite this effort, robotic production lines still have some limitations because there is a constant need for manual reprogramming every time it is necessary to create a new product. Nowadays, the transportation and machining actions are increasingly performed by robotic manipulators. Changes in some product characteristics lead to a new reprogramming effort which causes production process to be stopped.

Over the past few years, computer vision systems have been used to automatically distinguish work objects [5,17,2,19]. Normally, they resort to cameras usually placed on a fixed location. In this field, several different approaches can be found in the literature. However, almost all authors agree with the need for a vision system capable of satisfying some critical requirements like: accuracy, speed and reliability regarding changes in lighting.

In [8], an industrial machine vision application for object recognition based on CCD cameras, invariant moments and neural-network (NN) was presented to classify moving objects on the conveyor. They claim the achievement of a 100% accurate classifier. However, only five objects were used and any method that can prevent the overfitting of the data was taken during the NN training. Therefore, it is possible to expect the classification model to be less accurate relatively to the one that was obtained. Furthermore, there is another relevant issue related to the lighting and noise problems since the system is based on cameras and invariant moments (as the feature vector set), which are not immune to those aspects.

* Corresponding author.
E-mail addresses: andry.pinto@fe.up.pt (A.M. Pinto), luis.andre@fe.up.pt (L.F. Rocha), amoreira@fe.up.pt (A. Paulo Moreira).

0736-5845/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.rcim.2012.06.002

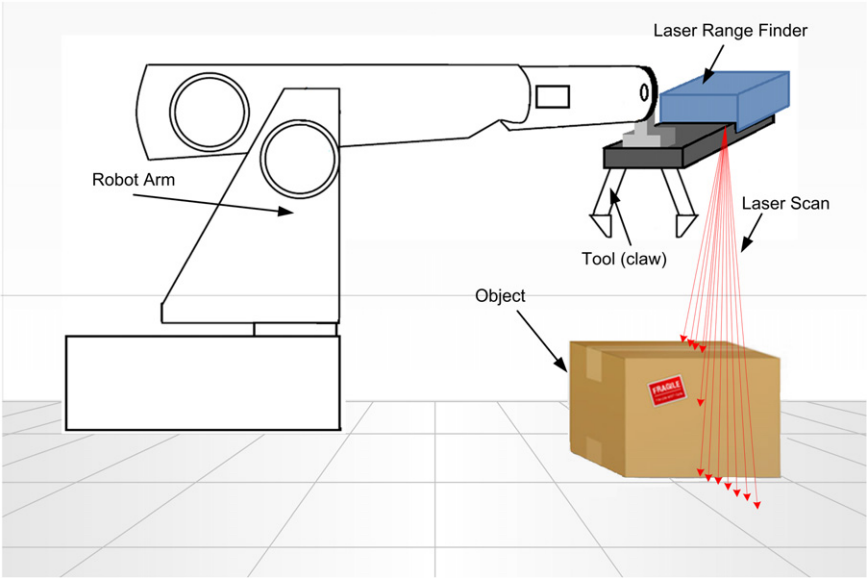

Fig. 1. Manipulators workspace scanning procedure.

In [15,16], a CCD camera attached to the end effector of the manipulator (eye-in-hand) is used as a ranging sensor to guide the robot manipulator, placed on the mobile base, to perform the pick-and-place operation. Their work focus on the uncalibrated method for the eye-in-hand vision to control the pick up procedure of the work object. However, no information on object recognition is provided. One important remark to make is also related to the lightning factors, because according to the authors, their work does not involve any special lighting agent as a light source for image processing.

In more recent works [7,3], it is already possible to find research with 2D laser range finders performing 3D scene reconstruction. However, this is usually done in mobile navigation robots and not with industrial systems. There are some solutions based on laser technology for the recognition of parts in industrial environments but normally they are laser beam sensors [18]. There are a few incipient works with LRF in this area [10,12,9]. The reason why they are incipient is because they focus on object reconstruction, using the LRF information, and not specifically on data mining, which is the identification of the object itself and the accuracy of the classification algorithm.

For our work, instead of using an architecture similar to the eye-in-hand process based on CCD cameras, where performance is significantly influenced by ambient lighting, we propose the use of an architecture based on a LRF. This is used as a computer vision system, coupled strategically in the manipulator tool (see Fig. 1).

With this structure, it is possible for the manipulator to perform an exploratory movement around the work environment to create a representative image (2D image) of the scene. Subsequently, image processing and machine learning methods [4,14] are applied and compared (*k*-nearest neighbor, support vector machine and neural networks) in order to obtain invariant features so that it is possible to perform a correct characterization and classification of the work objects. This processing will make it possible to identify and determine the object's *pose*. Hence the manipulator can correctly grab the object and transport it to a specific location (depending on the industrial application).

A more detailed description of this sequence (manipulator movement and image creation) will be provided later in Section 2. Section 3 explains which features are used in the characterization and the classification of the objects is performed in Section 4. Finally, in Section 5, some conclusions are presented on this work.

2. Exploratory scan

Today's industrial environment is very harsh and aggressive, where artificial vision systems based on ordinary cameras tend to be influenced by magnetic interference and variations in ambient light. These factors may affect the overall performance of the manipulator, reducing its efficiency or creating some operational restrictions.

Bearing this in mind, our first main objective is to extract the position and orientation (*pose*) of the work objects so that the manipulator can interact with it. To make the whole operation more flexible, autonomous and less influenced by outside interferences, the approach presented in this work consists of using a laser range finder (attached to the manipulator), as a way to obtain the images of the scene. Therefore to obtain these images, the manipulator only needs to scan the workplace, where at each time the distance between the laser and the possible work object present in the workstation are indicated. Subsequently, the image is scaled in shades of gray (dark gray indicates smaller distances, and vise versa for light gray). A 2D representation of the scene is obtained, where every pixel has information on depth.

After completing the exploratory scan it is necessary to recognize the object using a description database and its *pose*. Image processing methods become crucial when it comes to extracting features, thus making possible to distinguish and identify each object.

2.1. Image creation

2.1.1. Laser range finder

The laser range finders are sensors often used in the robotic applications. Those sensors are equipped with a mechanical system that makes a mirror spin with an angular resolution, allowing the IR light to cover in sequence a set of straight segments with different orientations. Because of the mechanical associated constraints the sensor performs a series of measures within a limited angle, that means an angle that does not reach 360°, which originate an area called blind or dead zone.

In this work, it is applied an LIDAR Hokuyo Automatic (URG-04lx)[1] that uses the difference of phase between the emitted and received infra-red rays to estimate the distances to the objects.

2.1.2. Concept

In order to understand the algorithm developed to create the image, it is fundamental to explain what is the meaning of exploratory movement.

Based on the assumption that a 2D laser range finder is used and it is coupled to the robotic arm, at each time, we only have a plane of distances that could be seen as array.

Therefore if we want to sense the workspace, it is necessary to produce some laser movement around the scene, designated as exploratory movement, that aim to reach and characterize all desirable points (table, ground, pieces, etc.). During the exploratory movement the laser provides a continuous range of distances (each distance *Di* has an angle θ_i) associated. Upon receiving the scan, the distances are converted to *XY* points, relatively to the laser reference in which the *X*-axes reflect the vertical distance and the *Y*-axes constitute the horizontal distance of the measured point (the relative separation).

In a simple matter, the *X*-value of each point (D_X) provides information on their depth, scaled in values between 0 and 255 using the height of the laser in the workstation. Otherwise, the *Y*-value (D_Y) offers information on the separation of some points concerning the center *j* of the laser scan, where $\theta_j = 0$ (Fig. 2). Considering that we are working with an array with the dimension of $1 \times ImageWidth$, the index is obtained by the *Y*-value.

Obviously not all the points provided by the laser are used to create the image (see Fig. 3). Instead, only those that are within an angle range of $[-\Delta\theta; \Delta\theta]$ are used. This interval is automatically determined from the height of the laser to the work plan, the width of the desired image and the image resolution.

After this point, we have the *X*-value and *Y*-value, vertical and horizontal distance (D_X and D_Y).

2.1.3. Movement description

Next we will explain how the manipulator executes the exploratory movement to create the image, and then how it picks up and transports the object to a specified location in the workstation.

1. The first step is to move the manipulator, which could be in an arbitrary location to an initial position (Pinit) of the exploratory movement.
2. Then the scanning process begins with the movement (at predefined velocity) until the Pend is reached (see Fig. 4).
3. After creating the image and identifying the object, the manipulator claw will be moved to Pgrab (position of the object) in order to pick up the object.
4. Finally, the robotic arm will move that object to Pput, which is a specified location to place the object (Fig. 5).

2.1.4. Control system

The control system was developed with the following objectives: control the movement of the robotic arm, receive the laser data and then, process only the desirable points (taking into account the laser height and the expected image size), apply a dynamic median filter to all the points, create the image (use the set of horizontal distances, D_Y, to interpolate the pixels that do not have direct correspondence), and communicate with a high level program that uses the image to characterize and identify the correct object that is present in the robot workstation. This program will return the orientation and position of the identified object, allowing the control system to operate the robotic arm (to finish the task of grabbing and transporting the object correctly).

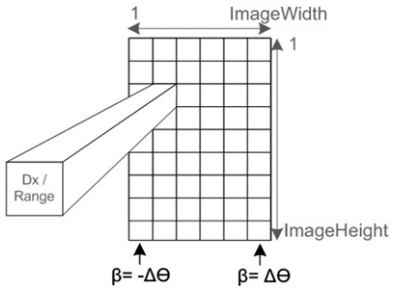

Fig. 2. Structure of the image created by scanning.

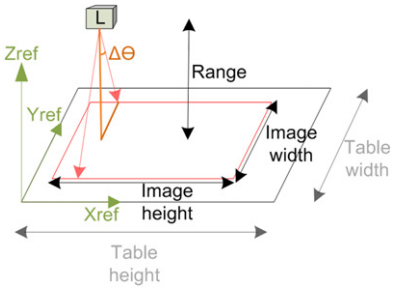

Fig. 3. Using the laser distances to characterize the environment.

2.2. System architecture

In conclusion, the architecture of the whole system that was developed can be seen in Fig. 6. Due to the large number of iterations between the subsystems, the interoperability, reliability and robustness of each element were fundamental to ensure the proper functioning of the entire system.

After creating the image, the control system interacts with the recognition system and this last one applies the most promising classification model (the one that has the best performance—see Section 4). Finally, after classifying the object, its position and orientation are sent back to the control system and the manipulator receives the object's pose information.

3. Features extraction

An image contains a significant amount of information that can be analyzed by a specific algorithm. Those algorithms execute some tasks allowing automatic systems to get information on some fundamental characteristics of the scene. One of the most important aspects in image processing is object extraction and recognition.

3.1. Invariant moments

Determining invariant moments[2] is an interesting procedure when evaluating the object's shape, particularly because they do

[1] http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx_ug01.html

[2] Specific method used to obtain features from the object represented in the image, based on the statistics (to characterize the probability density function).

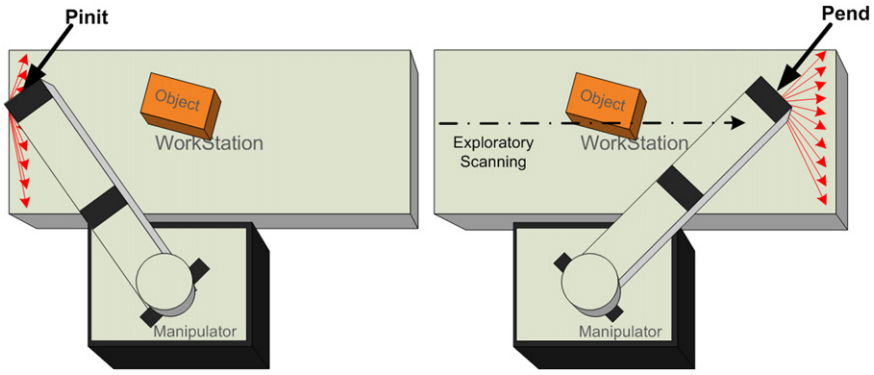

Fig. 4. Exploratory scanning: initial (Pinit) and final (Pend) positions.

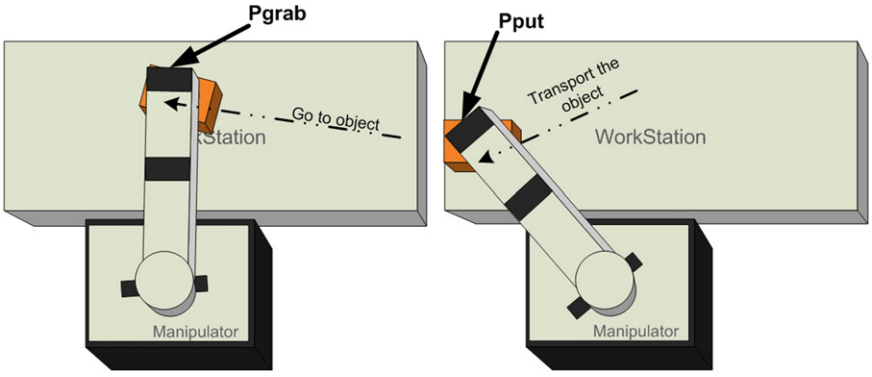

Fig. 5. Manipulator movement: the grab (Pgrab) and the predefined position (Pput).

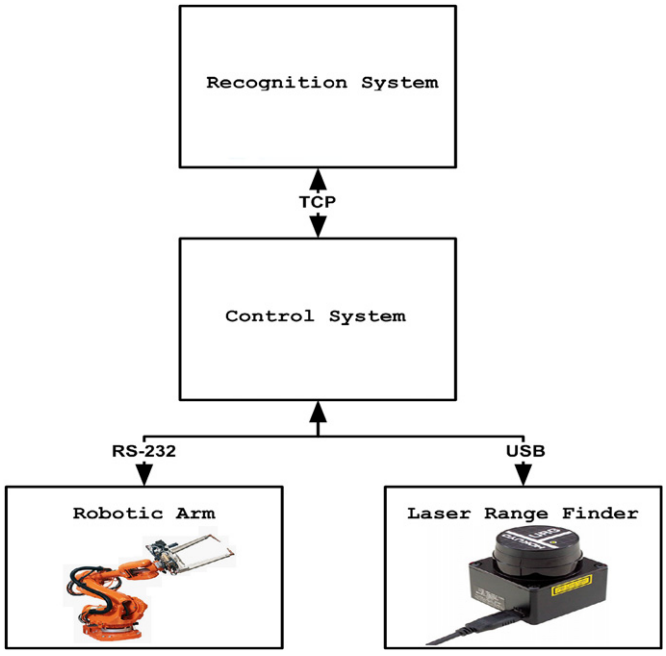

Fig. 6. System architecture.

not depend on scaling, translation and rotation. For this reason, they give the impression to be the most promising method and has been used extensively to describe the objects (based on a set of parameters that represent particular properties) [6]. To discriminate different objects, it is simply necessary to distinguish the parameters that belong to each class, preserving the image's information [13].

3.1.1. Geometric moments

Using $p_{pq}(x,y)=x^p y^q$ as the basis function the geometric moment, m_{pq}, will be defined by (considering the discrete form)

$$m_{pq}=\sum_{i=1}^{M}\sum_{j=1}^{N} x_i^p y_j^q f_{XY}(x_i,y_j) \tag{1}$$

Geometric moments of low orders are widely used to determine intuitive parameters, for instance the zero order moment that represents the "*mass*" of the image or area in binary images, m_{00}, and the first order moments (m_{10} and m_{01}) that provide information on the "*centroid*" location, $\overline{x}$ and $\overline{y}$:

$$\overline{x}=\frac{m_{10}}{m_{00}}, \quad \overline{y}=\frac{m_{01}}{m_{00}} \tag{2}$$

3.1.2. Central geometric moments

If the object's centroid coincides with the origin of the coordinate system, the moments will be invariant to the translation and rotation [6]. This can be achieved by calculating the central geometric moments:

$$\mu_{pq}=\sum_{i=1}^{M}\sum_{j=1}^{N}(x_i-\overline{x})^p(y_j-\overline{y})^q f_{XY}(x_i,y_j) \tag{3}$$

Calculating the second order moments, it is possible to determine the covariance matrix, J and the rotation angle,[3] ϕ

$$J=\begin{bmatrix}\frac{\mu_{20}}{m_{00}} & \frac{\mu_{11}}{m_{00}}\\ \frac{\mu_{11}}{m_{00}} & \frac{\mu_{02}}{m_{00}}\end{bmatrix} \tag{4}$$

[3] Angle between the x-axis and the axes associated to the higher eigenvector of J.

$$\phi = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right) \quad (5)$$

Eccentricity can also be calculated with the eigenvalues (λ_1,λ_2), of the covariance matrix: $Eccentricity = \sqrt{1-\lambda_2/\lambda_1}$. This value represents a relation between the semi-major and semi-minor axes length of the image.

3.1.3. Normalized central geometric moments

To make the moments invariant to translation, rotation and scaling, it is necessary to perform a proper normalization of the central geometric moments. The formula used more often for this normalization is [6]

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{((p+q)/2)+1}} \quad (6)$$

The result, η_{pq}, is called *normalized central geometric moments*, and is normally used to determine the rotation invariant moments, also known as Hu's invariant moments.

3.1.4. Hu's invariant moments

With these moments, we have invariance under translation, changes in scale, and also rotation of the object

$$I_1 = \eta_{20}+\eta_{02}$$

$$I_2 = (\eta_{20}-\eta_{02})^2+4\eta_{11}$$

$$I_3 = (\eta_{30}-3\eta_{12})^2+(3\eta_{21}-\eta_{03})^2$$

$$I_4 = (\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2$$

$$I_5 = (\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})((\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2) + (3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})(3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2)$$

$$I_6 = (\eta_{20}-\eta_{02})((\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2) + 4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03})$$

$$I_7 = (3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})((\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2) - (\eta_{30}-3\eta_{12})(\eta_{21}+\eta_{03})(3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2)$$

There is also an extended eight invariant moment defined as

$$I8 = \eta_{11}((\eta_{30}+\eta_{12})^2-(\eta_{03}+\eta_{21})^2)-(\eta_{20}-\eta_{02})(\eta_{30}+\eta_{12})(\eta_{03}+\eta_{21})$$

3.2. Considered features

The integration of the features will result in a feature vector, designed as an observation. Therefore, it is necessary to design a classifier that receives a feature vector and distinguishes the object in the image.

3.2.1. Algorithm

To make the training and testing of several classification models possible, it was created an algorithm to extract all the desirable features from the gray-scaled images.

The desirable features are the following:

- I1…I7, the seven invariant moments of Hu;
- I8, an extended eight moment of Hu;
- area, the area of the binary image (after the threshold);
- perimeter is not a rotation invariant feature (in image processing), but it was also considered for analysis purposes;
- eccentricity is the relation between the major- and minor-axis of the object present in the image.

The algorithm causes some transformations on the image: it begins with a binarization of the grayscale image based on a global image threshold using Otsu's method, followed by a *dilation* and *erosion* of the binary image. This is a very important step because it reduces the noise in the image and makes possible the extraction of the features explained previously.

The extraction of Hu's eight invariant moments and object's area, perimeter and eccentricity, represented in the image, will originate a characteristic vector (a full observation) that are used as a base for the classification method developed in Section 4.

In the feature extraction, the "*centroid*" position and orientation (rotation angle) are also computed (see Sections 3.1.1 and 3.1.2) to allow the manipulator to grab the identified object.

3.2.2. Results using laser range finder

To demonstrate the feature extraction algorithm, some work examples are shown in order to prove that the desirable features represent the object characteristics suitably (as we see in the following section).

Each observation sample consists of two images: one reflects the result of the laser scanning and the other reflects the result of three extractions. It is important to mention that all images were taken setting the speed of the manipulator equal to 600 mm/min and with a perpendicular distance of 58 cm from the platform where each object was placed.

First, the features of the same object (with different positions and orientations) are compared.

As it is visible in Table 1, there is a set of features that remain approximately constant (note that there is some noise in the representation of the images). Features I1, I2, perimeter and eccentricity are apparently more stable, in which the values do not vary significantly between the images and should yield a good characterization of the object during the classification process.

Below is the feature extraction result for different objects. Their classification is addressed in Section 4. The aim of this work is to identify which is the object present in the operating environment of the robot. The object can be one of the seven pre-defined objects: turn screw, industrial spray-paint, industrial painting tank, small water bottle, artificial wood bar, wood block and cork block.

As we can see, there are some objects that are easily identifiable, but there are others that are difficult to classify, such as the distinction between the artificial wood bar, wood block and the cork block.

In Table 2, we can see how eccentricity is important because the block or bar of wood and cork have very similar features. However, eccentricity is significantly different (Figs. 6–9).

3.3. Results using CCD camera

To prove the increase of light robustness of our system over the CCD camera approach, we performed some experimental tests using an ImagingSource RGB Camera fixed in a particular place of the workspace environment. The images obtained are presented in Fig. 10. In the CCD image tests we used the same methodology as for the laser range finder approach, in order to faithfully compare and interpret the results.

In Fig. 10, we present the original RGB camera images and the corresponding Otsu's segmentation results since the segmentation step is the most sensitive to environment lightning changes. After this step, Hu's feature extraction and classification process are similarly applied to both laser range finder and CCD camera

Table 1
Features extraction—different observations for the same object.

Features extraction										
I1	I2	I3	I4	I5	I6	I7	I8	Area	Perim.	Eccent.
0.435	0.161	5.04e−08	3.34e−08	1.03e−15	1.32e−08	−6.56e−16	−1.06e−09	7914	611	4.857
0.407	0.137	2.33e−08	5.20e−09	1.18e−17	7.52e−10	3.58e−17	8.90e−10	8057	649	4.503
0.371	0.110	2.67e−08	1.77e−08	3.54e−16	5.65e−09	−7.49e−17	8.06e−10	9388	683	4.087

Table 2
Features extraction—example for all the objects considered.

Features extraction										
I1	I2	I3	I4	I5	I6	I7	I8	Area	Perim.	Eccent.
0.907	0.794	1.54e−06	1.37e−06	3.16e−13	1.22e−06	3.58e−13	4.34e−08	8936	947	10.39
0.397	0.130	6.70e−09	5.96e−09	4.96e−18	2.14e−09	−2.36e−17	4.87e−11	8216	650	4.409
0.230	0.025	6.74e−11	7.58e−11	3.18e−21	1.211e−11	2.23e−21	2.11e−14	10 170	563	2.119
0.341	0.089	4.32e−08	2.27e−08	4.85e−16	6.71e−09	−2.13e−15	5.11e−10	8585	659	3.728
0.202	0.011	1.11e−07	1.91e−08	1.73e−16	1.97e−09	−4.68e−16	−2.31e−10	8447	657	1.483
0.507	0.110	1.13e−05	1.13e−06	−3.90e−13	−2.54e−07	2.79e−12	1.32e−07	7007	1064	1.949
0.335	0.080	1.66e−07	1.12e−07	1.01e−14	3.11e−08	−6.15e−15	3.36e−09	3018	571	3.324

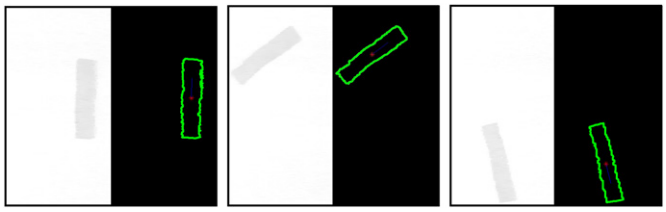

Fig. 7. Samples of three different observations for the same object in the robotic environment.

approaches. Fig. 10a illustrates an ideal case, where high contrast between the work object and the background is presented. Therefore, the application of Otsu's segmentation method is performed with a successful result, however for the rest of the images this segmentation was unsuccessful which forbad the feature extraction process. Fig. 10b and c represents the low contrast problem between the work object and the background, and finally, Fig. 10d illustrates the environment's light problem.

With these experiments we assuredly prove that CCD cameras are unreliable in an unstructured light environment and with a low contrast between object and background color cases.

4. Classifiers methods

4.1. Feature selection and model selection/assessment

In this section, the concepts of feature selection, model selection and assessment are introduced, as well as some algorithms and experiments applied to our classification problem. Note that the feature selection and model assessment are used to improve the classification process and to overcome the situation of poor generalization error caused by the model overfitting to the dataset, used in the training process.

4.1.1. Feature selection

Feature selection is a process often used in data mining applications. A subset of features is selected from all the available data according to certain criteria. By reducing the number of features and aspects such as redundant, irrelevant, or noisy data it is possible to speed up the learning algorithms, and improve their performance [20]. The best subset contains the minimum number of features that contribute to increase accuracy, since that when there are many irrelevant features the learning models tend to overfit the data and to be less comprehensible. This is an important preprocessing stage where the final goal is to reduce the dimension of the data by finding a small set of important features that offers good classification performance and reduce the complexity of the classification models.

There are two main approaches to reduce the number of features: *feature selection* and *feature transformation* [11]. Feature transformation methods turn data from the original high-dimensional feature space into a new space with reduced dimensionality.

In contrast, feature selection algorithms select a subset of features from the original feature set, keeping the original meaning of the features. The selection criteria usually involve the minimization of a specific measure of predictive error for models to fit different subsets. Algorithms search for a subset of features that optimize the predictive models, subject to some constraints such as required/excluded features and subset size.

We intend to keep the original meaning of the features and determine which of those features are really important. To accomplish this, a feature selection algorithm is desirable. Feature selection algorithms are characterized by three main strategies: filter, embedded models and wrapper methods [11,20]. Wrapper methods consist in searching for the features that fit the chosen learning algorithm better (evaluating the feature subsets based on the performance of the classifier). These can be significantly slower comparatively to filter methods if the learning algorithm takes a longer period of time to run. Usually they are more accurate than filter methods in a particular classifier. However, the selected features may not be the most appropriated for other classifiers [11].

The selection of a good subset of features could be performed in three ways: *exhaustive search* (2^D possible combinations, where D is the total number of features), *random search* methods and *heuristic search* strategies.

In the exhaustive search, all combinations of the feature subset are tested. Although it is possible to find the best or the optimal feature subset, it is expensive and in some cases computationally impractical because the search base is larger. Random search methods add some randomness to the searching process, but it is not guaranteed that the best feature subset is reached.

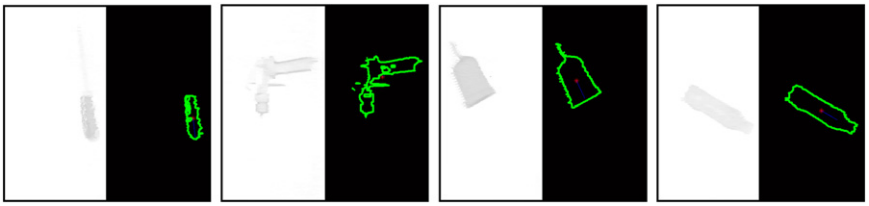

Fig. 8. Sample of a turn-screw, spray-paint, painting-tank and water bottle in the robotic environment.

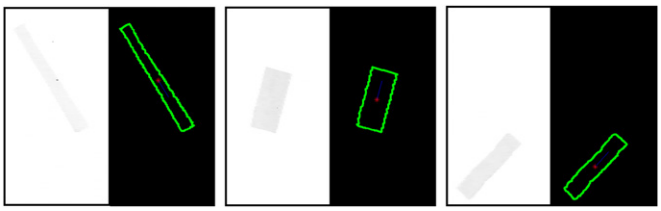

Fig. 9. Sample of an artificial wood bar, wood block and cork block in the robotic environment.

Feature selection is an NP-hard optimization problem, and for that kind of problems heuristic methods (where certain guidelines are used for the selection) are commonly used. There is one problem with the heuristic approach: there may be a high order combination of relevant feature subsets and for that the optimal solution is also not guaranteed. There are several heuristic methods (forward selection, backward selection, combined selection, etc.), but for this, the simulated annealing was the chosen selection algorithm where a worst neighborhood solution can be momentarily accepted depending on some conditions (try to escape from local minimum).

One quick way to decide on the number of features required is to define an evaluation criterion (normally the classifier error rate, i.e., the number of misclassified observations divided by the number of observations), and the selection algorithm conducts a search for a good subset. For the purposes of this work, a wrapper method was implemented in the *k*-nearest neighbor, support vector machine and neural networks based on heuristic simulated annealing.

4.1.2. Model selection and assessment

To overcome the problem of overfitting the data and the choice of model and parameters adjustment, some model selection strategies and assessment are usually applied. Model selection is used to estimate the performance of different models in order to choose the best one, and the model assessment is used to estimate the generalization error of the new data having chosen the final model.

To accomplish the model selection and assessment, the dataset is split into three parts: the training set is used to fit the models; the validation set is used to estimate prediction error for model selection, and the test set is used to assess the generalization error for the final chosen model.

In the classification problems, the main methods for model selection and assessment are leave-one-out, cross-validation with random sub-sampling and *k*-fold cross-validation [1].

For our approach we have selected *k*-fold cross validation that has the big advantage of that every data point that belongs to the training data is in a validation set exactly once and in the training set $k-1$ times. Usually, the training error is lower than the actual test error, and to estimate the generalization error of the model it is necessary to apply independent test samples (validation sets).

In this case, and due to the lack of observations, where 2/3 of the full data belong to the train-data, this advantage is even more important since it allows the train data to be used in order for it to be optimized. With this procedure we try to avoid the overfitting of the models and achieving a more reliable performance evaluation of the models by using independent samples of the train data used (test data).

However, there are disadvantages to this method. In fact the training algorithm has to rerun from scratch *k* times, which means that it takes *k* times and as much computational cost to make a single evaluation.

4.2. Practical results

In this work we aggregate the feature selection and model assessment in order to achieve a classifier that has low complexity and a low generalization error.

The simulated annealing selects a features vector and then the cross-validation algorithm adjusts the best model parameters. At the end of the *k*-fold, the best parameters and the respective accuracy goes back to the feature selection phase, and the process is repeated until the stop conditions (the temperature concept inherent to the algorithm or the maximum iteration number) are reached. The generalization error can be predicted in the final phase of our algorithm, which is the performance testing. Due to the heuristic approach, the cross-validation algorithm runs Nmax (max number of iterations) times and consequently the process presented in Fig. 11 takes a longer period of time to compute the best feature subset and the parameters for each classifier. Although this procedure is done in off-line mode it does not interfere with the industrial robot's performance.

With this approach we identify the features that contribute the most to the classification process as well the best parameters of the classification model. We also compare the balance between the complexity and expected accuracy of each type of classification model *k*-nearest neighbor (kNN), support vector machine (SVM) and neural network (NN).

Next, we present the results of applying this algorithm to different types of classifiers. In this work, three types of classifiers were implemented, the kNN, the SVM and the NN. It is important to note that the feature subset vector is a binary array were each "1" represents the activation of the corresponding feature and "0" is a non-activation of the feature in the training phase. The following results represent the feature vectors with lower training errors, and the respective tuning parameters of each classifier. Finally the generalization error achieved after applying the test-set is also analyzed and the best models are chosen.

4.2.1. k-Nearest neighbors (kNNs)

The first approach considered was the *k*-nearest neighbors. By knowing the *k* closest neighbors and the classes that they belong to, we use mode function to select the class that we are going to associate to each test sample. The mode function is normally used to eliminate noise that could be found in the class features. The drawback in this algorithm is that, as *k* increases and approaches the *n* value (size of the data), the performance of the classifier will become a statistical baseline and all unknown data will belong to

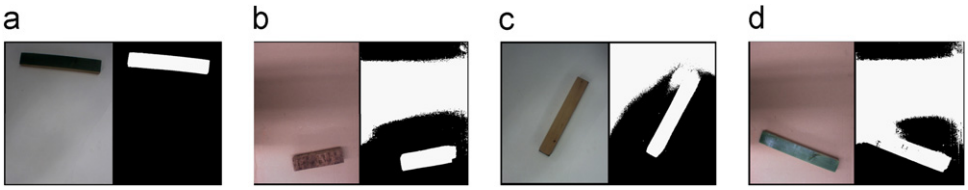

Fig. 10. Samples of objects obtained by the CCD camera approach. In the left side of each sample we have the original camera image. In the right side we have Otsu's threshold method.

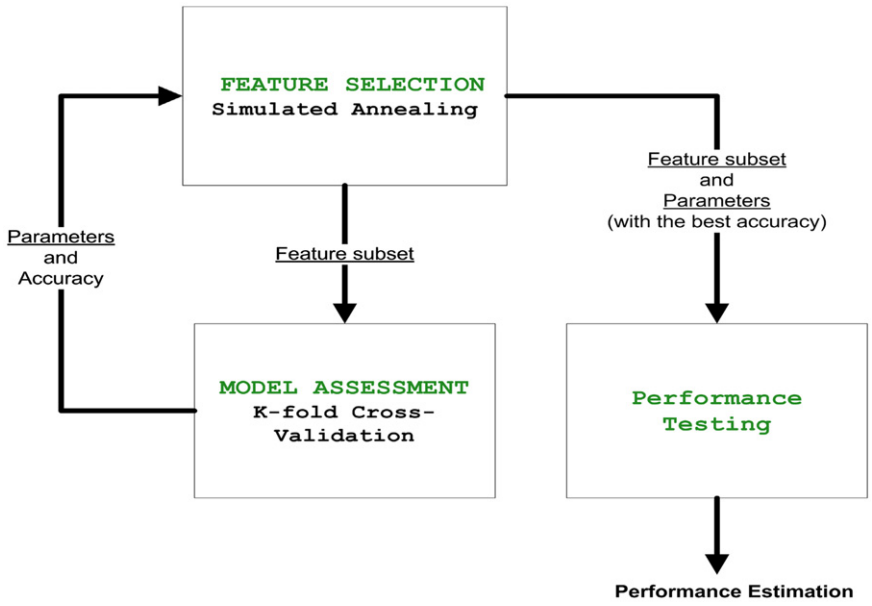

Fig. 11. The feature selection with model assessment.

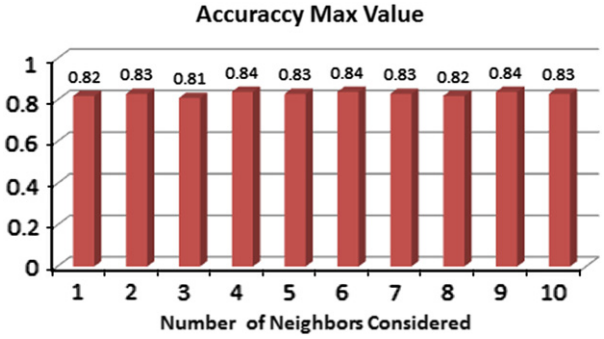

Fig. 12. Result of the application of the kNN algorithm.

the class, most frequently represented in the training set. On the contrary, if the k value is too low (for example $k<2$) the algorithm becomes very sensible to noise. This is not true only if the dataset is completely separable and the margin between each class is large.

In Fig. 12, the accuracy obtained for the different values of neighbors is presented, with the Euclidean distance as the selected metric. By analyzing the results, we can verify that the objective is not very sensitive to changes performed in the k value (number of neighbors). This can lead to an interpretation error that the data is suitably divided into classes. Nevertheless, as we can see the value of the accuracy is around 83%. These accuracy values were obtained for the features selected using the simulated annealing process introduced before, i.e., after the first solution is generated using a pre-defined combination of features, the algorithm will search other combinations of features that will possibly increase the value of the objective function (accuracy). In Table 3 are presented the combination of features that retrieved the same maximum value of accuracy used to classify the test sample according to its neighbors. As we can see, features I1, I2, I3 and eccentricity are presented in most of the results, which means that they are more important to the kNN classification algorithm.

Table 3
kNN—best feature combination for the mode function.

k value	Feature selection											No. of features
	I1	I2	I3	I4	I5	I6	I7	I8	Area	Perimeter	Eccentricity	
3	1	1	1	0	1	0	0	0	0	0	1	5
8	1	1	0	0	0	0	0	0	0	0	1	3

4.2.2. Neural networks

Neural networks are much more complex algorithms than the kNN, with much more parameters to tune (example: neuron activation function, number of hidden layers, number of neurons for each hidden layer, etc.). Therefore, we have decided to assign some parameters, which are activation function of the hidden layers (sigmoid function) and the activation function of the output layer (softmax). The softmax activation function was used since we defined that the output layer would have seven neurons corresponding to the seven classes. The softmax function deals with this classification in a probabilistic way allowing the output layer to have data from the different classes. In an ideal situation,

Table 4
Best feature combination for one hidden layer (NHL).

NHL	Feature selection											No. of features
	I1	I2	I3	I4	I5	I6	I7	I8	Area	Perimeter	Eccentricity	
1	0	0	0	1	0	0	0	0	1	0	1	3

Table 5
Best feature combination for two hidden layers (NHLs).

NHL	Feature selection											No. of features
	I1	I2	I3	I4	I5	I6	I7	I8	Area	Perimeter	Eccentricity	
2	0	0	0	0	0	0	0	0	1	0	1	2

only one neuron in the output layer should have a value different from zero, whose index will correspond to the classified class.

In the approach developed we changed the number of the hidden layer between {1; 2}, and the number of neurons in each layer between {2; 5} in the first hidden layer and {2; 3} in the second hidden layer. We considered these values because the neural network with one hidden layer can represent any continuous function and with two hidden layers it is possible to represent any discontinuous function. The number of neurons in each layer were chosen taking into account some initial tests. As we can see, with the cross validation we try to obtain a more reliable performance measurement of the models and, at the same time, tune the parameter presented before.

Next, the results obtained for only one hidden layer and for two hidden layers are presented.

For only one hidden layer the best result was obtained for four neurons in the hidden layer and the feature combination is presented in Table 4, with a validation accuracy of 94%. The accuracy value, resulting from the application of these tuned parameters in the test set (result of the division of the initial data, 2/3 training data; 1/3 testing data), was equal to 95%. The best result obtained with two hidden layers was of two neurons in the first hidden layer and three neurons in the second hidden layer. The feature combination is presented in Table 5, with a validation accuracy of 96%. The accuracy value, obtained for these parameters in the test set, was equal to 95.5%. As can be seen, the validation accuracy is quite similar to the generalization accuracy (applying the test set), which indicates that our training methodology does not overfit the training data and the performance obtained under real conditions should be identical.

4.2.3. Support vector machine

Support vector machines (SVMs) are useful tools for classification purposes. However it is a binary classification model. Therefore, there are two main SVM-based approaches to solve the multi-class problems: "One against" one and "One against all".

The literature is inconclusive on the best approach to solve multi-class problems. Therefore, a study was conducted in order to compare the performances achieved by each approach. The one against all approach is used quite often for SVM training and prediction because it is only necessary to compute N models, where each model represents the classification for each class (*belongs to class or not*). One of the biggest problems with this approach is the loss of significance in the dataset because there are 7 classes, and 22 observations for each class. Therefore, there will be 22 observations belonging to the class and 132 observations not belonging to the class. This large discrepancy between the number of observations may mislead or jeopardize the training process of SVM models. On the other hand one against one constructs one binary classifier for every pair of distinct classes, meaning that all together are $N(N+1)/2$ binary classifiers. After the vote of each of the $N(N+1)/2$ binary classifiers, the final result can be the class with the largest number of votes. Other approach is the iterative exclusion of the class that loses, i.e., all SVM models belonging to this class versus those that can be excluded from the classification process, where only the models of the victorious class should be considered. Finally, there is a tree type-like approach where the classification result is the strongest class. This was the approach considered and implemented in our work.

The results of applying the algorithm are explained in Fig. 11. Based on those parameters, it is possible to analyze: the feature subset, the C-value (soft margin parameter) and the generalization error (the real expected accuracy of the model).

Another important parameter for SVM models is the kernel function. In this work we use both the polynomial, $K(X_i,X_j) = (X_i,X_j)^D$ and the Radial Basis Function (RBF), $K(X_i,X_j) = \exp(-\gamma\|X_i-X_j\|^2)$, where the different parameters were selected (polynomial order, and the gamma for the RBF) according to the best performance achieved during the testing phase.

For each result, the feature subsets we obtained were the C-value, the kernel parameter and the performance accuracy of the model. In the following graphs it is possible to see the training performance, which is an optimistic accuracy, and the testing performance by applying the independent test set to evaluate the real expected accuracy for the SVM model.

4.2.4. Radial basis function

In our SVM models, the gamma value for the Radial Basis Function (RBF) is equal to $\alpha/(SizeTrainingData)$, where the value of

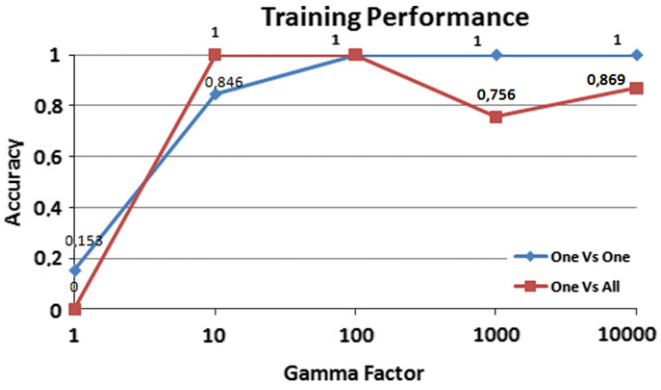

Fig. 13. Training performance obtained as a function of the gamma factor.

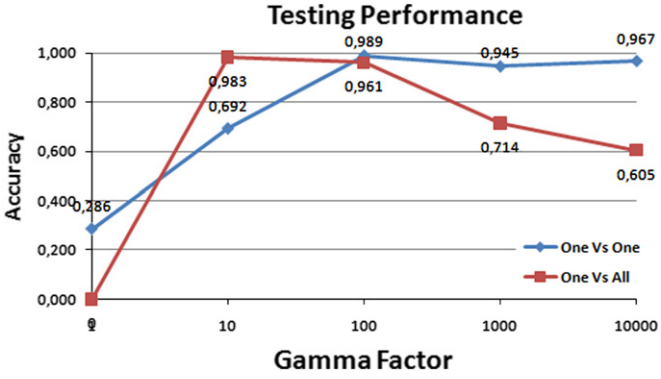

Fig. 14. Testing performance obtained as a function of the gamma factor.

Table 6
Best feature combination for RBF kernel function with *C*-value = 0.25.

Feature selection											No. of features
I1	I2	I3	I4	I5	I6	I7	I8	Area	Perimeter	Eccentricity	
0	0	0	0	0	0	1	0	1	1	1	4

Table 7
Best feature combination for polynomial kernel function with *C*-value = 0.25.

Feature selection											No. of features
I1	I2	I3	I4	I5	I6	I7	I8	Area	Perimeter	Eccentricity	
0	0	0	0	0	0	0	0	1	0	1	2

gamma factor, α, was changed according to the following graphics. In Figs. 13 and 14, it is possible to see that there are different performances (training and test) when the OneVsOne and OneVsAll strategies are used. Generally, the OneVsOne strategy obtains better results, specially when $\alpha = 100$ and the generalization accuracy has a maximum of 98.9%, with feature selection presented in Table 6.

4.2.5. Polynomial

In the second polynomial order, the maximum generalization accuracy (95.4%) was obtained with the feature selection presented in Table 7 (Fig. 15).

Comparing the performance between the two kernel functions (see Figs. 14 and 16), we can infer that the polynomial function offers better global results than those that are provided by the RBFs. However, for both kernels the soft margin parameter (*C*-value) is equal to 0.25. In our experiments we noted that the best results have been obtained when the *C*-value was $\in [0.25; 4]$.

Analyzing the contribution of each feature to the classification process, it is possible to conclude that, for the results with higher generalization accuracy, the total number of active features can be considered as a playoff factor.

5. Conclusion

In this work we have used an innovative technique that allows the manipulators to recognize and identify work objects in their workstations.

As an opposition to the traditional data acquisition systems (CCD camera and laser beam scanner), this method uses a 2D laser range finder and an exploratory movement in order to produce a grayscale 3D image. The use of this laser makes possible for the manipulator to cover a wider area (comparatively to the single line laser) and, at the same time, providing higher immunity to the variation of luminosity, a problem that limits camera based systems.

The image obtained portrays the environment of the manipulators, i.e., the object presented in their workstation. Later on different machine learning methods were implemented including *k*-nearest neighbor, neural networks and support vector machine, in order to identify and recognize the objects portrayed in the image. Note that although the tests have been conducted with an inexpensive laser (scan at 10 Hz) however, there are available more accurate lasers to scan up to 500 Hz.

For this research, 7 different objects were used (some of them very similar and more difficult to differentiate) as well as a feature extraction algorithm based on the invariant moments.

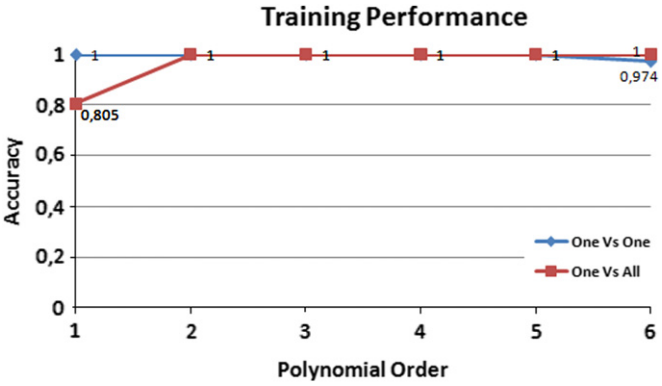

Fig. 15. Training performance obtained as a function of the polynomial degree.

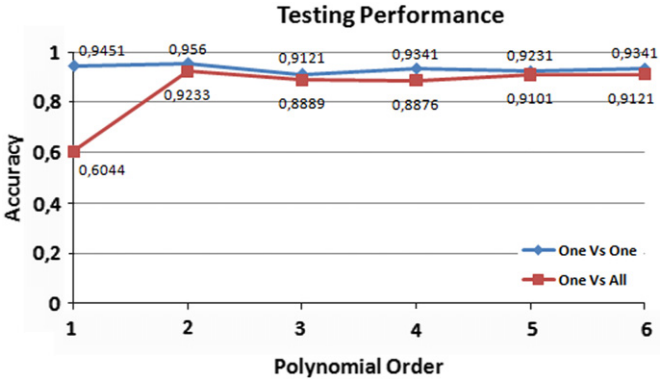

Fig. 16. Testing performance obtained as a function of the polynomial degree.

As it can be verified, performances of 83.5% were obtained for kNN, 95.5% for the NN and 98.9% for the SVM (generalized accuracy). These high performances are largely due to the feature selection algorithm based on the simulated annealing algorithm, and the model assessment (*k*-fold cross-validation), since it makes possible to identify the most important features in the recognition process, as well as the adjustment of the best parameters to the classification methods.

We also found that features like area and eccentricity are present in almost all of the results, and based on this fact, we can conclude that these features are the most important for the classification process (NN and SVM). Another curious result is the perimeter feature because it also appears to allow SVM classifier models to be characterized properly (using RBF as kernel function), although it is not invariant to rotation.

In short, an automatic system that is robust and reliable was developed which allows the object to be recognized. Furthermore, it is possible to obtain the position and orientation of the object so that it can be operated correctly for machining and transportation tasks.

References

[1] Arlot S. A survey of cross-validation procedures for model selection. Statistics Surveys—Project Euclid 2010;4:40–79.
[2] Bebis GN, Papadourakis GM. Object recognition using invariant object boundary representations and neural network models. Pattern Recognition 1992;25(1):25–44.
[3] Ben-Tzvi P, Charifa S, Shick M. Extraction of 3D images using pitch-actuated 2D laser range finder for robotic vision. In: IEEE international conference on robotic and sensors environments, vol. 4; October 2010. p. 1–6.
[4] Bishop CM. Pattern recognition and machine learning. New York: Springer; 2006.
[5] Caelli T, West G, Robey M, Osman E. A relational learning method for pattern and object recognition. Image and Vision Computing 1999;17(5–6):391–401.

[6] Flusser J, Suk T, Zitov B. Moments and moment invariants in pattern recognition. Chichester, UK: John Wiley and Sons Ltd.; 2009.
[7] Klimentjew D, Arli M, Zhang J. 3D scene reconstruction based on a moving 2D laser range finder for service-robots. In: IEEE international conference on robotics and biomimetics, vol. 4; December 2009. p. 19–23.
[8] Koker R, Oz C, Ferikoglu A. Development of a vision based object classification system for an industrial robotic manipulator. In: The 8th IEEE international conference on electronics, circuits and systems; August 3, 2001. p. 1281–4.
[9] Lopes P, Oliveira E. Model based 3D object recognition using an accurate laser range finder. In: International conference on industrial electronics, control, and instrumentation, vol. 3; July 1993. p. 1696–1701.
[10] Luo R, Scherp R. 3D object recognition using a mobile laser range finder. In: IEEE international workshop on intelligent robots and systems: 'Towards a New Frontier of Applications', vol. 2; July 1990. p. 673–7.
[11] Masaeli M, Fung G, Dy JG. From transformation-based dimensionality reduction to feature selection. In: 27th international conference on machine learning, Haifa, Israel, vol. 1; June 2010. p. 751–8.
[12] Okamoto K, Langer W, Mengel P. 3D object recognition system using laser range finder. In: Fifth international conference on advanced robotics 'Robots in Unstructured Environments', vol. 2; June 1991. p. 1245–50.
[13] Paschalakis, Lee S, Kent P. Pattern recognition in grey level images using moment based invariant features. In: Seventh international conference on image processing and its applications, 1999 (Conf. Publ. No. 465), vol. 1; July 1999. p. 245–9.
[14] Trevor Hastie RT, Friedman J. The elements of statistical learning: data mining, inference, and prediction. New York: Springer; 2001.
[15] Tsay T, Lai Y, Hsiao Y. Material handling of a mobile manipulator using an eye-in-hand vision system. In: International conference on intelligent robots and systems (IROS), vol. 3; October 2004. p. 30–3.
[16] Tsay T, Lai Y, Hsiao Y. Material handling of a mobile manipulator using an eye-in-hand vision system. In: International conference on intelligent robots and systems (IROS), vol. 1; October 2010. p. 43–7.
[17] Wang J, Lu J, Zhang Y, Xu W. Learning filters for object recognition with linear support vector machine. Procedia Engineering 2011;15(0):1657–1661.
[18] Wei G-Q, Hirzinger G. Active self-calibration of hand-mounted laser range finders. In: IEEE international conference on robotics and automation, vol. 4; April 1997. p. 2789–94.
[19] Yuan C, Niemann H. Neural networks for appearance-based 3-D object recognition. Neurocomputing 2003;51(0):249–264.
[20] Zhao Z, Morstatter F, Sharma S, Alelyani S, Anand A, Liu H. Advancing feature selection research. Technical report. ASU Feature Selection Repository; 2010.