Efficient grasping of novel objects through dimensionality reduction

Benjamin Balaguer and Stefano Carpin

***Abstract*— A learning method capable of empowering a robot to successfully grasp a novel object through vision has recently been demonstrated, and generated much interest in the robotics community. In this paper we carefully analyze this new approach and apply dimensionality reduction techniques to decrease the number of features that need to be computed in order to classify whether a given pixel in an image is associated with a good or bad grasping point. Exploiting the ideas behind principal component analysis, we formulate two hypotheses about possible ways to eliminate certain features from training and classification. We then experimentally verify that the feature reduction significantly improves speed while retaining classification accuracy. Overall, the combination of the two hypotheses leads to a speedup factor of almost ten. The hypotheses are validated on third party synthetic data and also demonstrated on a seven degrees-of-freedom manipulator.**

I. Introduction

One of the major stumbling blocks on the way to a massive use of robotic devices assisting humans in a variety of daily tasks is found in the current limits in robots' ability to grasp and manipulate objects. Robotic manipulators helped the very development of the discipline, however they almost exclusively carry out highly repetitive tasks in carefully conditioned operating environments. Evidently, it would be highly profitable if robots were capable of grasping and manipulating a variety of objects under different conditions and limited knowledge.

A major development in this direction was recently reported by Saxena et al. [15], who developed a robotic system capable of grasping novel objects based on vision alone. Their approach relies on machine learning and exploits a huge training set of synthetic images labeled with so-called *good grasping points*. As described later on, the algorithm learns to identify good grasping points in the image-space of a novel object by computing a high dimensional feature vector for every pixel in the image, and applying logistic regression for classification. The feature vector characterizing a pixel in the novel images is obtained by applying a battery of filters in a 5×5 patch surrounding the pixel to be classified. As reported by the authors, a feature vector in $\mathbb{R}^{459}$ is computed for classification at every pixel of an image. This high dimensionality has two significant drawbacks. First, the training stage requires fitting a probabilistic model of high dimensionality that leads to time consuming computations in addition to requiring significant amounts of memory. In principle this is not an overwhelming problem as long as the training stage is performed rarely. However, as evidenced in the conclusions, we are pursuing a long range investigation where the robot needs to frequently be re-trained, and this time/space complexity becomes a major nuisance. Second, in order to identify good grasping points at run time one needs to compute these high dimensional feature vectors over an entire image. As a consequence, the amount of frames per second that can be processed is severely limited. This issue is particularly relevant when considering the case of a robot attempting to grasp an object that is not only novel, but also moving.

After having re-implemented Saxena's original algorithm, and having noticed the aforementioned limitations, we considered the possibility to carefully analyze the algorithm and to apply dimensionality reduction techniques in order accelerate the algorithm. The reduction is obtained through selection, i.e. many features are removed altogether from the training and classification stage, as opposed to methods that achieve reduction through feature combination. The results presented in this paper confirm that significant speedups through dimensionality reduction are indeed possible, and we eventually produced a refined version of the algorithm that achieves a high classification precision while relying on feature vectors with only 54 elements (as opposed to the 459 originally mentioned). The improvement is not only theoretical, but supported by practical experiments performed with a Barett WAM robot equipped with a stereo camera.

The paper is organized as follows. Section II briefly discusses related literature. Saxena's algorithm is shortly presented in section III, and, in section IV, we present an analysis based on principal component analysis that leads to two hypotheses capable of reducing the feature vector size. Section V experimentally validates the two hypotheses individually and jointly both in terms of simulated data and on a real robotic platform. Finally, conclusions and future work are addressed in section VI.

II. Related Work

Research related to robotic grasping and manipulation is vast, and we therefore here touch only a few selected contributions relevant to place our work into context. Specifically, due to the feature-based nature of our approach, we skip literature concerning model-based techniques for grasping. Piater builds upon an already-established mechanical framework that tries a variety of grasps for an object until a stable one is found. More specifically, in [13], this mechanical framework is enhanced by utilizing visual features from an overhead camera as a learning tool for good grasps. While the paper introduces a good series of concepts such as the need for task decomposition and learning and the focus on visual

School of Engineering, University of California, Merced, CA, USA, {bbalaguer,scarpin}@ucmerced.edu

features that remove the need for scene reconstruction or geometric reasoning, it focuses exclusively on simple objects (e.g. triangle, circle, square), and experiments are limited to simulation. A similar work has been published in [10], where the authors present an algorithm intended at finding grasps of unknown planar objects not limited to primitive shapes. The paper contributes a good algorithm for its intended application while coming up with important cornerstones such as the necessity of vision and sensing for grasping in unstructured environments. It, however, has some limiting assumptions, namely the fact that the input image is only comprised of the object contour and that the objects are extrusions of these contours. This work is subsequently implemented on a real platform in [9], where the authors identify the decoupling between finding stable grasps (i.e. visual processing) and physically grasping the object. In addition, they correctly identify the visual processing step as being independent from the end-effector configuration. Last but not least, the authors revisit their framework [11], in greater detail, contributing more realistic examples and pointing out the very desirable characteristic that their system is modular with respect to the manipulator's hand configuration.

Moving away from the limitations of two-dimensional grasps of planar objects, Anglani et al. propose a grasping algorithm by utilizing a controller capable of learning in a trial-and-error methodology [1]. Even though the paper is focused on the different problem of visual servoing, it encompasses some nice and surprising results such as running the learning phase of the algorithm in simulation, transporting the results on the real platform, and achieving good experimental results. The biggest limitation, however, is that the presented approach only works, as presented and implemented, for a spherical object of known size randomly placed in the environment. A similar paper, also exploiting visual servoing, tries to remove some of the most constraining assumptions made [14]. More specifically, the authors propose an environment-independent algorithm that does not rely on information about the objects in advance. These assumptions are however weakened by using an operator to draw a box around the object to be picked up rather than an autonomous algorithm. In our opinion, the highest impact approach to solve this specific problem was recently proposed by Saxena et al. [15]. The authors come up with the idea of image features as a representation of good grasping points. The main idea behind image features stems from the fact that different objects are grasped similarly by humans and that image features should be a good representation of grasping points. This paper is heavily influenced on this work and a thorough description of the algorithm is presented in the next section.

It is crucial to recognize the importance of human grasping as an insight to come up with viable solutions to robotic manipulation and, as such, we briefly summarize some interesting research about human subjects and grasping. Through a case study, Goodale et al. found that there exists a dissociation between recognizing objects and grasping them [6]. This work is substantiated in [4], where the author mentions that several neural pathways are used during a grasping task and, more specifically, that separate neural activities encode object features and move the fingers appropriately. In addition, the author reviews a variety of human and monkey studies that establish a correlation between object features and grasping parameters. Another interesting publication describes the irrelevance, for humans, of maintaining visual contact with the hand and the object during a reaching or grasping phase [7]. In other words, humans do not need to use visual-servoing techniques to grasp objects and we do not either on our robot implementation.

III. Grasping novel objects

Given the aforementioned related work, our primary motivation for working with a feature-based approach is that, when properly implemented, they are manipulator-independent, they can account for untrained objects, they attempt to replicate visual cues used in human grasping, they can use a single visual sensor (i.e. cheap sensor), and they do not make apriori assumptions on the objects or the environment. In this section we shortly recap what we consider to be the best feature-based algorithm to date, from Saxena. The reader is referred to [15] for a more detailed description, also including suggestions about integrating depth information (at the cost, however, of increasing the size of the feature vector). We purposefully do not take into account depth information because it rarely can be obtained for all pixels in an image and, as such, could introduce bias resulting in classification errors.

A. Training

The starting point for the learning algorithm is a huge set of synthetic images where good grasping points have already been identified. We define a good grasping point as any point on an object that a human would use to grasp the object. Consequently, objects have many good grasping points that are manually labeled for the training data. Objects in the training set include everyday entities, such as a cereal bowl, a pencil, an eraser, etc... Every image comes in two versions. The first one is the synthetic image, while the second is a binary version labeling pixels associated with good grasping points[1]. In order to learn how to discriminate pixels associated with good grasping points from bad ones, 17 filters are applied in a 5×5 patch surrounding a pixel. In addition, the same 17 filters are also applied to the pixel in two suitably scaled versions of the image itself, yielding a feature vector of size 459. This process is performed on every pixel of the image. The filters are applied to a YCbCr image as follows: six edge filters and nine Law's masks applied on the intensity channel of the image (i.e. Y), one average filter applied on the blue-difference chroma component (i.e. Cb), and one average filter applied on the red-difference chroma

[1]The whole data set is freely available for download on the authors' websites.

component (i.e. Cr). The feature vector is then obtained by concatenating the energy of these filters into a vector in $\mathbb{R}^{459}$. Therefore, the synthetic data leads to a set of (x_i, z_i) couples, where $x_i \in \mathbb{R}^{459}$ and z_i is a binary label indicating whether the associated pixel in the image is a good grasping point or not (with the value 1 associated to good grasping points). A parameter θ^* is then learned through maximum likelihood as follows:

$$\theta^* = \arg\max_{\theta} \Pi_i P(z_i|x_i;\theta). \quad (1)$$

B. Finding good grasping points

When the robot needs to grasp a novel object given an image of it, it starts computing the same filters for every pixel in the image, thus getting a feature vector $x_i \in \mathbb{R}^{459}$ for the ith pixel. The point is probabilistically classified as a good grasping point based on logistic regression, i.e.:

$$P(z_i = 1|x;\theta^*) = \frac{1}{1 + e^{-x^T\theta^*}}.$$

In order to appreciate the power of the technique, it is worth observing that the authors report remarkable results in terms of prediction accuracy both for objects similar to those in the training set, but also, and more importantly, for novel objects of classes not found in the data set. For example, it is shown that the system can predict how to grasp a coffee pot, a marker, and duct tape even though none of these objects were part of the training set. Consequently, we define, as was done by Saxena et al., a novel object as an object that was not part of the training data.
In our opinion, the weakest point of the presented solution, and the one that we address in this paper, comes from the authors' choice to rely on a highly dimensional vector of features. From a practical standpoint, a large feature vector is computationally expensive, both during training and execution time, as substantiated in section V. Features, moreover, are highly dependent on each other since they are both similar and spatially close together. This observation suggests that dimensionally reduction techniques would be prime candidates to reduce the size of feature vectors, and boost algorithm efficiency.

C. Modification to the Original Algorithm

In this paper, we use a slightly modified version of the algorithm. First, we remove the two features that are based on the color channels of the image. We believe that the color of an object should not affect how a robot grasps an object, as is the case for human grasping [16]. We also remove the features acquired on scaled versions of the original image since they do not capture sufficient information about different object sizes or views. Instead we suggests that it would be more beneficial to scale the images and treat them as new images (i.e. computing the full feature vector on the scaled images) to better account for different camera views representing smaller or bigger objects. In order to have a similar feature vector size, and to further test our dimensionality reduction theory, we add five more filters: a first-order 5×5 Sobel operator, a second-order 5×5 Sobel operator, a first-order 7×7 Sobel operator, a second-order 7×7 Sobel operator, and a Laplacian operator. As such, our final feature vector size is 500, with the original 15 filters from the algorithm added to the new 5 filters and performed in a 5×5 window around each pixel.

IV. An experimental study aimed at dimensionality reduction

Dimensionality reduction techniques have become mainstream tools in machine learning when high dimensional data sets hide an intrinsic lower dimensionality. The number of tools developed is high, and very often tailored to the specific problem being tackled. The reader is referred to [12] for a general introduction about the topic. One of the most common, yet powerful, techniques is Principal Component Analysis (PCA) [3]. PCA has already been used in the recent past in the context of robotic grasping, leading to the well-known concept of *eigen-grasps* [5]. PCA can be formulated in various and eventually equivalent ways. In essence, given a set of data points $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n]$ from which the mean has been subtracted in order get a 0 mean data set, we seek a change of bases capturing the dimensions associated with the highest variance. For the specific problem at hand, the matrix $\mathbf{X}$ has d rows and n columns, where each column corresponds to a d dimensional feature extracted from the training images. PCA is performed by solving an eigenvalue problem on the matrix $\mathbf{X}\mathbf{X}^T$. Arguably the most important aspect of PCA is that by sorting the eigenvectors according their associated eigenvalues (in decreasing order), it is possible to select a subset of eigenvectors, say $\mathbf{e}_1, \ldots, \mathbf{e}_m$, retaining a prescribed level of energy from the original data set. Once such eigenvectors have been identified, they can be used in various ways. The most straightforward approach consists in using them as bases to express a novel feature vector as a linear combination of the various $\mathbf{e}_i$ eigenvectors. This corresponds to projecting the original data set $\mathbf{X}$ along the directions identified by the eigenvectors (perhaps the most notorious example of this procedure is face recognition [8]). Alternatively, one can analyze the eigenvectors directly to identify patterns outlining which components of the original feature vectors contribute to more variability.
Figure 1 plots the ratio between the first 100 eigenvalues and the largest one (λ_{Max}). This chart was obtained by analyzing feature vectors of size 500.

It is evident from the plot that only a small subset of dimensions contribute to the variability found in the set of features. In particular, the first 9 eigenvectors retain 99% of the energy found in the data set. More insightful information can be discovered by examining the eigenvectors themselves. Figure 2 shows a plot for the components of the first eigenvector normalized to the largest one.

A periodic pattern is evident, as well as the fact that certain components have normalized values close to 0. A similar trend is evidenced in the other 8 eigenvectors accounting for 99% of the energy. Two aspects are important:

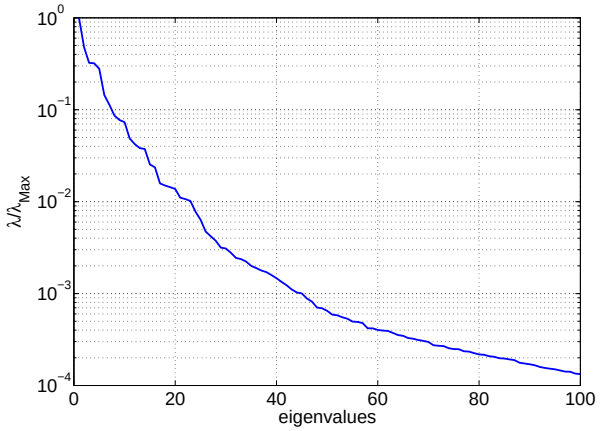

Fig. 1. Spectrum of the first 100 eigenvalues normalized to the value of the largest eigenvalue λ_{Max}. The reader should note the scale is logarithmic.

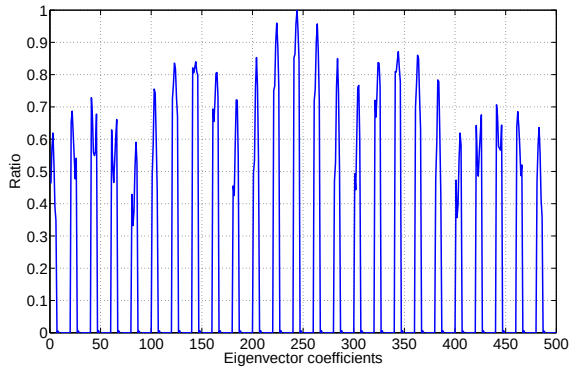

Fig. 2. Plot of the normalized coefficients of the eigenvector associated with the largest eigenvalue.

- the data shows a periodic trend with period 20 (i.e. the number of applied filters). In particular, it is always the same set of six filters that have normalized coefficients significantly larger than 0, and always the same filters that have negligible coefficients. This suggests that feature selection is possible.
- the repetitive trend stems from the fact that each feature is produced by concatenating together the energy of the filters applied to a 5×5 patch centered around the pixel to be classified. Figure 2 seems to suggest that the size of the window may be larger than what is needed in order to account for most of the variability in the data.

These two observations respectively lead to the formulation of two hypotheses:

Hyp 1 it is possible to identify good grasping points relying on a subset of filters, which are, based on the eigenvector analysis, the six edge filters.

Hyp 2 it is possible to identify good grasping points by only processing a 3×3 window around a candidate point rather than a 5×5 patch.

It is worth outlining that if the first hypothesis is verified the dimension of the feature space drops from 500 to 150 (6 filters applied to a 25-pixel patch) and if the second hypothesis is verified it drops to 180 (20 filters applied to 9 pixels). Moreover, if both hypotheses hold, the dimension of the feature space drops to 54 (6 filters on 9 pixels). PCA is often used to identify a suitable change of bases that are used to project high dimensional data along the directions of the most significant eigenvectors. However, this approach is not viable in our scenario because the high dimensional feature vector needs to be projected into a lower dimensional subspace at runtime, a time consuming process. Consequently, we focus on avoiding the extraction of many, possibly insignificant, features by relying on feature selection.

Having observed such a potentially dramatic decrease in the number of features, one may wonder whether an even more radical simplification is possible. Referring to literature related to dimensionality reduction for face detection [2], it makes sense to explore whether Linear Discriminant Analysis (LDA, also known as Fisher Discriminant Analysis) may be used in order to exploit the fact that features are assigned to two classes, namely good or bad grasping points. LDA is a technique similar to PCA, in the sense that it also involves a change of bases. However, rather than looking for directions that maximize variation within the data, it looks for directions that maximize *between-class* covariance, while minimizing *within-class* covariance. Therefore, at least in principle, LDA has the potential to overcome PCA since it also exploits the training labels that are instead ignored by PCA. The limit however, is that, for the problem at hand, we are dealing with only two classes. Therefore, LDA will attempt to find a linear separation between the two classes[2]. Not surprisingly, a preliminary investigation of this idea evidences that a linear separation leads to a significant compromise in terms of accuracy. The confusion matrix obtained processing a set of 330035 labeled features is as follows:

$$\begin{bmatrix} 178864 & 26853 \\ 10232 & 114086 \end{bmatrix}.$$

The fraction of false positives is of particular concern (about 13%) because it may drive the robot to try to grasp objects in points that are not appropriate. What can be concluded is that the training data cannot be linearly separated while retaining a sufficient accuracy and that LDA appears not to be suitable to further reduce the dimensionality of the feature vector.

V. Experimental results

In order to validate the two hypotheses formulated on the basis of eigenvector analysis, we have performed three series of experiments. First, we compute the accuracy of finding good grasping points on the synthetic data using the different hypotheses as well as real data collected from a camera. Second, we evaluate the tradeoff between speed and accuracy when the number of features is progressively

[2]In general for a classification problem involving c classes, LDA will determine $c-1$ separation hyperplanes.

reduced. Finally, we implement the proposed accelerated techniques on a real robotic system.

A. Accuracy with Synthetic and Real Data

The entire synthetic data is comprised of 13247 images, divided into the following nine object classes: cereal bowl, eraser, martini glass, mug, stapler, tea cup, pencil, two tea cups, and two mugs. Each object class has a number of images ranging from 120 to 2001. We first train our algorithm using 20% of the entire synthetic data, a number chosen based on both the speed of the training and the empirical observation that 10%-20% captured enough varied information about the data. The training is performed for the original algorithm (500 features), hypothesis 1 (150 features), hypothesis 2 (180 features), and the combination of hypotheses 1 and 2 (54 features).

Our goal is to deduce the accuracy of the dimensionally-reduced data compared to the full data. Given a new image, we start by assigning each pixel two good grasping point probabilities: H_i, based on the training from one of our three hypotheses, and G_i, based on the training from the full data. We then take the 15 pixels with highest H_i probabilities and compare them to the G_i probabilities in a 5×5 window centered around the ith pixel. If any pixel in the 5×5 window is within 2% of H_i, the grasping point of our hypothesis is clasified as accurate. We use a 5×5 window to account for the fact that good grasping points tend to be co-located and a 2% threshold for the inherent variability in determining good grasping points from percentages.

We first run our accuracy measure on synthetic data, the results of which is shown in Table I. As can be seen from the table, the results corroborate our hypotheses. More specifically, we can observe that hypothesis two (i.e. reducing the size of the window) has very little effect on the accuracy. Hypothesis one (i.e. reducing the number of filters) and, as a result, the combination of hypothesis one and two have slightly lower accuracies, explained by the possibility that a small portion of the objects might have been strongly influenced by the removed features.

Having verified the validity of our hypotheses on the same objects that were trained on, we move on to novel objects by training on synthetic data and executing our algorithm on real data. The real data, collected directly from our robot's camera, is significantly different than the training data, being comprised of a bottle, a calculator, a very small cup, a hammer, a shampoo bottle, and duck tape. Each object is associated with a set of 30 different images portraying different poses and light conditions. We use the same training (i.e. 20% of training data) as the previous experiment.

The result of the experiment is shown in Table II, which shows a very similar trend as the one in Table I. More specifically, reducing the window size (Hypothesis 2) has very little effect on accuracy while reducing the filters (Hypothesis 1) has a slightly more, yet reasonable, negative effect on accuracy.

Object	Hyp 1	Hyp 2	Hyp 1 & 2
Cereal Bowl	96.28%	97.84%	96.10%
Eraser	94.70%	97.22%	94.85%
Martini	97.37%	98.43%	96.61%
Mug	97.89%	95.99%	96.50%
Stapler	96.48%	98.31%	96.15%
Tea Cup	97.11%	96.71%	95.87%
Pencil	95.53%	98.01%	95.61%
Two Mugs	93.39%	98.12%	85.69%
Two Tea Cups	96.21%	97.24%	93.99%
All	**96.14%**	**97.74%**	**95.68%**

TABLE I

ACCURACY MEASURE OF THE HYPOTHESES TRAINED AND EXECUTED ON SYNTHETIC DATA. RESULTS ARE SHOWN FOR EACH HYPOTHESIS, EACH OBJECT CLASS, AND THE COMBINATION OF ALL THE OBJECT CLASSES (LAST ROW).

Object	Hyp 1	Hyp 2	Hyp 1 & 2
Bottle	99.25%	97.18%	99.26%
Calculator	90.44%	98.44%	92.00%
Cup	81.06%	95.96%	82.75%
Hammer	89.74%	98.12%	90.57%
Shampoo	90.44%	99.11%	88.57%
Tape	80.56%	92.59%	78.26%
All	**88.73%**	**97.22%**	**89.30%**

TABLE II

ACCURACY MEASURE OF THE HYPOTHESES TRAINED ON SYNTHETIC DATA AND EXECUTED ON REAL IMAGES. RESULTS ARE SHOWN FOR EACH HYPOTHESIS, EACH OBJECT CLASS, AND THE COMBINATION OF ALL THE OBJECT CLASSES (LAST ROW)

This series of experiments clearly indicate that our approach to dimensionality-reduction, in this context, works for objects that have been trained on and that are completely novel. It is important to note that objects have multiple good grasping points and that techniques only need to find a few good ones to be successful. Figure 3 attempts to illustrate this fact with a couple of representative examples of our real images representing novel objects. As can be seen from the pictures, the best grasping point generated by our methods is part of the subset of points generated by the original method.

B. Speed-accuracy tradeoff

As suggested earlier, the principal reason for dimensionality-reduction is to speed up the entire process in order to, eventually, grasp moving objects. While lowering the time that it takes to train on the data is a welcomed benefit, it is not a crucial part of this algorithm (i.e. it is only performed once) and, as such, we focus on the running time of the algorithm. Figure 4 shows the speed of the algorithm as a function of the feature vector size. Displayed timing information refers to a C++ implementation using OpenCV for image processing, and executed on a 3GHz Linux system. The time is measured from the acquisition

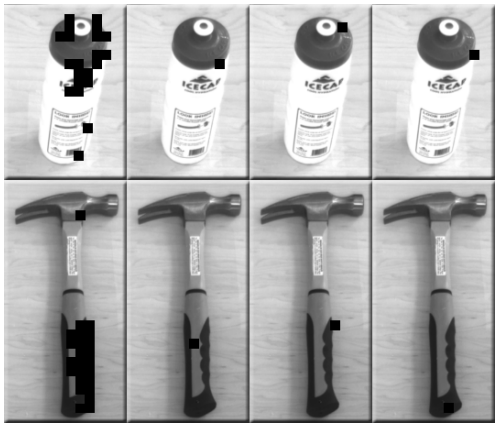

Fig. 3. For two objects, a bottle (1st row) and a hammer (2nd row), black pixels show all the good grasping points with a confidence level of 98% or more for the original method (1st column), along with the best grasping point (i.e. the grasping point chosen by the robot) for hypothesis 1 (2nd column), hypothesis 2 (3rd column), and hypothesis 1 and 2 combined (last column).

of a 640×480 image to the determination of all good grasping points within that image. Evidently, and as supported by the plot, a decrease in the number of features results in a linear decrease in the algorithm's running time. The dimensionality-reduction's influence on speed is substantiated by two operations of the algorithm, namely the application of filters (i.e. less filters to apply) and the logistic regression (i.e. smaller vectors to multiply). As a side note, the speed decrease for training is even more significant since determining the maximum likelihood involves a variety of matrix inversions.

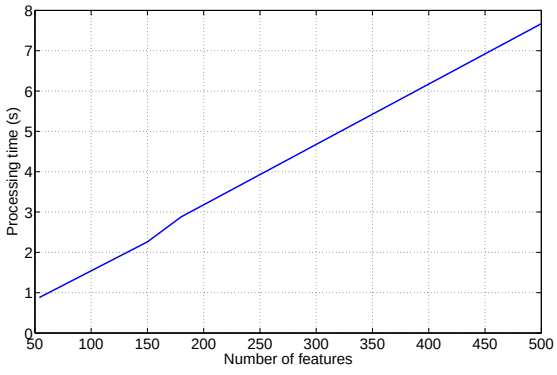

Fig. 4. Processing time to identify good grasping points as a function of the number of features. The plot refers to a 640×480 images.

C. Validation on a WAM manipulator

The proposed algorithm has been implemented on the robotic torso displayed in Fig. 5. The robot is composed of two WAM robotic arms, and is complemented by a Bumblebee 2 stereo camera mounted on two servos providing pan and tilt capabilities. Both arms are equipped with the Barrett Wrist and the Barrett Hand, thus providing 7 degrees of freedom per arm (excluding the degrees of freedom controlling spread and closure of the fingers).

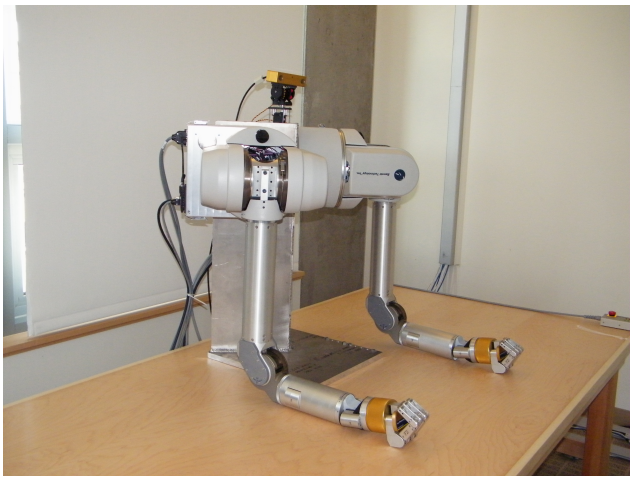

Fig. 5. *George* is a humanoid torso composed by two WAM arms and a Bumblebee 2 stereo camera.

For the experimental purposes relevant to our validation, we used only one arm, namely the right one, and the camera was kept at a fixed position, so that possible variations in performance can be attributed to the algorithm identifying grasping points, and not to changes in the operating conditions. The control software acquires one image from the camera, computes a set of good grasping points according to the techniques formerly described, and then selects the upper-rightmost one. The upper-rightmost one is chosen because the object will be approached from the right by the right arm and to provide enough clearance from the table. The pixel's 3D coordinate is inferred from a typical three-step stereo process involving 1) the undistortion and rectification of the images so that they are row-aligned, 2) finding pixel correspondences between the left and right images using a block matching algorithm, and 3) triangulating the 3D location of a pixel. The 3D point is then transformed into an appropriate robot coordinate and passed to the inverse kinematics module that computes an appropriate posture to approach the object. We performed a simple approach-and-closure technique where the manipulator would be moved to the good grasping point and the end-effector would fully close. For this experiment, 2 trained object (e.g. an eraser and a mug) and 5 novel objects (plastic and steel water bottles, a shampoo bottle, an hexagonal mug, and a box) were presented under different locations, orientations, and lighting conditions. For all the objects except the mugs, the grasping success rate was the same regardless of the technique used. The mugs had differing outcomes, mainly due to our simple approach-and-closure technique that did not take into account the proper finger positioning related to the mug's handle. A companion video associated with this submission shows a few examples of the robot approaching and grasping the water bottle and the hexagonal mug with the original method and the three different hypotheses. It is worthwhile to mention that, in order to preserve the same

operating conditions for each method, clear tape was used on the table to make sure that the objects were positioned in the exact same manner. In other words, the video should be viewed as an example and we stress the fact that the algorithm works equally well for other objects of different and unknown rotations and orientations.

The implementation on the real robot suggested some possible interesting directions for future research. Given a target grasping point, the problem of computing a good position from which the grasping point can be used is not trivial. The reader should note these are two different problems. This paper deals with the problem of computing good grasping points, i.e. to determine where contact between the hand and the object to grasp should happen. Once one of these points is chosen, there is the additional problem of moving the hand to a vantage point from which chances of successfully grasping the object at the given point are maximized. As the focus of this paper is about efficiently computing good grasping points, we have opted for a simplified motion strategy, i.e. to approach the object through a simplified sequence of elementary moves of the right arm. The aforementioned decision to always commit to the rightmost grasping point is a consequence of this strategy. However, following an approach similar to the one of identifying good grasping point, one can envision using a learning algorithm to come up with a good position from which the object can be grasped.

VI. Conclusions

In this paper we have presented an accurate study of a recently proposed algorithm for computing good grasping points from images [15]. After having implemented the algorithm and performed an analysis based on principal components, we formulated two hypotheses. The first is that good grasping points can be reliably inferred using only a small number of filters, and the second postulates that these points can be identified using only a small patch around the point of interest. We have experimentally verified that not only these two hypotheses hold separately, but also jointly, i.e. good grasping points can be identified using a few filters applied to a small patch. More specifically, we have determined that, out of the numerous different filters, only the edge filters significantly contributed to the classification of good grasping points. This finding is sensible since robots and humans alike tend to mostly grab objects by their edges. Moreover, we verified that the possibly competing LDA technique seems inappropriate for the task at hand. The joint verification of these hypotheses leads to a dramatic reduction in the dimension of the feature space, namely from more than 450 down to 54. Our accuracy measure shows an attractive tradeoff between loss of accuracy and reduction of the dimensions of the feature space. This finding has two main consequences. Firstly, the overall computation time to identify good grasping points in a 640×480 image drops from about 8 seconds to below one second, thus paving the way to grasp objects while in motion. Secondly, and not less importantly, the training time also dramatically drops, mostly when it comes to compute the maximum likelihood shown in equation 1. This finding is particularly relevant for our future research, where we envision the robot to perform the training step frequently in order to integrate the experience it acquires while successfully or unsuccessfully trying to grasp new objects. The study builds upon public available training data and has been validated on a real robot. Validation on the real robot also suggests that a similar approach may be valuable in order to learn how to approach an object given a good grasping point.

Acknowledgments

The authors thank Dr. Miguel Carreira-Perpiñán for useful discussions about dimensionality reduction.
This work is partially supported by the National Science Foundation under grant BCS-0821766.

References

[1] A. Anglani, F. Taurisano, R. de Giuseppe, and C. Distante. Learning to grasp by using visual information. In *Computational Intelligence in Robotics and Automation*, pages 7–14, 1999.

[2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[3] C.M. Bishop. *Pattern analysis and machine learning*. Springer, 2006.

[4] U. Castiello. The neuroscience of grasping. *Nature Reviews Neuroscience*, 6:726–736, 2005.

[5] M. Ciocarie, C. Goldfeder, and P. Allen. Dimensionality reduction for hand-independent dexterous robot grasping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3270–3275, 2007.

[6] M. Goodale, A. Milner, L. Jakobson, and D. Carey. A neurological dissociation between perceiving objects and grasping them. *Nature*, 349:154–156, 1991.

[7] M. Jeannerod. The timing of natural prehension movements. *Motor Behavior*, 16(3):235–254, 1984.

[8] M. Kirby and L. Sirovich. Application of the karhunen-loéve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[9] A. Morales, P. Sanz, and A. del Pobil. An experiment in constraining vision-based finger contact selection with gripper geometry. In *IEEE/RSJ Intelligent Robots and Systems Conference*, 2002.

[10] A. Morales, P. Sanz, and A. del Pobil. Vision-based computation of three-finger grasps on unknown planar objects. In *IEEE/RSJ Intelligent Robots and Systems Conference*, 2002.

[11] A. Morales, P. Sanz, A. del Pobil, and A. Fagg. Vision-based three-finger grasp synthesis constrained by hand geometry. *Robotics and Autonomous Systems*, 54(6):496–512, 2006.

[12] M. Carreira-Perpi nán. A review of diemension reduction techniques. Technical Report CS-96-09, University of Sheffield, 1997.

[13] J. Piater. Learning visual features to predict hand orientations. In *ICML Workshop on Machine Learning of Spatial Knowledge*, 2002.

[14] A. Remazeilles, C. Dune, E. Marchand, and C. Leroux. Vision-based grasping of unknown objects to improve disabled people autonomy. In *Robotics: Science and Systems Manipulation Workshop: Intelligence in Human Environments*, 2008.

[15] A. Saxena, J. Driemeyer, and A.Y. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2):157–174, 2008.

[16] P. Weir, C. MacKenzie, R. Marteniuk, and S. Cargoe. Is object texture a constraint on human prehension?: kinematic evidence. *Motor Behavior*, 23:205–210, 1991.