Assessing Image Features for Vision-Based Robot Positioning [⋆]

GORDON WELLS and CARME TORRAS
Institut de Robòtica i Informàtica Industrial (CSIC-UPC) Edifici NEXUS, Gran Capità 2-4, Barcelona 08034, Spain; e-mail: gwells@iri.upc.es, ctorras@iri.upc.es

Abstract. The development of any robotics application relying on visual information always raises the key question of what image features would be most informative about the motion to be performed. In this paper, we address this question in the context of visual robot positioning, where a neural network is used to learn the mapping between image features and robot movements, and global image descriptors are preferred to local geometric features. Using a statistical measure of variable interdependence called Mutual Information, subsets of image features most relevant for determining pose variations along each of the six degrees of freedom (dof's) of camera motion are selected. Four families of global features are considered: geometric moments, eigenfeatures, Local Feature Analysis vectors, and a novel feature called Pose-Image Covariance vectors. The experimental results described show the quantitative and qualitative benefits of performing this feature selection prior to training the neural network: Less network inputs are needed, thus considerably shortening training times; the dof's that would yield larger errors can be determined beforehand, so that more informative features can be sought; the order of the features selected for each dof often accepts an intuitive explanation, which in turn helps to provide insights for devising features tailored to each dof.

Key words: feature selection, global image descriptors, mutual information, robot neurocontrol, variable interdependence, visual robot positioning.

1. Introduction

Vision-based robot positioning involves mapping a set of image features to robot movement commands. The mapping between these two continuous domains is highly nonlinear and, depending on the type of features used, very difficult – if not impossible – to derive analytically. This is especially the case when global image descriptors are employed as features. For local geometric features such as points and lines, explicit analytical relationships may be found relating their image coordinates with camera pose. Knowledge of their location in the observed scene may also be exploited to choose features which will be highly correlated (in the general sense) with movements of the robot-mounted camera. This is not possible, however, for features computed from overall pixel intensities. Not only must

[⋆] Some preliminary results of this work were presented at the IEEE Intl. Conf. on Robotics and Automation, Leuven, May 1998, pp. 2819-2826. This research has been partially supported by the research grant CICYT TAP97-1209 of the Spanish Science and Technology Council.

the feature-to-movement mapping be implicitly estimated by some means, but the feature correlations also must be numerically assessed.

In a previous work [35], we described a prototype visual positioning system based on global image descriptors and a neural network which learned the mapping between the features and the robot movements. By directly mapping descriptors to robot movements, the difficult and often unreliable steps of feature matching, camera calibration, and scene modelling were avoided. The choice of descriptors used, and their correlation with camera displacements, was judged only intuitively, however. We have therefore performed a systematic study in order to more quantitatively determine the relevance of the image features used for controlling each of the robot's degrees of freedom (dof), and their influence on the positioning errors achieved. The results of this study are the object of this paper.

The aforementioned prototype, developed in collaboration with Thomson CSF within the project CONNY, is aimed at the visual inspection of objects that cannot be precisely positioned. The set-up consists of a 6-dof robot arm with a camera mounted on its end-effector, and the goal is to move the camera so as to make the observed image coincide with a given reference image for its subsequent inspection. The training procedure consists of moving the robot end-effector (with the attached camera) from the reference position to nearby random positions, and then applying the back-propagation algorithm to learn the association between the computed image features and the motion performed. In operation, the robot is commanded to execute the inverse of the motion that the network has associated to the given input.

Feature selection can be carried out a priori, through the application of statistical techniques that essentially seek inputs as variant as possible with the output [11, 20, 26], or a posteriori through the use of the neural network itself, by either cell pruning or regularization [4]. The latter method would be extremely costly in our case, since we wish to consider a very large set of possible features and assess their relevance for predicting each of the 6 dof. This would lead to large networks with prohibitive training times. Therefore, we have chosen to perform feature evaluation (and selection) prior to learning, using a statistical dependence measure based on entropy, which is called the Mutual Information (MI) criterion [22].

The paper is structured as follows. Section 2 provides an overview of previous work in visual robot positioning, with an emphasis on the image features used. Section 3 describes in detail the different families of features we have computed and Section 4 contains the results obtained with each of them separately. In Section 5, the MI criterion is introduced and it is applied to the selection of feature subsets. A discussion of the experimental results obtained is presented in Section 6 and, finally, in Section 7 some conclusions as well as the envisaged future research are outlined.

2. Visual Robot Positioning

The aim of visual positioning is to achieve a desired robot pose[⋆] relative to one or more objects in the environment, using information extracted from images of the robot's workspace. When dealing with robot manipulators the desired pose is understood to be that of the end-effector, while, in the case of mobile robots, it refers to the pose of the robot itself. Visual information may be obtained from one or more cameras, mounted either on the robot or else at some fixed location in the environment. Moreover, the control scheme may be a static look-and-move one or else one based on dynamic visual servoing. For extensive reviews on the existing works in this area, the reader is referred to [5, 10, 13, 35]. In this paper, we consider only the case of positioning a robot manipulator on the basis of the information supplied by a single camera mounted on the robot's end-effector.

2.1. PREVIOUS WORK

The case just mentioned is usually tackled by defining a set of geometric image features (typically, points, lines and circles) and then deriving an interaction matrix relating 2D shifts of these features in the image to 3D movements of the camera [8]. In operation, the features in the captured image have to be matched to those in the reference image, in order to find the offsets to which the interaction matrix should be applied. This often requires precise camera calibration [9] and hand-eye calibration [17]. Recently, efforts have been devoted to extending this approach for use with uncalibrated cameras [23, 33].

This geometry-based approach has the advantage of lying on very solid mathematical grounds (projective, affine and Euclidean geometry). However, so far, the processing of complex objects in cluttered scenes at reasonable rates has proven elusive. This is partly due to the difficulty of reliably detecting simple geometric features within images obtained in real-world situations. Object shape and texture, occlusion, noise and lighting conditions have a large effect on feature visibility. Thus, some authors have begun to explore the use of more global image characteristics, as described in Section 2.2.

The advantages of applying neural networks to this task are the direct learning of the interaction matrix, as well as the possibility of avoiding the costly matching of features in the current and reference images. The former advantage was already achieved in a system developed by Hashimoto *et al.* [16], while the latter entails the extraction of global descriptors from the image which both preserve positioning information and permit direct comparison, as investigated in the present work.

2.2. IMAGE FEATURES

Practically all visual positioning and visual servoing applications described to date have relied on the use of local structural features extracted from images, such as

[⋆] Pose is an abbreviation of "position and orientation". The pose of an object in 3D space has six components, while that in 2D space has three components.

points, lines, rectangles, regions, etc. The main advantage of using local features is that they correspond to specific physical features of the observed objects and, once correctly located and matched, provide very accurate information concerning the relative pose between the camera and scene. For instance, the image coordinates of 4 non-coplanar points are sufficient to uniquely determine the pose of an arbitrary object [18]. Example applications using point features may be found in the works of Giordana *et al.* [12], and Hashimoto *et al.* [15].

Object corners and the centroid coordinates and area of holes in an object were used as features by Wilson *et al.* [36]. Rives and Borrelly [30] used edge features to track pipes with an underwater robot, and the road-following vehicle of Dickmanns *et al.* [7] based on edge tracking is well known. Edge contours were tracked in real-time by Wunsch and Hirzinger [37] for manipulation of free-floating objects by a space robot. The projection variations of a circle pattern were used by Kabuka and Arenas [21] in a robot docking application, and the centroid and diameter of circles was used by Harrell *et al.* [14] for fruit tracking with an orange harvesting robot. Vanishing points (intersection of two nearly parallel lines) and line orientations were used by Zhang *et al.* [38] for robot navigation. Chaumette *et al.* [3] and Espiau *et al.* [8] derived variations of a tracking method for points, circles and lines.

Visual positioning based on local features requires that they be reliably extracted and matched under a particular set of working conditions. When effects such as occlusion, noise, uncertainty, and lighting variations interfere, this cannot always be ensured in real-world situations. Many recent applications make use of more global image features to attempt to partially overcome some of these difficulties. Jang *et al.* [19] and Bien *et al.* [2] showed how a number of different global image features could be used within a general visual control framework. Listed features included geometric moments, image projections on a line, random transforms, region templates, and Fourier transforms. Results are given only for simplified tracking tasks using as features the target centroid and area. Optic flow has been applied by Shirai *et al.* [31], Allen *et al.* [1] and Papanikolopoulos [27]. Sipe *et al.* [32], Nayar *et al.* [24] and Deguichi [6] used eigenfeatures for 3-dof positioning and tracking.

Our previous work demonstrates how global image encodings such as Fourier descriptors [34] and geometric moments [35] may be applied to complex, real images to position a 6-dof industrial robot. Although the first results obtained with this prototype were very encouraging, the precision attained was not yet as desired for some of the degrees of freedom, and varied considerably among them. These results motivated the present study.

3. Global Image Features for Pose Estimation

3.1. GEOMETRIC MOMENT DESCRIPTORS

Several image descriptors based on geometric moments may be derived which are useful for robot positioning. While, for pattern recognition applications, moment *invariants* are typically used to recognize image features regardless of the viewing position, for visual servoing it is desired that the moments have a *variant* relationship with respect to the camera pose. Here, eight descriptors involving moments were chosen which characterize several statistical variations in the object's projection in the image when the camera pose is varied on any of its 6 axes.

The general formula for geometric image moments is given by

$$m_{ij} = \sum_x \sum_y x^i y^j f(x, y), \tag{1}$$

where m_{ij} is the moment of order $i + j$, x and y are the coordinates of each pixel in the image, and $f(x, y)$ is the grey-level value of the pixel between 0 and 255. By giving different values to orders i and j, several important statistical characteristics of the image may be encoded. For example, m_{00} is the total "mass" of the image, and m_{02} and m_{20} are the moments of "inertia" around the x and y axes, respectively.

Two important descriptors are the x and y coordinates of the image centroid, which is clearly variant with camera translation parallel to the image plane:

$$\bar{x} = m_{10}/m_{00}, \qquad \bar{y} = m_{01}/m_{00}. \tag{2}$$

To represent the rotation of the object in the image plane, the angle of rotation of the principal axis of inertia may be used. This quantity may be derived from the eigenvectors of the inertia matrix

$$\begin{bmatrix} \bar{m}_{20} & -\bar{m}_{11} \\ -\bar{m}_{11} & \bar{m}_{02} \end{bmatrix}, \tag{3}$$

where $\bar{m}_{11}$, $\bar{m}_{20}$ and $\bar{m}_{02}$ are central moments, defined with respect to the centroid of Equation (2) as:

$$\bar{m}_{ij} = \sum_x \sum_y (x - \bar{x})^i (y - \bar{y})^j f(x, y). \tag{4}$$

The scaling of the object, due primarily to camera translation along the optical axis, may be quantified by the radii of the major and minor inertia axes. These are derived from the eigenvalues, λ_1 and λ_2, of matrix (3):

$$r_1 = \sqrt{\frac{\lambda_1}{m_{00}}}, \qquad r_2 = \sqrt{\frac{\lambda_2}{m_{00}}}. \tag{5}$$

The zero-order moment m_{00} may also be used as a descriptor sensitive to scaling.

The orientation of the major principal axis, θ, is derived (see [29]) from the values of the second moments and the angle of the principal axis nearest to the x axis, ϕ, given by

$$\phi = \frac{1}{2}\tan^{-1}\left(\frac{2\bar{m}_{11}}{\bar{m}_{20} - \bar{m}_{02}}\right). \tag{6}$$

The coefficients of skewness for image projections onto the x and y axes are computed from the third- and second-order moments:

$$Sk_x = \frac{\bar{m}_{30}}{\bar{m}_{20}^{3/2}}, \qquad Sk_y = \frac{\bar{m}_{03}}{\bar{m}_{02}^{3/2}}. \tag{7}$$

3.2. EIGENFEATURES

In the field of computer vision, Principal Component Analysis (PCA) is best known as a method for image compression, feature detection, and pattern recognition. Recently, however, several authors have demonstrated its usefulness for pose estimation [24, 32, 6].

Given a set of multidimensional data samples, the aim of PCA is to determine a reduced orthonormal basis whose axes are oriented in the directions of maximum variance of the data in each dimension, and then project the data onto this new basis. These "principal" axes are the eigenvectors of the data's covariance matrix. By decorrelating the data components in this way, redundancy between them is reduced, and the data is effectively compressed into a more compact representation. Individual data points can then be accurately approximated in just a small subspace of the components with the highest variance. The projected data components are often called eigenfeatures, KL (Karhunen–Loève) features, or PCA features.

To project a set of M brightness images onto a K-dimensional eigenspace, the N-pixel images, of the form

$$\mathbf{x} = [x_1 x_2 \ldots x_N]^T \tag{8}$$

are first normalized so that their overall brightness is unity: $\hat{\mathbf{x}} = \mathbf{x}/\|\mathbf{x}\|$. The average image, $\bar{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M}\hat{\mathbf{x}}_i$, is subtracted from each image, and the resulting vectors are placed columnwise in an image set matrix:

$$\mathbf{X} = [\hat{\mathbf{x}}_1 - \bar{\mathbf{x}}\ \hat{\mathbf{x}}_2 - \bar{\mathbf{x}} \ldots \hat{\mathbf{x}}_M - \bar{\mathbf{x}}]. \tag{9}$$

The covariance matrix of the image set is then obtained as

$$\mathbf{C} = \mathbf{X}\mathbf{X}^T. \tag{10}$$

A set $\mathbf{E} = [\mathbf{e}_1 \mathbf{e}_2 \ldots \mathbf{e}_N]$ of N eigenvectors of $\mathbf{C}$ and their N corresponding eigenvalues λ_i may be computed such that

$$\mathbf{C}\mathbf{e}_i = \lambda_i \mathbf{e}_i. \tag{11}$$

These eigenvectors represent the directions of maximum brightness variance of the images in the set.

The projection $\mathbf{y}_i$ of an arbitrary image $\hat{\mathbf{x}}_i$ onto the subspace $\mathbf{E}_K$ spanned by the $K < N$ eigenvectors corresponding to the K largest eigenvalues of $\mathbf{C}$ is given by

$$\mathbf{y}_i = \mathbf{E}_K^T(\hat{\mathbf{x}}_i - \bar{\mathbf{x}}). \tag{12}$$

In this way, the N-dimensional image is reduced to a set of K eigenfeatures corresponding to the elements of $\mathbf{y}_i$.

3.3. POSE-IMAGE COVARIANCE VECTORS

In order to more directly relate feature variations with the displacements of each pose component, a set of vectors similar to eigenspace may be defined based on the correlation of image brightness variations not with themselves, but directly with the pose displacements of the camera.

For the image set matrix $\mathbf{X}$ of (9), and a matrix $\mathbf{P} = [\mathbf{\Delta p}_1\mathbf{\Delta p}_2 \dots \mathbf{\Delta p}_\mathbf{M}]^T$ of associated 6-dimensional pose displacement vectors $\mathbf{\Delta p}_i = [\mathbf{p}_i - \mathbf{p}_{\text{reference}}]^T$, we define a set $\mathbf{M}$ of 6 Pose-Image Covariance (PCI) vectors as

$$\mathbf{M} = \mathbf{X}\mathbf{P}^\mathbf{T}. \tag{13}$$

The projection $\mathbf{y}_i$ of an arbitrary image $\mathbf{x}_i$ onto the PCI vectors is given by

$$\mathbf{y}_i = \mathbf{M}^T(\hat{\mathbf{x}}_i - \bar{\mathbf{x}}). \tag{14}$$

3.4. LOCAL FEATURE ANALYSIS VECTORS

Local Feature Analysis (LFA) [28] is a recent technique designed to obtain compact, topographic representations of images based on statistically derived local features and their positions. Although LFA was originally developed as a method for compactly representing image ensembles of object classes, such as human faces, based on reduced sets of static features in the images, we have explored its usefulness for representing sets of images whose appearance varies with pose.

The LFA representation is derived directly from the eigenvectors of PCA. Any number $1 \leqslant K \leqslant N$ of eigenvectors may be used, the reconstruction error of an image reconstructed with LFA features being exactly equal to that of PCA reconstruction with the same number of eigenmodes. Multiplying the K eigenvectors $\mathbf{e}_i$ in $\mathbf{E}_K$ by the whitening factor $1/\sqrt{\lambda_i}$ normalizes the variance of data components projected onto eigenspace to unity. This ensures that data projections onto the resulting vectors,

$$\hat{\mathbf{e}}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{e}_i, \tag{15}$$

are minimally correlated, while maintaining the same degree of decorrelation of the eigenvectors themselves. The LFA space is then defined as

$$\mathbf{K} = \mathbf{E}_K \hat{\mathbf{E}}_K^T, \tag{16}$$

where $\hat{\mathbf{E}}_K$ is the matrix of normalized eigenvectors. The LFA matrix **K** has dimensions $N \times N$. Each column of **K** is a unique N-element receptive field centered at the pixel location in the image corresponding to its column index in **K**, and represents the normalized covariance of that pixel with all other pixels in the image. These vectors act as feature detectors matched to the feature that is expected near their respective centers.

The projection $\mathbf{y}_i$ of an arbitrary image $\mathbf{x}_i$ onto the LFA subspace,

$$\mathbf{y}_i = \mathbf{K}^T (\hat{\mathbf{x}}_i - \bar{\mathbf{x}}), \tag{17}$$

is therefore an image of the same size, each pixel of which reflects the activation of the receptive field centered at that location in the image. However, since the receptive fields in a local region are correlated, representing a feature redundantly, only the one which best represents each feature need be retained. A small subset of LFA vectors centered at chosen features of interest is therefore sufficient to characterize a particular image. The set of scalar-valued features is obtained by projecting the images onto the individual columns of **K** corresponding to the chosen feature locations.

For pose estimation, it is desirable to find features that are maximally covariant with displacements of pose. In a similar manner as used to derive the PCI vectors, the covariance of the pose vectors with the LFA projections may therefore be used to select a subset of feature locations.

A matrix **M** of six covariance vectors of the pose matrix **P** with the projections **Y** of the image set **X** of (9) onto the LFA space may then be defined as

$$\mathbf{M} = \mathbf{Y}\mathbf{P}^T. \tag{18}$$

The locations of the maxima in these six vectors may be used to select the corresponding LFA receptive fields for each pose component, as described in the next section.[⋆]

4. Pose Estimation Using Neural Networks

Vision-based positioning of a robotic manipulator may be implemented using a control scheme like the one shown in Figure 1. In our neural network-based approach, the control law takes the form of a neural network which, when input the feature offsets between the currently observed image and a prespecified desired, or “reference” image, outputs the movement command required to return the robot to

[⋆] Another alternative could be to use the image projections on the covariance vectors **M** directly as features, similar to the PCI vectors, although this possibility was not explored here.

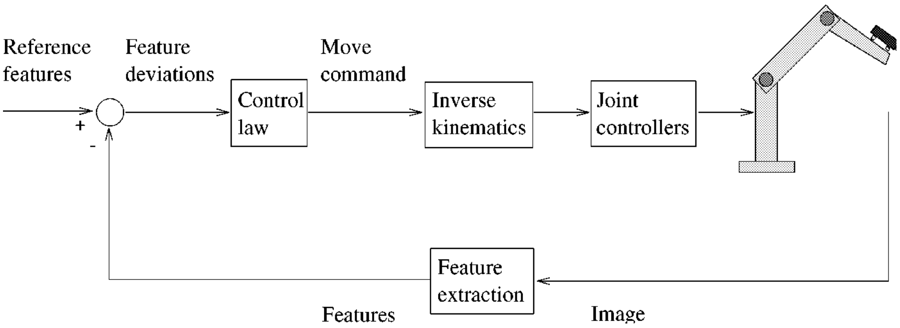

Figure 1. Visual positioning control scheme.

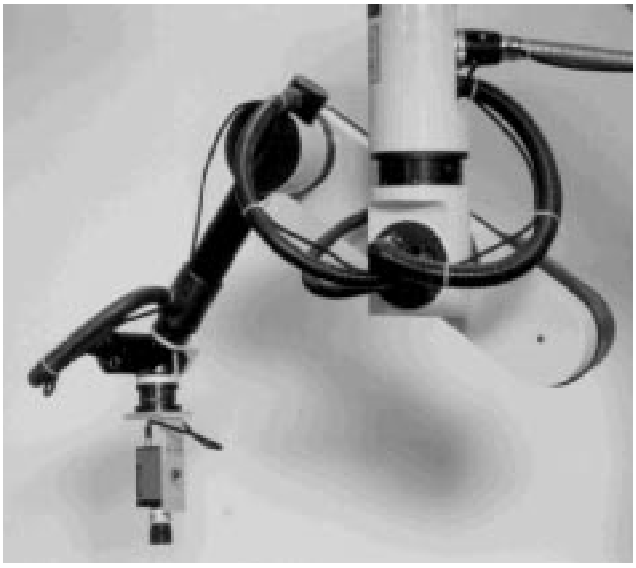

Figure 2. The GT Productique 6-dof robot with wrist-mounted camera, used to acquire the image set.

the reference position. While, in a previous article [35], we presented the results of positioning trials using trained networks, here we concentrate on the effectiveness of the network training for the different feature sets studied.

Feedforward neural networks were used to learn the direct mapping between feature variations and pose displacements when a robot-mounted camera was moved away from the reference position. A black-and-white CCD camera mounted on the wrist of a 6-dof GT Productique industrial robot (see Figure 2) was used to acquire a set of 1000 training images of a scene, consisting of an automobile cylinder head on a white table. Images were taken at random poses within a range of -25 to 25 mm translation and -15 to 15 degrees rotation with respect to the reference position for the three coordinate axes, where x, y, z are the horizontal, vertical, and optical axes, respectively. The reference position was chosen so as to

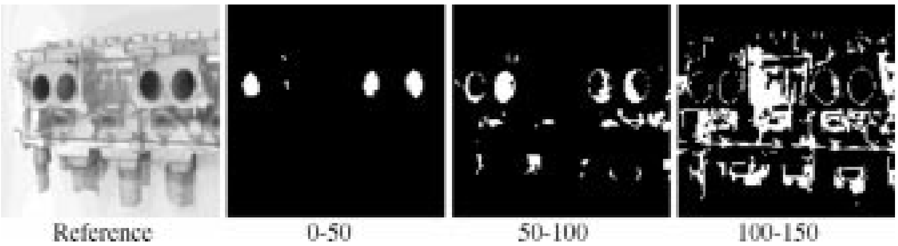

Figure 3. The reference image and intensity slices for three grey-level intervals.

have the camera centered perpendicularly above the object's surface, at a distance of approximately 550 mm. The center of rotation was a point centered 550 mm in front of the camera on the optical axis (near the object's surface), which helped maintain the object within the field of view. A translational component is therefore derived from the rotation around the two horizontal axes, makint the actual maximum ranges of T_x and T_y equal to $\pm[25+550\sin(15)] = \pm 167$ mm. The 6-element pose displacement vector was stored with each image as the desired outputs for the neural networks.

The Levenberg–Marquardt algorithm was used to train backpropagation networks implemented using the Matlab Neural Networks Toolbox commercial software package. All networks had 30 hidden nodes and the number of input nodes equal to the length of the feature vector used in each case. A separate network was trained for each degree of freedom, each network having a single output for one of the 6 pose components. A set of 250 training examples was reserved as a test set. All networks were trained until no further reduction in the RMS error for the test set could be achieved.

Geometric moment descriptors were computed for three versions of each image filtered by "intensity slicing", consisting of thresholding the images within three different minimum and maximum intensity ranges (0–50, 50–100, 100–150). The resulting images contained features roughly localized around holes and other structural regions of the object, and segmented from the bright background, as seen in the example of Figure 3. The input feature vector for the neural network was composed of the moment descriptors computed for all three filtered versions of each image (denoted by the superscripts 50, 100, 150), totalling 24 features: $\mathbf{f} = (\bar{x}\ \bar{y}\ \theta\ r_1\ r_2\ m_{00}\ S_x\ S_y)^{50,100,150}$.

For the network trained with PCA features, the image projections onto the first 24 eigenvectors of the image set were the inputs. Several of the eigenvectors are shown in Figure 4. The SLAM software library [25] was used to compute the eigenvectors and image projections.

The 6 PCI vectors for the image set are shown in Figure 5. Note that each one (except perhaps T_z) has an apparent visible symmetry related to its corresponding pose component, and that the vector pairs for translational and rotational movements on opposite x and y axes have a similar symmetry. Unlike for GM and PCA features, the networks trained with PCI features had only 6 inputs.

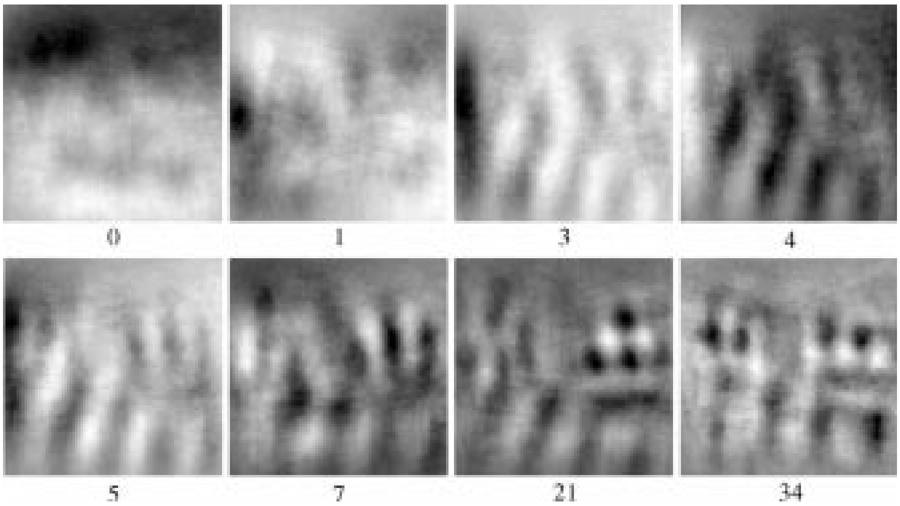

Figure 4. Some eigenvectors of the image set.

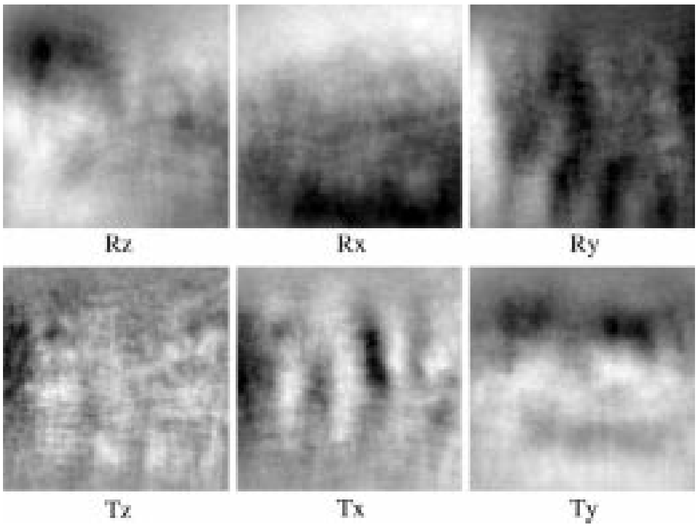

Figure 5. The six PCI vectors of the image set.

A subset of LFA vectors was chosen for each pose component by selecting those vectors centered in the regions of highest covariance with the pose component. The Pose-LFA Covariance (PCLFA) vectors were first computed as in (18). The resulting vectors for the image set used are shown in Figure 6. Since covariance has maximum values at -1 and 1, the vectors were normalized to this range, and their absolute values computed. The centers of the brightest image regions were then located by filtering the image with a center-weighted subwindow mask and choosing those subwindows with the maximum overall intensity. The weights of the mask were computed as the inverse pixel distance of each component pixel of

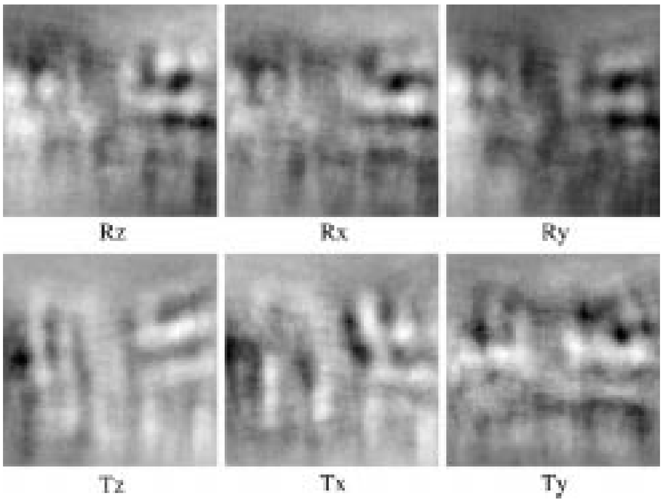

Figure 6. The six Pose-LFA Covariance vectors of the image set. The LFA projections were computed using 85 eigenvectors.

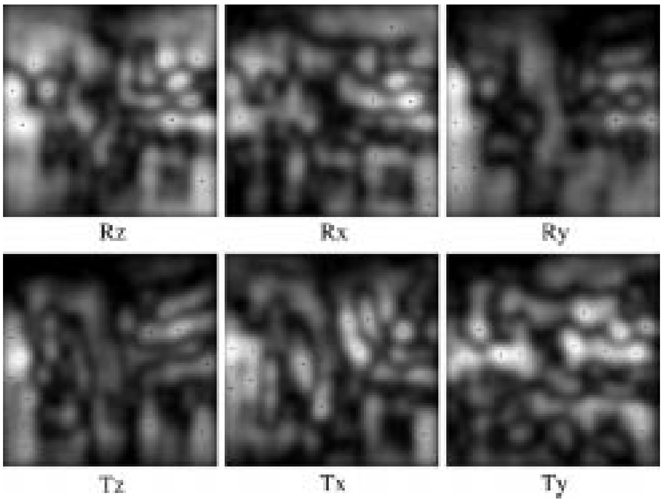

Figure 7. The Pose-LFA Covariance vectors filtered with a 12 × 12-pixel center-weighted mask. The twelve 12 × 12-pixel subwindows with maximum intensity in each image have their centers marked with crosses. These locations were used as indices for selecting twelve LFA vectors for each pose component.

the subwindow to the subwindow centroid. A 12 × 12 mask was chosen based on visual inspection of the average size of the bright regions in the PCLFA images. The centroid locations of the 12 brightest subwindows in each PCLFA image were chosen as LFA vector indices for the corresponding pose component. The filtered PCLFA images and chosen LFA vector locations are shown in Figure 7. The first LFA vector chosen for each pose component is shown in Figure 8.

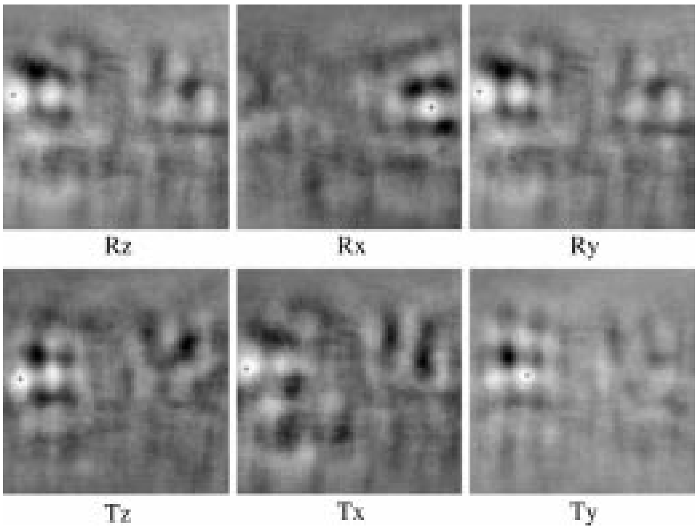

Figure 8. The first LFA vector selected for each pose component.

Table I. Final RMS test-set errors for neural networks trained with GM, PCA, PCI, and LFA features using 35 and 85 eigenvectors.

Features	R_z	R_x	R_y	T_z	T_x	T_y
GM	0.03	0.05	0.07	0.14	0.07	0.05
PCA	0.10	0.20	0.22	0.41	0.29	0.29
PCI	0.09	0.11	0.18	0.32	0.25	0.14
LFA (35 ev)	0.15	0.25	0.19	0.39	0.29	0.24
LFA (85 ev)	0.21	0.39	0.24	0.44	0.32	0.29

Two different experiments were performed with LFA vectors. In the first, the LFA vectors were computed using the 35 eigenvectors previously used to obtain the PCA features. In the second, 85 eigenvectors were used to see if any possible improvement might result. Networks were trained for each of the two sets of resulting LFA vectors.

The final RMS test-set errors for the networks trained with the 4 feature types are shown in Table I. The values correspond to the lowest errors achieved for 3 training trials of each network. The best results, for all 6 coordinate axes, were clearly obtained for GM features. Error values are somewhat lower for PCI features than for PCA features. The worst results were obtained for LFA vectors. Surprisingly, error values were even higher when the LFA vectors were computed based on 85 eigenvectors than when only 35 were used.

5. Feature Selection Using Mutual Information

The aim, when training a neural network, is to map a subset $\mathbf{f}$ of a collection $\mathbf{F} = \{f_1, f_2, \ldots, f_m\}$ of prospective input variables (features) to a set $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ of output variables. This is only possible, however, to the extent that some statistical dependence exists between $\mathbf{y}$ and $\mathbf{f}$. The more informative the input variables are about the outputs, the lower the network's final training error will be, assuming that training is efficient in all other respects. Choosing the subset of inputs which is maximally informative about the outputs can therefore help to optimize learning with the available input variables and, with a minimum number of features, reduce training times.

A useful statistical measure, from Information Theory, for quantifying the dependence between variables is the *mutual information*. For a single input variable f and an output y, which are both assumed to be random variables, the mutual information (MI) is given by

$$I(f, y) = \sum_f \sum_y P(f, y) \log_2 \frac{P(f, y)}{P(f)P(y)}, \tag{19}$$

where the summations are computed over the suitably discretized values of f and y. Although probability density estimates, and consequently MI values, are dependent on the number of chosen discretization intervals, in our work the order of selected features was found to be the same for any number between 5 and 20, so 10 intervals were used.

By measuring the "peakedness" of the joint probability between the variables f and y, the mutual information captures arbitrary (linear and nonlinear) dependencies between them. It is equivalent to the Kullback–Leibler distance, or cross entropy, between the joint distribution and the product of the marginal distributions, and measures the degree to which knowledge provided by the feature vector decreases the uncertainty about the output.

Clearly, the MI of a subset of input variables with an output variable is not equal to the sum of the individual MI values of its component inputs, since the output may be dependent on some function of two or more inputs, but apparently independent of any single input separately. Consequently, the maximally informative subset of inputs may only be found by computing the MI of all possible subsets of inputs with outputs. However, for a given number n of candidate features, the number of possible subsets is 2^n and, therefore, computationally expensive to consider when n is large. An alternative is to build a suboptimal set based on the MI values of individual variables using some sort of heuristic.

Since the features themselves may be dependent on each other, selecting features based only on their individual MI with the outputs can lead to input sets with redundant features which add little to the MI of the set as a whole. The goal is to find a subset of inputs whose MI is maximum with respect to the outputs, but minimal with respect to each other. The chosen heuristic should aim to fulfill

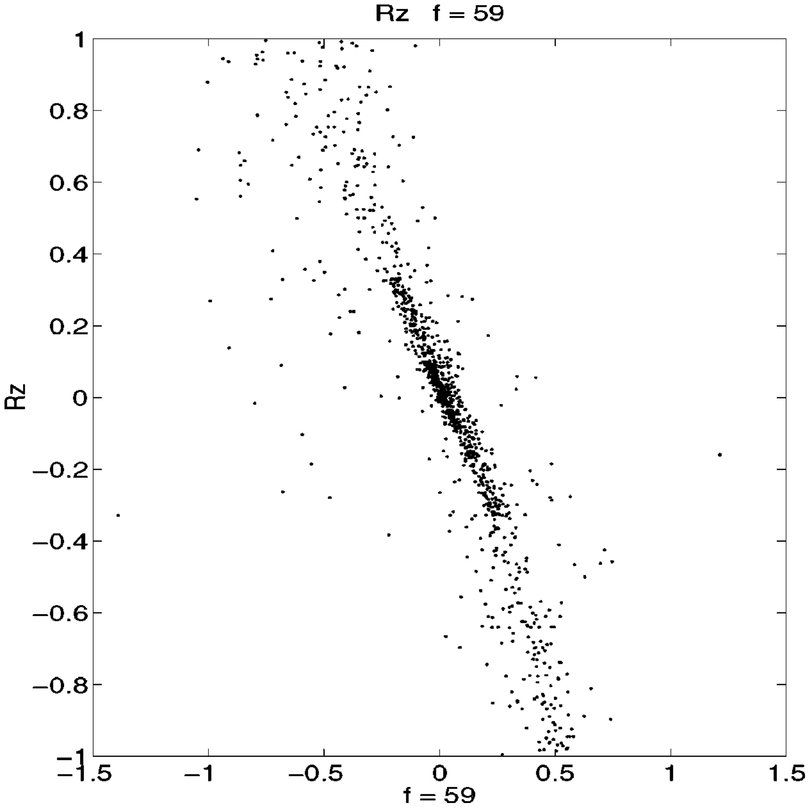

Figure 9. Output variable R_z plotted against input feature θ^{150} (labelled as feature 59), showing a clear dependence between these two variables.

these criteria as best as possible. One such method [39] consists of choosing, at each selection step, the candidate feature f_i with the smallest Euclidean distance from the point of maximum $I(f_i, y)$ and minimum $\sum_k I(f_i, f_k)$ in the 2D space formed by these two variables, where $\sum_k I(f_i, f_k)$ is the sum of the MI values of the candidate feature f_i with the already selected features f_k in the set. This is illustrated for one selection step in Figures 9 and 10.

The MI selection procedure was applied to a set of 65 features previously computed for the image set, containing 24 GM, 35 PCA, and 6 PCI features. In view of the high error values obtained on all axes for networks trained with LFA features alone, and the still unexplained worsening of results when the LFA representation was computed using more eigenmodes, it was chosen not to include this class of features in the set. For the set used, the first 24 features selected for each pose component, in order of decreasing MI, are listed in Table II.

Minimum feature subsets were found for each pose component by building subsets of features with MI values above minimum threshold values, which were varied between 0 and 1. As the threshold increased, the number of features in the

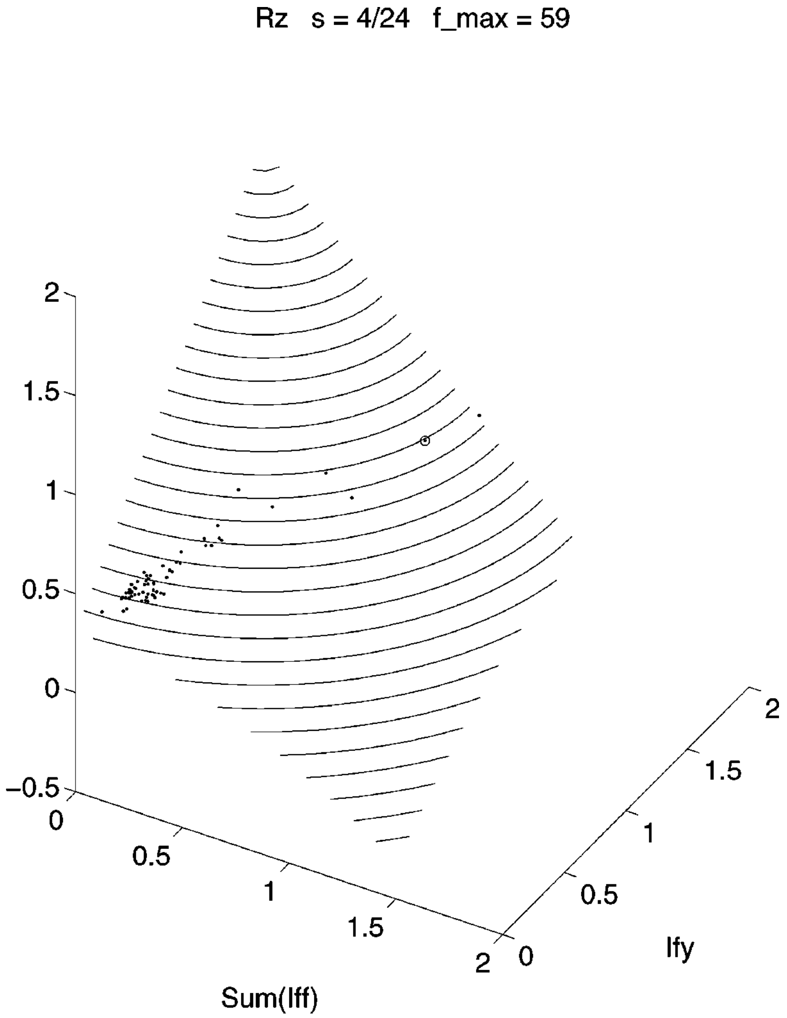

Figure 10. A 3D representation of the weights assigned to each feature during the MI selection process. $I(f_i, y)$ is plotted against $\sum_k (f_i, f_k)$ in the x–y plane, and weights are represented by the z axis. Contour lines mark curves of constant weight values, computed as the distance to the "ideal" point where x is zero and y is maximum. Here, three features f_k have already been selected, and the candidate feature f_i with the highest weight is circled, in this case θ^{150} (labelled as feature 59).

subsets decreased. The RMS test-set error values for networks trained with each subset were plotted against the MI threshold values, as shown in Figure 11, and the subset with the lowest error value was selected. An important observation from this graph is that MI values are much lower overall for the 3 translational components than for the 3 rotational components, resulting in higher RMS errors even for larger feature sets.

The feature sets with the lowest RMS errors are summarized in Table III.

Table II. The first 24 MI-selected features from the set of 65 GM, PCA and PCI features, in order of decreasing MI.

R_z	R_x	R_y	T_z	T_x	T_y
θ^{50}	r_2^{100}	r_1^{50}	e_{21}	e_7	$\bar{y}^{50}$
PCI_{R_z}	$\bar{y}^{150}$	r_1^{150}	r_1^{100}	S_x^{100}	θ^{50}
θ^{150}	S_y^{100}	m_{00}^{150}	S_y^{150}	e_{34}	e_{30}
S_y^{50}	m_{00}^{150}	$\bar{x}^{100}$	r_2^{50}	e_{17}	e_{18}
e_1	S_x^{150}	e_5	e_{31}	m_{00}^{50}	e_{10}
e_6	m_{00}^{100}	S_x^{50}	S_x^{150}	e_{13}	e_{21}
θ^{100}	$\bar{y}^{100}$	m_{00}^{100}	e_{29}	S_x^{150}	PCI_{R_x}
S_y^{150}	PCI_{T_z}	r_2^{150}	m_{00}^{50}	e_{10}	e_{15}
e_8	θ^{50}	$\bar{x}^{50}$	S_x^{100}	m_{00}^{150}	e_{11}
e_{31}	r_2^{150}	e_{11}	e_{32}	e_{22}	e_{31}
e_y	e_0	e_2	e_{18}	$\bar{y}^{100}$	S_y^{100}
e_{20}	e_{21}	e_{21}	r_2^{150}	r_1^{100}	e_{26}
e_{29}	S_x^{100}	$\bar{x}^{150}$	m_{00}^{100}	e_{15}	e_{29}
e_{11}	e_{19}	e_4	e_{28}	S_y^{100}	e_{20}
S_y^{100}	S_y^{150}	e_8	$\bar{x}^{150}$	e_{19}	PCI_{T_z}
e_{21}	PCI_{R_x}	r_1^{100}	e_9	$\bar{x}^{150}$	m_{00}^{50}
e_{33}	r_2^{50}	S_y^{100}	e_{26}	e_{29}	e_{17}
e_{32}	e_{29}	S_y^{50}	e_{13}	e_{14}	e_{22}
e_{26}	m_{00}^{50}	e_3	$\bar{y}^{100}$	e_{11}	S_x^{150}
r_2^{150}	S_y^{50}	e_{10}	S_y^{50}	e_{28}	$\bar{y}^{150}$
e_{28}	e_{18}	S_x^{150}	e_{16}	e_8	e_{34}
e_{16}	$\bar{x}^{100}$	PCI_{R_y}	S_x^{50}	e_{31}	e_7
S_x^{150}	e_{30}	S_x^{100}	e_{30}	S_y^{150}	e_{28}
e_{25}	e_7	$\bar{y}^{50}$	S_y^{100}	e_5	r_2^{100}

Table III. RMS test-set error for minimum feature subsets computed with MI for GM, PCA and PCI features. The number of features in each subset is indicated below the error values.

	R_z	R_x	R_y	T_z	T_x	T_y
RMS error	0.03	0.05	0.08	0.19	0.16	0.09
N features	5	10	16	24	24	7

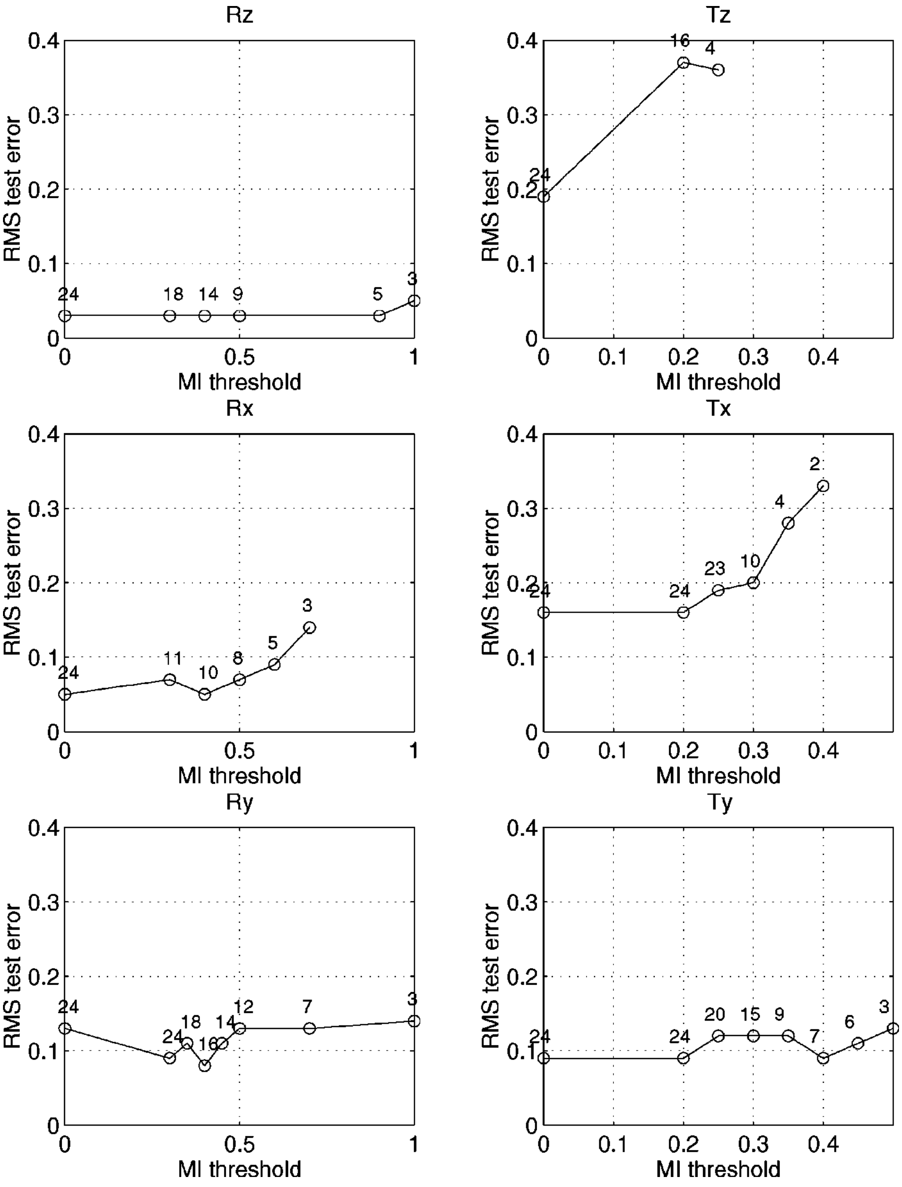

Figure 11. RMS test error vs. MI threshold. On top of each circle the number of features with an MI value greater than the corresponding threshold is recorded. The RMS error obtained by training a neural network with only those features is then plotted. Note that the MI scale for the translational axes is half that of the rotational axes, indicating that features are less informative overall for the translational dof's.

6. Discussion

The results presented in the last two sections may be analyzed in several ways. Let us first discuss the quantitative benefits resulting from the application of the MI criterion.

By comparing Tables I and III, one can immediately see that the effect of the MI selection varies considerably depending on the degree-of-freedom (dof) considered. Thus, in the case of R_z and R_x, the best precision is attained with only 5 and 10 features, respectively, instead of the 24 used when the same precision was attained with geometric moments alone. This best precision could not be reached with any other single feature type. The benefits are also noticeable in the case of T_y and R_y, for which a very good precision is attained with only 7 and 16 features, respectively. Finally, in the case of T_z and T_x, the MI selection cannot outperform the use of geometric moments alone,* although it outperforms the other two types of features.

The reason for this large disparity in the effect of MI selection becomes evident when looking at Figure 11. The MI of the features considered varies largely with each dof. This variation is even larger than one could perceive at first glance, since a different scale has been used in the abcissas for rotations and translations. For all rotational dof's, at least 10 features have an MI greater than 0.4, and at least 5 of them also surpass the 0.6 threshold. None of the translational dof's lead to similar MI values, although T_y falls close behind, while T_z is undoubtably the worst. This is, in fact, another interesting outcome of the MI assessment: the features we are using do not provide enough information on T_z variation, which indicates that other features should be sought (possibly with the help of the MI criterion) in order to diminish the error for this dof.

Many interesting qualitative observations about the results may be also be made by studying in detail the features selected for each dof (see Table II). For example, the orientation angles of the first principal axis for the images of Figure 3 ($\theta^{50}, \theta^{100}, \theta^{150}$) would seem to have a clear relationship with R_z and, therefore, it is not surprising that they are among the seven features most highly ranked. The skewness coefficients of the projection on the y axis for the same images ($S_y^{50}, S_y^{100}, S_y^{150}$) also seem intuitively relevant and, consistently, they are within the 15 most highly ranked. Note also that the image ordering for these two types of features is maintained. The PCI vector for this dof is ranked second, which seems very natural, and more so when one looks at its image-like representation (see Figure 5). Note that, with the possible exception of PCI_{T_z}, each PCI vector resembles a particular eigenvector (e.g. PCI_{R_z} and e_1, PCI_{R_x} and the inverse of e_0, PCI_{R_y} and e_4, PCI_{T_x} and e_3, PCI_{T_y} and e_0). What is more difficult to interpret is the role played by the eigenfeatures. In principle, those with the least concentric symmetry

* This is counterintuitive, but not contradictory, since the MI is not computed for all possible feature subsets (which is computationally unfeasible), but instead features are selected one by one on the basis of their individual MI values corrected by a redundancy term.

should be selected and, up to what can be seen in Figure 4 (e_1, in particular), this is consistent with the performed selection.

In the case of R_x, the length of the second principal axis would seem to be a good indicator of this dof and, effectively, r_2^{100}, r_2^{150}, and r_2^{50} are within the first 17 features selected, with r_2^{100} in the 1st place. The three skew coefficients S_y^{100}, S_y^{150}, and S_y^{50} are intuitively within the first 20 features selected. The relationship of the y coordinate of the centroid with R_x variations is also visibly evident in the image set, which makes the presence of $\bar{y}^{150}$ and $\bar{y}^{100}$ in the first 7 features selected seem logical. The corresponding PCI vector, PCI_{R_x}, is the 16th feature selected but, curiously, PCI_{T_z} is ranked much higher at number 8. The reason for this becomes clear when the weight assignments are plotted as in Figure 10: Although PCI_{R_x} has the highest MI value with R_x of all the remaining features (0.8), it also has the highest sum value of its MI with the already selected features (0.7). PCI_{T_z} is therefore selected first, because its information/redundancy balance is more favorable (0.4 and 0.4, respectively).

Similar observations may be made regarding R_y. The length of the first principal axis is selected, with r_1^{50} and r_1^{150} in 1st and 2nd place, and r_1^{100} as number 16. The corresponding PCI vector, PCI_{R_y}, is number 22, and the three skew coefficients S_x^{50}, S_x^{150}, and S_x^{100} are within the first 23 selected features, with S_x^{50} in 6th place. The x coordinate of the centroid is visibly related with R_y in the image set, and $\bar{x}^{100}$, $\bar{x}^{50}$, and $\bar{x}^{150}$ are within the first 13 features selected.

Understandably, $\bar{y}^{50}$ is the first feature selected for T_y, which has an MI value of 1.1. The MI values of all the remaining features are 0.5 or less, and, unlike the first feature, show a negligible relationship with the output when plotted as in Figure 9. When their weight assigments are graphed as in Figure 10, they appear tightly clustered together, which explains the unexpected selection of features such as θ^{50} and PCI_{T_z}, and the absence of other expected ones such as PCI_{T_y}. The information content of all except the first feature is simply too low to permit any meaningful discrimination between them, indicating that other feature types must be sought for this dof.

The high test errors obtained for T_x reflect the unintuitive set of features selected for this dof. While the x-coordinate of the centroid would be expected in the set, $\bar{x}^{50}$ and $\bar{x}^{100}$ are missing. Only $\bar{x}^{150}$ is selected as number 16, but the seemingly less related feature $\bar{y}^{100}$ is ranked higher, in 11th place. The 1st feature selected is e_7, which has a vague visible relationship with movements on T_x. As seen in Figure 11, the explanation lies in the fact that the MI values of all features with T_x are below 0.5, indicating that they provide practically no information concerning this dof.

Lastly is T_z, with the worst test-set performance. Although the highly ranked features r_1^{100}, r_2^{50}, m_{00}^{50}, r_2^{150}, and m_{00}^{100} would appear to be particularly sensitive to the changes of scale resulting from movements on T_z, they, like all the other features, have MI values so low (0.3 or lower) as to indicate they are virtually unrelated, thus precluding any further analysis of the selected feature set. Different feature types are clearly needed for this dof.

Overall, there is a clear tendency of geometric-moment descriptors to appear among the most relevant features, especially in those cases in which a satisfactory precision is reached, which is consistent with the lower test-error values obtained for these features alone. The selection in many cases of apparently unrelated features may be explained by the fact that, in absolute terms, the MI values for all except the few most highly ranked features for the best-learned dof's are simply very low. An MI value of 0.1, for example, is typical of totally unrelated variables.

Aside from feature relevancy alone, there are two clear sources of the low precision values obtained for T_x and T_z. Since the observed object (the cylinder head) in the image set used is longer on the x axis than on the y axis, portions of the object fall outside the field of view on the left and right sides of the image for the majority of the poses. This effect reduces the sensitivity of the computed image features to displacements in T_x. In the case of T_z, the relatively low sensitivity of the image variations to movements along the optical axis are evident upon visual inspection of the image set, to the point of being mostly overshadowed by the changes of scale caused by movements on the other dof's. Improving the precision for these two pose components in future work will require finding features which overcome these limitations.

7. Conclusions

In this paper we have used the Mutual Information criterion to evaluate the sensitivity of several types of global image features to camera translations and rotations. Contrary to the goal of object recognition applications, where invariant features are sought, here we look for image features that are as variant as possible with camera pose.

We have presented the novel application of two new types of global features for pose estimation: LFA features based on Local Feature Analysis, and a new pose-image covariance feature called PCI. We also give results of using eigenfeatures for 6-dof positioning for the first time, as well as several geometric-moments descriptors of a higher order than those used in our previous works. In particular, geometric moment descriptors (including the newly introduced skew coefficients) and the new PCI vectors were ranked as highly informative of the rotational movements by the MI criterion.

Using the MI selection procedure as a preprocessing step before training a neural network, we have shown how mixed sets of all four feature types may be assembled which provide the maximal information for pose estimation with a minimal number of features. In this way, we have achieved a considerable reduction in the number of features needed to obtain a given learning precision for the three rotational degrees of freedom, which has reduced neural-network training times by reducing the number of inputs.

Besides providing a way to automate the feature selection process, MI permits foreseeing which degrees of freedom will yield larger errors, thus allowing one to look for more informative features before actually training the neural network. As for the three translational degrees of freedom, MI analysis has provided a quantitative explanation of why their precision values obtained so far have been consistently low. The significantly low MI of these dof's with all the features considered indicates that other features must be devised which are more highly correlated with translational movements.

Although the study has been carried out in the context of visual positioning of a robot arm, this same methodology clearly may be applied to other visually-guided robotic arrangements, such as mobile robots and underwater robots, as a tool for selecting features as well as to guide in the design of new ones.

Our future work will center on the search for more informative features for estimating the translational dof's, and a sensitivity analysis of both the camera and the trained neural networks in order to quantitatively account for other possible sources of positioning error.

References

1. Allen, P., Timcenko, B., and Michelman, P.: Hand-eye coordination for robotic tracking and grasping, in: K. Hashimoto (ed.), *Visual Servoing*, World Scientific, Singapore, 1993, pp. 33–70.
2. Bien, Z., Jang, W., and Park, J.: Characterization and use of feature-Jacobian matrix for visual servoing, in: K. Hashimoto (ed.), *Visual Servoing*, World Scientific, Singapore, 1993, pp. 317–363.
3. Chaumette, F., Rives, P., and Espiau, B.: Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing, in: *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, Sacramento, 1991, pp. 2248–2253.
4. Cibas, T., Fogelman, F., Gallinari, P., and Raudys, S.: Variable selection with neural networks, *Neurocomputing* **12** (1996), 223–248.
5. Corke, P.: Visual control of robot manipulators – a review, in: K. Hashimoto (ed.), *Visual Servoing*, World Scientific, Singapore, 1993, pp. 1–31.
6. Deguichi, K.: Visual servoing using eigenspace method and dynamic calculation of interaction matrices, in: *Proc. 13th Intl. Conf. on Pattern Recognition*, Vol. 1, 1996, pp. 302–306.
7. Dickmanns, E., Mysliwetz, B., and Christians, T.: An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles, *IEEE Trans. Systems Man Cybernet.* **20** (1990), 1279–1284.
8. Espiau, B., Chaumette, F., and Rives, P.: A new approach to visual servoing in Robotics, *IEEE Trans. Robot. Automat.* (1992), 313–326.
9. Faugeras, O.: *Three Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Boston, 1993.
10. Feddema, J.: Visual servoing: A technology in search of an application, in: *Notes for Workshop M-5 (Visual Servoing), IEEE Intl. Conf. on Robotics and Automation*, San Diego, 1994.
11. Fukunaga, K.: *Statistical Pattern Recognition*, 2nd edn, Academic Press, 1990.
12. Giordana, N., Bouthemy, P., and Chaumette, F.: 2D Model-based tracking of complex shapes for visual servoing tasks, in: *Notes for Workshop WS2 (Robust Vision for Vision-based Control of Motion), IEEE Intl. Conf. on Robotics and Automation*, Leuven, 1998.

13. Hager, G. and Hutchinson, S.: Visual servoing: Achievements, issues and applications, in: *Notes for Workshop M-5 (Visual Servoing), IEEE Intl. Conf. on Robotics and Automation,* San Diego, 1994.
14. Harrell, R., Slaughter, D., and Adsit, P.: A fruit-tracking system for robotic harvesting, *Machine Vision Appl.* **2** (1989), 69–80.
15. Hashimoto, K., Akoi, A., and Noritoyu, T.: Visual servoing with redundant features, in: *Proc. 35th Conf. on Decision and Control*, 1996.
16. Hashimoto, H., Kubota, T., Kudou, M., and Harashima, F.: Self-organizing visual servo system based on neural networks, *IEEE Control Systems* (1992), 31–36.
17. Horaud, R. and Dornaika, F.: Hand-eye calibration, *Intl. J. Robotics Research* **14**(3) (1995), 195–210.
18. Horaud, R., Conio, B., and Lebouleux, O.: An analytic solution for the perspective 4-point problem, *Computer Vision, Graphics and Image Process.* **44** (1989), 33–44.
19. Jang, W. and Bien, Z.: Feature-based visual servoing of an eye-in-hand robot with improved tracking performance, in: *Proc. IEEE Conf. on Robotics and Automation*, Sacramento, 1991, pp. 2254–2260.
20. Janabi, F. and Wilson, W.: Automatic selection of image features, *IEEE Trans. Robot. Automat.* **13**(6), (1997), 890–903.
21. Kabuka, M. and Arenas, A.: Position verification of a movile robot using standard pattern, *IEEE J. Robotics Automat.* **3**(6) (1987), 505–516.
22. Li, W.: Mutual information functions versus correlation functions, *J. Statist. Phys.* **60**(5/6) (1990), 328–387.
23. Mohr, R., Boufama, B., and Brand, P.: Understanding positioning from multiple images, *Artificial Intelligence* **78**(1/2) (1995), 213–328.
24. Nayar, S., Nene, S., and Murase, H.: Subspace methods for robot vision, *IEEE Trans. Robotics Automat.* **12**(5) (1996), 750–759.
25. Nene, S., Nayar, S., and Murase, H.: SLAM: Software Library for Appearance Matching, Technical Report CUCS-019-94, Dept. of Computer Science, Columbia University, USA, 1994.
26. Papanikolopoulos, N.: Selection of features and evaluation of visual measurements during robotic visual servoing tasks, *J. Intelligent Robotic Systems* **13** (1995), 279–304.
27. Papanikolopoulos, N.: Adaptive control, visual servoing and controlled active vision, in: *Notes for Workshop M-5 (Visual Servoing), IEEE Intl. Conf. on Robotics and Automation*, San Diego, 1994.
28. Penev, P. and Atick, J.: Local feature analysis: A general statistical theory for object representation, *Network* **7**(3) (1996), 477–500.
29. Prokop, R. and Reeves, A.: A survey of moment-based techniques for unoccluded object representation and recognition, *Graphical Models and Image Process.* **54**(5) (1992), 438–460.
30. Rives, P. and Borrelly, J.: Real-time image processing for image-based visual servoing, in: *Notes for Workshop WS2 (Robust Vision for Vision-based Control of Motion), IEEE Intl. Conf. on Robotics and Automation*, Leuven, 1998.
31. Shirai, Y., Okada, R., and Yamane, T.: Robust visual tracking by integrating various cues, in: *Notes for Workshop WS2 (Robust Vision for Vision-based Control of Motion), IEEE Intl. Conf. on Robotics and Automation*, Leuven, 1998.
32. Sipe, M., Casasent, D., and Neiberg, L.: Feature space trajectory representation for active vision, in: S. Rogers (ed.), *Applications and Science of Artificial Neural Networks III*, Vol. 3077, Society of Photo-Optical Instrumentation Engineers, 1997, pp. 254–265.
33. Sturm, P.: Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction, in: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Puerto Rico, 1997, pp. 1100–1105.

34. Venaille, C., Wells, G., and Torras, C.: Application of neural networks to image-based control of robot arms, in: *Proc. 2nd IFAC Symp. on Intelligent Components and Instruments for Control Applications (SICICA)*, Budapest, 1994, pp. 281–286.
35. Wells, G., Venaille, C., and Torras, C.: Vision-based robot positioning using neural networks, *Image and Vision Comput.* **14** (December 1996), 715–732.
36. Wilson, W., Williams, C., and Janabi, F.: Robust image processing and position-based visual servoing, in: *Notes for Workshop WS2 (Robust Vision for Vision-based Control of Motion), IEEE Intl. Conf. on Robotics and Automation*, Leuven, 1998.
37. Wunsch, P. and Hirzinger, G.: Real-time visual tracking of 3D objects with dynamic handling of occlusion, in: *Proc. IEEE Intl. Conf. on Robotics and Automation*, Albuquerque, 1997.
38. Zhang, Z., Weiss, R., and Hanson, A.: Automatic calibration and visual servoing for a robot navigation system, in: *Proc. IEEE Conf. on Robotics and Automation*, Atlanta, 1993, pp. 14–19.
39. Zheng, G. and Billings, S.: Radial basis function network configuration using mutual information and the orthogonal least squares algorithm, *Neural Networks* **9**(9) (1996), 1619–1637.