Real-time Hand Posture Recognition Using Range Data

S. Malassiotis, M.G. Strintzis

PII: S0262-8856(07)00209-0
DOI: 10.1016/j.imavis.2007.11.007
Reference: IMAVIS 2647

To appear in: *Image and Vision Computing*

Received Date: 16 May 2005
Revised Date: 14 September 2007
Accepted Date: 12 November 2007

Real-time Hand Posture Recognition Using Range Data

S. Malassiotis and M. G. Strintzis

Informatics & Telematics Institute, Thessaloniki, Greece,

Email: malasiot@iti.gr

Abstract

A hand posture recognition system using 3D data is described. The system relies on a novel 3D sensor that generates a dense range image of the scene. The main advantage of the proposed system, compared to other gesture recognition techniques, is the capability for robust unconstrained recognition of complex hand postures such as those encountered in sign language alphabets. This is achieved by explicitly utilizing 3D hand geometry. Moreover, the proposed approach does not rely on color information, and guarantees robust segmentation of the hand under varying illumination conditions, and scene content. Several novel 3D image analysis algorithms are presented, covering the complete processing chain: 3D image acquisition, arm segmentation, hand-forearm segmentation, hand pose estimation, 3D feature extraction, and gesture classification. The proposed system is extensively evaluated.

1 Introduction

Even though, in the last two decades we have witnessed a rapid evolution of computing, communication and display technology, the physical human-computer interface has largely remained unchanged since the first workstations. However, traditional interface devices, keyboards and mice, are inadequate for modern applications such as interaction with complex three-dimensional environments and sign language recognition. Recently, several innovative controllers and sensors have been investigated with a view towards a more ”natural” interaction with the machine. Several of these new systems, such as glove-based devices, compromise convenience by requiring the user to be instrumented with encumbering devices in order to achieve high expressiveness.

The use of gesture recognition provides an attractive alternative to the cumbersome interface devices for human-computer interaction that are typical today. Vision-based recognition of hand gestures in particular, promises natural, unobtrusive, human-computer interaction [25]. This is based on analyzing signals acquired by imaging sensors such as video, infrared or ultrasonic, inferring the geometry and motion of the hand and finally mapping to a set of predefined gestures. An important potential application of this technology comes from the possibility to develop advanced interfaces for the interaction with virtual objects. These objects can be images on a computer screen. The user can ”manipulate” the objects by moving his/her hand and performing actions like ”grasping” and ”releasing”. The computer uses gesture recognition to reproduce the user actions on the virtual object and the result of the operation is shown in the graphical interface so as to provide feedback to the user. Application is in simulation, robot teaching, graphical interface control, device control and Virtual Reality. Another important application is the interpretation of gestures from the sign-language alphabet to aid natural interaction of hearing impaired people with computing devices [30].

Vision-based recognition of hand gestures, especially dynamic hand gestures, is an extremely challenging interdisciplinary task due to following three reasons: (1) hand gestures are rich in diversity, ambiguity, and space-time variation; (2) the human hand is a complex non-rigid object; (3) computer vision itself is an ill-posed problem; (4) it is difficult to obtain estimates faster than standard video frame rate.

Much of recent research effort has been concentrated on gesture recognition, i.e. the interpretation of dynamic hand trajectory patterns. Color-based [30] or model-based tracking techniques [26] together with temporal pattern classifiers such as Hidden Markov Models have demonstrated acceptable performance. The problem of posture recognition, that is the interpretation of finger configuration, has received less attention.

Model-based techniques (see [11] for a recent review) rely on fitting articulated 3D models of the hand on 2D image features. Use of a 3D model allows exploitation of geometric (and kinematic) constraints on finger configuration which helps dealing with the ambiguity caused by perspective projection of a 3D structure on a 2D image. However if only a single frame is available then the problem has still multiply solutions. One approach around this problem is to perform an expensive global search in a database of hand templates labelled by the corresponding finger configuration parameters [1, 2], and resorts to measuring the similarity between the input image and the template image. Template images are synthesized using the 3D model. Other techniques rely on the detection of high-level features such as the fingertips and using inverse kinematics [5] or non-linear regression [15] to solve for the unknown configuration parameters. Markers are used to facilitate feature detection. Model-based techniques have demonstrated good results but on the expense of computational complexity which makes them not suitable for real-time application.

In many cases posture recognition is performed on a limited set of predefined postures and therefore full 3D finger configuration estimation is not necessary. In this case one can work in a bottom-up fashion using image features directly. The main difficulties in this case is the selection of features which are invariant to illumination and pose variations (e.g. orientation histograms, silhouettes), and to deal with self-occlusions, clutter backgrounds and within-class variations of postures. The most recent techniques in this domain [31, 37] rely on extensive training databases to efficiently model within-class variability.

The majority of reported techniques facilitate hand segmentation using a uniform background and/or assume that the hand is the only object visible in the scene. Good results on hand segmentation may be obtained using color information as soon as the skin-color model has been trained for the specific sensor/environment [36].

Many of the above difficulties have more tractable solutions if 3D information is exploited. Several researchers have proposed using more than one camera, or exploiting 3D information acquired by passive stereo sensors. In [34] a gesture recognition system based on a range sensor is proposed. The algorithm is capable of recognizing a limited set of simple manipulative gestures, containing static finger configuration, while the hand segmentation problem is not addressed. To cope with the problem of occlusions, a multi-viewpoint hand gesture tracking system is proposed in [35]. The best viewpoint is selected based on the estimated hand rotation in 3D. Then, 2D shape Fourier descriptors are extracted and used to recognize a limited set of simple postures. A finger tracking scheme relying on a multiple camera configuration is proposed in [16]. The system combines various cues such as color, edges, 3D shape and motion for robust detection of fingertip position and orientation. A similar system, relying however on dense stereo measurements, is described in [18]. The hand pose estimation problem is also addressed in [8]. A 3D model of the hand is used that is iteratively fitted to dense 3D data. A stereo-based gesture recognition technique is described in [14]. The orientation of the arm and location of the hand is estimated by means of sparse 3D data. This is subsequently used to drive a color-based hand segmentation algorithm. Moment-based 2D shape gesture classification is finally performed on the perspectively unwarped color images. In [20] depth data from a time-of-flight camera are used for gesture recognition. A limited set of hand postures are recognized but constraints are posed on the placement and orientation of the hand with respect to the body. Finally, methods for fitting articulated 3D models on 3D point clouds have been proposed in [3, 9], which are however expensive for real-time applications.

In this paper a novel hand posture recognition system using 3D data is described. The system relies on a novel 3D sensor that generates a dense range image of the scene. The main novelty and advantage of the proposed system, compared to the 3D gesture recognition techniques mentioned above, is the use of 3D data for posture classification, and the capability for robust recognition of complex hand postures such as those encountered in sign language alphabets over unconstrained environments. The proposed approach does not rely on color information, and guarantees robust segmentation of the hand under various illumination conditions and scene contents. Finally, using novel 3D image analysis algorithms, the paper addresses the complete processing chain, unlike other techniques which, by means of simplifying assumptions, bypass one or more of the following stages: 3D image acquisition, arm segmentation, hand-forearm segmentation, hand pose estimation, 3D feature extraction, and posture classification. Apart from demonstrating very satisfactory classification results the system achieves real-time performance on conventional hardware. The system is also evaluated for use in a keyboard-less application interface scenario.

On the following section we briefly describe the employed 3D acquisition setup. 3D images are then processed and the arm is segmented from the rest of the body.

This is described in section 3. Then in section 4 we present algorithms that further segment the hand from the arm. A pose compensation algorithm (section 5) is subsequently applied on the cloud of 3D points belonging to the detected hand region. This results in normalized depth images which are classified to a set of predefined posture classes as described in 6.The performance of the algorithms is evaluated with extensive experiments in section 7 while section 8 concludes the paper.

2 3D data acquisition

A sensor capable of real-time acquisition of 3D dynamic scenes is employed in this paper. It is based on low cost devices, an off-the-shelf CCTV-color camera and a standard video projector. The sensor relies on active illumination of the scene with a colored illumination pattern by means of a video or slide projector. A color camera captures the resulting image and by analyzing the deformation of the pattern on the object surface the 3D coordinates of each point on the surface may be computed (see fig. 1). This computation is performed sufficiently fast to allow real-time 3D image acquisition. Unlike similar structured light system, this device may capture dynamic scenes, thanks to the special color-encoded light pattern used. The reader is refereed to [33] for further technical details.

In our experiments the system was optimized for an application scenario, where subjects are located about one meter from the camera, and so that the upper torso and their arms are contained in the working space of the sensor ($75cm \times 75cm \times 75cm$). Inside this volume the average depth accuracy is about $1mm$. The spatial resolution of the range images is equal to the color camera resolution in the one direction while in the other direction it is dependent on the width of the color stripes of the projected light pattern (see fig. 1) and the bandwidth of the surface signal.

For a low-bandwidth surface such as the human body the resolution is thus close to the resolution of the color camera (768×576).

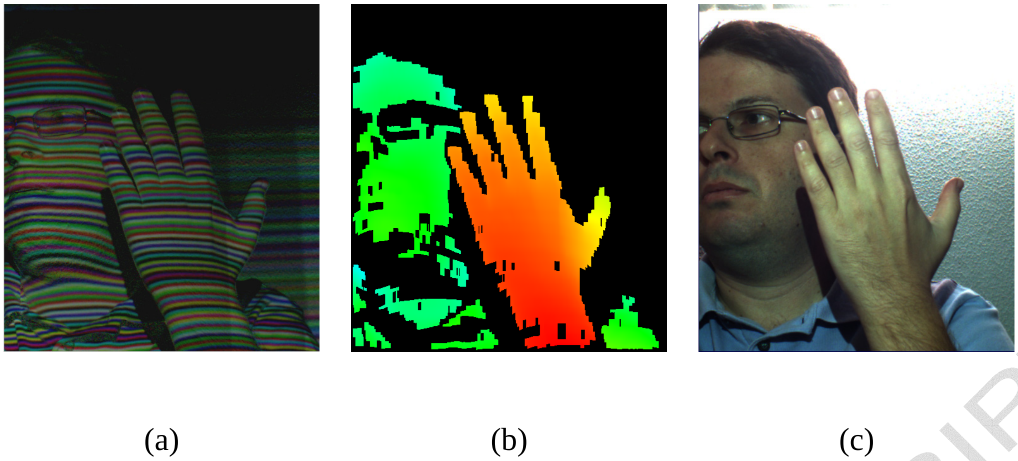


Fig. 1. Typical colour (b) and range image (c) acquired by the 3D sensor. The range image depth resolution is 12 bits per pixel, and therefore is color-encoded to show the complete range of values. The image on the left (a) is the striped-image used to compute depth information.

The acquired range images contain artifacts and missing points mainly over areas that cannot be reached by the projected light. Instead of filtering or interpolating 3D data, a process that may lead to further artifacts, we prefer making subsequent processing stages robust to the above artifacts.

Given the camera-projector calibration parameters we may reconstruct the 3D coordinates of image points. The projection equation is:

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

where P is a 3×4 projection matrix containing implicit and explicit camera calibration parameters, $\mathbf{x} = (X, Y, Z)$ are the coordinates of the 3D point, $\mathbf{X} = (x, y)$ are the projected point coordinates, and s is a scaling constant [12]. Therefore, for a range image pixel $\mathbf{X}$ with depth value $Z = Z(\mathbf{X})$ it is possible to estimate the X, Y coordinates of $\mathbf{x}$. Using a simple pinhole camera model:

$$P = \begin{bmatrix} f_x & 0 & x_0 & 0 \\ 0 & f_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

we easily obtain

$$\mathbf{x} = \begin{bmatrix} (x - x_0)Z/f_x \\ (y - y_0)Z/f_y \\ Z \end{bmatrix} \tag{3}$$

on the coordinate frame fixed on the camera. In the following when the term “3D data” will be used this will actually denote the 3D coordinates of the corresponding range image pixel values.

Finally, we note that it is possible to acquire color images of the object quasi-synchronously with the corresponding 3D images. This may be achieved by rapidly alternating a striped light pattern with white light giving rise to a striped image (which is further processed to give 3D data) followed by a normal color image. In this way we may achieve a frame rate of 12 frames (color + depth) per second. Although in principle one may exploit color information to increase classification accuracy, in this paper we concentrate on 3D image sequences only so that the method will be applicable to a wide variety of sensors.

3 Arm Segmentation

An important step in vision-based gesture recognition is the segmentation of user hands from the background, i.e. the user's body and other objects in the scene. The problem is usually simplified by several assumptions on the scene content, illumination, motion and camera configuration, by controlling the environment and by limiting the working space. Under these constraints, skin color-based segmentation, motion detection and background subtraction based on a previously trained background model have demonstrated relatively good performance [25]. The benefit of using depth information is that robust image segmentation may be achieved without posing any constraints to the environment or the users of the system. This is a very important requirement for natural human-computer interaction.

An initial segmentation of the scene may be obtained by means of thresholding the depth values, assuming that the background, user body, and arms span distinct ranges of depth. If the above assumption holds, then the histogram of depth values consists of well separated modes, and an optimal threshold is estimated. In many situations, however, this is not the case, especially when the user's body is slanted with respect to the camera plane. We have used a thresholding technique [24] in order to subtract the background that may be reasonably assumed to be well separated from the user.

The segmentation of the subject's arms is achieved by means of a hierarchical unsupervised clustering procedure. This is based on the observation that the various parts of the body, such as the arms, torso and head, form compact 3D clusters in space. Clustering techniques such as K-means lead to poor results since clusters are elongated and linearly transformed with respect to each other, while maximum-likelihood techniques such as the Expectation-Maximisation approach rely on good initial values for the cluster centers and orientations. The proposed agglomerative (hierarchical) segmentation algorithm consists of the following steps:

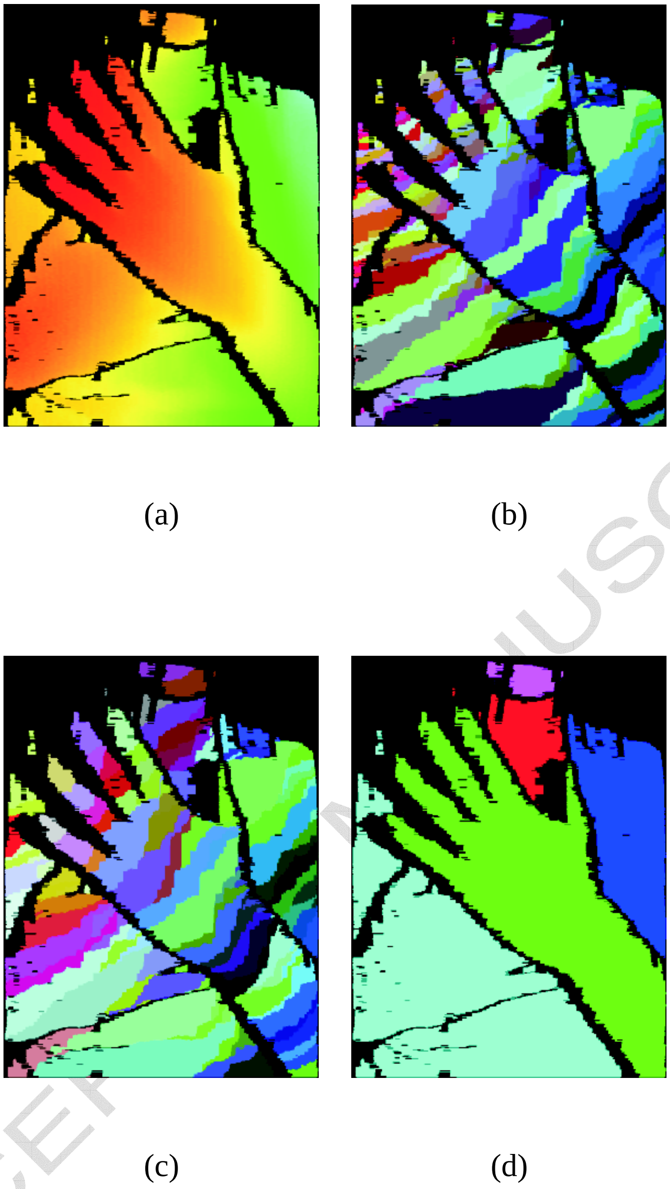


Fig. 2. Range image clustering procedure. (a) Original image, (b) initial clustering, (c) refinement, (d) final clustering.

(1) An initial clustering is obtained by sequentially scanning the depth image, and classifying each pixel according to the distance from previously classified pixels in its neighborhood. The distance measure used is the Euclidian metric

in the Z component. This procedure leads to a large number of small regions (fig. 2b).

(2) The aim of this step is to favor larger regions, by merging small regions into larger ones. The smallest cluster $\mathcal{D}_i$ is selected and merged with the cluster $\mathcal{D}_j$ which minimizes the inter-cluster variance $S_B(i,j)$, given by

$$\begin{aligned} S_B(i,j) &= \mathit{Trace}\{n_i(\mathbf{m_i}-\mathbf{m})(\mathbf{m_i}-\mathbf{m})^T + n_j(\mathbf{m_j}-\mathbf{m})(\mathbf{m_j}-\mathbf{m})^T\} \\ &= n_i\|\mathbf{m_i}-\mathbf{m}\|^2 + n_j\|\mathbf{m_j}-\mathbf{m}\|^2 \end{aligned} \tag{4}$$

where $\mathbf{m_i}, \mathbf{m_j}$, and $\mathbf{m}$ are the centers of clusters $\mathcal{D}_i$, $\mathcal{D}_j$ and total cluster respectively. The iterative procedure is terminated as soon as a specific number of clusters is reached (fig. 2c).

(3) Finally, hierarchical merging of adjacent clusters is performed. Two clusters $\mathcal{D}_k$, $\mathcal{D}_l$ are merged if the total scatter of the combined cluster is minimized. The total scatter $S_T(k,l)$ measure is given by:

$$S_T(k,l) = \mathit{trace}\{\mathbf{S}_T(k,l)\} = \mathit{trace}\{\sum_{\mathbf{x}\in\mathcal{D}_k,\mathcal{D}_l}(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^T\}$$

where $\mathbf{S}_T(k,l)$ is the total scatter matrix. The procedure is repeated iteratively and terminated when a specific number of clusters is reached (fig. 2d).

The clusters corresponding to the arms may be subsequently selected by employing prior knowledge. From the 3-4 clusters closer to the camera, we select those that have a relatively elongated shape, corresponding to the arms. The distance of a cluster from the camera is measured by:

$$d = \min_{\mathbf{x}_i} Z_i$$

while the elongation criterion is

$$\lambda_{min} < \lambda_2/\lambda_1 < \lambda_{max}$$

where λ_1, λ_2 are the largest eigenvalues of the cluster's scatter matrix, and $\lambda_1 > \lambda_2$. Their ratio is a measure of the width of the cluster to its length, and for human arms this is limited to a range of a-priori determined values.

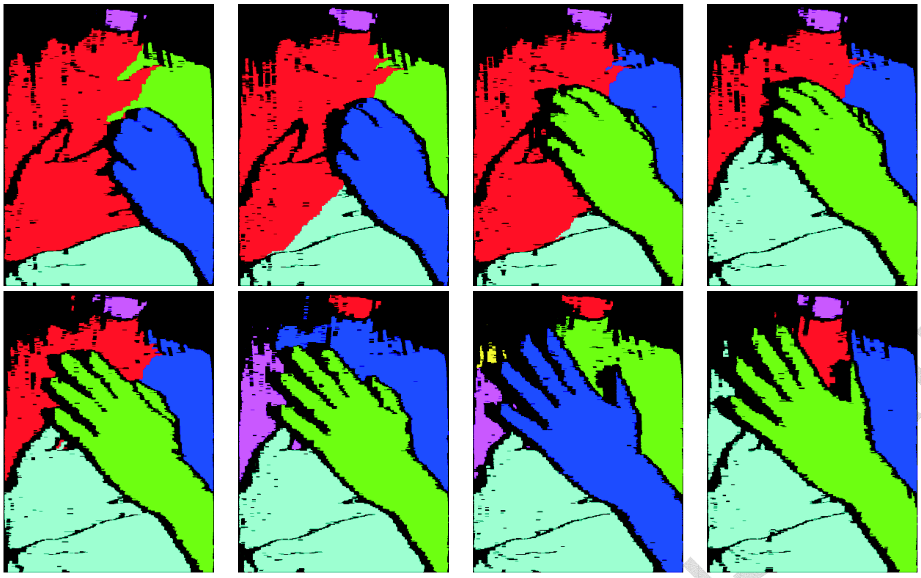


Fig. 3. Range image clustering results.

Figure 3 illustrates segmentation results where one arm is adjacent to the body and the other is very close to it (10-15 cm) .

4 Hand - Forearm Segmentation

The segmentation of the palm and fingers from the forearm is important for the accurate estimation of the hand pose and the subsequent feature extraction procedure. Our approach relies on statistical modelling of the arm points in 3D space. This is similar to the approach adopted in [18] for the segmentation of the arm from the body.

The probability distribution of a 3D point $\mathbf{x}$ is modelled as a mixture of two Gaussians:

$$P(\mathbf{x}) = P(\text{hand})P(\mathbf{x}|\text{hand}) + P(\text{forearm})P(\mathbf{x}|\text{hand}) \tag{5}$$

$$= \pi_1 N(\mathbf{x}; \mu_1, \mathbf{\Sigma}_1) + \pi_2 N(\mathbf{x}; \mu_2, \mathbf{\Sigma}_2) \tag{6}$$

where π_1, π_2 are prior probabilities of the hand and forearm respectively, and

$$N(\mathbf{x}; \mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{3/2}|\mathbf{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right].$$

Maximum-likelihood estimation of the unknown parameters π_k, μ_k, $\mathbf{\Sigma}_k$, $k = 1, 2$ from the 3D data is obtained by means of the Expectation-Maximisation algorithm [19]:

$$\begin{aligned}
p_{kn} &= \frac{\pi_k N(\mathbf{x}_n; \mu_k, \mathbf{\Sigma}_k)}{\sum_i N(\mathbf{x}_n; \mu_i, \mathbf{\Sigma}_i)}, \\
\mu_k &= \frac{\sum_n \mathbf{x}_n p_{kn}}{\sum_n p_{kn}} \\
\mathbf{\Sigma}_k &= \frac{\sum_n (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T p_{kn}}{\sum_n p_{kn}} \\
\pi_k &= \frac{\sum_n p_{kn}}{\sum_n \sum_k p_{kn}}
\end{aligned}$$

where p_{kn} are the posterior probabilities of the the state k given the data and the model parameters. The convergence of the above iterative procedure relies on good initial parameter values. In our case these may be obtained by exploiting prior knowledge of the arm geometry. Let $u_i,\ i = 1, \ldots, 3$ be the eigenvectors of the arm scatter matrix $\mathbf{R}_T = trace\{\sum_{\mathbf{x}}(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T\}$, computed from the data points $\mathbf{x}_i$ belonging to the arm, ordered according to the magnitude of the corresponding eigenvalues. Initial estimates of the unknown parameters were selected by:

$$\mu_1 = \mathbf{m} + \rho_1 s_{min} \mathbf{u}_1,\ \ \mu_2 = \mathbf{m} + \rho_2 s_{max} \mathbf{u}_1$$

where

$$s_{min} = \min_{\mathbf{x}_i}\{(\mathbf{x}_i - \mathbf{m})^T \mathbf{u}_1\},\ s_{max} = \max_{\mathbf{x}_i}\{(\mathbf{x}_i - \mathbf{m})^T \mathbf{u}_1\},$$

$$\boldsymbol{\Sigma}_k = \mathbf{U}\boldsymbol{\Lambda}_k\mathbf{U}^T, \;\; \boldsymbol{\Lambda}_k = diag(\rho_k^2\lambda_1, \lambda_2, \lambda_3),$$

$$\pi_k = \rho_k, \;\; k = 1, 2,$$

where $\mathbf{U}$ is the orthogonal eigenvector matrix of $\mathbf{R}_T$ and λ_i, $i = 1, \ldots, 3$ the corresponding eigenvalues, while ρ_1, ρ_2 are constants related to the relative length of the hand and forearm with respect to the arm (in the experiments $\rho_1 = 1/3$, and $\rho_2 = 2/3$ were used).

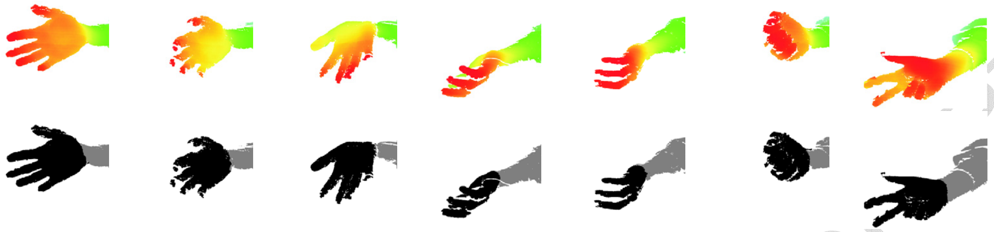


Fig. 4. Hand-Forearm Segmentation results

Classification of a 3D point $\mathbf{x}_n$ to the class k is dictated by the maximum likelihood criterion i.e. by selecting the class that maximizes p_{kn}. Experimental results demonstrate robustness of the algorithm under various orientations of the palm and finger configurations (see fig. 4).

5 3D Pose Estimation and Compensation

Availability of 3D information leads to efficient estimation of the orientation of the hand. This allows transformation of input point cloud into a canonical pose compensated depth image, where a variety of classification algorithms may be efficiently applied.

The pose estimation algorithm takes as input a cloud of 3D points which were classified as belonging to the hand. An estimate of the 3D orientation of the hand is obtained by computing the principal direction of the 3D points, given by the eigenvector $\mathbf{u}_1$ of the hand scatter matrix $\mathbf{H}_T$, corresponding to its largest eigenvalue. Geometrically, it is easy to see that the 3D line $\mathbf{p} = \mathbf{m} + \lambda \mathbf{u}_1$ passing from the center of mass $\mathbf{m}$ of the points with direction $\mathbf{u}_1$, minimises the orthogonal distance of the points from this line. Equivalently, the 3D plane $(\mathbf{p} - \mathbf{m})^T \mathbf{u}_3 = 0$, where $\mathbf{u}_3$ is the eigenvector corresponding to the smallest eigenvalue, is the best fitting plane, i.e. minimizes the orthogonal distance of the 3D points from this plane. To limit the effects of the fingers to the estimated orientation vector, each 3D point $\mathbf{x}_i$ is weighted by its distance $\|d\|$ from the center of the palm, estimated using the algorithm proposed in [7]:

$$w_i = \frac{1}{1 + k_w \|d\|^2}$$

where k_w is a constant. To cope with outliers that are due to 3D sensor noise, a robust LMS (Least Median of Squares) algorithm [27] was used.

Compensation of the 3D pose is performed by transformation of the 3D data using the estimated pose parameters, and then projection into the camera plane to create normalized (pose compensated) depth images, which are eventually used for classification. An orthonormal coordinate frame is defined, with center $\mathbf{m}$, and its x, y and z axis given by $\mathbf{u}_1, \mathbf{u}_2$ and $\mathbf{u}_3$ respectively. We wish to align this frame with the frame of the camera, and place the center of mass $\mathbf{m}$ at a distance $\tilde{z}$ from the camera center. This may be accomplished by the rigid transformation:

$$\mathbf{p}' = R^T \mathbf{p} + \mathbf{c} - R^T \mathbf{m} \tag{7}$$

where $\mathbf{p}$ is a 3D point on the camera frame, $\mathbf{p}'$ is the corresponding point in the normalized frame, R is the matrix with columns $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ i.e. $R = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$ and $\mathbf{c} = [0, 0, \tilde{z}]^T$. Given (7), pose compensated depth images are created by means of image warping. A 3D rectangular mesh is first defined using the 3D points. The vertex connectivity is directly available from the corresponding depth image.

The vertices of the mesh are then transformed using (7) and a scan-line Z-buffer rendering algorithm [13] is applied to the transformed mesh. The resulting Z-buffer contains the rectified depth image. Alternatively, one may use a more efficient 2D warping from the original to the compensated depth image, as proposed in [14]. However, this approach relies on the assumption that all points lie on a plane, and thus introduces distortions for certain finger configurations. where this assumption does not hold. We have alternatively applied a warping algorithm that is relatively

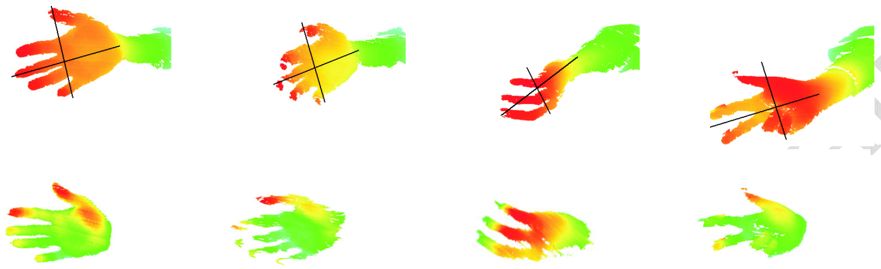


Fig. 5. 3D pose estimation and compensation results. The first row shows original depth images and estimated local coordinate frame (projections of x and y axis on the camera plane). The second raw shows the corresponding normalised depth images.

Experimental results of the proposed 3D pose estimation and compensation procedure are illustrated in fig. 5.

6 Hand Posture Classification

Several approaches have been recently proposed for the recognition of free form objects from range image data (See [4] for an extensive review). However, the problem of range classification of non-rigid objects such as the human hand has not been explicitly addressed. Object recognition techniques may be roughly divided to appearance-based and feature-based techniques.

Feature-based techniques, such as spin-images [17] and point-signatures [6] work by extraction of a view-invariant representation of the 3D object based on 3D surface curvature information. The reliance on curvature makes them sensitive to noisy or incomplete data. Also the computational complexity of these techniques is prohibitive for real-time application [4]. Appearance-based techniques on the contrary represent a 3D object by a series of depth images corresponding to different viewing angles. With this approach 3D object classification resorts to measuring the similarity of 2D images (depth images).

In this paper we have employed an appearance-based eigenspace technique. Principal component analysis (PCA) applied on a set of depth images was used to compute an orthogonal space of reduced dimension.

Let $\mathbf{f}_i, i = 1, \ldots, k$ be vectors constructed by lexicographical scanning of k training images, and $\mathbf{F} = [\mathbf{f}_1 - \tilde{\mathbf{f}}, \mathbf{f}_2 - \tilde{\mathbf{f}}, \ldots, \mathbf{f}_k - \tilde{\mathbf{f}}]$ be a $d \times k$ matrix with:

$$\tilde{\mathbf{f}} = \frac{1}{k} \sum_{i=1}^{k} \mathbf{f}_i.$$

Let $\mathbf{v}_i, i = 1, \ldots, k$ be the eigenvectors of $\mathbf{F}\mathbf{F}^T$ ordered by the descending magnitude of the corresponding eigenvalues $\sigma_1 > \sigma_2 > \ldots > \sigma_k$. Then, a feature vector $\mathbf{a}$ of reduced dimension r may be computed, by projecting $\mathbf{f}$ corresponding to an input image on the sub-space defined by the first r eigenvectors:

$$\mathbf{a} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r]^T (\mathbf{f} - \tilde{\mathbf{f}}).$$

Classification of an input depth image, to one of the predefined hand posture classes represented by the training set is performed using the k-nearest-neighbor rule [10]. This rule classifies a new input image and extracted feature vector $\mathbf{a}$ by computing the set of posture classes of the k nearest training samples and selecting from them the class which appears most often in this set. The proximity of input vector to training sample vectors is evaluated using the following distance measure:

$$D(\mathbf{a}, \mathbf{a}_i) = (\mathbf{a} - \mathbf{a}_i)^T \Lambda^{-1} (\mathbf{a} - \mathbf{a}_i), \;\; \Sigma = diag\{\sigma_1, \sigma_2, \ldots, \sigma_r\}. \tag{8}$$

The success of this approach relies on the use of a rich training set containing a representative subset of all possible variations of the hand. Then the eigenspace is guaranteed to capture the degrees of freedom of these variations [21]. However, the appearance of the hand of a user performing a specific posture (as captured by the range sensor), may change due to user specific finger configuration and hand geometry and also due to variations in the 3D pose of the hand. Although, user specific posture variations produce less pronounced appearance variations, and thus a small number of users are needed to train the system this is not the case with appearance variations which are due to different hand orientation. Training the system with all possible 3D pose variations, for all postures and all persons is a cumbersome procedure. In order to cope with this problem we have investigated three alternative techniques.

The first approach, which is similar to that in [22], is based on the generation of novel views of a hand posture using a prototypical view. A set of training examples is recorded (a few images per posture per person), segmented and rectified to an upright pose (as in sections 4, 5). Then for each image in the above training set a series of novel 3D views is generated by applying a 3D rigid transformation to the input 3D data. The 6D space of rigid transformations is sampled appropriately to cover the range of possible 3D variations. Then, the enriched training set is used to construct a global eigenspace.

The second approach bypasses the first recording step by utilizing a 3D articulated

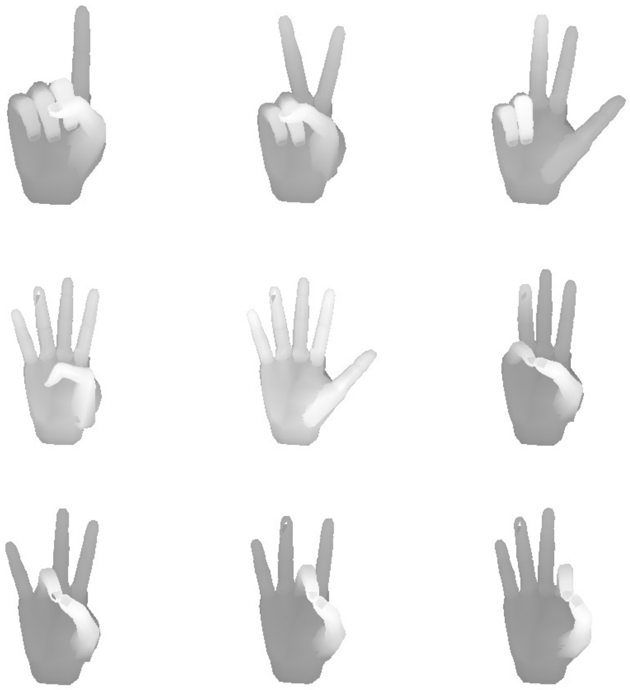


Fig. 6. Subset of the synthetic 3D hand models used for generating training images

hand model. This model is controlled by a set of parameters that correspond to the configuration and structure of the fingers. A data-glove device is used to capture the above parameters from the user in real-time. Then, small perturbations are introduced to the model parameters, in order to generate rigid and non-rigid variations of the given hand posture and simulate user specific attributes such as finger length. Then a subset of the generated models is selected, and depth images are generated for each of them by defining a virtual camera and measuring the distance of the 3D model surface from the camera plane (see fig. 6). The obvious advantages of the above training process is the efficiency in adding new postures and the direct control over the parameters of variability. Also, the training samples are free from any errors that may be introduced by the 3D sensor, such as occluded areas.

Finally, the third approach replaces the 3D pose generation step of the first approach by the 3D pose compensation algorithm described in section 5.

In the following sections an evaluation of the above techniques is described.

7 Experimental Results

A set of 20 hand postures was selected from the German sign language alphabet [23]. Ten of these postures correspond to the numbers 0 to 9, while the other ten correspond to letters. Two sets of images were recorded, one for training and one for testing.

Three volunteers participated in the recording of the training set. They were selected mainly according to the size of their hand (small, medium and large). The users were asked to rotate and translate their hands slowly. For each person and each posture about 50 images were acquired with the arm apart from the body and in horizontal orientation (over 3000 images). Both depth images and corresponding color image were acquired. The hand-forearm segmentation algorithm and the image rectification algorithms described in section 4, 5 were applied to the recorded images.

The test set was recorded similarly by employing two volunteers other than the three above. Two recording sessions were performed (we shall refer to them as Session A and Session B in the sequel) corresponding to different application scenarios. In the first session the arm of the user is vertical and apart from the body. In the second session the arm is in front of the body with the hand pointing upwards. On the average, 50 images per person, per posture, and per session have been acquired (over 4000 images). Also in this case both depth images and corresponding color image were acquired.

We used the above training and test set to evaluate the efficiency of the proposed algorithms on the basis of different classification techniques (section 6). More results, particularly in arm/hand segmentation as well as videos from the real-time operation of the system, are published in our web site [1] .

We have performed two experiments using the above data sets. In the first experiment we report posture classification results using 2D hand silhouettes only. This experiment aims at quantifying the benefit of using 3D data. In the second experiment an extensive evaluation of the 3D posture classification techniques described in section 6 is performed. The results demonstrate the superiority of the technique which is based on pose compensation.

7.1 Hand Posture Classification using 2D features

We have implemented a hand posture classification technique that is based on 2D hand silhouettes. Two hand silhouette extraction techniques were investigated. In the first technique (EFD-A in the sequel) the silhouette is obtained directly from the range images by means of contour following. In the second technique (EFD-B), we have applied color skin segmentation on the corresponding color images, instead of range images. In particular we have applied the skin segmentation algorithm proposed in [36] inside the region obtained by the hand detector in section 4. The skin-detection algorithm was trained on 1000 pre-segmented (hand only) color images in our training data set.

Once silhouette contours were extracted we compute a rotation, translation and scale invariant representation by means of Elliptic Fourier Descriptors (EFDs) [32]. The accuracy of the representation may be controlled by the number of EFD co-

[1] ”http://server-5.iti.gr/sotiris/gesture/gesture.html”

efficients used. However fewer coefficients also contribute to some robustness to noise.

Posture classification is achieved by estimating the minimum Euclidian distance between the EFDs extracted from an input image with the EFDs extracted for each image in a training set.

	Training database size					
	Session A			Session B		
EFD coef.	30%	60%	100%	30%	60%	100%
10	72	75	76	63	64	67
15	73	76	78	66	68	71
20	75	78	81	67	70	74

Table 1

Correct recognition rates for hand posture classification using 2D Elliptic Fourier Descriptors. Hand silhouettes were extracted using range images.

In table 1 the correct recognition rate achieved with the EFD-A technique is shown. The algorithm was tested for different combinations of the number of descriptors used to approximate the hand shape and the size of the database used for training (a percentage of the total database images is randomly selected). As expected, the rate increases when a more accurate approximation and a richer training set is used. The poor recognition rate of session B images is mainly due to inferior quality of the depth images (especially along the discontinuity boundary) acquired when the hand is parallel to the projected pattern stripes.

Marginally, better results were obtained when the EFB-B technique was applied

	Training database size					
	Session A			Session B		
EFD coef.	30%	60%	100%	30%	60%	100%
10	73	76	77	68	70	72
15	75	78	79	70	71	73
20	76	81	83	72	75	80

Table 2

Correct recognition rates for hand posture classification using 2D Elliptic Fourier Descriptors. Hand silhouettes were extracted using color images.

(table 2). The improvement may be attributed to the quality of hand masks obtained from the color data.

7.2 Hand Posture Classification using 3D data

In the following we report results obtained using the Principal Component Analysis algorithm on 3D data. All three alternative techniques used to cope with 3D pose variations were evaluated. In the following we shall refer to them as PCA-A (database enrichment by 3D transformation of prototype images), PCA-B (database containing synthetic postures and variations), PCA-C (3D Pose compensation).

Table 3 summarizes the results obtained using PCA-A for different combinations of the number of eigenvectors, the size of the training database (original database + variations), and the size of the training images. The variations were determined by uniform sampling the 3 degrees of freedom of hand orientation. In Var 1 the rotation around the x and y axis is sampled at -15, 0 and 15 degrees, while rotation around

	Session A					
	32×32			64×64		
eigenvectors	Var 1	Var 2	Var 3	Var 1	Var 2	Var 3
20	**77.7**	**82.2**	**88.5**	**80.2**	**84.9**	91.2
50	**80.4**	**84.8**	89.0	**82.7**	**88.3**	91.7
100	**82.5**	**85.7**	89.1	**85.5**	90.2	91.8
	Session B					
	32×32			64×64		
eigenvectors	Var 1	Var 2	Var 3	Var 1	Var 2	Var 3
20	**69.6**	**74.4**	**79.6**	**71.8**	**76.2**	82.4
50	**72.1**	**78.4**	80.3	**75.6**	**79.7**	83.0
100	**74.0**	**79.3**	80.4	**76.8**	81.2	83.0

Table 3

Correct recognition rates for hand posture classification using PCA-A approach. Number of variations introduced, Var 1 = 81 images/posture, Var 2 = 405 images/posture, Var 3 = 2025 images/posture. The cases where the achieved output frame-rate is over 5 frames per second are indicated in bold text

the z axis is uniformly sampled from -40 to 40 degrees with 10 degree steps, so giving totally $3 \times 3 \times 9 = 81$ images. Similarly, in Var-2 $5 \times 9 \times 9 = 405$ samples are used, while in Var 3 we have also introduced 5 variations (1cm) in hand translation along the X and Y axis ($5 \times 9 \times 9 \times 5 = 2025$).

As demonstrated by these results, by increasing the number of eigenvectors the recognition rate improves, reaching a threshold for large values (typically over 50). This threshold was shown to depend on the resolution of the images and size of the training database. The number of 3D pose variations has a significant effect in the classification efficiency but there is also a cut-off value; when more variations are introduced the within class variance increases, bringing in proximity posture clusters in the eigenspace. Finally, there is a weaker dependency of the recognition rate on the resolution of the images used for training. Since the computational complexity both for training and testing varies quadratically with the image resolution it is reasonable to compromise a small efficiency reduction to achieve real-time performance. Although, this approach leads generally to satisfactory results its computational complexity is relatively high. In table 3, the cases where the achieved output frame-rate is over 5 frames per second are highlighted.

The results obtained with the PCA-B approach are demonstrated in table 4. Similar observations with PCA-A results can be made regarding the effect of different parameters. The recognition rates however are about 10% worse. This is mainly a result of missing pixels in the original test images due to occlusions. In order to verify this argument we performed an experiment using synthetic test images, (with random rotation, and random finger joint perturbation), with various amounts of randomly located missing pixels. The recognition rates achieved are near 100% for no missing pixels but rapidly fall to 80% for 20% missing pixels. The same experiment was performed by artificially introducing missing pixels in the training data. Again, the accuracy drops with the amount of missing data in the test images but more gracefully this time. The problem of pattern classification with missing data has been addressed in the literature (e.g. using data imputation [29] or the EM algorithm for dealing with incomplete data [28]); however such techniques

	Session A					
	32 × 32			64 × 64		
eigenvectors	Var 1	Var 2	Var 3	Var 1	Var 2	Var 3
20	66.6	71.3	82.0	71.5	75.3	78.5
50	70.3	75.8	81.7	71.6	77.3	83.9
100	73.6	76.2	83.3	72.4	80.3	84.6
	Session B					
	32 × 32			64 × 64		
eigenvectors	Var 1	Var 2	Var 3	Var 1	Var 2	Var 3
20	63.2	64.7	68.5	61.6	70.1	73.0
50	65.9	68.1	70.3	66.9	71.6	73.8
100	67.1	70.9	70.5	67.6	72.8	74.2

Table 4

Correct recognition rates for hand posture classification using PCA-B approach. Number of variations introduced, Var 1 = 81 images/posture, Var 2 = 405 images/posture, Var 3 = 2025 images/posture

are characterized by high computational complexity, and are therefore non suitable for real-time applications. Training with missing data seems to partly alleviate the problem.

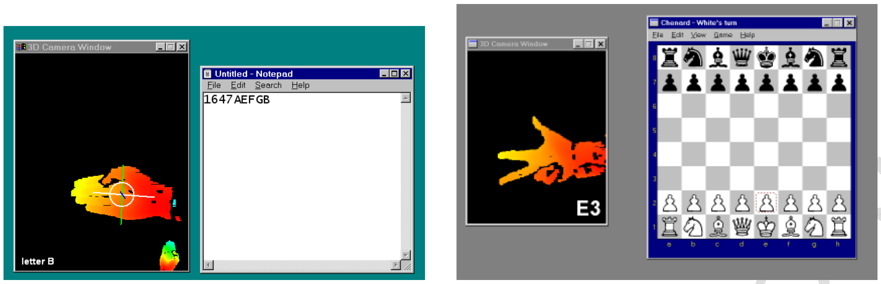
The best results were obtained using the PCA-C approach, both regarding recognition rates and computational efficiency. Since the variation of the training images is

	Session A					
	32 × 32			64 × 64		
eigenvectors	Var 1	Var 2	Var 3	Var 1	Var 2	Var 3
10	81.7	86.4	93.1	84.2	89.2	95.9
20	84.5	89.2	95.5	87.1	91.8	96.5
50	84.9	89.7	95.6	87.3	92.2	96.5
	Session B					
	32 × 32			64 × 64		
eigenvectors	Var 1	Var 2	Var 3	Var 1	Var 2	Var 3
10	73.7	77.5	84.0	75.6	80.7	86.3
20	75.9	80.6	85.3	78.6	83.1	86.7
50	76.0	81.3	85.3	79.3	83.7	87.2

Table 5

Correct recognition rates for hand posture classification using PCA-C approach. Number of variations introduced, Var 1 = 10 images/posture, Var 2 = 30 images/posture, Var 3 = 50 images/posture

now limited to finger configuration and size, an eigenspace with lower dimensionality was used. As demonstrated in table 5, increasing the number of eigenvectors over 20 does not lead to significant improvement of the recognition rates. Finally we developed a simple scheme to eliminate false-positives (i.e. non-hand images or irrelevant hand postures) by thresholding the distance measure function (8). An optimal threshold value is selected experimentally by the analysis of the Receiver Operating Characteristics curve (true positives versus false positives as a function of the threshold value) using cross-validation in a data set containing over 1000 non-hand images and training images. Using this approach a false positive rate less than 2% was achieved. Further reduction of false positives may be obtained by applying application specific syntactic/temporal constraints.



1647AEFGB

(a) (b)

Fig. 7. Screenshots of demo human-computer interface applications using the proposed 3D posture recognition system. (a) Keyboard-less typing, (b) Keyboard-less control interface to a chess game.

To demonstrate the proposed posture recognition system a prototype application was developed and evaluated. The system (PCA-C approach) was implemented on a PC Platform with a Pentium III 1 Ghz processor. Operating on subsampled images (180×144), without any optimization, a frame rate of 15 frames/sec was achieved. The application (figure 7a) provides the user with visual feedback by displaying in real-time the acquired range image, a textual description of the classification result as well as an icon depicting a synthetic image of the recognized posture. The alpha-numeric character corresponding to the recognized posture is passed to the underlying application window e.g. word-processing application, acting as a keyboard-less typing interface. A more impressive application scenario was also implemented (figure 7b), a keyboard-less command interface to a chess game. To move a piece the user has to perform four postures. A letter posture followed by a number posture e.g. ”E2” are performed to select a piece and another pair of letter, number postures e.g. ”E3” are performed to specify the new position of the piece. In this case a simple syntax checking strategy is utilized to reject false positives. The two demo applications were demonstrated in public exhibitions and the comments by the users were very positive, especially regarding the response of the system.

8 Conclusions and Future Work

We have demonstrated a complete system for the recognition of static hand postures based on a 3D sensor. The system relies only on range data, therefore is invariant to the content and illumination of the scene. This makes it suitable for operation in unconstrained environments. Also, it is tolerant to the 3D pose of the hand by including a pose compensation procedure. The classification of hand postures is achieved by representing the range images by a discriminative feature vector that incorporates 3D shape information. Experimental results demonstrate the efficiency and robustness of the system and the advantages of using 3D information instead of 2D silhouettes to obtain discriminative feature vectors. A sub-space classification technique with a 3D pose compensation stage was found to be the most appropriate both regarding accuracy in posture recognition as well as computational efficiency.

The utility of the proposed system lies mainly in the enhancement of human-computer interaction in a wide variety of applications. It is particularly useful in applications where tactile and/or verbal interaction is difficult or impossible, e.g. in medical operations, industrial and hazardous working environments, natural interaction with virtual displays in outdoor places etc. In addition if the system is used in conjunction with a dynamic gesture recognition system and other interaction modalities such as speech, the prospective applications are unlimited.

This paper presents a valuable first step towards real-time gesture-based interaction but there are several directions in which this work could be extended. One of them is the incorporation of temporal constraints by means of gesture recognition. Static posture and dynamic gesture recognition are commonly studied separately. Nevertheless, posture recognition may help in identifying the boundaries of individual gestures in a sequence. Also, given the trajectory of the moving hand the conditional distribution of a matching gesture and associated posture may be estimated. Arguably, future work should focus in a joint investigation of the two tasks. This will be valuable for challenging applications such as sign-language recognition. Another research direction is in 3D non-rigid hand tracking. This is a very challenging problem given the large number of degrees of freedom of the hand and the ambiguity in 3D reconstruction due to occlusions. The target application domain is enhanced virtual environment interaction (20+ degrees of freedom mouse) especially in desktop applications.

9 Acknowledgement

This work has been supported by the EU project HISCORE "High Speed 3D and Colour Interface to the Real World" (IST-1999-10087).

References

[1] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *Conference on Computer Vision and Pattern Recognition (CVPR 03)*, volume 1, Madison, Wisconsin, 2003.

[2] P. H. S. Torr B. Stenger, A. Thayananthan and R. Cipolla. Hand pose estimation using hierarchical detection. In *Intl. Workshop on Human-Computer Interaction*, 2004.

[3] M. Bray, E. Koller-Meier, P. Mueller, L. Van Gool, and N. N. Schraudolph. 3d hand tracking by rapid stochastic gradient descent using a skinning model. In *1st European Conference on Visual Media Production (CVMP)*, pages 59–68, March 2004.

[4] R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Comp. Vision, and Image Understanding*, 81:166–210, 2001.

[5] C.-S. Chua, H. Guan, and Y.-K. Ho. Model-based 3d hand posture estimation from a single 2d image. *Image and Vision Computing*, (20):191–202, 2002.

[6] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25:63–85, 1997.

[7] B. Deimel and S. Schroeter. *Improving Hand-Gesture Recognition via Video-Based Methods for the Separation of the Forearm from the Human Hand*. Technical Report Nr. 691/1998, University of Dortmund, 1998.

[8] Q. Delamarre and O. Faugeras. Finding pose of hand in video images: a stereo-based approach. In *Proceedings. Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 585–590, April 1998.

[9] H. Du and E. Charbon. 3d hand model fitting for virtual keyboard system. In *IEEE Workshop on Applications of Computer Vision*, pages 31–37, 2007.

[10] R. O. Duda, E. Hart, and D. G. Stock. *Pattern Classification*. Wiley, 2001.

[11] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. A review on vision-based full dof hand motion estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 3, pages 75–82, 2005.

[12] O. Faugeras. *Three Dimensional Computer Vision*. M.I.T. Press, Cambridge, MA, 1993.

[13] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1990.

[14] R. Grzeszcuk, G. Bradski, M. H. Chu, and J. Y. Bouguet. Stereo based gesture recognition invariant to 3d pose and lighting. In *Proceedings. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 826–833, 2000.

[15] H. Guan, R. S. Feris, and M. Turk. The isometric self-organizing map for 3d hand pose estimation. In *Automatic Face and Gesture Recognition Conference (FGR2006)*, volume 10.

[16] C. Jennings. Robust finger tracking with multiple cameras. In *Proc. Intl. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 152–160, Corfu, September 1999.

[17] A. E. Johnson and M. Hebert. Surface matching for object recognition in complex three-dimensional scenes. *International Journal of Computer Vision*, 16:635–651, 1998.

[18] N. Jojic, B. Brumitt, B. Meyers, S. Harris, and T. Huang. Detection and estimation of pointing gestures in dense disparity maps. In *Proceedings. Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 468–475, 2000.

[19] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[20] X. Liu and K. Fujimura. Hand gesture recognition using depth data. In *IEEE International Conference on Automatic Face and Gesture Recognition*, number 17-19.

[21] S. Malassiotis and M. G. Strintzis. Tracking the left ventricle in echocardiographic images by learning heart dynamics. *IEEE Trans. Med. Imaging*, 18(3):282–291, March 1999.

[22] H. Murase and S. K. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[23] Institute of German Sign Language and Hamburg University Communication of the Deaf. Alphabet sign language. http://www.sign-lang.uni-hamburg.de/fa/.

[24] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, 1979.

[25] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 19(7):677–695, July 1997.

[26] J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Intl Conf. on Computer Vision*, June 1995.

[27] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

[28] Sam Roweis. EM algorithms for PCA and SPCA. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 626–632. The MIT Press, 1998.

[29] D. Skocaj and A. Leonardis. Robust recognition and pose determination of 3-d objects using range images in eigenspace approach. In *Third International Conference on 3-D Digital Imaging and Modeling*, 2001.

[30] T. Starner and A. Pentland. Real-time americal sign language recognition using desk and wearable computer based video. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 20(12):1371–1375, December 1998.

[31] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Estimating 3d hand pose using hierarchical multi-label classification. *Image and Vision Computing*, 2007. to appear.

[32] O. D. Trier, A. K. Jain, and T. Taxt. Feature extraction methods for character recognition: A survey. *Pattern Recognition*, 29(4):641–662, 1996.

[33] F. Tsalakanidou, F. Forster, S. Malassiotis, and M.G. Strintzis. Real-time acquisition of depth and color images using structured light and its application to 3d face recognition. *Real-Time Imaging, Special Issue on Multi-Dimensional Image Processing*, 11(5-6):358–369, December 2005.

[34] K. Umeda, Y. Furusawa, and S. Tanaka. Recognition of hand gestures using range images. In *Proceeding of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, volume 3, pages 1727–1732, Victoria, Canada, October 1998.

[35] A. Utsumi and J. Ohya. Multiple hand gesture tracking using multiple cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 473–478, 1999.

[36] X. Yin and M. Xie. Finger identification and hand posture recognition for humanrobot interaction. *Image and Vision Computing*, (25):12911300, 2007.

[37] H. Zhou, D. J. Lin, and T. S. Huang. Static hand gesture recognition based on local orientation histogram feature distribution model. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, volume 10, pages 161–165, 2004.

Answer to Reviewer I

We wish to thank the referee for a detailed and constructive review. All of his/her comments were addressed. In particular:

1. *The introduction can be finished with a small paragraph about the structure of the rest of the paper..*

   We have introduced a paragraph outlining the paper as suggested.

2. *In tbl2 caption, please make clear what is bold, it is written in the text in p22*

   The explanation was inserted also in the table caption.

3. *On p8 the paper discusses a segmentation algorithm based on clustering. The authors mention the limitations of K-means and EM-like clustering, but does not consider agglomerative (hierarchical) which based on the algorithm they present would be a reasonable choice.*

   In fact the proposed algorithm may be consider very similar to agglomerative clustering. We have noted that in the revised manuscript.

4. *On p19, "The test set was recorded similarly by employing two volunteers in addition to the three above". This means that in most of the cases (3 people out of 5), the test was performed by the same person as the training. Does the recognition framework require or perform better when it is trained for the same hand? If yes, it should be more explicitly stated. This kind of mixed test set does not help to understand that.*

   What we meant was "two volunteers, other from the three above", so the test set is not a subset of the training set. Indeed, results may be much better if we train the algorithms with images of the user. We have clarified this in the revised manuscript.

5. *Experiments on noisy data. Recognition with randomly located noisy pixels (created synthetically) are better when the training is performed on similar type of noisy images. So this suggest that the framework more robust to noise when it is trained on noisy data. Shall it be always trained on noisy data? Will it still recognize non-noisy images, or it just learned the data with the noise? These should be discussed.*

We assume that the reviewer is referring to PCA-A and PCA-B experiments where the system is trained with rotated original (noisy) images and synthesized images respectively. Note that, these techniques will not have a trouble with noise in general (e.g. additive Gaussian noise) but instead with missing pixels. This mainly a problem with the PCA algorithm (and other template-based techniques) which cannot handle missing pixels well, and especially reconstruct the sharp discontinuity at the boundary. The problem was investigated in detail by Leonardis et. al and modified algorithms have been proposed which are however computationally expensive. The problem is alleviated as demonstrated by our experiments when training is with images containing missing pixels. The experiments showed that by increasing the amount of missing pixels the accuracy drops, both when training is performed with noisy and clean data. This was clarified in the revised manuscript.

6. *p23-24: "An optimal threshold value is selected experimentally by the analysis of the ROC curve ..." This suggest that the threshold was set on the test set. It should be set on the training set by, e.g., cross-validation.*

   Indeed, the threshold was determined by cross-validation on a data set consisting of the training hand images and 1000 non-hand images. This is clarified in the revised manuscript.

7. *On the first page of the paper it is indicated that this paper has been submitted on 27 January 2003. The latest references are from 2001 in the document. During the last 5 years there are a few papers which could be mentioned regarding to gesture/sign language recognition, as well as techniques related to depth-map (e.g., stereo sensors). Even if the paper has been submitted for so long, I would suggest the authors to refresh the state-of-the-art review, which would consequently improve the quality of the text before the final version. (Note: the future work also includes directions of research, such as sign-language recognition which are investigated in the last few years.*

   The literature was updated with many recent research results.

8. *On p13, first paragraph: "The problem is usually addressed by: ... (a) ... (b) .. (c) ...". Please give example references to these.*

   This was a quite generic statement and it would not be appropriate to put specific references, so we chosen to remove it in the revised manuscript.

9. *My major concern about the paper, is the missing comparison to state-of-the-art. I understand that these comparisons cannot be made by downloading a database, because of the sensing methods, however, a baseline technique can be evaluated on a similar dataset. Even if the baseline is very simple, the reader would get a feeling that how the proposed framework performs. The results and the example segmentations look very good and promising, but numerical comparison is necessary. The sentence from the abstract: "The main advantage of the proposed method, compared to other gesture recognition techniques, is ..." is also asking for comparison.*

   Techniques using skin-color information is a popular and a successful in hand recognition (tracking, etc). While the authors correctly acknowledge that on p7, they argue that depth information is much more useful. I would expect that the combination of the two leads even better performance, yet if the authors drop color information because of limitations (that is what I understood from the paragraph on p7), I think a comparison experiment in a less-constrained environment would be very nice to see.

   We have addressed the concern of the reviewer by performing an additional experiment using skin-color segmentation. We had fortunately recorded along with depth images the corresponding color images and therefore we could run skin-color segmentation on this images thus making the results of this experiment comparable with those obtained from 3D data. Classification was based on EFD features.

Answer to Reviewer II

We thank the reviewer for the very careful review, and his constructive and helpful comments. We have reorganized and rewritten several paragraphs and we believe that the revised manuscript is now easier to follow. In particular:

1. *Section 1.*

   We have introduced an outline of the paper, giving also an idea of the building blocks of the system.

2. *Section 2.*

   This section has been rewritten and new example images were introduced. We have briefly described the working principle, and clearly described the use of color striped images to compute depth information but also the possibility for synchronous acquisition of associated color images. We believe that now there should be no confusion regarding the dual use of color images. More recent publications describing the 3D sensor have been cited. These are also available online (http://server-5.iti.gr/sotiris/publications.html, see [19]) if the reviewer wishes to see further details regarding the sensor.

3. *Section 5: 3D Pose Estimation and Compensation*

   The section was rewritten to make it more readable addressing the comments of the reviewer. Maybe part of the confusion is due to the interchangeable use of 3D data and depth images. This is now clarified in section 2.

4. *"Section 6 is very confusing. Two feature-based techniques are introduced. These essentially attempt classification based on the surfaces represented by the depth images. The authors seem to have tried these as they state on page 15 that they led to poor performance when applied to their noisy data. What recognition rates were achieved?"*

   We do have results on these techniques but are not directly comparable to those presented in these paper. So we have removed the corresponding sentences and just retained a general remark about the sensitivity of the algorithms to noise and computational complexity.

5. *"Then the manuscript explains a 2D silhouette-based method employing EFSs. These 2D silhouettes are extracted from the 3D data. This discussion is immediately followed by the introduction of appearance-based eigenspace techniques*

that are referred to as PCA. For quite a while, I was unaware that the work had shifted back to 3D. Through several readings, I believed that the PCA approaches were applied to 2D images."

We have moved the description of the EFD 2D classification algorithm to the next section to avoid any confusion. 2D classification experiments were used to get a feeling of the benefit achieved by using 3D information.

6. *"Also included in this section is a discussion of three "approaches" (page 17) whose purposes are not made clear. Once again it took a great deal of effort to understand that these are approached to generating enhanced training sets. "*

   All three techniques aim to cope with the problem of pose variability. This is now clearly explained in page 19 of the revised manuscript.

7. *"Section 7 is entitled experimental results, but also attempts to explain the experimental methods. I feel that these methods are as confusing as the earlier sections and need to be rewritten so the preparation of the training and test sets are clear. There needs to be a clearer explanation that the experiments used 3D data, with the PCA methods that involved a variable number of eigenvectors, and three sets of data against which the test data were compared, which were produced using the three approaches described on page 17 and called PCA=A, PCA-B and PCA-C. But, as I understand it, there is only one PCA method, with the A, B and C referring to the way in which the training data are enhanced. Then we have Var A, Var B and Var C. The reuse of A, B and C adds confusion."*

   The section was reorganized and several paragraphs rewritten to address the concerns of the reviewers. We start by the description of the training and test set used. We continue with results on 2D posture classification and then with results on 3D posture classification. We have also changed Var A, Var B, and Var C with Var 1, Var 2, Var 3 as suggested by the reviewer.

8. *"I am not at all clear why Var A=81 images per posture, Var B=405 images per posture, and Var C = 2045 images/posture. How were these numbers determined?"*

   This is now explained in page 23 (last paragraph) of the revised manuscript.

9. *"Why are resolutions of 32X32 and 64X64 used? Why not 50X50 or 100X100 for example? Is there reason?"*

The selection is arbitrary. We only intended to examine the effect of training image size to classification accuracy and computational complexity. We clarify the issue of rectified image size in section 5 of the revised manuscript.

10. *"Why are there 10, 20 and 50 eigenvectors used with PCA-B and PCA-C and 20, 50 and 100 eigenvectors used with PCA-A?"*

    This is due to the fact that there are less variations in the training set which makes a lower dimensionality sufficient. This is explained in page 27 (first paragraph).