Journal of Ambient Intelligence and Smart Environments 6 (2014) 93–105
DOI 10.3233/AIS-130239
IOS Press

Hand shape classification using depth data for unconstrained 3D interaction

Luigi Gallo
Institute for High Performance Computing and Networking, National Research Council of Italy (ICAR-CNR), Via Pietro Castellino 111, 80131, Naples, Italy
E-mail: luigi.gallo@cnr.it

Abstract. In this paper, we introduce a novel method for view-independent hand pose recognition from depth data. The proposed approach, which does not rely on color information, provides an estimation of the shape and orientation of the user's hand without constraining him/her to maintain a fixed position in the 3D space. We use principal component analysis to estimate the hand orientation in space, Flusser moment invariants as image features and two SVM-RBF classifiers for visual recognition. Moreover, we describe a novel weighting method that takes advantage of the orientation and velocity of the user's hand to assign a score to each hand shape hypothesis. The complete processing chain is described and evaluated in terms of real-time performance and classification accuracy. As a case study, it has also been integrated into a touchless interface for 3D medical visualization, which allows users to manipulate 3D anatomical parts with up to six degrees of freedom. Furthermore, the paper discusses the results of a user study aimed at assessing if using hand velocity as an indicator of the user's intentionality in changing hand posture results in an overall gain in the classification accuracy. The experimental results show that, especially in the presence of out-of-plane rotations of the hand, the introduction of the velocity-based weighting method produces a significant increase in the pose recognition accuracy.

Keywords: Hand shape classification, static hand pose recognition, touchless interface, Kinect, medical visualization

1. Introduction

The design of effective touchless interfaces for interacting in smart spaces by using depth sensing technologies is now rapidly becoming a major challenge in human-computer interaction [1]. Recently, several natural user interfaces (NUIs) that allow users to perform omnifarious interaction tasks by means of body movements have been proposed, and their efficiency in solving complex problems has been evaluated in many real-world applications.

Nonetheless, 3D user interfaces able to fully exploit depth sensing technologies are still in their infancy. Interacting with 3D worlds is more complex than with 2D WIMP (window, icon, menu, pointing device) interfaces, since it requires the user to manipulate the position and orientation of 3D objects involving six degrees of freedom (DOF) [2]. Moreover, a truly 3D interaction requires the user not to be constrained to maintain a fixed position in the 3D space. On the contrary, most of the Kinect NUIs proposed so far are able to control only 2 DOF simultaneously, and correctly recognize hand and arm gestures only if the user stands fronto-parallel to the sensor and at a fixed distance from it. In fact, most systems are able to recognize gestures only if users are standing in a fixed place with hands extended [3]. At the heart of these difficulties there is the problem of recognizing static and dynamic hand gestures from low resolution depth images of a hand at different distances and differently oriented.

As an example, Fig. 1 depicts the sequence of actions required to perform a 3D rotation when static and dynamic hand postures are used to control the execution. First, the user chooses the grasping position while assuming the open hand posture (see Fig. 1a) and then grabs the 3D object by assuming the grasp position (closed fist, see Fig. 1b); then, while keeping this hand posture, he/she moves his/her arm to rotate it (see

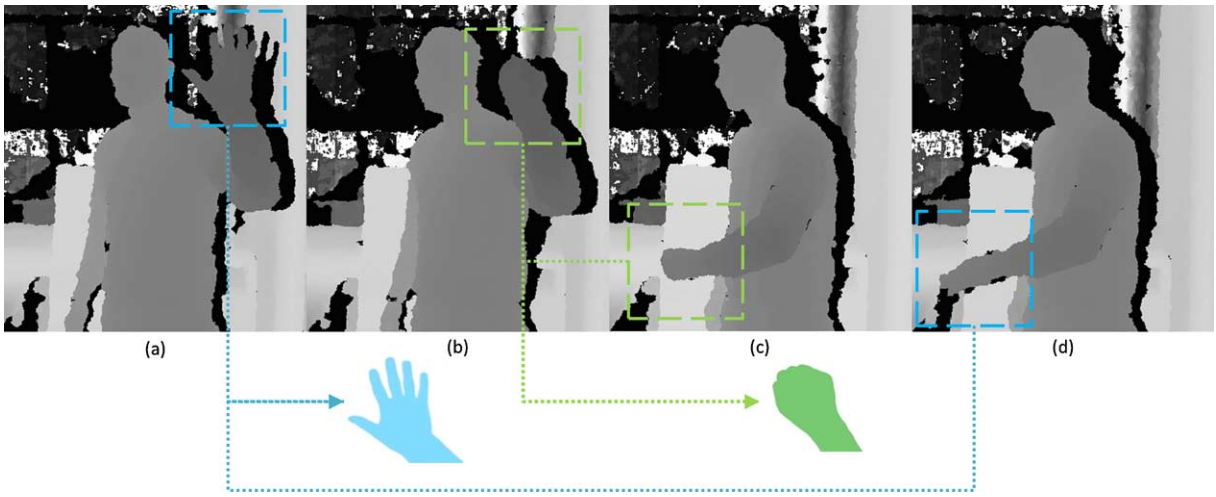


Fig. 1. Depth images of a 3D rotation sequence: (a) positioning; (b) grabbing; (c) rotating; (d) releasing.

Fig. 1c); finally, he/she releases the object by assuming the open hand posture (see Fig. 1d). Therefore, to perform this interaction task in a touchless way, the system should be able to track the user's hands, but also to correctly classify static hand postures (open hand and closed fist) independently from the hand position and orientation. In fact, even in the case of the user standing fronto-parallel to the sensor, a movement of the arm in the 3D space changes the shape of the hand as acquired by the sensor, making its classification problematic. Although voice input could be used as an alternative method to engage/disengage with the system, it would not work in noisy environments or when simultaneous interactions by more than one user are required, and could decrease the precision of the interaction since the user has to remain motionless until the speech recognition result is available.

These observations motivated us to investigate view-independent hand pose recognition approaches from a single, low resolution depth image. The majority of studies on this subject in recent literature approach view-independent static hand gesture recognition by using multiple sensors. However, most of these approaches require complex calibration steps, so reducing the portability of the system, and force the user to keep his/her hand in a small area of the 3D space. In contrast, our work has been focused on the design of a hand shape classification approach that allows users to move freely in the coverage area of the sensor.

This paper significantly extends the approach proposed in [4], in which a method for extracting and processing region-based statistical image features and a database containing pose compensated depth images of hands were presented. In more detail, the main contributions of this work are: i) a novel weighting method, which takes advantage of the velocity and orientation of the user's hand with respect to the sensor to improve the hand shape classification accuracy; and, ii) a user study, the results of which show that the velocity-based weighting approach increases the accuracy of the hand shape classification while performing 3D interaction tasks. As a case study, we have also implemented the proposed algorithms in an open-source software for 3D medical visualization, designing a touchless interface suitable for use in operating rooms or in medical education.

The remaining part of the paper is organized as follows: Section 2 describes the previously proposed methods for hand shape classification, and discusses the limitations of these methods when used for 3D interaction; Section 3 presents an overview of the hand shape classification chain, particularly focusing on the weighting method; Section 4 describes the main features and some implementation details of the touchless user interface for 3D medical visualization; Section 5 presents our experimental results both on the accuracy of the classifiers and on the impact of the velocity-based weighting method on hand pose recognition; finally, Section 6 concludes the paper.

2. Related work

In recent years, many largely different approaches have been proposed to recognize static hand postures. A review of the techniques and methods most com-

monly used can be found in [5]. In this section, we briefly introduce the different approaches, specifically focusing on view independent hand shape classification from a single image.

A common approach to static hand pose recognition relies on the use of a 3D hand model. Since the hand is an articulated deformable object, the complexity of such a model is high. The self-occlusion of the hand and the high computational cost are the main difficulties model-based approaches have to face. Moreover, to allow the recognition of a wide range of hand shapes, large image databases are required. It is worth noting that, if only a single frame is available, then the problem does not have a unique solution.

In contrast, appearance based approaches aim the recognition of hand postures by comparing the 2D image features extracted from the hand pose database with the one extracted in real-time from the video stream. The choice of the image features to use further contributes to classify the approach. Local invariant features, eigen values, or the whole image can be profitably used as image features. Among low level features, Fourier descriptors [6] and statistical moments (e.g. Hu moments [7], Flusser moments [8]) are the most commonly used. Approaches based on high level features such as fingers and fingertips have also been proposed [9]. Additionally, in many of such approaches, markers or instrumented data gloves [10] are used to provide a reliable recognition even in the presence of cluttered backgrounds. Recently, some markerless methods have also been proposed [11].

Although they are more efficient in computation time, appearance-based methods are view dependent, that is, the hand pose can be correctly recognized only if the camera is close to and in a fronto-parallel view with respect to the user. In contrast, in our work the goal has been to classify hand shapes to provide users with touchless interaction mechanisms in an unconstrained 3D space. In this context, the main challenges to cope with are the absence of constraints on the placement and orientation of the hand with respect to the body as well as limitations on the camera background.

Some vision-based view independent hand pose recognition systems have been proposed in the recent literature. In [12], a multi-angle hand gesture recognition system that makes use of three Support Vector Machine (SVM) classifiers is described. However, the proposed system requires three webcams, set respectively at the front of and to the left and right of the hand to properly work, so forcing the user to stay in a fixed position in the space. In [13], a neural network-based method for recognizing the position and posture of a user's hand in real time is described. This solution also requires multiple cameras to determine the position and orientation of the user's hand, and therefore the user must keep his/her hand in a restricted area of the 3D space. However, vision-based interfaces need to contend with problems related to lighting, cluttered background, distance of operation, etc. Moreover, problems arise when controls with a high degree of freedom are required. When camera input is used, 3D hand pose estimation is an ill-posed problem, since the 3D information of a hand is lost in a 2D image.

Considering the aforementioned limitations, the use of a single 3D sensor for hand pose recognition is an attractive alternative. Since it does not rely on color information, it has the capability of recognizing complex hand postures within unconstrained environments. Therefore, more recently the research community has been investigating depth cameras as an option to approach hand pose and motion recognition.

In [14], a Principal Component Analysis (PCA) based hand posture recognition system that makes use of single depth images is described. Similarly to this approach, in our work we use region-based statistical moments as image features and classification algorithms to improve the recognition accuracy when the user is far from the camera and the hand is not in front of and parallel to it. A static and dynamic hand shape classification that uses depth and translation invariant features extracted from a depth image and randomized classification forests is proposed in [15,16]. In [17], a clutter-tolerant shape and 3D pose estimation that works on depth data is proposed. Such a method is user-independent, and provides an estimation for the 3D pose orientation and for the full hand articulation parameters.

Differently from the aforementioned approaches, in this paper we focus on the view-independence of the hand shape classification from depth data. We present a complete view-independent static hand pose recognition chain, and introduce a novel weighting method that takes advantage of the user's hand orientation and velocity to increase the hand classification scheme.

3. The hand shape classification method

The proposed classification method provides an estimation of the hand shape by using a single depth image. In Section 3.1, we describe the Kinect depth

data, the pose compensation algorithm, the image features used and the classification strategy. Then, in Section 3.2, we describe how we use two classifiers, and orientation and velocity of the user's hand to improve the recognition accuracy.

3.1. Hand shape hypothesis

3.1.1. The depth sensor

The depth sensor we have used is the Microsoft Xbox Kinect™. The Kinect is equipped with a monochrome CMOS sensor and a laser-based IR projector. The latter is used to send a fixed speckle pattern towards the focused area. This pattern is then detected by the CMOS sensor and further used to calculate depth data by means of a triangulation against a hard-wired pattern. The aforementioned sensor also embeds a tilt motor for sensor adjustment, an RGB camera and a microphone array. The device features a $43°$ vertical field of view, a horizontal field of view of $57°$ and an operating distance range between 0.8 m and 3.5 m, with a spatial resolution of 3 mm for the plane on which the Kinect camera resides, and 10 mm for the axis orthogonal to this plane directed towards the user, within 2 m from the sensor. The resolution for the produced datastreams is 640×480 at 30 Hz, whereas the depth data have an 11 bit resolution.

A statistical analysis of the sensor precision dependence on distance is given in [18]. The reported results show that the relation between the distance of the target from the depth camera and the range/standard deviation of the measured depth values fits to a quadratic function. Therefore, as described in [19], since each depth image has a fixed resolution, the depth point density is inversely proportional to the square distance from the sensor, along the perpendicular camera axis. Moreover, the disparity image allows for 1024 levels of disparity. To capture the essence of this strict dependence of depth errors to the distance from the sensor, we divided the area focused by the sensor into three regions: the first region, with the highest depth accuracy, spans from 0.8 m to 1.2 m (depth resolution < 3 mm), the second spans from 1.2 m to 2.0 m (depth resolution ≤ 10 mm) while the third extends from the end of the second region to the limit of the operating distance range.

3.1.2. Hand segmentation and pose compensation

On each depth image provided by the Kinect, user data are segmented from the background by exploiting depth cues. A nearest neighbor filter is then applied to the hand position reported by the Kinect SDK skeleton tracking module in order to extract the point data concerning the relevant hand, thus producing a smaller 3D point cloud containing hand data only. Then, a weighting scheme is applied to account for the finger positions: the greater the distances of the hand points from the center of mass, the fewer are the corresponding applied weights. Finally, as performed in [14] by Malassiotis et al., principal component analysis is employed to estimate the hand orientation, by computing the eigenvectors of the hand scatter matrix.

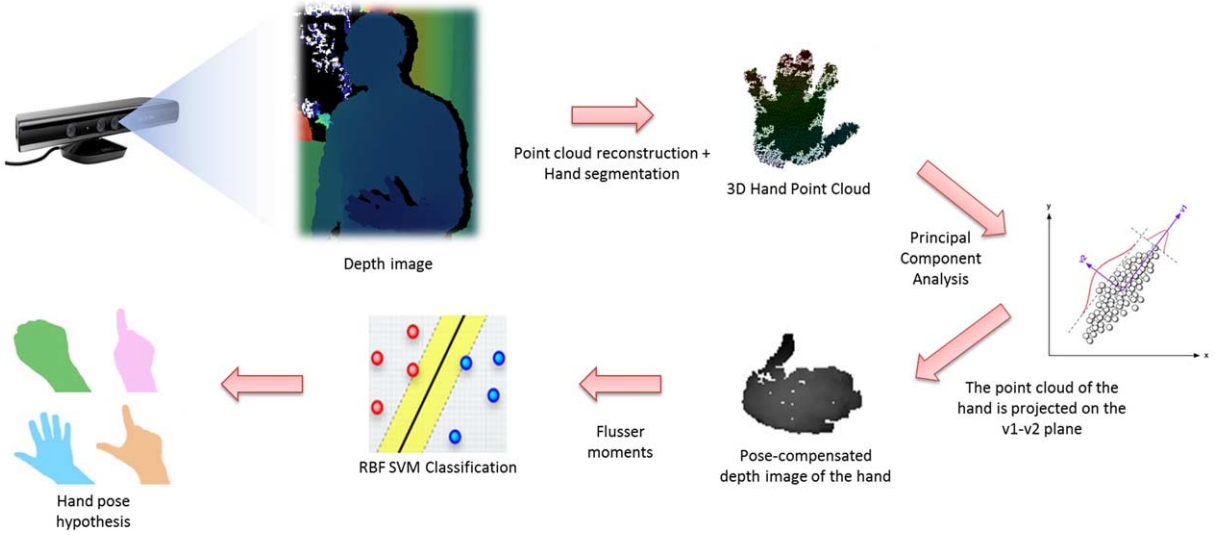


Fig. 2. Hand data capture and classification pipeline.

After the computation of the eigenvectors and the eigenvalues has been completed, the hand normalization is performed by transforming the hand point cloud data in such a way that the extracted principal directions are aligned to the frame of the depth sensor. This is achieved by means of the rigid transformation:

$$p_{normalized} = R^T p_{initial} + c - R^T m$$

where R^T is the transpose of the matrix holding the ordered eigenvectors as columns, $R = [e_1, e_2, e_3]$, with e_1 being the eigenvector corresponding to the biggest eigenvalue; c is the distance from the camera frame, and m is the center of mass of the initial hand point cloud.

Our hypothesis was that performing the hand shape classification on "pose-compensated" depth images would provide a means to recognize the hand shape even in the presence of the severe self-occlusions that are present especially during out-of-plane hand rotations.

Since no hand pose dataset was publicly available, to train and test the hand shape classifiers we recorded a database of greyscale pose-compensated depth images of four different hand postures: open hand, closed fist, L shape and finger pointing. Six subjects were required to perform two out-of-plane rotations (yaw and pitch, in the range of around $\pm 200^\circ$ and $\pm 120^\circ$, respectively) and one in-plane rotation (roll, in the range of around $\pm 240^\circ$). Due to the depth data precision dependence on distance, each subject had to repeat the acquisition process two times, with the first batch of acquisitions in the most accurate region near the sensor and the second batch of acquisitions in the second region, farther from the sensor ($\simeq$1.0 m and $\simeq$1.6 m, respectively). The sample acquisition rate was 30 fps.

Along with pose compensated depth images, the database also holds a registry containing, for each sampled frame, the distance of the tracked hand and the angle between the normal of the hand plane and the direction running from that hand and the depth sensor. Further information on the hand posture database is reported in [4].

3.1.3. Feature extraction and processing

In order to characterize the pose-compensated hand posture images, we have chosen to use moment invariants, a particular type of region-based statistical feature derived by Hu [7], as image features. However, the hand posture recognition performance of the Fourier descriptors (FD) and moment invariants was compared in [20] and, in all the experiments, FD resulted in a higher classification accuracy. The reason for our choice is that we found that countour-based features such as FD or orientation histograms, differently from region-based features such as Hu invariants and Zernike moments, require clear images to provide us with a significant discriminative power. On the contrary, depth images coming from a Kinect, particularly if taken farther than 1 m from the camera, contain artifacts and missing points within those areas that are not reached by the projected light.

Hu's seven moment invariants are defined as 2D object $f(x, y)$ descriptors invariant to translation, rotation and scale transformations. However, as reported by Flusser in [21], Hu's system of moment invariants is dependent and incomplete. Therefore, a new set of independent and complete invariants, namely Flusser moment invariants, was proposed:

$$\begin{array}{ll} \psi_1 = c_{11} & \psi_2 = c_{21}c_{12} \\ \psi_3 = Re(c_{20}c_{12}^2) & \psi_4 = Im(c_{20}c_{12}^2) \\ \psi_5 = Re(c_{30}c_{12}^3) & \psi_6 = Im(c_{30}c_{12}^3) \end{array}$$

With c_{pq} being the complex moment of order $p + q$ of an integrable image function $f(x, y)$, defined as:

$$c_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x + iy)^p (x - iy)^q f(x, y) dx dy$$

In our work we used only the five *true* invariants defined by Flusser: ψ_1, ψ_2, ψ_3, ψ_4 and ψ_5. We did not consider ψ_6 since it is a *skew* invariant, that is, it distinguishes between the mirrored images of the same object. This is undesirable in our case since the pose normalization step can produce images reflected across the x- or y-axis.

To mitigate the problem of outliers among the computed moments, a percentile filter was applied to the values with a double pass algorithm. During the first pass of the algorithm, each sample in the posture database was analyzed and the values of two percentiles were computed for each feature. During the second pass, if any of the values for the moment invariants of the current sample were lower than the lowest percentile or greater than the highest percentile, then the whole sample was discarded from the set used to train the classifier. The percentile values were chosen based on observation. They were different for each fea-

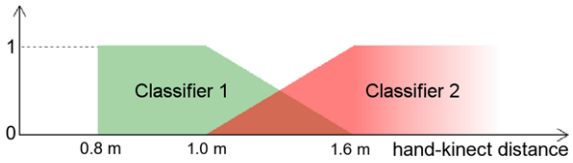


Fig. 3. Calculus of the classifier score.

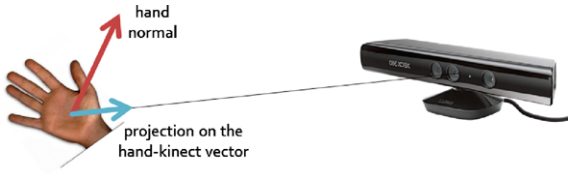


Fig. 4. Calculus of the accuracy score.

ture and range from 0th to 12th for the lowest percentile and from 85th to 92th for the highest percentile. The computed percentiles were saved and used for online hand posture recognition tasks.

To classify hand shapes on single frames by using Flusser moments, we trained two SVM classifiers on depth images obtained at different distances, with a Radial Basis Function (RBF) kernel due to the non-linearity of the Flusser moment values. All the required parameters were set to default values, except for the cost (C) and gamma (γ) parameters. To find such values a grid-search approach was adopted: as described in [22], many (C, γ) value pairs, generated through an exponentially growing sequence, were tested and the pair which produced the best cross-validation accuracy was picked. To handle multi-class classification, we used the One-Against-One (1A1) strategy. The hand shape classification rates are reported in Section 5.1.

3.2. Hand posture filtering

To improve the hand shape classification rate considering a single depth frame and to reduce its dependance on the hand-sensor distance, we devised a weighting method that takes advantage of the velocity and orientation of the user's hand with respect to the sensor. The hand posture filtering approach consists in assigning a score to each hand shape hypothesis, by computing and then multiplying three factors, each varying in the range $[0, 1]$:

$$shapeScore = classScore \times accScore \times velScore$$

The *classScore* takes into account that two classifiers are used at the same time to recognize the hand shape, each trained on depth images obtained at different distances and with a degree of certainty that varies according to the hand-sensor distance.

$$classScore = |\alpha \cdot score_1 \pm (1 - \alpha) \cdot score_2|$$

where factors are summed or subtracted if the shape hypothesis provided by the two classifiers is the same or not, respectively, while α is defined as:

$$\alpha = \begin{cases} 1 & \text{if } d \leq 1 \\ \frac{1.6-d}{0.6} & \text{if } 1 < d < 1.6 \\ 0 & \text{if } d \geq 1.6 \end{cases}$$

In more detail, as depicted in Fig. 3, if the hand-sensor distance is inside the interval $]1.0$ m$, 1.6$ m$[$, the scores provided by the classifiers are merged into a single score. Specifically, a different weight is assigned to each score (α and $1-\alpha$, respectively) according to the actual hand-sensor distance; then, if both the classifiers give the same hand shape hypothesis, the two scores are added; if not, they are subtracted. If the hand-sensor distance is lower than 1.0 m, then only the first score is used; whereas, if such a distance is higher than 2.0 m, only the second score is used.

The *accScore* is higher when the hand palm is orthogonal to the hand-kinect direction. In fact, the less the hand palm is orthogonal to that direction, the more the sensor view of the hand is occluded, and so the hand shape hypothesis is uncertain. Therefore, the hand orientation with respect to the sensor is here considered as an indicator of the degree of certainty of the hand shape hypothesis. To assign the score, we normalize the projection of the hand normal on the hand-kinect vector as depicted in Fig. 4.

The *velScore* is a score aimed at integrating the voluntariness of the user in changing the hand shape into the recognition chain. The use of the hand velocity as a voluntariness indicator [23] derives from Fitts' law, which formalized an intuitive trade-off in aimed human movements: the faster we move, the less precise our movements are. In this context, we consider unlikely an intentional change in the user's hand pose while he/she is performing a fast movement. Since the faster we move the less precise our movements are, we speculate that the probability of a hand pose transition in a frame is inversely proportional to the hand velocity in that frame. Therefore, we compute a score which is highest when the user's hand movements are slower than 0.1 m/s, and null when they are faster than

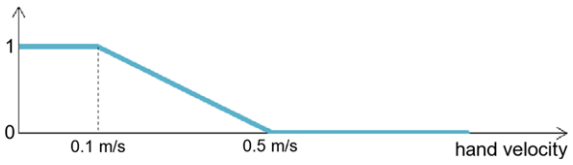


Fig. 5. Calculus of the velocity score.

0.5 m/s, as depicted in Fig. 5. The slope and the parameters of the weighting function were chosen through observation.

Since we obtain a classification label and a hand shape score for each frame, we compute the hand shape hypothesis for the actual frame by adding the scores for each hand shape hypothesis in the last 30 frames (about 1 s at 30 fps) multiplied by a time-weighting factor, and then choosing the one with the highest score.

4. Case study

A significant example of the successful application of touchless technologies to tackle real life problems can be seen in the medical field. Here, in fact, specific solutions are required in the design of both usable and practical user interfaces [24]. Particularly in operating rooms, surgeons need to access medical images without having to physically touch any control since they cannot leave the sterile field around the patient [25]. For this reason, touchless interfaces are advantageous in that they can preserve sterility around the patient without forcing surgeons to rely on a proxy, who may not share the same level of professional vision [26].

Therefore, as a case study, we implemented the hand shape classification pipeline by extending the open-source MITO project [27], a software tailored for 3D medical image visualization. In more detail, we designed an interface that allows the user to point to, crop and rotate 3D reconstructions of anatomical data with up to 6 DOF. In addition to the hand shape classification chain described in this paper, we used the technique described in [28] to make the mouse pointer always bind to the visible surfaces of 3D objects, and the filtering technique described in [29] to enhance the precision of distal pointing and to smooth the 3D rotations.

The interface is built upon five independent modules, connected as a pipeline by means of a signal/slot paradigm (see Fig. 6). Each module behaves as a data producer and/or data consumer and can be easily replaced in the pipeline. Module connectivity is implemented by means of a signal/slot paradigm provided by Boost C++ Libraries [30]: the signals represent callbacks with multiple targets (publishers), whereas the slots represent callback receivers (subscribers) and are called when signals are emitted. The processing pipeline also makes use of the Point Cloud Library (PCL) [31], a comprehensive C++ framework containing a wide range of state-of-the-art algorithms for 3D point cloud processing.

The **KinectSDKJointsProvider** is the first component in the pipeline and relies on the Microsoft Kinect SDK to segment users from depth data and produce a 3D point cloud for each user. It also fills a data packet, *JointsDataUnit*, which contains a 3D point cloud for the tracked user and the estimated positions of joints, and propagates it along the pipeline.

Next, the **HandExtractor** component, which consumes the *JointsDataUnit* packets, uses a nearest neighbor algorithm to extract a smaller 3D point cloud only containing the data for each of the users' hands. It also creates and propagates two *HandDataUnit* packets, one for each hand, containing the 3D point cloud for one hand and a flag defining which hand is being described (left/right).

As the third step along the pipeline, there is the **HandPostureProcessor**, which consumes the *HandDataUnit* packets. It employs principal component analysis on the 3D point cloud of the user's hand to estimate the hand palm orientation, by using a weighting scheme to account for the finger positions. The hand data is transformed (pose-compensated) so that the estimated principal directions are aligned to the frame of the camera. Then, Flusser moments are extracted from the new depth map, which was built from the pose-compensated point cloud. Two hand shape hypotheses are formulated using the extracted features in the two multi-class SVM-RBF classifiers. Finally, the component creates and propagates a *HandPostureDataUnit* packet, which contains the identifiers of the recognized postures, the actual hand-sensor distance and the estimated principal directions.

Then, the **HandPostureFilter** analyzes *HandPostureDataUnit* packets and produces the final hand posture hypothesis. It collects and analyzes 30 posture packets before deciding on a posture hypothesis, computing an aggregated score for each posture as described in Section 3.2. Then, the component creates and propagates a new *HandPostureDataUnit* packet.

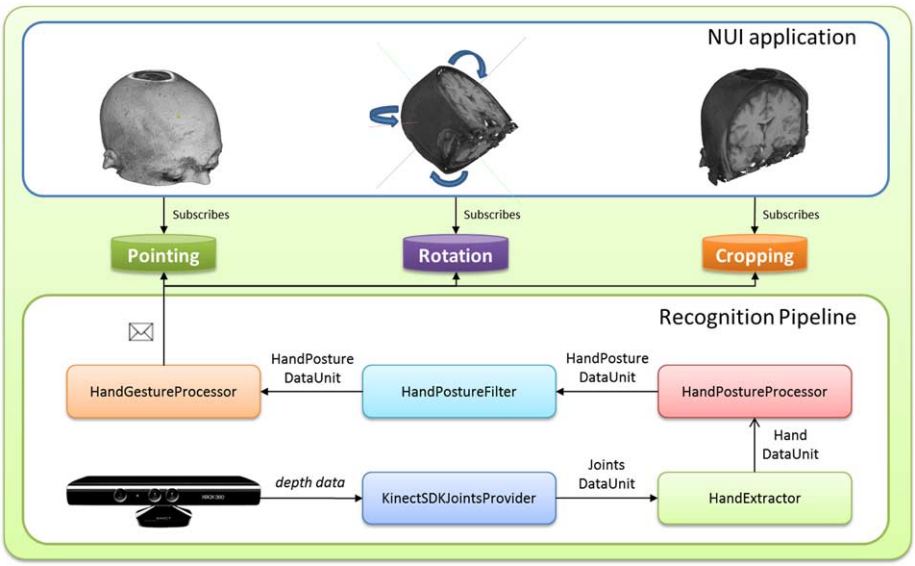


Fig. 6. The recognition modules of the touchless interface for 3D medical visualization.

Table 1
10-fold cross-validation results

Hand-sensor distance	Classification rate
0.8 m–1.2 m	**81.18%**
1.2 m–2.0 m	**73.53%**

The last module is the **HandGestureProcessor**, which consumes the *HandDataUnit* and *HandPostureDataUnit* packets. The component is in charge of collecting the posture packets and joint positions to produce high level event signals to which an application can subscribe (see Fig. 6). It manages a finite state machine for each user and, depending on the state of the FSM, emits the relevant signals (e.g. *beginRotationSignal*, *endRotationSignal*, etc.) to the registered slots.

To cope with the limitations of floating point data representation (see [32] for further details), we scaled each moment invariant in the $[-1, +1]$ range, by considering the statistical distribution of values for each one of the five Flusser moments.

To assess the near-real-time interactivity of the interface, which is a strict constraint for a touchless interface because of the lack of perceptible tactile stimuli, we logged the performances of the hand shape classification pipeline on a standard PC running the 64 bit version of Microsoft Windows 7 and equipped with a 3.20 GHz Intel® Core™ i7-930K CPU and 8 GB of RAM. The time spent in each step of the pipeline was averaged over fifty frames. The results showed that the whole recognition pipeline, comprising the segmentation, normalization, features extraction and shape classification steps for both hands, once the 3D point cloud is available, does not take more than 5 ms to complete.

Table 2
Confusion matrix

	Closed fist	Open hand	L shape	Finger pointing
Closed fist	**79.22%**	10.89%	2.92%	6.97%
Open hand	7.27%	**82.25%**	6.67%	3.80%
L shape	2.47%	6.32%	**85.20%**	6.01%
Finger pointing	9.56%	5.67%	7.89%	**76.89%**

5. Evaluation

The evaluation is divided into two parts. The first part deals with the analysis of the hand shape classification rates on single depth images, whereas the second deals with the evaluation of the score weighting method we designed to improve the classification accuracy.

5.1. Evaluation of the classifier

For each classifier a ten fold cross-validation was performed. The resulting classification rates are presented in Table 1, and the confusion matrix in the 0.8 m–1.2 m interval is shown in Table 2.

The overall classification accuracy for hand postures in the 0.8 m–1.2 m interval was found to be 81.18%, while in the 1.2 m–2.0 m interval it decreased to 73.53%. The classification rate in the whole 0.8 m–2.0 m interval, achieved by using only one of the two classifiers depending on the hand-sensor distance, is 75.94%.

5.2. Evaluation of the filtering approach

We were interested to assess whether or not the application of the velocity-based filtering to the classification pipeline improves the hand shape estimation and so eases the execution of complex 3D interaction tasks. In more detail, we were interested to assess if using the user's hand velocity to understand the intentionality of a change in the hand posture results in an overall gain in the classification and so to a reduction of the completion time of touchless interaction tasks. In order to investigate this hypothesis, we conducted a user study.

5.2.1. Evaluation set-up

Twelve subjects participated in the study as volunteers. All (9 male, 3 female) were recruited from students in Computer Engineering. Their ages ranged from 22 to 27, averaging at 24. All were right handed. None of them had a good competence in using a touchless interface. The subjects were randomly assigned to two groups. The first group used the complete score weighting method described in Section 3.2, whereas the second did not use the velocity score. To counteract the effect of fatigue or other outside factors on the experiment, we used a complete counterbalanced measures design.

The subjects were required to perform three tasks, while standing in front of the Kinect sensor at a distance of 1.5 m. The three tasks consisted in rotating a 3D-reconstruction of a human head from MRI-scans (see Fig. 7), which is visualized in a close-up view (centered at 0.3 m from the user) in a semi-immersive virtual environment, using a projection screen with a 3 m width. The update rate was controlled at 60 Hz. All the participants had a training time of five minutes to familiarize themselves with the interface. In this period of time, a tutor explained the goal of the test and how to perform a 3D rotation using static hand postures to grab and release the object and hand and arm movements to rotate it.

The subjects were required to perform roll, pitch and yaw rotations of the 3D object, with a fixed center of rotation, by using only one hand. The roll rotation is an in-plane rotation, since it requires a rotation along the longitudinal axis (from the 3D object to the user), whereas yaw and pitch are out-of-plane rotations, since the object is rotated along the vertical and lateral axis, respectively.

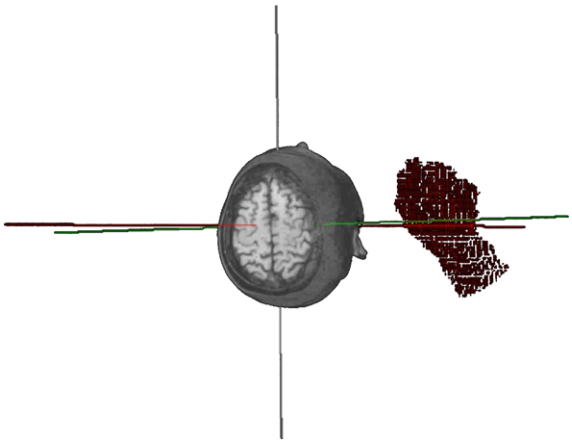


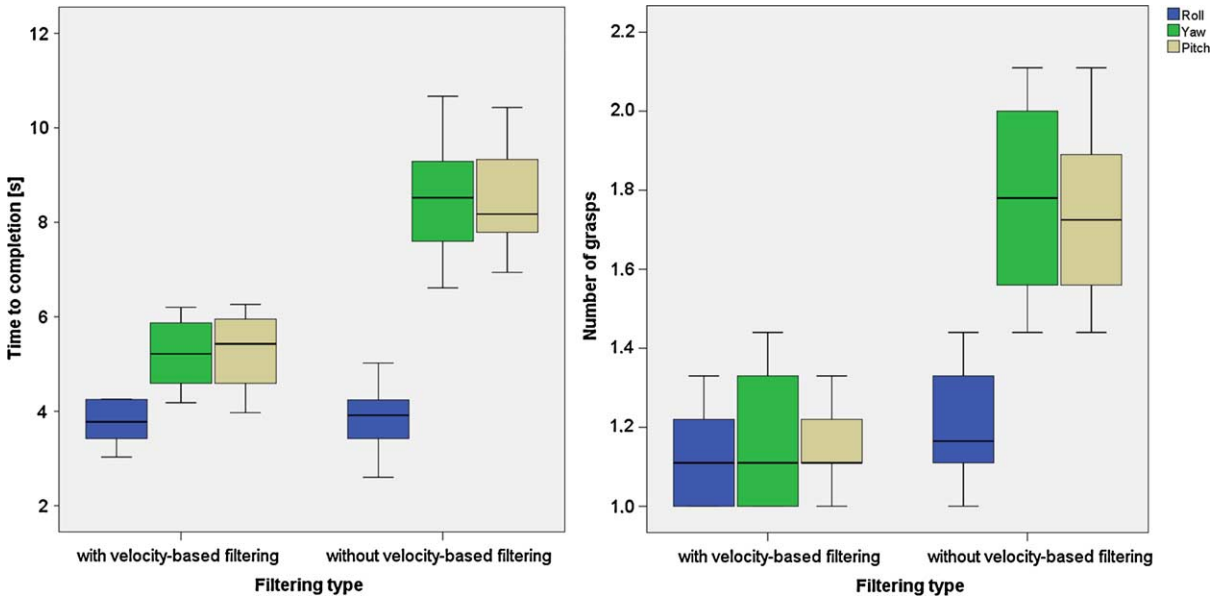
Fig. 7. Rotation of the 3D object with the NUI.

In more detail, the execution of a rotation task required the participant to seize the object by assuming the grasping posture (closed fist), then to rotate it by moving her/his hand around the center of the object, and finally to release it by assuming the ungrasping posture (open hand). For each task, the subjects were required to rotate the object by $30°$, $90°$ and $150°$. As suggested in [33], in each task the initial orientation of the object was chosen so that there would be no coincidence between a principal axis of the viewer's visual frame, the rotation axis and the object's major limb.

Since all the rotation tasks require only one degree of freedom to be completed, during the tasks we let participants control only one degree of freedom at a time. The task completion time, computed as the average of the three trials per rotation, and the number of grasps of the object were used as quantitative measures. A rotation was considered completed only if it was performed with an accuracy of 95%. The experiment consisted of 108 trials in total (12 subjects $\times$ 1 posture filtering approach per subject $\times$ 3 tasks $\times$ 3 rotations per tasks). Every user was able to complete the trials.

5.2.2. Experimental results and analysis

We performed a mixed-design ANOVA with *filtering type* as a between-subjects factor. The performance was measured by three repeated measures corresponding to the three levels of the within-subject factor *rota-*



(a) Time to completion

(b) Number of grasps

Fig. 8. Box and whisker diagrams of time to completion and number of grasps.

tion axis. We were interested in testing both a main effect of the posture filtering method as well as the interactions between the posture filtering and rotation tasks.

Figures 8a and 8b show the box-plots of completion time and number of grasps for the three rotation tasks. The results indicated that the between group variable filtering type was statistically significant on the time to completion ($F_{1,10} = 29.85$, $p < .001$) and on the number of grasps $F_{1,10} = 31.44$, $p < .001$). That is, the time to completion and the number of grasps of the three rotation tasks differed significantly as a function of the filtering condition. In more detail, the results indicated that the group which had used the weighting method with the velocity score achieved the lowest time to completion ($M = 4.744$ vs $M = 6.953$) and the lowest number of grasps ($M = 1.147$ vs $M = 1.574$). Therefore, averaged over the three task completion time measures, the posture filtering method leads to a reduction of approximately 32% in the completion time when compared with the recognition without using the posture filtering step, and to a reduction of approximately 27% in the number of grasps. However, the results were highly different according to the particular rotation task.

In fact, the analysis also revealed a significant main effect of the within-subjects variable rotation axis ($F_{2,20} = 51.025$, $p < .001$) on the time to completion. The results indicated that, averaged across the two filtering conditions, the shortest time to completion was achieved for the roll rotation ($M = 3.802$). The time spent for the roll rotation was statistically different from the time spent for the yaw rotation ($F_{1,10} = 87.788, p < .001$), whereas the difference between the yaw and pitch rotations was not statistically significant.

By considering the number of grasps, this result is further confirmed. The analysis revealed a significant main effect of the within-subjects variable rotation axis ($F_{2,20} = 12.495$, $p < .001$) on the number of grasps. Again, averaging across the two filtering conditions, the number of grasps used during the roll rotation ($M = 1.165$) was statistically different from that for the yaw rotation ($F_{1,10} = 14.863, p < .01$), whereas there was no significant difference in the number of grasps between the yaw and pitch rotations ($M = 1.472$ and $M = 1.444$, respectively).

These results were expected. In fact, yaw and pitch are both out-of-plane rotations, whereas roll is an in-plane rotation. This means that, for the roll rotation, the hand is always unoccluded, facilitating the hand shape recognition and so reducing the the number of grasps necessary to complete the rotation and the time to complete the whole rotation task. Therefore, both the filtering approaches performed differently when applied to in-plane rotations and to out-of-plane rotations due to the complexity required to recognize the

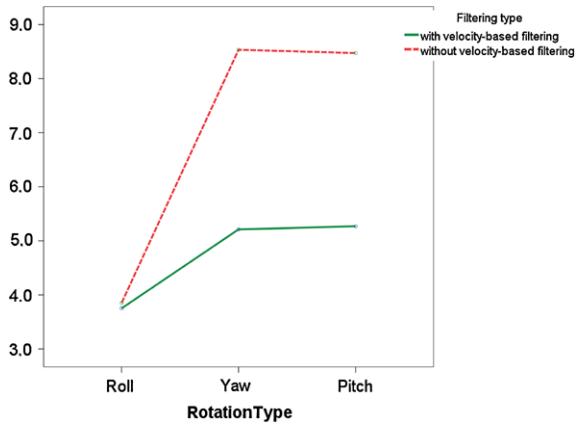


Fig. 9. Estimated marginal means of time to completion for the three rotations tasks.

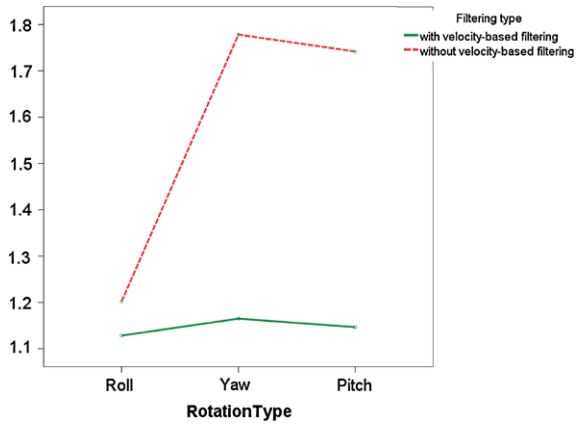


Fig. 10. Estimated marginal means of number of grasps for the three rotations tasks.

hand shape when it is rotated arbitrarily with respect to the camera. However, the variability of the completion time and of the number of grasps was significantly different between the two filtering approaches used.

In fact, the analysis also revealed a statistically significant interaction between the factors filtering type and rotation axis both on time to completion ($F_{2,20} = 13.551, p < 0.01$) and on the number of grasps ($F_{2,20} = 10.2, p < 0.01$). These results, summarized in Figs 9 and 10, suggest that the time to completion and the number of grasps across the three rotation tasks are dependent on the type of filtering used (weighting method with or without the velocity score).

Although there was a general increase in the number of grasps and in the time to completion between the roll rotation and the yaw and pitch rotations, the rate of increase is significantly different for the group that did not use the velocity-based shape filtering. In greater detail, the analysis revealed that, when the velocity score was not used, users grasped the object more times to complete the rotation during the yaw and pitch rotations compared to the roll rotation. On the contrary, when it was used, the average number of grasps was almost the same between the three rotation tasks. This may explain why the velocity-based filtering method showed a significantly lower time to completion, which was almost constant over the three rotation tasks.

On one hand, this finding confirms that, whatever rotation task is executed, the efficiency of the rotation increases when a velocity-based weighting method is used. On the other, it also suggests that the presence of the velocity-based method considerably reduces the time to completion of out-of-plane rotations. The analysis suggests that this improvement is due to the fact that the subjects were able to grab the object and then rotate it precisely without releasing the grasp, since the velocity-based filtering reduces the number of erroneous classifications.

6. Conclusions

In this article we have presented a novel method for view-independent static hand pose recognition from depth data. Since the system relies only on depth data, it is invariant to the content and illumination of the scene, so making it suitable for use in unconstrained environments. Considering the challenging nature of the task, achieving real-time hand shape classification from low resolution depth images, with the user being free to perform out-of-plane rotations of the hands and to move in the viewing area of the Kinect, the results are promising. We have detailed the 3D image analysis algorithms and their implementation, and demonstrated the real-time applicability of the whole shape classification chain.

Furthermore, we have discussed the results of a user study in which the hand velocity was evaluated as an indicator of the user's intentionality in changing the hand posture. This study is the first to investigate the relationship between the velocity and shape of the hand in aimed movements. These results have implications for the design of applications where users are expected to interact through the use of camera-based gesture recognition technology without being constrained to maintain a fixed position in the 3D space. In partic-

ular, the analysis of the weighting method for filtering the shape hypotheses could be very beneficial to the research community and spur further research in this area.

Acknowledgements

The view-independent hand shape classification method described in this work has been awarded with the Second Place Award at the ChaLearn Gesture Demonstration Competition at ICPR 2012. We would like to thank the organizers, Isabelle Guyon and Vassilis Athitsos, and the four judges, Alex Balan, Hugo Jair Escalante, Paul Doliotis and Jeffrey Margolis, for their suggestions that have helped us to improve this work significantly.

References

[1] G. Goth, Brave NUI world, *Communications of the ACM* **54**(12) (2011), 14–16.

[2] L. Gallo, A study on the degrees of freedom in touchless interaction, in: *SIGGRAPH Asia 2013 Technical Briefs (SA)*, ACM, New York, NY, USA, 2013. doi:10.1145/2508363.2525481.

[3] J.P. Wachs, M. Kölsch, H. Stern, and Y. Edan, Vision-based hand-gesture applications, *Communications of the ACM* **54**(2) (2011), 60–71.

[4] L. Gallo and A.P. Placitelli, View-independent hand posture recognition from single depth images using PCA and Flusser moments, in: *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems (SITIS)*, IEEE, 2012, pp. 898–904.

[5] A. Erol, G. Bebis, M. Nicolescu, R. Boyle, and X. Twombly, Vision-based hand pose estimation: A review, *Computer Vision and Image Understanding* **108**(1–2) (2007), 52–73.

[6] E. Persoon and K. Fu, Shape discrimination using Fourier descriptors, *IEEE Transactions on Systems, Man and Cybernetics* **7**(3) (1977), 170–179.

[7] M. Hu, Visual pattern recognition by moment invariants, *IRE Transactions on Information Theory* **8**(2) (1962), 179–187.

[8] J. Flusser and T. Suk, Pattern recognition by affine moment invariants, *Pattern Recognition* **26**(1) (1993), 167–174.

[9] K. Oka, Y. Sato, and H. Koike, Real-time fingertip tracking and gesture recognition, *IEEE Computer Graphics and Applications* **22**(6) (2002), 64–71.

[10] L. Gallo, A glove-based interface for 3D medical image visualization, in: *Intelligent Interactive Multimedia Systems and Services (IIMSS)*, Springer-Verlag, Berlin Heidelberg, 2010, pp. 221–230.

[11] T. Lee and T. Hollerer, Handy AR: Markerless inspection of augmented reality objects using fingertip tracking, in: *IEEE International Symposium on Wearable Computers (ISWC)*, IEEE Computer Society, Washington, DC, USA, 2007, pp. 83–90.

[12] Y.-T. Chen and K.-T. Tseng, Multiple-angle hand gesture recognition by fusing SVM classifiers, in: *IEEE International Conference on Automation Science and Engineering (CASE)*, IEEE, 2007, pp. 527–530.

[13] Y. Sato, M. Saito, and H. Koik, Real-time input of 3D pose and gestures of a user's hand and its applications for HCI, in: *IEEE Virtual Reality Conference (VR)*, IEEE Computer Society, Washington, DC, USA, 2001, pp. 79–86.

[14] S. Malassiotis and M.G. Strintzis, Real-time hand posture recognition using range data, *Image and Vision Computing* **26**(7) (2008), 1027–1037.

[15] C. Keskin, F. Kiraç, Y.E. Kara, and L. Akarun, Randomized decision forests for static and dynamic hand shape classification, in: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012, pp. 31–36.

[16] C. Keskin, F. Kiraç, Y.E. Kara, and L. Akarun, Hand pose estimation and hand shape classification using multi-layered randomized decision forests, in: *12th European conference on Computer Vision (ECCV) – Volume Part VI*, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 852–863.

[17] P. Doliotis, V. Athitsos, D. Kosmopoulos, and S. Perantonis, Hand shape and 3D pose estimation using depth data from a single cluttered frame, in: *Advances in Visual Computing*, 2012, pp. 148–158.

[18] A. Maimone, J. Bidwell, K. Peng, and H. Fuchs, Enhanced personal autostereoscopic telepresence system using commodity depth cameras, *Computers & Graphics* **36**(7) (2012), 791–807.

[19] K. Khoshelham and S.O. Elberink, Accuracy and resolution of kinect depth data for indoor mapping applications, *Sensors* **12**(2) (2012), 1437–1454.

[20] S. Conseil, S. Bourennane, and L. Martin, Comparison of Fourier descriptors and Hu moments for hand posture recognition, in: *European Signal Processing Conference (EUSIPCO)*, EURASIP, 2007, pp. 1960–1964.

[21] J. Flusser, On the independence of rotation moment invariants, *Pattern Recognition* **33**(9) (2000), 1405–1410.

[22] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, A practical guide to support vector classification, Tech. rep., Department of Computer Science, National Taiwan University (July 2003).

[23] L. Gallo, M. Ciampi, and A. Minutolo, Smoothed pointing: A user-friendly technique for precision enhanced remote pointing, in: *2010 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, IEEE, 2010, pp. 712–717.

[24] A. Blandford, G. De Pietro, L. Gallo, A. Gimblett, P. Oladimeji, and H. Thimbleby, Engineering interactive computer systems for medicine and healthcare (EICS4Med), in: *ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS)*, ACM, New York, NY, USA, 2011, pp. 341–342.

[25] L. Gallo, A. Placitelli, and M. Ciampi, Controller-free exploration of medical image data: Experiencing the Kinect, in: *International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE Computer Society, Washington, DC, USA, 2011, pp. 1–6.

[26] R. Johnson, K. O'Hara, A. Sellen, C. Cousins, and A. Criminisi, Exploring the potential for touchless interaction in image-guided interventional radiology, in: *ACM CHI Conference on Human Factors in Computing Systems (CHI)*, ACM, New York, NY, USA, 2011, pp. 3323–3332.

[27] iHealth lab, MITO: Medical imaging toolkit, http://sourceforge.net/projects/mito/ (2013).

[28] L. Gallo, A. Minutolo, and G. De Pietro, A user interface for VR-ready 3D medical imaging by off-the-shelf input devices, *Computers in Biology and Medicine* **40**(3) (2010), 350–358.

[29] L. Gallo and A. Minutolo, Design and comparative evaluation of smoothed pointing: A velocity-oriented remote pointing enhancement technique, *Int. J. of Human-Computer Studies* **70**(4) (2012), 287–300.

[30] B. Kempf, The Boost.Threads Library, Dr. Dobb's Journal, www.drdobbs.com/cpp/the-boostthreads-library/184401518 (May 2002).

[31] R.B. Rusu and S. Cousins, 3D is here: Point Cloud Library (PCL), in: *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ, USA, 2011, pp. 1–4.

[32] D. Goldberg, What every computer scientist should know about floating-point arithmetic, *ACM Computing Surveys (CSUR)* **23**(1) (1991), 5–48.

[33] L. Parsons, Inability to reason about an object's orientation using an axis and angle of rotation, *Journal of Experimental Psychology: Human Perception and Performance* **21**(6) (1995), 1259–1277.