Ensemble to improve gesture recognition

Loris Nanni[1], Alessandra Lumini[2], Fabio Dominio[1], Mauro Donadeo[1], Pietro Zanuttigh[1]

[1]Department of Information Engineering - University of Padova, Via Gradenigo, 6 - 35131- Padova - Italy

[2]DISI, University of Bologna, via Venezia 52, 47023 Cesena - Italy.

E-mail: nanni@dei.unipd.it;

Abstract

Automatic hand gesture recognition plays a fundamental role in current research with the aim of empowering a natural communication between users and virtual reality systems. Starting from an existing work, based on the extraction of two different descriptors from the depth maps followed by their classification with a stand-alone multi SVM classifier, in this paper we improve the gesture recognition system performances and reliability and we evaluate different classification approaches. To this purpose, we first compare the performance of different descriptors and analyze their correlation for assessing their complementarity, and then we demonstrate the advantage gained by their fusion by the Wilcoxon Signed-Rank test.

In particular, the novelties of this paper are a new method for extracting features from the curvature image and the design of a very effective ensemble of classifiers to solve the problem.

Keywords: hand gesture; texture descriptor; ensemble of classifiers; depth data; Kinect.

1. Introduction

Automatic hand gesture recognition [23] is a challenging problem that is attaining a growing interest due to its many applications, in particular in the field of natural interfaces for human-computer interaction. While hand gesture recognition from 2D digital images has been investigated for many years, the use of range cameras or depth cameras for automatic gesture recognition is still at an early stage of existence. The goal of this research is to build a gesture recognition system exploiting the 3D information provided by a depth camera framing the user hand.

Until recently, most hand gesture recognition approaches were based on the analysis of images or videos only framing the hand [23][24]. The bidimensional representation provided by images and videos is, however, not always sufficient to capture the complex movements and inter-occlusions characterizing hand gestures.

A more accurate description is given by three dimensional representations, which are now easily obtainable thanks to the recent introduction of low cost consumer depth cameras. Innovative devices, such as Time-Of-Flight cameras and Microsoft's Kinect™ [25] have made depth data acquisition available to the mass market, thus opening the way to novel gesture recognition approaches based on depth information.

Several different approaches have been proposed for this task, and most of them share a common basic approach consisting in firstly extracting a set of features from depth data and then applying machine learning techniques to the extracted features in order to recognize the performed gestures. In [30] silhouette and cell occupancy features are used to build a shape descriptor to be fed into a classifier based on action graphs. Suryanarayan et Al. [34], instead, extract 3D volumetric shape descriptors from the hand depth and then exploit Support Vector Machines for the classification. Volumetric features and a SVM classifier are also used by [36]. Doliotis et al. [23] instead extract the trajectory of the hand that is then used as input for a Dynamic Time Warping (DTW) algorithm. Other approaches [21], [22] use features based on the convex hull of the hand’s silhouette. Finally, [32] and [33] compare the histograms of the distance of hand edge points from the hand center in order to recognize the gestures.

Starting from a previous work [19], this paper follows this rationale as well and exploits a hand gesture recognition scheme combining two types of hand shape features to be fed into a classifier, namely, distance features describing the positions of the fingertips and curvature features computed on the hand contour.

In [19] a simple combination of two SVM classifiers, one for each of the two types of features, was used. We hereby propose to improve the previous system performance by studying the correlation of the features and designing a more performing ensemble based on the fusion of different classifiers.

The proposed ensemble design accounts for the following observations:

- Since most of the features extracted are greatly correlated and belong to a high dimensionality space we design a Random subspace ensemble (RS) of classifiers [6]. In fact is well known in the literature the advantage of using feature selection/transformation techniques [5] in order to reduce the number of the features (which can cause the curse of dimensionality problem, i.e. with a fixed number of training samples, the predictive power of a system reduces as the dimensionality increases) and to solve the problem of the correlation among them. Several papers (e.g. [7]) have shown that random subspace ensembles seem to be well-suited to classify high dimensionality data (e.g. gene expression data), as they reduce the high dimensionality of the data by randomly selecting subsets of features.
- Another remarkable approach for dealing with correlation of features and improving the performance of an ensemble of classifiers is to design heterogeneous ensembles; in this work we test both a random subspace of SVM classifiers (as stated above) and a boosting approach based on decision trees (ROT) [16].
- A further observation is that a recognition system can reach better performance in term of accuracy and computational requirements by designing a parallel ensemble instead of training a single classifier based on the concatenation of features. In this work we propose a different classification system for each set of descriptor which are finally fused by the sum rule.

- Finally, a new set of features is considered, obtained directly from the curvature information. The proposed descriptor is motivated by the encouraging results [8] obtained in extracting textural information from other descriptors represented in a matrix format. In this work we represent the curvature as a matrix and extract textural descriptors to represent each pattern. In particular we examine the local phase quantization (LPQ) texture descriptors [4]. LPQ extract local information considering a set of neighbor pixels of a given center, therefore they are able to handle the correlation among the original features (in this case the curvature information). We believe that our technique exploits a new source of information for representing a gesture image.

The paper is organized as follows: Section 2 introduces the proposed gesture recognition system. Section 3 contains the experimental results and finally Section 4 draws the conclusions.

2. Method overview

The proposed recognition pipeline, depicted in Fig. 1, is based on three main steps: in the first one the depthmap region corresponding to the hand is segmented from the background, then the relevant features are extracted from the hand depth data only.

For each descriptor an ensemble of SVM or an ensemble of boosted decision tree is trained. Then these sets of classifiers are combined by sum rule [2].

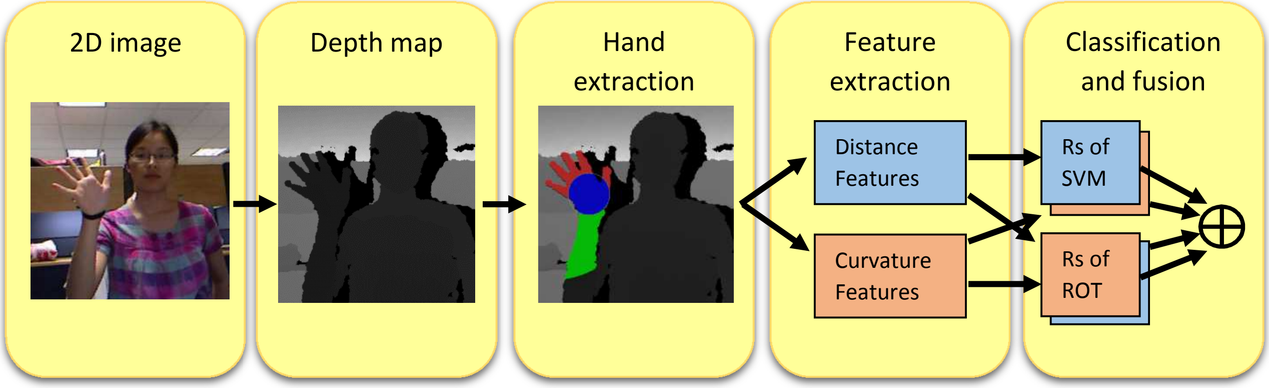


Figure 1. Proposed ensemble.

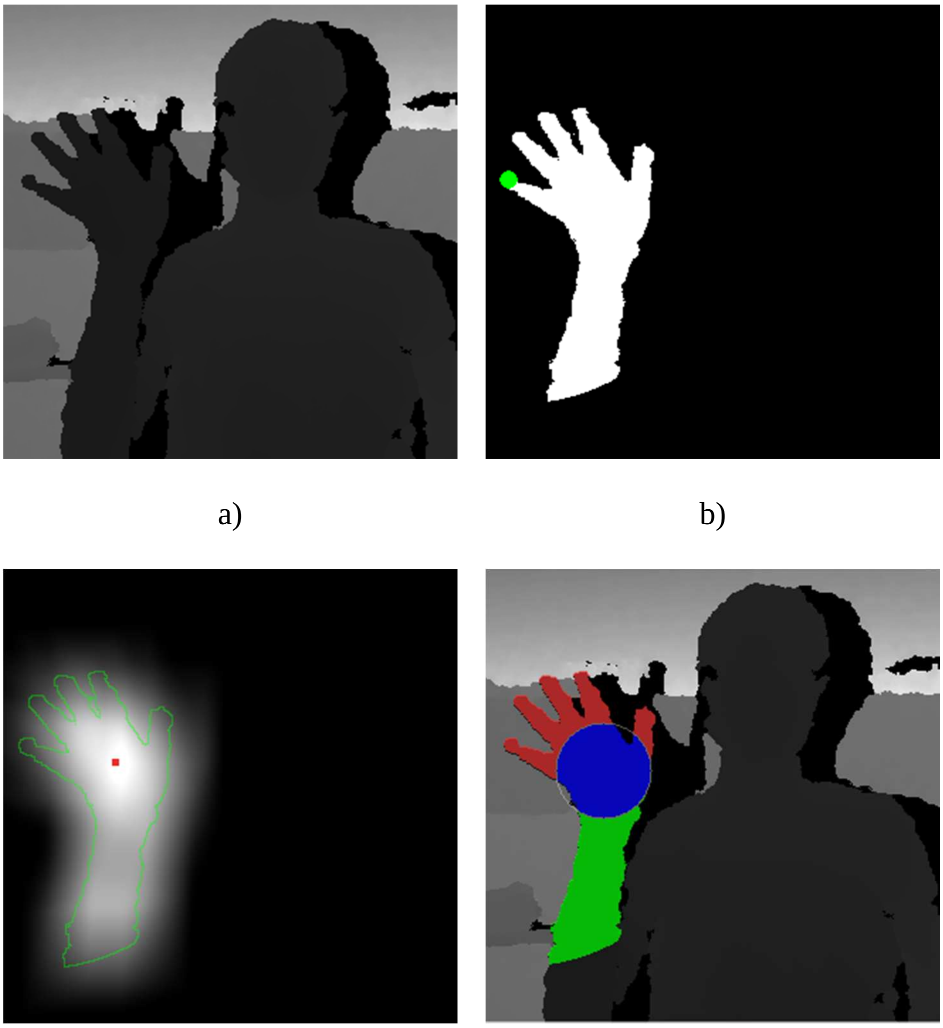
2.1 Hand segmentation and palm recognition

The first step, as pointed out in Section 2, is the extraction of the depth samples corresponding to the hand region from the depth map. For this purpose we used the approach introduced in [19] and [20], that we briefly resume here.

The proposed method starts by detecting the closest point X_{min} to the camera in the acquired depth map, as exemplified in Figure 2b. Note how the algorithm automatically avoids to select as X_{min} an isolated artifact due to noise by verifying the consistency of the selected point with close samples [19]. Once X_{min} is found, the set *H* of all the points X_i with depth $D(X_i)$ within a range [X_{min} , $X_{min} + T_h$] and with a Euclidean distance in 3D space from X_{min} smaller than T_{h2} is computed:

$$H \quad = \{X_i \mid D(X_i) < \ D(X_{min}) \ + \ T_h \ \wedge \|X_i \ - X_{min}\| < \ T_{h2}\}$$

where T_h and T_{h2} are thresholds whose values depend on the hand size (e.g., in our experiments we set T_h= 10cm and T_{h2} = 30cm). Further checks are also performed on the hand color and size [19,20] in order to discard points X_i not belonging to the hand. Note how this approach allows to reliably segment the hand from the scene objects and the other body parts (as shown in Figure 2b). Parts of the wrist and the forearm may be included at this level into *H*, but they will be removed in the following step.



a) b) c) d)

Figure 2. Extraction of the hand: a) original depth image; b) depth mask (the green point is X_{min}); c) blurred depth mask with C dotted in red; d) segmented hand regions. *(best viewed in colors)*

A 2D mask corresponding to the hand samples in the depth image space is then built and filtered by a Gaussian filter with a very large standard deviation that depends on the distance of the hand from the Kinect [19].

The maximum of the filtered image, which is the starting point of the next step, is now detected. Since the Gaussian filter support is larger than the hand and the palm is larger than the forearm and denser than the finger region, the computed maximum lies somewhere close to the center of the palm region (see Figure 2c). In case of multiple points with the same maximum value the closest to X_{min} is selected.

The following step of the proposed method is the detection of the largest circle, centered on the maximum point cited above, that can be fitted on the palm region, as described in [19]. A more refined version of this procedure [20] uses an ellipse in place of the circle in order to better approximate the shape of the palm, especially when the hand is tilted respect to the Kinect. We will denote with *C* the center of the circle or ellipse, i.e., the center of the palm region.

For the distance features extraction, the samples inside the circle (or the ellipse) are associated now to the palm region and a 3D plane is fitted on them by using a robust estimator exploiting SVD and RANSAC. Principal Component Analysis (PCA) is then applied to the hand samples in order to extract the main axis, that roughly corresponds to the direction of the vector going from the wrist to the fingertips. Note how the direction computed in this way is not very precise and only gives a general indication of the hand orientation.

On the basis of the computed data, *H* is now segmented in three sets [19] as shown in Figure 2d:

- *P* containing points of the palm.
- *W* containing the points of the wrist and the forearm that will be discarded.
- *F* containing the points corresponding to the fingers region.

Edge detection is then applied to the points of *H*-*W* in order to build the set *E* containing the hand contour points.

2.2 Feature Extraction

2.2.1 Distance features

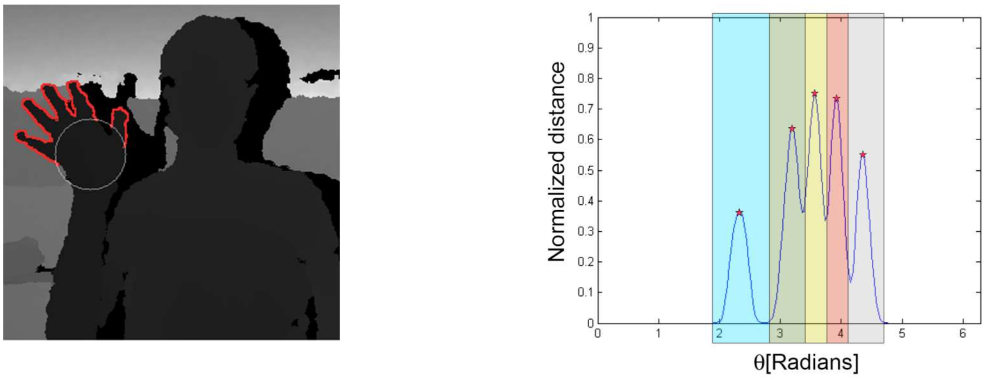
The first feature set, introduced in [19] on the basis of a previous idea from [32], represents the distance of the samples in *F* from the hand centroid *C*.

For each sample $\mathrm{X_i}$ in *F* we compute its distance $\mathrm{d(X_i)}$ in 3D space from the centroid and the angle between the projections on the palm plane of the PCA principal axis and of the vector connecting the point to the centroid. Then a histogram representing the maximum of the distance from the centroid for each angular direction is built:

$$\mathrm{L}(\theta) = \max_{\theta-\frac{\Delta}{2}<\theta(\mathrm{X_i})<\theta+\frac{\Delta}{2}} \mathrm{d(X_i)}$$

Where $\theta(\mathrm{X_i})$ is the angular direction corresponding to X_i (using the axes computed by the PCA as reference system) and Δ is the quantization step for the histogram computation (Δ=2° in the proposed implementation). For each gesture *g* in the database, a reference histogram $\mathrm{L_g^r}$ is built and a set of angular regions corresponding to the direction of the various fingers that are used in each gesture is defined as shown in Figure 3b. These regions correspond to each finger in each gesture and will be used for computing the distance features.

In order to precisely extract the regions corresponding to the various fingers, it is necessary to align the different histograms with the template on which the regions are defined. For this purpose we search for the maximum of the correlation between the acquired histogram and the translated version of the reference histogram of each gesture (the possibility of flipping the histogram to account for the fact that the hand could have either the palm or the dorsum facing the camera is considered as described in [19]). This gives us the translational shift aligning the acquired histogram with the reference one. Note how there can be a different alignment for each gesture. This approach basically compensates for the limited precision of the direction computed by the PCA, and allows to precisely alignthe reference and the computed histograms in order to define the regions corresponding to the various features of the gesture, thus solving one of the main issues of directly applying the approach of [32]. Figure 4 shows some examples of the computed histograms for three different gestures. Note that the fingers raised in the various gestures (or the fact that there are no raised fingers for the fist) are clearly visible from the plots.



a) b)

Figure 3. Histogram of the distances in 3D space of the edge samples from the center of the palm with the features regions: a) finger edges computed from *F* ; b) corresponding histogram L(θ) with the regions corresponding to the different features.

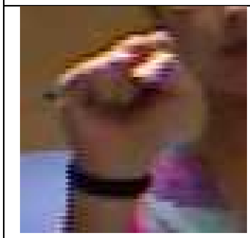
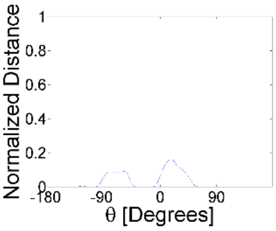
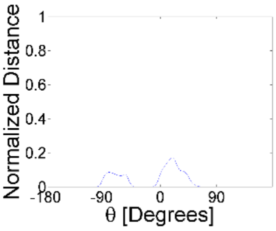
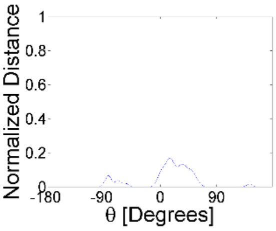
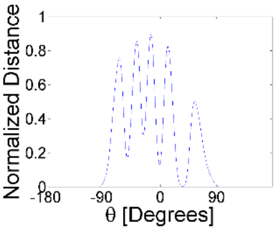
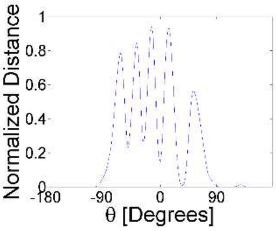
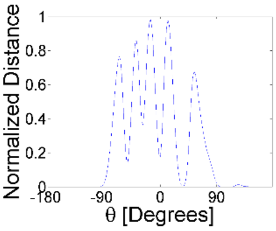
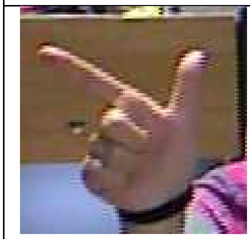
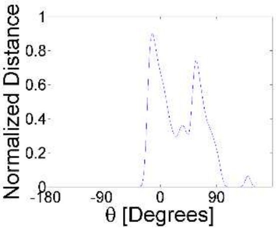
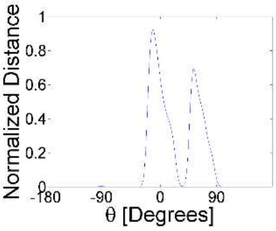
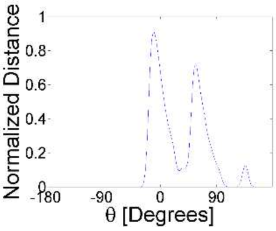
Sample image of the gesture	Distance histograms corresponding to various repetitions of the gesture		
			
			
			

Figure 4. Examples of distance histogram for some sample frames from different gestures.

The set of distance features F^l associated to each acquired sample is thus made by a feature value for each finger j in each gesture $g \in 1,..,G$ (note how not all the fingers are of interest in all the gestures). The feature value $f^l_{g,j}$ associated to finger j in gesture g is the maximum of the aligned histogram in the angular region corresponding to the finger j in gesture g (see Figure 3b), i.e. :

$$f^l_{g,j} = \frac{\max\limits_{\theta^{min}_{g,j}<\theta<\theta^{max}_{g,j}} L^g_a(\theta) - r_f}{L_{max}}$$

Where $\theta_{g,j}^{min}$ and $\theta_{g,j}^{max}$ are the extremes of the region corresponding to finger j in gesture g, $L_a^g(\theta)$ is the aligned version of the computed histogram, r_f is the radius of the circle (or the distance from the ellipse border) and L_{max} is the length of the middle finger used for normalization purposes between people with hands of different size. Note how if r_f was not subtracted from all the features the feature values would jump from 0 to r_f as soon as edge points cross the circle.

In this way up to G*5 features are built for each acquired sample (their actual number is smaller since not all the fingers are of interest in all the gestures). For example, in the dataset of [33] used in the experimental results there are 10 different gestures, and 24 features have been used.

2.2.2 Histogram of curvature features

The second descriptor represents the curvature of the edges of the hand shape in the depth map. The proposed algorithm is based on integral invariants [31]. It takes as input the hand edge points E and the mask M_h representing the hand samples in the depth map (i.e., the mask in Figure 2b).

For each point X_i in E we build a set of S circular masks $M_s(X_i),\ s = 1,..S$ centered on X_i with radius varying from 0.5 cm to 5 cm. The ratio $V(X_i, s)$ between the number of samples inside each circular mask that belong also to the hand mask and the total number of samples in the mask for is then computed, i.e.:

$$V(X_i, s) = \frac{|M_s(X_i)\ \cap M_h|}{|M_s(X_i)|}$$

Note how the radius value s actually corresponds to the scale level at which the feature extraction is performed, and that differently from [31] and other approaches, the radius is defined in metrical units, thus making the descriptor invariant with respect to the distance of the hand from the camera.

The values of $V(X_i, s)$ range from 0 (extremely convex shape) to 1 (extremely concave shape) with $V(X_i, s)$= 0.5 corresponding to a straight edge. The [0,1] interval is then quantized into B bins of equal size. Let $V_{b,s}$ be the set of the finger edge points $X_i \in E$ with the corresponding value of $V(X_i, s)$ falling in each bin (we will denote with b the bin index), namely:

$$V_{b,s} = \left\{ X_i \,\middle|\, \frac{b-1}{B} < V(X_i, s) \leq \frac{b}{B} \right\}$$

Finally, the curvature features are given by the cardinalities $|V_{b,s}|$ for each bin b and radius value s normalized with respect to the hand contour length, i.e.:

$$f^c_{b,s} = \frac{|V_{b,s}|}{|E|}$$

In this way a second feature vector F^c containing $B*S$ features is built. As expected, the value of the curvature features depends on the number of fingers not folded on the palm region and on their arrangement, thus giving an accurate description of the hand gesture. An example of curvatures vectors, arranged in a 2D matrix and visualized as B/W images, is reported in Fig. 5.

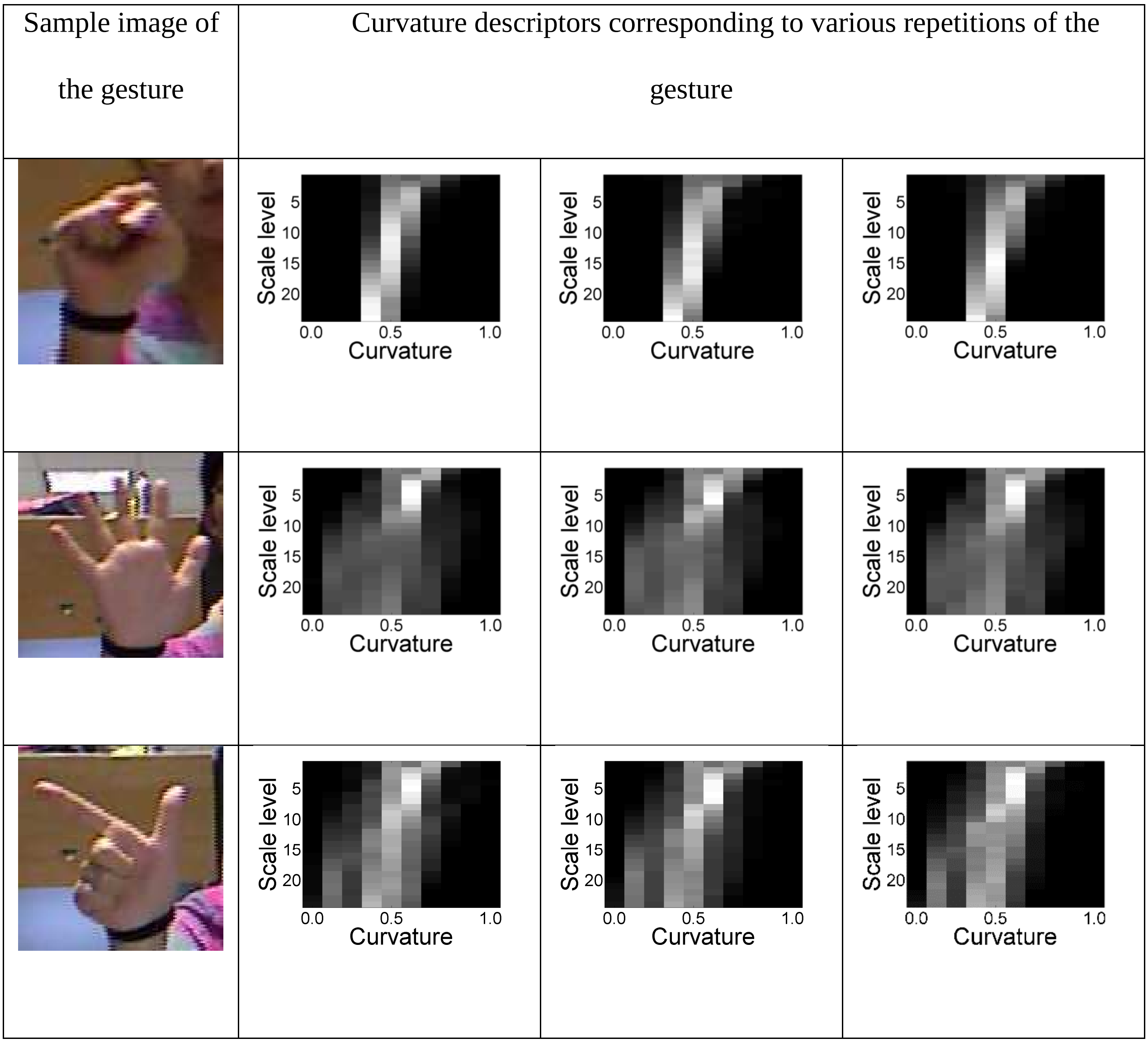


Figure 5. Examples of curvature descriptors for some sample frames from different gestures.

2.2.3 A texture based descriptor for curvature

As curvatures are fundamental data for hand representation, we propose a new set of features based on curvature information. According to the approach proposed in [8] we rearrange curvature data as matrices and extract features related to texture: since texture descriptors are designed to take into account local variation, we believe that they are well suited to extrapolate relevant information concerning shape variations due to gestures. First the feature vector representing hand curvature is rearranged as a matrix by random assignment, with 50 different random reshapings (see Fig. 6), and then texture descriptors (LPQ, in this work) is used for extracting the features. Local texture pattern features (such as LPQ [4]) are based on the idea that local neighborhoods contain many discriminative information that can be exploited by coarsely quantizing the appearance of local neighborhoods. Using different reshaping arrangements it is possible to observe and encode different aspects of curvature variations from a single curvature vector: then a different SVM is trained for each descriptor and the results are combined using the mean rule.

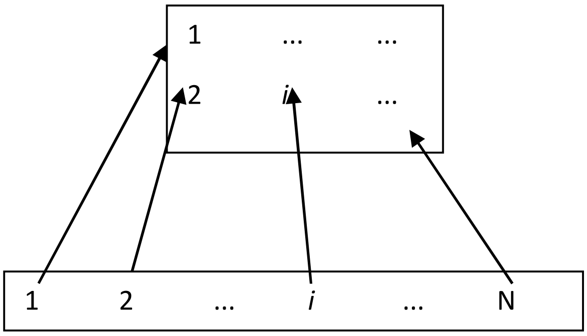


Figure 6. Reshaping a vector into a matrix.

Local Phase Quantization (LPQ), proposed as a texture descriptor by Ojansivu and Heikkila [4], is a local pattern descriptor that is based on the blur invariance property of the Fourier phase spectrum. First the phase information is computed locally from the 2-D short-term Fourier transform (STFT) evaluated in a neighborhood window for every pixel position. Then the phases of only the four low-frequency coefficients (corresponding to the 2-D frequencies) are considered, decorrelated and uniformly quantized in an eight-dimensional space. Finally, a histogram of the frequency of the resulting code words is constructed and used as descriptor. In this work the original MATLAB code[1] shared by the inventors of LPQ has been used for feature extraction. We have concatenated the descriptors obtained with radius 3 and 5.

[1] http://www.cse.oulu.fi/CMV/Downloads/LPQMatlab

2.3 Classification

2.3.1 Random Subspace Ensemble of Support Vector Machines

Automatic hand gesture recognition represents a difficult learning task, because of the high dimensionality and low cardinality of the data and the presence of redundancy in the feature set. In order to deal to the dimensionality curse, we proposed to apply random subspace (RS) ensembles [6] as an alternative to feature selection. RS is a method for ensemble design which reduces the dimensionality of the data by randomly sampling subsets of features: to construct a random subspace ensemble with *L* classifiers, *L* sets of samples of size *M* are drawn without replacement from a uniform distribution over the set of samples; then a classifier is trained on each feature subset using the whole training set (or sometimes a bootstrap).

Given a training set composed by *n* samples $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iP})$, $\mathbf{x}_i \in \Re^P$, RS randomly selects $M < P$ features from the original feature space and creates a new training set. This procedure is repeated *L* times and *L* classifiers are trained from each of the new training sets. The final classification is obtained by combining the scores of the *L* classifiers (the sum rule is employed in this work).

A great advantage of RS ensembles compared to many other ensemble methods is that they can be coupled to any general purpose classifier and they need only two parameters, *L*, the ensemble size and *M*, the size of the feature sample. In this work we use support vector machines (SVM) as classifiers and we set *L* and *M* to standard value, i.e. respectively to L=P*50%, M=50.

SVM [15] are binary classifiers that performs classification by cutting the feature space $\Re^P$ into two regions, one for each class, by a *P*-dimensional hyperplane that has the largest possible distance (margin) from the training vectors of the two classes. The mathematical formulation for SVM training requires to find the hyperplane that maximize the margin, then classification is performed assigning an unknown sample according to its position with respect to the learned hyperplane.

In order to handle classes that are not linearly separable, kernel functions may be used for remapping the training vectors into a novel higher dimensional space where they are linearly separable In this work we have used the LibSVM toolbox[2].

2.3.2 Random Subspace Ensemble of RotBoost with NPE (RSR)

In order to exploit the diversity of classifiers we designed another ensemble based on RS, the Rotation Boosting [10][14], i.e. a method for constructing ensembles, coupled with a dimensionality reduction method, i.e. the Neighborhood Preserving Embedding (NPE) [12].

Rotation Boosting, or RotBoost [14], is an ensemble classifier technique obtained integrating two successful ensemble classifier generation techniques, i.e. AdaBoost [15] and Rotation Forest [10], that both apply a learning algorithm to a set of permutated training sets.

The difference between AdaBoost and Rotation Forest lies in the ways used to perturb the original training set. AdaBoost iteratively constructs successive training sets by reweighting the original one in order to better predict the samples misclassified in the previous step. Rotation Forest builds each classifier on a training set that is obtained by randomly split into *S* subsets the feature space and reducing its dimensionality by applying Principal Component Analysis (PCA), with the aim of promoting diversity through feature extraction while preserving accuracy by keeping all the extracted principal components.

The original RotBoost algorithm [14] is a simple integration of Rotation Forest and AdaBoost: first dimensionality reduction by PCA is applied and a Rotation Matrix is calculated to map original data into a new feature space (as in RotationForest), then, base classifiers are built by applying the AdaBoost technique. In this work according to result reported in [16][17], we adopt a variant of

[2] LibSVM toolbox http://www.csie.ntu.edu.tw/~cjlin/libsvm/

RotBoost[3] obtained by coupling the ensemble with RS and by using the neighborhood preserving embedding (NPE) feature transform instead of PCA for dimensionality reduction.

PCA is an unsupervised method for dimensionality reduction aimed to find a linear mapping which preserves the maximal variance of data; unfortunately PCA fails to discover the underlying nonlinear structure of the data. On the contrary, Neighborhood Preserving Embedding (NPE) [4] [12] is a technique for dimensionality reduction which aims at preserving the local neighborhood structure on data manifold. NPE, proposed as a linear approximation Locally Linear Embedding, begins by building a weight matrix to describe the relationships between data points: each point is described as a weighted combination of its neighbors; then an optimal embedding is selected such that the neighborhood structure is preserved in the reduced space.

3. Experimental Results

In order to evaluate the performances of the proposed approach, two different datasets have been used. The first is the database provided by [32] and contains 10 different gestures performed by 10 different people as shown in Fig. 7. Each gesture is repeated 10 times for a total of 1000 different depth maps with related color images. The data has been acquired with Microsoft's Kinect camera.

The second dataset (Fig. 8) has been acquired in our laboratory and contains 12 different gestures performed by 14 different people [18]. The gestures are a small sub-set of the American Sign Language gestures. Each gesture is repeated 10 times as in the previous case for a total of 1680 different depth maps with the corresponding color images.

[3] Source code [16] available at http://www.dei.unipd.it/wdyn/?IDsezione=3314&IDgruppo_pass=124.

[4] MATLAB code available from http://www.cad.zju.edu.cn/home/dengcai/Data/DimensionReduction.html.

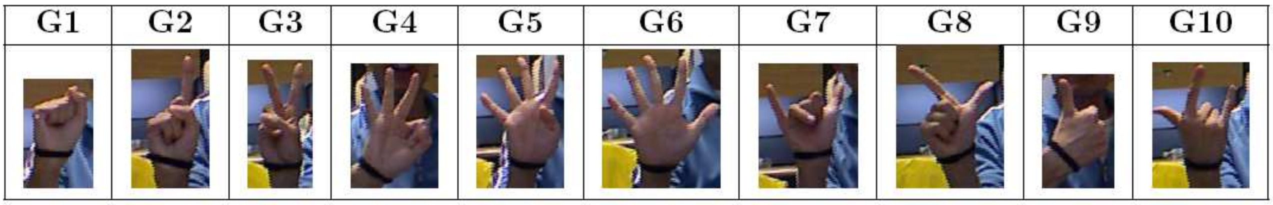


Figure 7. Sample images of the 10 different gestures in the database of [32]

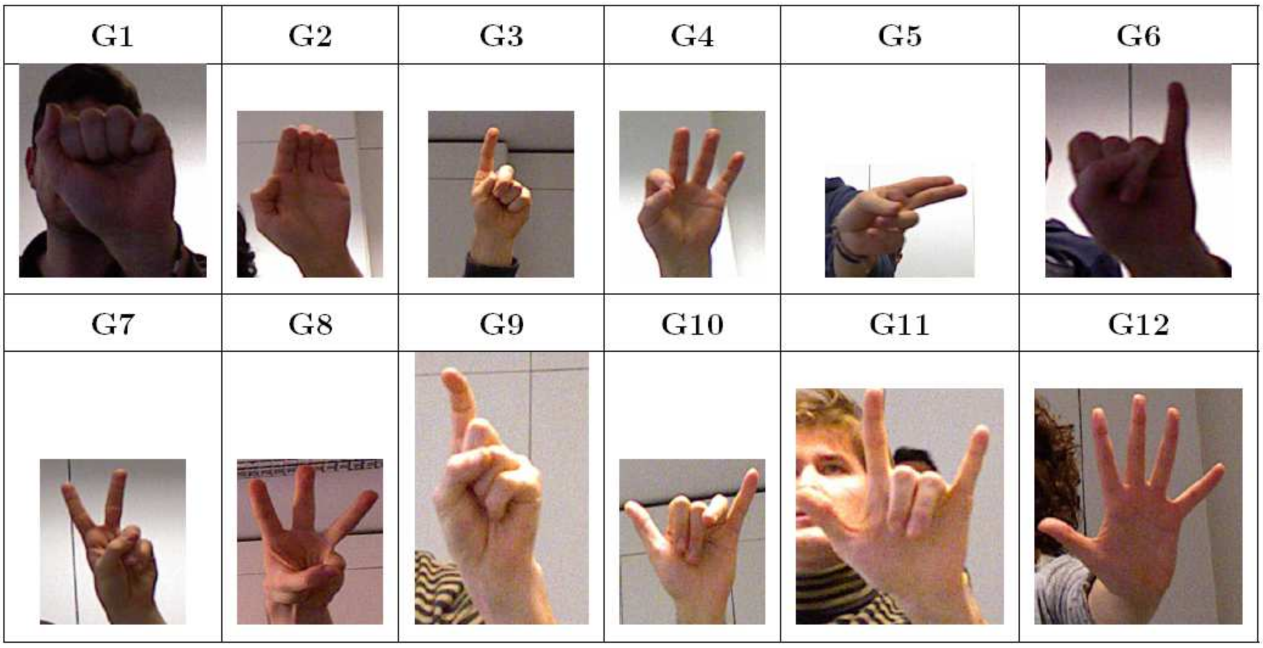


Figure 8. Sample images of the 12 different gestures in our database

The performance indicators used to compare the different approaches are accuracy (ACC), i.e. the proportion of true results (both true positives and true negatives) in the population, and error under the ROC curve [2] (EUC), i.e. 1-AUC. AUC is a one dimensional performance indicator obtained as the area under the curve (ROC) which plot the fraction of true positive rate vs. the false positives rate at various threshold settings. The AUC be interpreted as the probability that the system will assign a higher score to a randomly chosen positive sample than to a randomly chosen negative sample. In a multiclass problem as hand gesture recognition, the one-versus-all EUC is considered for each class [9] and the reported EUC value is obtained by averaging all class values.

As testing protocol we have used the leave-one-out user, i.e. *N-1* user belong to the training set and one belongs to the test set. Note that this protocol is different from the one used in [19], the “Baseline” results refer to the method of [19] but evaluated with the protocol used for this work. The average performance is reported.

The following methods are compared in Table 1 coupled with different feature vectors:

- Baseline, method proposed in [19], based on a one VS one multiclass SVM;
- SVM, a stand-alone SVM classifier;
- RS SVM, a random subspace ensemble of SVM classifiers (as described in section 2.2.1);
- RS ROT, a random subspace ensemble of rotation boosting (as described in section 2.2.2);
- HET, the heterogeneous ensemble obtained by a fusion with sum rule between RS SVM and RS ROT;

When a single feature set is present in the second column (“Feature set”) then the performance is obtained using only one feature set; when a sum or a weighed sum of feature sets is indicated, the performance is obtained combining by (weighed) sum rule the previous.

		ACC		EUC	
Approach	**Feature set**	*Dataset 1*	*Dataset 2*	*Dataset 1*	*Dataset 2*
Baseline [19]	Distance	87.2	62.6	1.7	9.5
	Curvature	92.1	86.4	0.8	2.7
	Distance + Curvature	97.2	87.1	0.3	1.8
SVM	Distance	83.9	50.9	1.8	10.0
	Curvature	92.1	82.7	0.5	2.0
RS SVM	Distance	86.9	57.2	1.1	7.1
	Curvature	92.4	84.0	0.5	1.8
	Distance + Curvature	96.7	85.3	0.3	1.2
	Distance + 2*Curvature	97.5	86.8	0.3	1.2
RS ROT	Distance	88.8	60.5	0.9	5.7
	Curvature	93.9	84.9	0.4	1.3
	Distance + Curvature	97.4	86.6	0.2	1.1
	Distance + 2*Curvature	97.4	87.5	0.1	1.1
HET	Distance	89.0	60.1	0.9	5.5
	Curvature	94.6	86.2	0.3	1.3
	Distance + Curvature	97.5	87.2	0.2	0.9
	Distance + 2*Curvature	**97.9**	**88.7**	**0.1**	**0.9**

Table 1. Comparison among methods studied in this work.

The results reported in Table 1 clearly show that:

- Random subspace permits to improve performance of stand-alone SVM;
- RS ROT works well in this problem;
- The fusion between RS ROT and RS SVM outperforms both the single approaches;
- The fusion of the classifiers trained using different descriptors greatly improves the performance.

The difference between Baseline and SVM is that in [19] a grid search to optimize parameters (for maximizing accuracy) was performed for each run of the fold, separately in each dataset. On the contrary, in this work we chose to not perform SVM parameters optimization[5], to avoid overtraining,

[5] We use standard SVM parameters: radial basis function kernel, γ=0.1, C=1000 for both the datasets

since both the datasets are small and were collected from the same research group in the same location.

The second experiment is aimed at evaluating the novel texture based descriptor for curvature introduced in section 2.1.2. In Table 2 the performance are evaluated as a function of different feature sets and their weighed fusion: the curvature set and the novel texture based descriptor for curvature (named *Texture*). For *Curvature* we use a RS of SVM, for *Texture* we train a different SVM for each reshaping then we combine them by sum rule. Our results confirm that combining the two approaches permits to outperform *Curvature.*

		ACC		EUC	
Approach	**Feature set**	*Dataset 1*	*Dataset 2*	*Dataset 1*	*Dataset 2*
HET	Curvature	92.4	82.7	0.5	2.0
	Texture	91.0	80.9	0.8	1.8
	Texture + Curvature	93.5	84.3	0.5	1.6
	Texture + 2*Curvature	93.4	85.0	0.5	1.6
	Texture + 3*Curvature	**93.5**	**85.1**	**0.5**	**1.6**
	Texture + 4*Curvature	93.3	84.9	0.5	1.6

Table 2. Comparison among the curvature feature set, the novel texture based descriptor for curvature and their fusion.

Finally, for a statistical validation of our experiments we have used the Wilcoxon signed rank test [1], obtaining that HET coupled with both the descriptors (Distance + Curvature) outperforms with a p-value of 0.05 all the other methods. Moreover, we analyzed the relationship among the different feature sets according to Q-statistic [13], in order to evaluate the error independence between the classifier trained using those features.

In this problem the Q-statistic value between RS SVM trained with distance and RS SVM trained with curvature is 0.34, it is low enough to validate the idea of combining the different approaches.

4. Conclusions

In this paper we have proposed an hand gesture recognition system based on distance and curvature features computed on the hand shape that improves both in accuracy and reliability the method of [19]. The introduced novelties are an ensemble based on two different descriptors, extracted from 3D information provided by a depth map, two different novel classification systems and a new texture based descriptor extracted from the curvature image. Our improved system has been tested, using the same parameters and the datasets of [19] obtaining very good performances, as reported in Tables 1 and 2.

We have planned several future improvements of the existing approach. They include adding new features based on the depth map and on the 2D image, testing other texture descriptors for improving the performance of the new curvature based descriptor and finally extending the proposed approach to the recognition of dynamic gestures.

References

[1] J. Demsar. Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7 (2006) 1-30.

[2] Duda RO, Hart PE, Stork D. (2000) Pattern Classification, Wiley, 2nd edition 2000.

[3] Qin ZC (2006). ROC Analysis for Predictions Made by Probabilistic Classifiers, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 5:3119-312.

[4] Ojansivu V & Heikkilä J (2008) Blur insensitive texture classification using local phase quantization. Proc. Image and Signal Processing (ICISP 2008), 5099:236-243.

[5] K. Huang, R. Murphy: Boosting accuracy of automated classification of fluorescence microscope images for location proteomics. BMC Bioinformatics 2004, 5(78).

[6] Ho T.K.: The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.

[7] Alizadeh, A; et al.. (2001) Towards a novel classification of human malignancies based on gene expression, J. Pathol., (195), 41–52, 2001.

[8] Loris Nanni, Sheryl Brahnam, Alessandra Lumini. Matrix representation in pattern classification. In Expert Systems with Applications, 39 (3): 3031-3036, 2012.

[9] Landgrebe, T. C. W., and Duin, R. P. W., “Approximating the multiclass ROC by pairwise analysis,” Pattern Recognition Letters, vol. 28, pp. 1747–1758, 2007.

[10] Rodríguez JJ, Kuncheva LI, Alonso CJ (2006) Rotation forest: a new classifier ensemble method. IEEE Trans Pattern Anal Mach Intell 28(10):1619–1630

[11] Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139

[12] He H, Cai D, Yan S, Zhang H-J (2005) Neighborhood preserving embedding, Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on Date of Conference: 17-21 Oct.

[13] L.I. Kuncheva, Whitaker C.J., Measures of Diversity in Classifier Ensembles and their Relationship with the ensemble accuracy, Machine Learning, 51, pp. 181-207, 2003

[14] Zhang, C.-X., and Zhang, J.-S., “RotBoost: a technique for combining Rotation Forest and AdaBoost” Pattern Recognition Letters, vol. 29, no. 10, pp. 1524-1536, 2008.

[15] Cristianini, N., and Shawe-Taylor, J., An introduction to support vector machines and other kernel-based learning methods, Cambridge, UK: Cambridge University Press, 2000.

[16] L. Nanni, S. Brahnam, C. Fantozzi and N. Lazzarini (2013) Heterogeneous Ensembles for the Missing Feature Problem, 2013 Annual Meeting of the Northeast Decision Sciences Institute, New York, April 2013.

[17] Nanni, L., Brahnam, S., Lumini, A., and Barrier, T., "Data mining based on intelligent systems for decision support systems in healthcare," Intelligent Support Systems in Healthcare Using Intelligent Systems and Agents, Sheryl Brahnam and Lakhmi C. Jain, eds.: Springer, 2010.

[18] F.Dominio, M.Donadeo, P.Zanuttigh (2013) Combining multiple depth-based descriptors for hand gesture recognition , PRL to appear 2014

[19] F.Dominio, M.Donadeo, G.Marin, P.Zanuttigh, G.M. Cortelazzo (2013) Hand gesture recognition with depth data, Artemis Workshop, ACM multimedia.

[20] Giulio Marin, Marco Fraccaro, Mauro Donadeo, Fabio Dominio, Pietro Zanuttigh (2013), Palm area detection for reliable hand gesture recognition, accepted to IEEE Multimedia Signal Processing Workshop (MMSP)

[21] Li, Y., June 2012. Hand gesture recognition using Kinect. In: Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on. pp. 196 -199.

[22] Pedersoli, F., Adami, N., Benini, S., Leonardi, R., Oct. 28 - Nov. 2 2012. XKin - eXtendable hand pose and gesture recognition library for Kinect. In: Proceedings of ACM Conference on Multimedia 2012 - Open Source Competition. Nara, Japan.

[23] Doliotis, P., Stefan, A., McMurrough, C., Eckhard, D., Athitsos, V., 2011. Comparing gesture recognition accuracy using color and depth information. In: Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments. PETRA '11. ACM, pp. 20:1-20:7.

[24] Zabulis, X.; Baltzakis, H. & Argyros, A., Vision-based Hand Gesture Recognition for Human Computer Interaction, 34, The Universal Access Handbook, Lawrence Erlbaum Associates, Inc. (LEA), 2009

[25] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, Time-of-Flight Cameras and Microsoft Kinect, SpringerBriefs in Electrical and Computer Engineering. Springer, 2012.

[26] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. IEEE Trans. on PAMI, 24(4):509-522, apr 2002.

[27] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. ACM Trans. on Intelligent Systems and Technology, 2:27:1-27:27, 2011.

[28] C. Keskin, F. Kirac, Y. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In ICCV Workshops, pages 1228-1234, nov. 2011.

[29] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs,W. J. Kress, I. Lopez, and J. V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In Proc. of ECCV, October 2012.

[30] A. Kurakin, Z. Zhang, and Z. Liu. A real-time system for dynamic hand gesture recognition with a depth sensor. In Proc. of EUSIPCO, 2012.

[31] S. Manay, D. Cremers, B.-W. Hong, A. Yezzi, and S. Soatto. Integral invariants for shape matching. IEEE Trans. on PAMI, 28(10):1602-1618, 2006.

[32] Z. Ren, J. Meng, and J. Yuan. Depth camera based hand gesture recognition and its applications in human-computer-interaction. In Proc. of ICICS, pages 1-5, 2011

[33] Z. Ren, J. Yuan, and Z. Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In Proc. of ACM Conference on Multimedia, pages 1093-1096. ACM, 2011.

[34] P. Suryanarayan, A. Subramanian, and D. Mandalapu. Dynamic hand pose recognition using depth data. In Proc. of ICPR, pages 3105-3108, aug. 2010

[35] J. P. Wachs, M. Kolsch, H. Stern, and Y. Edan. Vision-based hand-gesture applications. Commun. ACM, 54(2):60-71, Feb. 2011

[36] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In Proc. of ECCV, 2012.