Combination of Depth and Texture Descriptors for 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, 47521 Cesena - Italy.

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

Abstract

Automatic hand gesture recognition is a challenging problem that is attaining a growing interest due to its applications in natural interfaces for human-machine interaction, automatic sign-language recognition, computer gaming, robotics and healthcare. This chapter briefly reviews existing approaches for automatic hand gesture recognition and proposes a novel system exploiting together color and depth data. The proposed approach is based on a set of four descriptors extracted from the depth map and three texture descriptors extracted from the 2D image, while the classification is performed by an ensemble of support vector machines and decision trees. A main novelty for feature extraction is a method based on the histogram of gradients used for describing the curvature image obtained from the depth map.

Another novelty is the evaluation of different colorimetric spaces for improving the recognition performance of the texture descriptors: the best performance is obtained using the lightness band of the L*c*h* color space.

In the experimental Section the performances of different "stand-alone" descriptors are firstly compared and their correlation is analyzed for assessing their complementarity, and eventually the advantage gained by their fusion is demonstrated by the Wilcoxon Signed-Rank test.

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

1. Introduction

Automatic hand gesture recognition [19] is a challenging problem that is attaining a growing interest due to its many applications in different fields like natural interfaces for human-machine interaction, automatic sign-language recognition, computer gaming, robotics and healthcare applications. While many different approaches have been developed for hand gesture recognition from 2D color images, the use of range cameras or depth cameras for this task is a novel research field. In this chapter a gesture recognition system is presented, exploiting together color data and the 3D geometry information provided by a depth camera framing the user hand.

Until recently, most vision-based hand gesture recognition approaches were based on the analysis of images or videos framing the hand. Complete reviews of the field may be found in literature ([19], [20]). The bidimensional representation is not, however, always sufficient to capture the complex poses and inter-occlusions characterizing hand gestures.

Three dimensional representations are instead a more informative description that represent the actual shape of the hand in the framed pose. Furthermore, nowadays 3D data is easily obtainable thanks to the recent introduction of low cost consumer depth cameras. Devices such as Time-Of-Flight cameras and Microsoft's Kinect™ [21] 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 in order to exploit depth for hand gesture recognition. The general pipeline is the same for most of them, i.e. first a set of features is extracted from depth data and then various machine learning techniques are applied to the extracted features in order to recognize the performed gestures.

Kurakin et al [22] use silhouette and cell occupancy features for building a shape descriptor that is then fed into a classifier based on action graphs. Volumetric features are extracted from the hand depth and then classified by Support Vector Machines (SVM) in both the approaches of [26] and [27]. In the work of Doliotis et al. [19] the trajectory of the hand is extracted from depth data and used inside a Dynamic Time Warping (DTW) algorithm.

Randomized Decision Forests (RDFs) have also been used for the classification step in hand gesture recognition in [30] and [31]. The latter also combines together color and depth information to improve the classification results. Another key observation is that depth data allows to perform an accurate segmentation of the hand shape. For instance, [32] and [17] extract features based on the convex hull and on the fingertips positions from the silhouettes obtained in this way. A similar approach is exploited in XKin [18], an open-source hand gesture recognition software. Histograms of the distance of hand edge points from the hand center have been used in [24] and [25] by Ren et al.

Many gestures may not be recognized by only considering a static pose, and the recognition of dynamic gestures is attracting a large interest. Biswas and Basu [33] exploit the trajectory of the hand centroid in the 3D space for dynamic gesture recognition. A joint depth and color hand detector is used to extract the trajectory that is then fed to a Dynamic Time Warping (DTW) algorithm in [19]. Finally, Wan et al. [34] use the convex hull on a single frame together with the trajectory of the gesture.

In [15] two feature descriptors, one representing the distance of the fingertips from the hand centroid and the other the curvature of the hand contour computed on depth data, are used inside an SVM classifier. A more performing ensemble is proposed in [14], where two additional features were added to the approach of [15], namely, one describing the elevation of the fingers from the palm plane and the other describing the shape of the palm area.

Starting from [14][15][36] this work follows this rationale as well and exploits a hand gesture recognition scheme combining seven types of hand shape features extracted from both the depth map and the color image framing the hand. Moreover, it proposes the choice of ad hoc colorimetric space for the feature extraction from the color image and the design of a performing ensemble that takes into account the correlation among features and is based on the fusion of different classifiers.

The proposed ensemble is made of seven descriptors: four are extracted from the depth map (distance, elevation, curvature, palm area) and three are extracted from the 2D image (local phase quantization, local ternary patterns, histogram of gradients). After accurate testing it has been found that feature extraction from different colorimetric spaces is useful for improving the performance of the texture descriptors, therefore in this work the lightness band of the L*c*h* color space is used for the extraction of texture descriptors from the 2D image.

Furthermore, a novel method for representing the curvature image before using it for training a given classier is proposed based on the histogram of gradients.

The classification task is carried out by means of a heterogeneous ensemble of classifiers: a random subspace of SVM classifiers and a boosting approach based on decision trees (ROT) [12].

The reported results clearly show the usefulness to combine different descriptors; the proposed ensemble outperforms previous works in the same datasets.

The chapter 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, starts from a 2D color image and the depth map of the framed scene, and is based on the following main steps:

- *reprojection*: the depth data is projected over the 2D color image using previously computed calibration information in order to have properly aligned color and depth data in the same reference system.
- *hand segmentation*: the region corresponding to the hand is segmented from the background using both color and depth information;
- *color space transformation*: the RGB input image is converted in the L band of L*c*h* color space. This band has been selected among many others as the one which obtained best classification performance for the selected texture descriptors;
- *geometric feature extraction*: distance, elevation, curvature and palm features are extracted from the hand region of the depthmap;
- *texture feature extraction*: texture feature are extracted from the hand region of the 2D image and from a matrix representation of the curvature;
- *classification and fusion*: each descriptor is classified by an ensemble of SVM and an ensemble of boosted decision trees, then these sets of classifiers are combined by weighted sum rule [2].

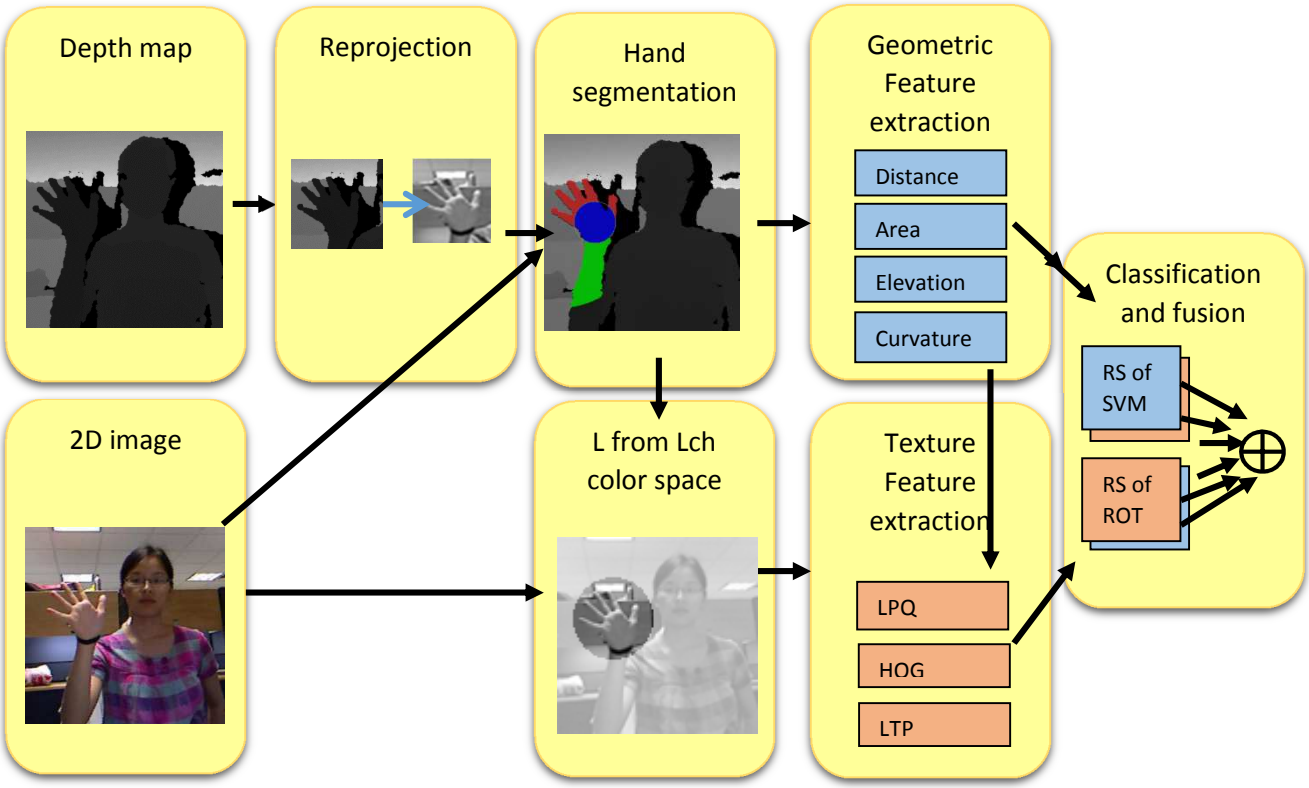


Figure 1. Overview of the proposed approach.

2.1 Reprojection

The depth data acquired by the Kinect is firstly projected on color image for the correct alignment of color and depth data. In order to perform this task it is necessary to compute the positions of the depth samples in the 3D space and then to project them back to the 2D color image reference system. This requires the calibration data for the depth and color cameras of the Kinect that have been previously computed by using the calibration method proposed in [35]. By the end of such alignment a color and a depth value are associated to each sample.

2.2 Hand segmentation and palm recognition

The first step is the extraction of the depth samples corresponding to the hand region from the depth map. For this purpose, the approach introduced in [14] and [15] is used and here briefly resumed.

The proposed method starts by extracting from the acquired depth map the closest point to the camera, that will be denoted with X_{min} (see Figure 2b).The algorithm automatically avoids to select as X_{min} an isolated artifact by verifying that the selected point has close samples with a similar depth according to the approach described in [15]. After the selection of X_{min}, the set H of all the points X_i with a depth value $D(X_i)$ included in the range $[X_{min}, X_{min} + T_h]$ and with Euclidean distance from X_{min} in 3D space smaller than a threshold T_{h2} is computed:

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

T_h and T_{h2} are suitable thresholds whose values depend on the user's hand size (in the experimental results T_h= 10cm and T_{h2} = 30cm).

In the next step the color of the samples is checked in order to verify if it is compatible with the skin color. Finally, the detected hand size must be compatible with the hand's one [16]. This approach allows to reliably segment the hand from the other scene objects and body parts (as shown in Figure 2b). However a drawback of this scheme is that some parts of the wrist and of the forearm may be included in the extracted region.

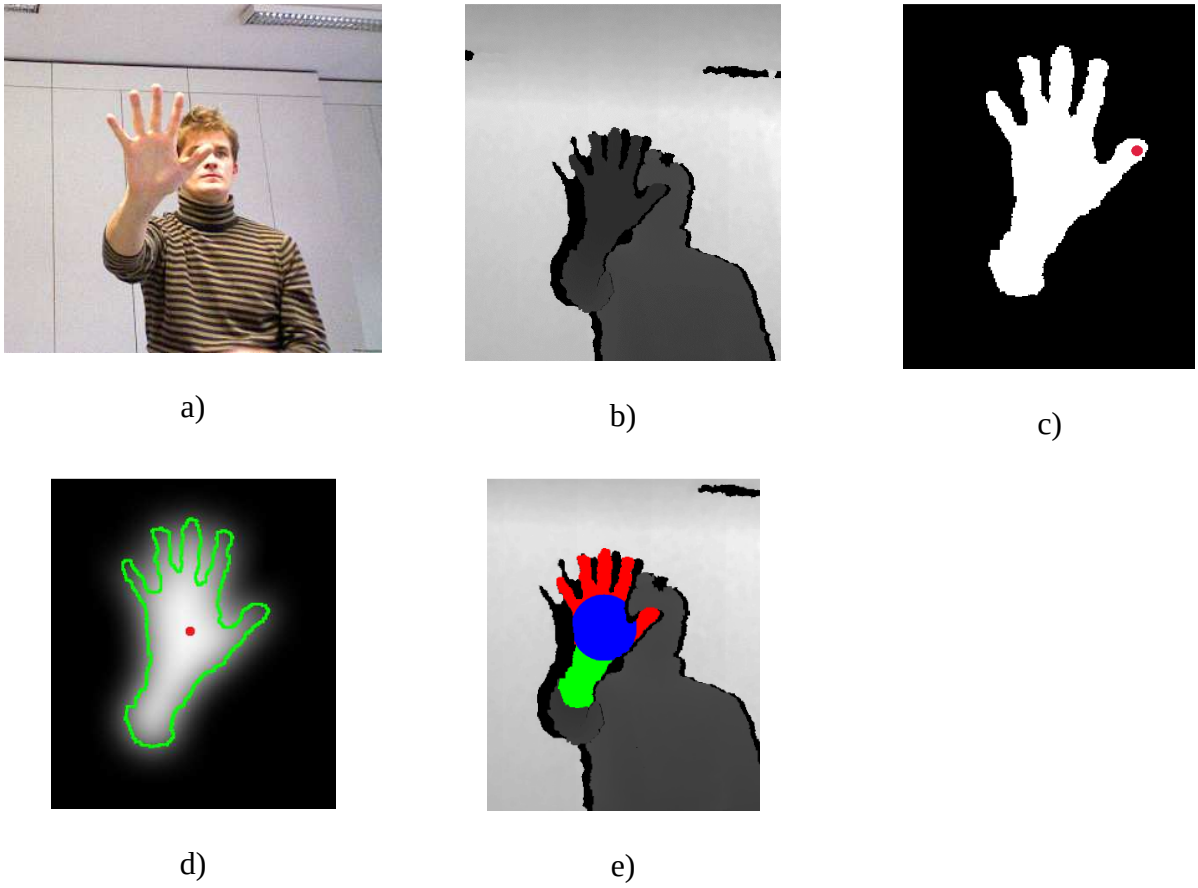


Figure 2. Extraction of the hand: a) RGB image; b) acquired depth image; c) computed hand mask (the red point correspond to X_{min}); d) blurred depth mask with C dotted in red; e) segmented hand regions. *(best viewed in colors, some images have been cropped to highlight the areas of interest)*

A 2D mask corresponding to the hand samples in the depth image space is then built and the related binary image is filtered by a low pass Gaussian filter with a large standard deviation. The value of the standard deviation is adaptive and varies with the distance of the hand from the Kinect, as described in [15].

The maximum of the filtered image, which is the starting point of the next step, is now detected. Since the filter support is larger than the hand and the palm is larger than the forearm and denser than the finger region, the computed maximum typically 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 (denoted with *C*), that can be fitted on the palm region, as described in [15]. A more refined version of this procedure [16] uses an ellipse in place of the circle in order to better approximate the shape of the palm, especially when the hand is not perpendicular with the optical axis of the depth camera.

The samples inside the circle (or inside the ellipse) belong to the palm. A plane in 3D space is fitted on them by using a robust estimator exploiting Singular Value Decomposition and RANSAC. The axis that roughly corresponds to the direction of the vector going from the wrist to the fingertips is then estimated by applying Principal Component Analysis (PCA) to the hand samples. This is a rough estimation that gives only a general idea of the hand estimation but it is a good starting point for the feature extraction algorithm.

On the basis of the computed data, the set *H* is subdivided into three sets as shown in Figure 2d:

- The palm points *P*
- The wrist points *W* (this set contains both wrist and part of the forearm and will be discarded).
- The finger points *F*

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

2.3 Color space transformation

Several colorimetric spaces have been evaluated for improving the performance of the texture descriptors. In particular the best performance is obtained using the lightness band of the L*c*h* color space. The CIE L*c*h* color space (Figure 3) is a device independent color model which is essentially in the form of a sphere with three axes.

The three components represent:

- *Lightness* (a vertical axis from 0 to 100, i.e., absolute black to absolute white)
- *Chroma* (a horizontal axes from 0 at the center of the circle to 100, i.e., from neutral grey, black or white, to “color purity”)
- *Hue* (a circular expressed in degrees from 0° to 360°, representing different colors 0°=red, 90°=yellow, 180°=green, 270°= blue).

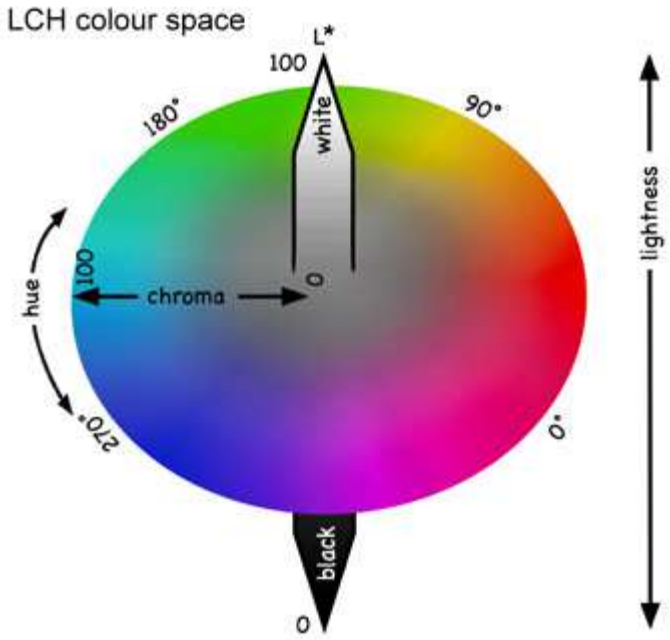


Figure 3. L*c*h* color space *(best viewed in colors)*

2.4 Geometric Feature Extraction

2.4.1 Distance features

Distance features, introduced in [15] on the basis of a previous idea from [24], represent the distance of the finger samples in *F* from the hand centroid *C*.

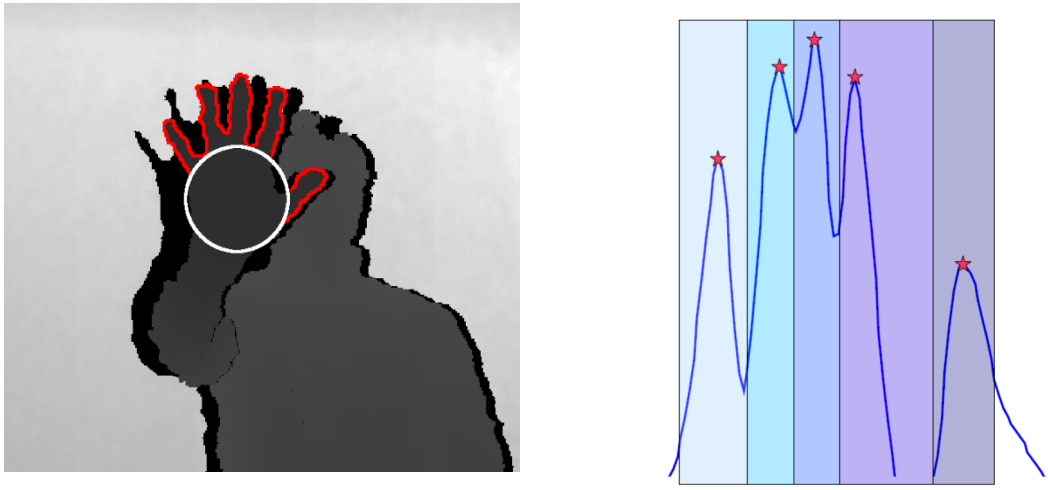
For each sample X_i in *F* the Euclidean distance $d(X_i)$ in 3D space from the centroid is computed. The various samples are sorted on the basis of the angle $\theta(X_i)$ between the projection on the palm plane of the PCA principal axis and of the vector connecting each point X_i to the centroid. A histogram representing the maximum of the distance from the centroid for each angular direction is then built:

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

Where Δ is the quantization step for the histogram computation (Δ=2° has been used). For each gesture *g* in the database, a reference histogram L_g^r is built. A set of angular regions corresponding to the direction of the various fingers that are used in each gesture is then defined on this histogram (see Fig.4b). These regions correspond to the position of 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 computed histograms $L(\theta)$ with the template on which the regions are defined. For this purpose, the maximum of the correlation between the acquired histogram and the translated version of the reference histogram of each gesture is computed. The computation is also performed with the flipped version of the histogram in order to account for the fact that the hand could have either the palm or the dorsum facing the camera, and that both the left and the right hand could have been used. The maximum between the two cases is selected and the corresponding translation gives the translational shift required to align the acquired histogram with the reference one (together with the flipping if it was selected). 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 align the reference and the computed histograms. In this way, the regions corresponding to the various features of the gesture are precisely defined. Fig. 5 shows some examples

of the computed histograms for three different gestures. The plots clearly show the different fingers arrangements in the various gestures.



a) b)

Figure 4. Histogram of the 3D distances of the edge samples from *C*. The colored areas are the features regions: a) finger edges computed from *F* ; b) corresponding histogram L(θ) with the regions corresponding to the different features.

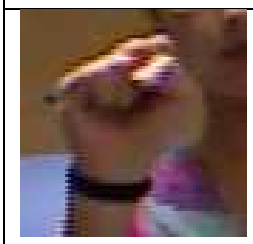
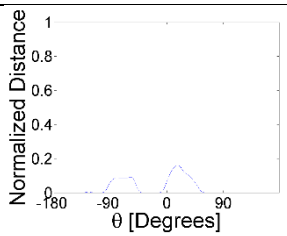
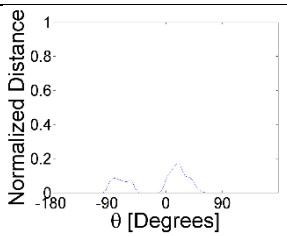
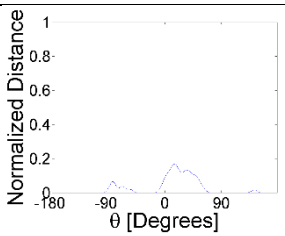
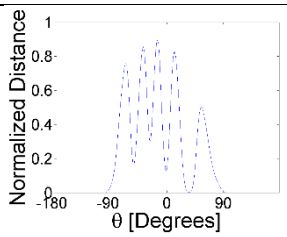
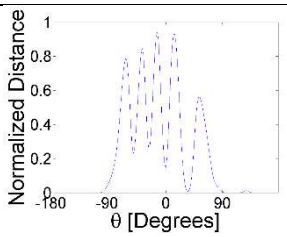
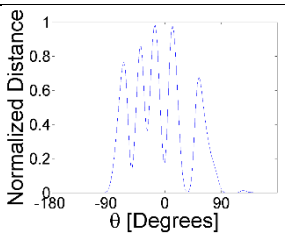
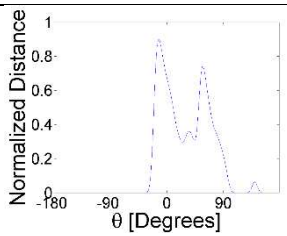
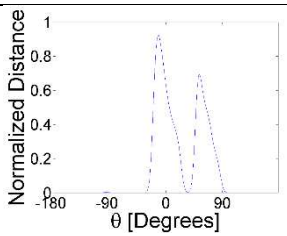
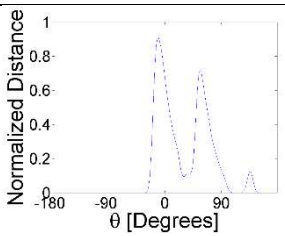
Sample image of the gesture	Distance histograms corresponding to various repetitions of the gesture		
			
			
			

Figure 5. Examples of distance histogram for some sample frames from different gestures.

The distance features set F^l contains a feature value for each finger j in each gesture $g \in 1,..,G$. The number of raised fingers is different in each gesture, and so also the number of feature values is different. The feature value $f_{g,j}^{l}$ 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 Fig. 4b), i.e, :

$$f_{g,j}^{l} = \frac{\max_{\theta_{g,j}^{min} < \theta < \theta_{g,j}^{max}} L_a^g(\theta) - r_f}{L_{max}}$$

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 $\boldsymbol{C_f}$ to the ellipse border). The length L_{max} of the middle finger is used to normalize with respect to the hand size in order to make the approach independent from the size of the hands of different people. The radius r_f need to be subtracted from all the features for avoiding the jump from 0 to r_f of the values when the edge crosses the circle border.

In this way up to G*5 features are built for each acquired sample (the actual number is smaller since not all the fingers are of interest in all the gestures). For instance, the dataset used in the experimental results taken from the work of Ren et al. [25] contains 10 different gestures and 24 features have been used, about a half of the 50 features that there would be if all the fingers were used in all the gestures.

2.2.2 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 [23] and exploits the hand edge points E and the mask M_h representing the hand samples in the depth map (Figure 2b).

For each point X_i in E 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 is built. 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. Differently from [23] 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 $V(X_i, s)$ characterize the curvature of the region around sample X_i. The minimum value $V(X_i, s) = 0$ corresponds to an extremely convex shape, $V(X_i, s) = 0.5$ to a straight edge and the maximum $V(X_i, s) = 1$ to an extremely concave shape. The [0, 1] interval is 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:

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

where *b* is the bin index. Curvature features are given by the cardinalities of the sets $|V_{b,s}|$ for each bin *b* and radius value *s* normalized with respect to the hand contour length, i.e.:

$$f_{b,s}^{c} = \frac{|V_{b,s}|}{|E|}$$

In this way a feature vector F^c containing *B*S* features is built. As expected, the value of the different curvature features depends on the positions of the 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 with color maps, is reported in Fig. 6.

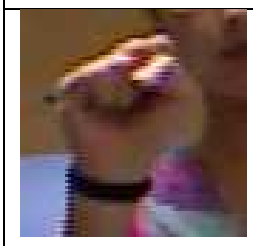
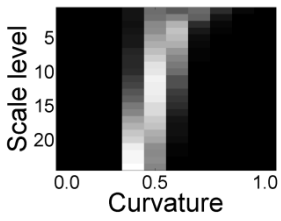
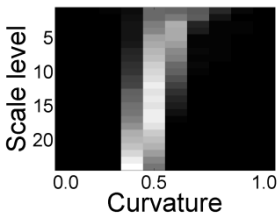
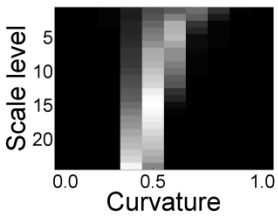
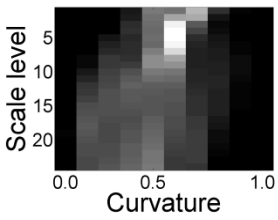
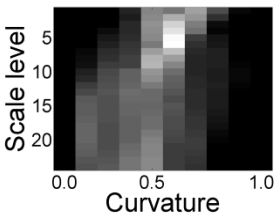
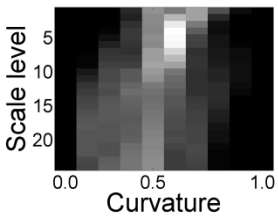
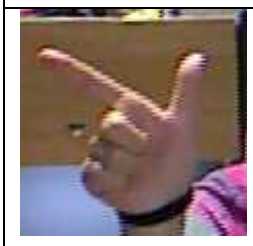
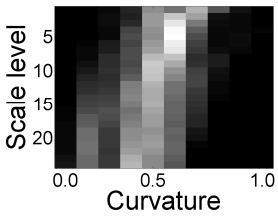
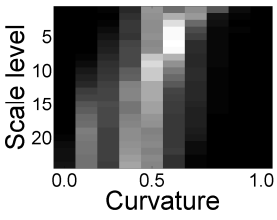
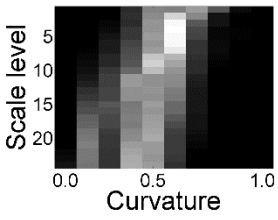
Sample image of the gesture	Curvature descriptors corresponding to various repetitions of the gesture		
			
			
			

Figure 6. Examples of curvature descriptors for some sample frames from different gestures.

2.2.3 Elevation features

The construction of the elevation features is based on the same scheme employed for the distance features in Section 2.2.1.: first an histogram representing the distance of each sample in $\mathcal{F}$ from the palm plane $\boldsymbol{\pi}$ is built, namely, for each sample $\mathbf{X_j}$ in $\mathcal{F}$ its distance from plane $\boldsymbol{\pi}$ is computed:

$$\mathrm{A} = \mathrm{e}_{\mathrm{X_j}} = \mathrm{sgn}((\boldsymbol{X_j} - \boldsymbol{X_j^{\pi}}) \cdot \boldsymbol{i_z^{\pi}}) \, | \boldsymbol{X_j} - \boldsymbol{X_j^{\pi}} |, \qquad X_j \in \mathcal{F}$$

where $\mathbf{X_j^{\pi}}$ is the projection of x_j on π. The sign of $\mathrm{e}_{\mathrm{X_j}}$ accounts for the fact that $\mathbf{X_j}$ can belong to any of the two hemi-spaces defined by $\boldsymbol{\pi}$, i.e., $\mathbf{X_j}$ can either be on the front or behind $\boldsymbol{\pi}$.

Then, following the scheme used for distance features, for each angular sector (represented by a quantized value Θ_q) the point with the greatest absolute distance from the plane is selected, thus producing an histogram E(Θ):

$$E(\Theta_q) = \begin{cases} \max\limits_{I(\Theta_q)} e_{X_j} & \left|\max\limits_{I(\Theta_q)} e_{X_j}\right| > \left|\min\limits_{I(\Theta_q)} e_{X_j}\right| \\ \min\limits_{I(\Theta_q)} e_{X_j} & otherwise \end{cases}$$

The quantization uses the same intervals used for distance feature in Section 2.2.1. The histogram E(Θ) corresponding to the performed gesture is then aligned to the various reference gestures in *G* using the alignment information already computed in Section 2.2.1, and it is subdivided in different regions corresponding to the various fingers as done for the distance features. Let $\mathrm{E_g}(\Theta)$ be histogram E(Θ) aligned with the $\mathrm{g^{th}}$ gesture template. The elevation features are then computed according to:

$$\mathrm{f}_{g,j}^{e} = \begin{cases} \frac{1}{L_{max}} \max\limits_{I(\Theta_{g,j})} E^g(\Theta) & \left|\max\limits_{I(\Theta_{g,j})} E^g(\Theta)\right| > \left|\min\limits_{I(\Theta_{g,j})} E^g(\Theta)\right| \\ \frac{1}{L_{max}} \min\limits_{I(\Theta_{g,j})} E^g(\Theta) & otherwise \end{cases}$$

Where the intervals $I(\Theta_{g,j})$ are again the same used for distance features. Note that the alignments computed in Section 2.2.1 are used here both to save computation time and because the correlations from distance data are more reliable than the ones computed on elevation information. Finally note that the vector $\mathbf{F}^{\boldsymbol{e}}$ of the elevation features has the same structure and number of elements of the vector $\mathbf{F}^{\boldsymbol{l}}$ of the distance features.

2.2.4 Palm area features

The last set of features describes the shape of the palm region *P*. Note that *P* corresponds to the palm area, but it may also include finger samples when the fingers are folded over the palm. The palm region is partitioned into six different areas, defined over the plane $\boldsymbol{\pi}$(see Fig. 6). The circle or ellipse defining the palm area is firstly divided into two parts: the lower half is used as a reference for the palm position since it is not occluded by the fingers in most gestures, and a 3D plane $\boldsymbol{\pi_p}$ is fitted on this region.

The upper half is divided into 5 regions A_j, j = 1, .., 5 roughly corresponding to the regions close to the different fingers as shown in Fig. 7, i.e., each region corresponds to the area of the palm that is affected by the position of the associated finger and where the finger can potentially fold.

The various area features account for the deformation the palm shape undergoes in the corresponding area when the related finger is folded or is moved. In particular, notice how the samples corresponding to the fingers folded over the palm are associated to *P* and are not captured by distance or elevation features, but they are used for the computation of palm area features.

The areas positions on the plane depend on the following parameters:

- The palm area, represented by the center $\mathbf{C_f}$ and the radius $\mathbf{r_f}$ of the circle (or by the two axes of the ellipse if this representation is used)
- The widths of the various fingers. A standard subdivision of the upper half of the circle has been used for the experimental results, but it can also be optimized on the basis of the specific user's hand.
- The direction i_x^{π} corresponding to $\Theta = 0$.

Since the center $\mathbf{C_f}$ and radius $\mathbf{r_f}$ or axes have already been computed in Section 2.1, the only missing element is the alignment of the Θ directions. Again, the correlation of the distance histograms computed in Section 2.2.1 is used to align the regions template with the hand direction $\mathbf{i}_x^{\boldsymbol{\pi}}$. The templates are also scaled by r_f (or scaled and stretched according to the two axes of the ellipse).

The templates are then aligned with the acquired data using the alignment information computed in the previous steps. In this way an area feature set is extracted for each candidate gesture. The areas aligned with the template of each gesture will be denoted with A_j^g, where g indicates the corresponding gesture. The set of points $X_i \in P$ associated to each of the regions A_j^g is then computed. Finally, the distance between each sample $X_i \in A_j^g$ and $\boldsymbol{\pi_p}$ is calculated for each region A_j^g.

The feature corresponding to the area A_j^g is the average of the distances of the samples belonging to the area from plane $\boldsymbol{\pi_p}$:

$$f_{g,j}^a = \frac{\sum_{X_i \in A_j^g} \| X_i - X_i^\pi \|}{\left| A_j^g \right|}$$

The area features are collected in a vector $\mathbf{F^a}$, made by G x 5 area features, one for each finger in each possible gesture, following the same rationale used for $\mathbf{F^l}$ and $\mathbf{F^e}$. The entries of $\mathbf{F^a}$ are finally normalized in order to assume values within range [0, 1] as the other feature vectors.

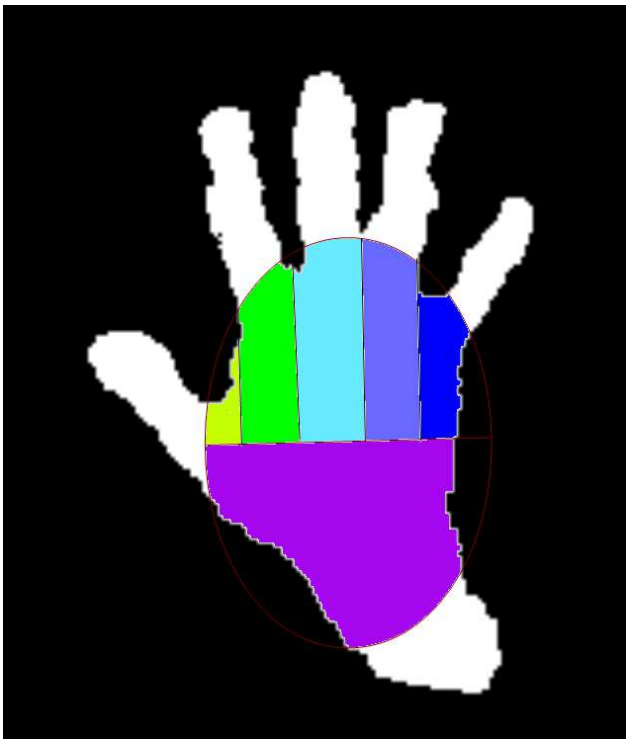


Figure 7. Regions used for the computation of the palm area features.

2.5 Texture feature extraction

In this work three texture descriptors are extracted both from the 2D image and from a matrix representation of the curvature. Three well-known descriptors are used: Local Phase Quantization (LPQ) [3], Local Ternary Patterns (LTP) [28] and Histogram of Gradients (HoG) [29].

2.5.1 Extracting texture feature from curvature

Since curvature is very significant for hand representation, in this work two different types of descriptors are extracted from curvature information: the standard geometrical measures of curvature (explained in section 2.2.2) and other numerical descriptors designed to take into account local variation. Following the approach proposed in [5], a matrix representation is used for curvature data, obtained by simply rearranging the linear feature vector as a matrix: in this way, relevant information can be extracted from the Curvature using well-known textural descriptors, which have the ability to well represent shape variations due to gestures. In order to make results independent from the evaluation order, 50 different random reshapings are used (see figure 8) to rearrange the curvature vector as a matrix and then texture descriptors are extracted from each resulting matrix. Since local texture features measure local variations, they are suited to measure the discriminative information present in the local neighborhoods of each pixel (i.e., the curvature value in this case). The use of different reshaping arrangements makes possible to observe and encode different aspects of curvature variations from a single curvature vector. Each descriptor extracted from a reshaped matrix is used to train a SVM classifier to be combined in an ensemble using the sum rule. Due to computational issue, for this descriptor, the ensemble of decision trees is not used.

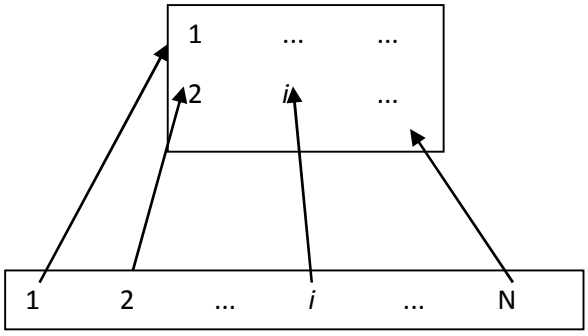


Figure 8. Reshaping a vector into a matrix.

2.5.2 Local phase quantization

Local Phase Quantization (LPQ) [3] is a texture descriptor based on quantizing the Fourier transform phase in local neighborhoods of a point. Histograms of LPQ labels computed within local regions are used as texture descriptor similarly to the well-known Local Binary Patterns [37].The extraction of LPQ is performed using a short-term Fourier transform on local neighborhoods at each pixel; this transform is efficiently computed for the whole image by a 1-D convolution for the rows and columns successively. Then, only four complex low-frequency coefficients are considered and quantized by observing the signs of the real and imaginary parts of each component. In this work the final descriptor[1] is obtained by the concatenation of histograms obtained with two settings of the parameter radius R which denotes the neighborhood size (R = 3, R = 5).

[1] The MATLAB code for LPQ is freely available at http://www.cse.oulu.fi/CMV/Downloads/LPQMatlab

2.5.3 Histogram of gradients

The histogram of oriented gradients (HoG) [37] is based on the idea that local shapes can be characterized rather well by the distribution of local intensity gradients. HoG descriptor is extracted by dividing the image into small cells and calculating a local 9-bin equi-width histogram of gradient directions (discretized in 9 bins) over the cells. For better invariance to illumination and shadows histograms are contrast-normalized within a larger region (blocks of cells) and their combination is the final descriptor.

2.5.4 Local ternary patterns

Local binary pattern (LBP) [37] is a widely used texture descriptor based the encoding of the pixel differences between the neighboring pixels and the center pixel in a local region of an image. Due to its sensitivity to noise, several variants have been proposed [38], including the Local Ternary Pattern (LTP) [28] which encode the pixel difference between the center pixel $\mathbf{p_c}$ and the neighboring pixels $\mathbf{p_n}$ using a ternary code, according to a threshold τ: 1 if $\mathbf{p_n} \geq \mathbf{p_c} + \tau$; –1 if $\mathbf{p_n} \leq \mathbf{p_c} - t$; else 0. LTP is less sensitive to noise as the small pixel difference is encoded into a separate state. To reduce the dimensionality, the ternary code is split into two binary codes: a positive LBP and a negative LBP. The final LPT descriptor is the concatenation of the histograms computed from positive and negative LBP. In this work a multi-resolution version of LTP is obtained by the concatenation of descriptors evaluated at different neighborhood sizes: (P = 8; R = 1) and (P = 16; R = 2). Two implementation of LTP are tested: LTPu, where the uniform bins are considered, and LTPri, where rotation invariant bins are considered. The interested reader can see [37] for more details on uniform and rotation invariant bins.

2.6 Classification

2.6.1 Random Subspace Ensemble of Support Vector Machines

Due to the high dimensionality of the descriptors and the low cardinality of the sample dataset, automatic hand gesture recognition is a difficult classification task. In order to deal with this "dimensionality curse" problem a random subspace (RS) ensemble [4] is used for classification, since PS has proven to be effective in these cases. RS is a valid approach for designing ensembles based on the perturbation of features: each classifier is trained on a training set obtained by reducing the dimensionality of the data by randomly subsampling the features.

Given a collection of *m* training samples $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iN})$, $\mathbf{x}_i \in \Re^N$, RS randomly selects $K < N$ features from the original feature space and creates a new training set by projecting each sample into $\Re^K$. This procedure is repeated *L* times where *L* is the number of final classifiers combined by the sum rule to obtain the final decision. In this work the two RS parameters are fixed $L = N / 2$, $M = 50$ and support vector machines (SVM) [11] from LibSVM toolbox[2] are used as classifiers.

As already shown in [36] the RS ensemble of SVM outperforms the standard SVM classifier, therefore stand-alone SVM has not been tested in this work.

2.6.2 Random Subspace Ensemble of RotBoost with NPE (RSR)

In order to exploit the diversity of classifiers another ensemble is tested: a variant [12][13] of Rotation Boosting (RotBoost) [7][10] coupled with the Neighborhood Preserving Embedding (NPE) [8], which is a dimensionality reduction method.

RotBoost is designed as the integration of AdaBoost [11] and Rotation Forest [7], two ensemble generation techniques that apply a learning algorithm to a set of permutated training sets. 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 training set by randomly split into *S* subsets the original feature space and reducing its dimensionality by applying Principal Component Analysis (PCA).

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

A variant of RotBoost[3] [12][13] is used in this work, obtained by coupling the ensemble with RS and by using the neighborhood preserving embedding (NPE) feature transform instead of PCA for dimensionality reduction. First dimensionality reduction by NPE 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 a RS to the AdaBoost technique.

Neighborhood Preserving Embedding (NPE)[4] [8] is a technique for dimensionality reduction which aims at preserving the local neighborhood structure on data manifold; it has proven to be more effective than PCA in discovering the underlying nonlinear structure of the data and less sensitive to outliers than other feature transform. NPE starts by building a weight matrix to describe the relationships between samples: each sample 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. It is useful to highlight several aspects of NPE (see [8] for more details):

- NPE is linear (it is a linear approximation of Locally Linear Embedding) so it is fast and suitable for real-time applications;
- NPE can be performed in either supervised or unsupervised mode. When the labels of the training patters are available they can be used for building a better weight matrix.

NPE procedure is based on three step:

- constructing an adjacency graph, using a *K* nearest neighbors method;
- Computing the weights of the edge between the nodes of the graph;
- Computing the Projections: a linear projections is computed

[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.

3. Experimental Results

In this section the experiments performed on two different datasets are discussed. The first [24], named REN, contains 10 different gestures performed by 10 different people (Fig. 9) and acquired with Microsoft's Kinect. Each gesture is repeated 10 times for a total of 1000 different samples.

The second dataset [14], named SELF, has also been acquired with the Kinect, and is a self-collected dataset which contains 12 different gestures (a small subset of the American Sign Language gestures) performed by 14 different people (Fig. 10). Each gesture is repeated 10 times for a total of 1680 samples (depth maps and the corresponding color images).

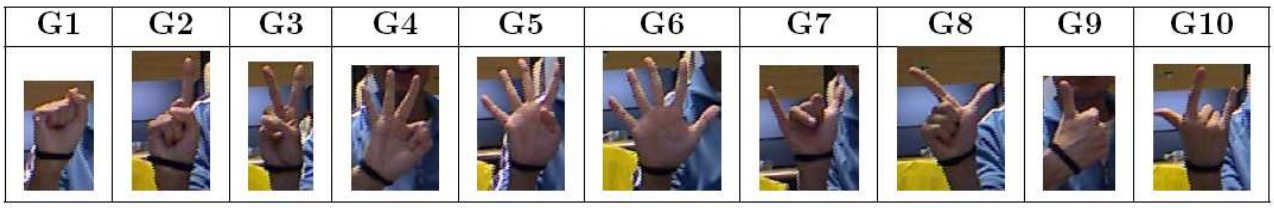


Figure 9. Sample color images of the 10 different gestures in REN.

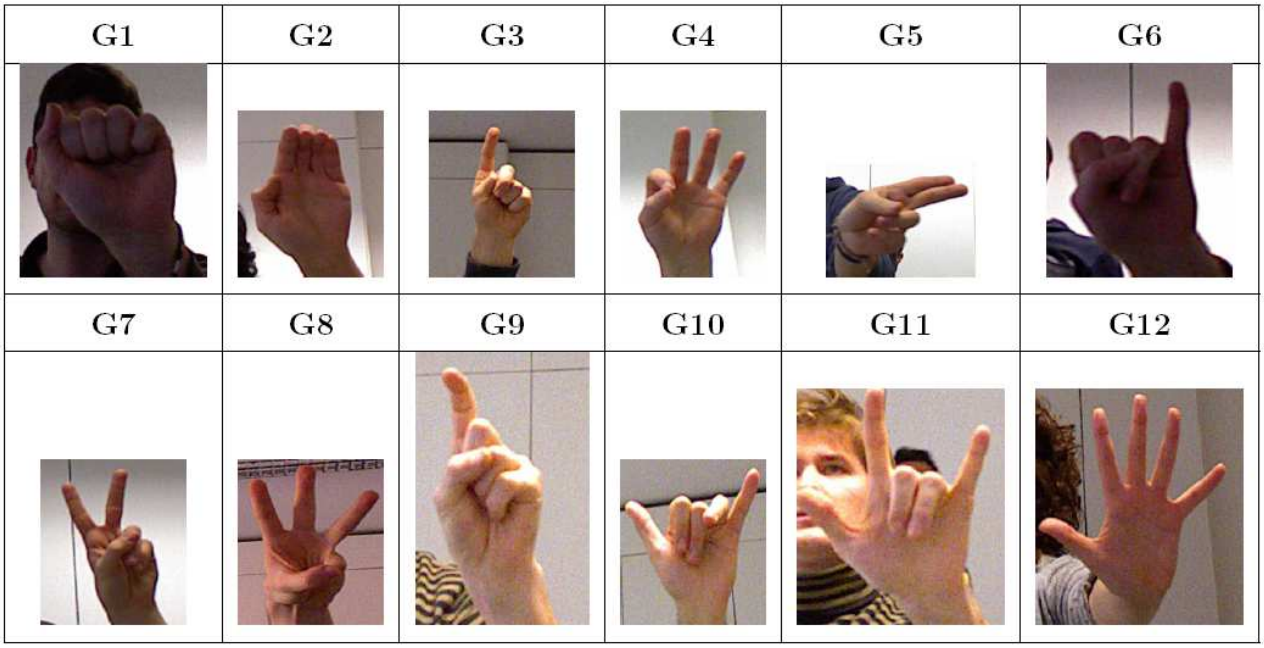


Figure 10. Sample color images of the 12 different gestures in SELF.

The experiments have been carried out according the “the leave-one-out user” in both the datasets: the samples from *N-1* users are used for the training set and the samples from the remaining user for the test set. Then the obtained results are averaged among all the *N* experiments obtained by varying the user in the test set.

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 plots the fraction of true positive rate vs. the false positives rate at various threshold settings. The AUC may 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 [6] and the reported EUC value is obtained by averaging all class values.

The first experiment aimed at selecting the most appropriate color space for performing texture feature extraction (using a stand-along SVM classifier). In this experiments an exhaustive search among 14 different color spaces is performed in order to find out the most appropriate space for this case study. In Table 1 the accuracies obtained by the three texture descriptors presented in Section 2.5 are evaluated as a function of the input color band (only the best five color bands are reported, considering the *HOG* descriptor). As stated in Section 2.3 the best choice is the band L of L*c*h*. From the reported results it is clear that the *L* bands work well for all the descriptors.

	Ren			*Self*		
Color Band	*LTP*	*LPQ*	*HOG*	*LTP*	*LPQ*	*HOG*
Gray Values from RGB	70.3%	77.0%	94.1%	60.7%	66.7%	85.2%
*L of L*c*h**	71.6%	77.4%	95.3%	63.9%	67.3%	87.0%
L of Luv	71.6%	77.4%	95.3%	63.9%	67.3%	87.0%
L of Lab	71.6%	77.4%	95.3%	63.9%	67.3%	87.0%
S of HSV	26.2%	70.4%	92.6%	17.0%	55.4%	86.8%
Cr of YCbCr	55.2%	57.1%	93.5%	49.6%	57.6%	85.8%

Table 1. Comparison among different color bands in terms of accuracy.

The second experiment aimed at validating the use of texture descriptors on the curvature extracted from the 2D image. In table 2 the performance obtained by texture descriptors coupled with the matrix representation of the curvature are reported (LTP is not reported since it is not suited for representing directly the curvature image which is not a grey level image).

The classification task is performed by means of an ensemble SVM classifiers, which combines results obtained according to different rearrangements of curvature data (see section 2.5.1).

		ACC		EUC	
Classification Approach	**Feature set**	*REN*	*SELF*	*REN*	*SELF*
Ensemble of SVMs	Curvature	92.4	82.7	0.5	2.0
	LPQ	91.0	80.9	0.8	1.8
	HOG	94.7	84.6	0.4	1.4
	HOG + Curvature	94.5	86.2	0.4	1.4
	2× HOG + Curvature	94.9	86.5	0.4	1.3
	3× HOG + Curvature	94.8	86.0	0.4	1.3
	4× HOG + Curvature	94.8	85.6	0.4	1.3

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

The best results among those reported in Table 2 (and all other combination of weights tested) are obtained coupling standard approach for representing curvature and 2×HOG; this ensemble is named CurvTexture in the following.

The third experiment aimed at comparing the performance of different descriptors presented in Section 2 coupled with two classification approaches: a random subspace ensemble of SVM classifiers, and a random subspace ensemble of rotation boosting. Moreover, the fusion of the two heterogeneous ensembles is reported (HET). In Table 3 the performance in terms of accuracy (ACC) and EUC are reported for the above cited approaches on both the datasets (*REN* and *SELF*) and for the following ensembles:

- CurvTexture, the weighted sum rule between HOG extracted from the curvature image and Curvature (see the second experiment).
- 2DTexure, weighted sum rule of 4×HOG+LTP+LPQ (evaluated on the color image).

		ACC		EUC	
Classification Approach	**Descriptors**	*REN*	*SELF*	*REN*	*SELF*
RS SVM	Distance	86.9	57.2	1.1	7.1
	Curvature	92.4	84.0	0.5	1.8
	Palm	60.9	45.3	9.2	17.6
	Elevation	60.5	46.2	8.1	11.8
	CurvTexture	94.9	86.5	0.4	1.3
	2DTexure	95.5	88.1	0.4	1.2
RS ROT	Distance	88.8	60.5	0.9	5.7
	Curvature	93.9	84.9	0.4	1.3
	Palm	64.0	48.2	7.7	11.4
	Elevation	61.1	48.7	7.8	9.4
	2DTexure	94.7	88.0	0.6	1.0
HET	Distance	89.0	60.1	0.9	5.5
	Curvature	94.6	86.2	0.3	1.3
	Palm	63.6	48.0	7.6	11.7
	Elevation	61.5	49.2	7.1	9.5
	2DTexure	**95.3**	**89.3**	**0.4**	**0.9**

Table 3. Comparison among classification methods and descriptors studied in this work.

The results reported in table 3 clearly show that RS ROT works well in this problem, although the fusion between RS ROT and RS SVM outperforms both the single approaches. As stated above, in order to avoid a large table the results obtained using stand-alone SVM are not reported, since it is outperformed by RS-SVM, as already shown in [36].

In Table 4 the performance of the complete approach obtained as the weighed fusion of different descriptors are reported and compared with other works from the literature. F1 and F2 denote different fusion weights:

- F1=2×Distance + 4×Curvature + Palm + Elevation, is a fusion of the only geometric descriptors;
- F2= 2×Distance +4×Curvature + Palm + Elevation+4×HOG+LTP+LPQ, involves both geometric and texture descriptors.

		ACC		EUC	
Approach	**Classification Approach**	*REN*	*SELF*	*REN*	*SELF*
[15]	-	97.2	87.1	0.3	1.8
[14]	-	97.0	93.5	---	---
[36]	-	97.9	88.7	0.1	0.9
This work	RS SVM (F1)	98.2	89.8	0.2	0.9
	RS SVM (F2)	99.3	95.0	0.1	0.5
	RS ROT (F1)	98.9	91.4	0.1	0.6
	RS ROT (F2)	99.8	94.6	0.1	0.4
	HET (F1)	99.1	92.6	0.1	0.6
	HET (F2)	**99.9**	**96.0**	**0.1**	**0.3**

Table 4. Comparison among the approaches studied in this work and the literature.

The results reported in Table 4 clearly show that the proposed weighted fusion of the classifiers, trained using different descriptors, greatly improves the performance reported in the literature. The

proposed approach has been compared with three of our works, e.g. [14], [15] and [36]. The approach presented in [15] uses two different geometry feature set, i.e., distance and curvature features. A more performing approach is presented in a newer work [14], that uses all the 4 different geometric feature descriptors presented in this paper and a more refined version of the hand recognition and feature extraction scheme. The proposed scheme has also been compared with [36], where a more advanced SVM classifier has been employed. However, notice that the hand recognition and feature extraction scheme of this work and of [36] are based on [15], that presents a simpler version of the approach with respect to the one proposed in [14]. In particular, notice that the improved hand recognition and feature extraction scheme of [14] greatly improve the performance in the SELF dataset but slightly reduce the performance in the REN dataset.

Another difference with two of our previous work is that in [14] and [15] 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, since both the datasets are small, and all the images of a given dataset are collected in the same laboratory (one in Padua and the other in US).

Finally, for a statistical validation of the experiments the Wilcoxon signed rank test [1] is used, obtaining that the ensemble HET (F2) outperforms with a p-value of 0.05 all the previous methods both in the REN and SELF datasets.

Moreover, the relationship among the different descriptors according to the Q-statistic [9] has been analyzed in order to evaluate their error independence (highest independence is obtained when Q-statistic is 0). Table 5 reports the Q-statistic among several couples of descriptors tested in this work using RS-SVM as classifier. The results in Table 5 show a high independence among several couples

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

of descriptors. Note how it is likely to discover independence among couples of weak classifiers, while it is rarer to find it on strong methods. From Table 5 it is evident the high independence between Curvature and 2DTexture which are very performing descriptors (this is the reason of the good accuracy of their fusion).

Descriptors	Distance	Curvature	Palm	Elevation	2DTexure
Distance	---	0.34	0.29	0.35	0.25
Curvature	---	---	0.32	0.27	0.37
Palm	---	---	---	0.21	0.22
Elevation	---	---	---	---	0.32
2DTexure	---	---	---	---	---

Table 5. Q-statistic among the different descriptors.

4. Conclusions

In this chapter a hand gesture recognition system is proposed based on 7 different set of features computed on the hand shape and color that improve both in accuracy and reliability the methods of [14][15][36]. The main novelties here introduced are: an ensemble based on different descriptors, extracted from both the 3D information provided by a depth map and the color data; a new texture based descriptor extracted from the curvature image that improves the similar approach proposed in [36]. As in [36] two different classification systems have been used for improving the performance. The proposed system has been tested using the same datasets used in [14][15][36] obtaining very good performances outperforming previous works, as reported in Tables 3 and 4.

Several future works have been planned for a further performance improvement:

- new features based on the depth map and the inclusion into the proposed system of the more refined feature extraction scheme used in [14];
- new features based on texture descriptors, in particular the bag of words approach will be studied [39];
- 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] Ojansivu V & Heikkilä J (2008) Blur insensitive texture classification using local phase quantization. Proc. Image and Signal Processing (ICISP 2008), 5099:236-243.

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

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

[6] 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.

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

[8] 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.

[9] 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

[10] 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.

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

[12] 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.

[13] 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.

[14] F.Dominio, M.Donadeo, P.Zanuttigh, Combining multiple depth-based descriptors for hand gesture recognition, Pattern Recognition Letters, (accepted for publication), available online 24 October 2013

[15] F.Dominio, M.Donadeo, G.Marin, P.Zanuttigh, G.M. Cortelazzo, Hand gesture recognition with depth data, ACM multimedia Artemis Workshop, Barcelona, Spain, Oct. 2013

[16] G. Marin, M. Fraccaro, M. Donadeo, F. Dominio, P. Zanuttigh, Palm area detection for reliable hand gesture recognition, IEEE Multimedia Signal Processing Workshop (MMSP), Pula, Italy, Oct. 2013

[17] 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.

[18] 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.

[19] 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.

[20] 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

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

[22] 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.

[23] 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.

[24] 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

[25] 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.

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

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

[28] Xiaoyang Tan and Bill Triggs, Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions, IEEE Transactions on Image Processing, 19(6), pp. 1635-1650, 2010

[29] N. Dalal and B. Triggs (2005) (In CVPR'05). . An effective pedestrian detector based on evaluating histograms of oriented image gradients in a grid.

[30] Keskin, C., Kraç, F., Kara, Y.E., Akarun, L., 2012. Hand pose estimation and hand shape classification using multi-layered randomized decision forests, in: Proc. of the European Conference on Computer Vision (ECCV), pp. 852–863.

[31] Pugeault, N., Bowden, R., 2011. Spelling it out: Real-time asl fingerspelling recognition, in: Proceedings of the 1st IEEE Workshop on Consumer Depth Cameras for Computer Vision, pp. 1114–1119.

[32] Wen, Y., Hu, C., Yu, G., Wang, C., 2012. A robust method of detecting hand gestures using depth sensors, in: Haptic Audio Visual Environments and Games (HAVE), 2012 IEEE International Workshop on, pp. 72–77.

[33] Biswas, K., Basu, S., 2011. Gesture recognition using microsoft kinect, in: Automation, Robotics and Applications (ICARA), 2011 5th International Conference on, pp. 100–103.

[34] Wan, T., Wang, Y., Li, J., 2012. Hand gesture recognition system using depth data, in: Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on, pp. 1063–1066.

[35] Herrera, D., Kannala, J., Heikkilä, J., Joint depth and color camera calibration with distortion correction", IEEE Trans. Pattern Anal. Mach. Intell. 34, pp. 2058–782, 2012

[36] Loris Nanni, Alessandra Lumini, Fabio Dominio, Mauro Donadeo, Pietro Zanuttigh (2013) Ensemble to improve gesture recognition International Journal of Automated Identification Technology, to appear

[37] Ojala, T., Pietikäinen, M. and Harwood, D. (1996), A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recognition 19(3):51-59.

[38] L. Nanni, A. Lumini and S. Brahnam, Local Binary Patterns variants as texture descriptors for medical image analysis, Artificial Intelligence in Medicine, vol.49, no.2, pp.117-125, June 2010.

[39] L. Nanni, A. Lumini and S. Brahnam (2014), Ensemble of different local descriptors, codebook generation methods and subwindow configurations for building a reliable computer vision system, Journal of King Saud University, available http://dx.doi.org/10.1016/j.jksus.2013.11.001.