Real-time Fingertip Localization Conditioned on Hand Gesture Classification

Xavier Suau, *Student Member, IEEE,* Marcel Alcoverro, *Student Member, IEEE,* Adolfo López-Méndez, Javier Ruiz-Hidalgo, *Member, IEEE* and Josep R. Casas, *Member, IEEE,*

***Abstract*—A method to obtain accurate hand gesture classification and fingertip localization from depth images is proposed. The Oriented Radial Distribution feature is utilized, exploiting its ability to globally describe hand poses, but also to locally detect likely fingertip positions. Hence, hand gesture and fingertip locations are characterized with a single feature calculation. We propose to divide the difficult problem of locating fingertips into two more tractable problems, taking advantage of hand gesture as an auxiliary variable. Besides, the ColorTip dataset is proposed, a dataset for hand gesture recognition and fingertip classification on depth data. ColorTip allows automatic fingertip annotation through a wieldy and not costly footage. The proposed method is evaluated against recent works and datasets, achieving promising results in both gesture classification and fingertip localization.**

***Index Terms*—hand gesture recognition, fingertip classification, range camera, interactivity, dataset**

I. Introduction

UNTIL recent years, interaction between humans and computer systems has been driven through specific devices (*i.e.* mouse, keyboard). A great effort has been put into improving the user experience when interacting with such devices. Recent successful examples are Apple's Trackpad [1] or multi-touch devices, that allow interaction by combining simple movements with finger configurations. However, device-based interaction is always limited, since the user must be *touching* the device.

Touch-less interaction is an interesting way to provide a more immersive and intuitive experience. Inspired by the way to interact with currently available multi-touch devices, we propose a touch-less interactive paradigm where hand gestures and fingertip configurations are combined with simple movements. This is very convenient since gestures are easy to memorize and performed with one hand, while allowing a large combination of interaction possibilities. In order to provide a similar usability, a precise and real-time detection of fingertip locations is required.

Detection of fingers or hand gestures is a complex task, given the high number of degrees-of-freedom of a hand, the usual presence of self-occlusions and the large amount of possible gestures. During the last years, new consumer oriented cameras have appeared in the market, providing pixel-wise depth information in real-time (*i.e.* Kinect). Such depth cameras open the door to new research directions in the field of touch-less interactivity, enabling more precise and fast approaches to make novel interactive paradigms come true. However, few works have achieved performant fingertip detection results using Kinect [2]–[5] (see Section II), mostly due to resolution problems and noisy depth estimations around fingers.

The objective of this work is to locate fingertips in real-time, that is, to know *where* fingers are placed, and also classify them to know *which* finger is each. Instead of facing the problem from raw data, as could be done with a similar approach to [6], we propose to use an intermediate step to restrict the search space. We exploit the statistical correlation between gestures and fingertip locations to perform such a restriction. This is very intuitive, since fingertip locations are conditioned by hand gestures, and at the same time allows a highly efficient fingertip inference. We choose to use the hand gesture as a discriminative auxiliary variable in this intermediate step. Indeed, there exists a real necessity of detecting hand gestures, so we discard using other auxiliary variables without any semantic meaning. We remark that algorithmic decisions are strongly motivated by efficiency, since real-time is a strong objective for any interactive system.

In a second step, we infer the most probable fingertip locations conditioned on the obtained hand gesture. We propose a specific graph matching approach, which exploits fingertip structure, to undertake the fingertip localization task. Thus, both fingertip locations and hand gesture are obtained from the proposed overall scheme.

We propose a novel usage of the Oriented Radial Distribution (ORD) feature, presented in [7]. The ORD feature characterizes a point cloud in such a way that its end-effectors are given a high ORD value, providing a high contrast between flat and extremal zones. Therefore, ORD is suitable to both globally characterize the structure of a hand gesture and to locally locate its end-effectors. Such ORD property nicely fits in the above mentioned two-step method. We propose to use the overall ORD structure for the gesture classification task, and to use local ORD extrema to feed the graph matching step. Therefore, a single ORD calculation is enough for both tasks.

The proposed method is evaluated with a recent 3D feature benchmark, revealing the convenience of using ORD. Furthermore, the gesture classification step is assessed with the

Xavier Suau, Marcel Alcoverro, Adolfo López-Méndez, Javier Ruiz-Hidalgo and Josep R. Casas are with the Universitat Politècnica de Catalunya

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o}248138.

This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación, under project TEC2010-18094

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

ASL database provided by [8]. Fingertip localization results are successfully compared to a state-of-the-art Random Forest (RF) approach.

Despite the revolution that commercial depth cameras have brought, their recent irruption supposes a lack of public datasets. Ganapathi *et al.* [9] provide a body pose estimation dataset using a Time-of-Flight (TOF) camera. Pugeault and Bowden [8] propose a hand gesture dataset using Kinect, which is intended for American Sign Language (ASL) purposes.

We propose ColorTip [10], a depth-based dataset consisting of 7 subjects performing 9 different hand gestures (Fig. 2,1). Ground-truth annotations for hand positions, hand gestures, fingertip locations and finger labels are also provided. Finger positions are obtained using a colored glove during capture, enabling a non-costly color-wise segmentation. Furthermore, each subject performs two sequences (Set A and Set B), with increased intra-gesture variability in the latter. Up to the authors knowledge, there does not exist any depth-based dataset for hand gesturing containing such information variety.

Summarizing, in this work we propose the following main contributions:

- A practical touch-less interaction concept, combining finger configurations, hand gesture and simple movements.
- A real-time method to obtain locations and labels, as well as hand gestures, using Kinect. We propose to exploit the statistical correlation between hand gestures and fingertip locations.
- A novel use of the Oriented Radial Distribution feature, exploiting its global structure for hand gesture characterization and its local values for fingertip detection.
- ColorTip, a public dataset intended for hand gesture classification and fingertip localization.

II. Related work

Within the touch-less interactivity field, depth cameras are being used for many purposes, ranging from full body pose estimation [6], [9], [11], [12] to hand gesture classification and fingertip localization. Obtaining hand gestures with Nearest Neighbors (NN) classification has proven to be a promising approach when dealing with depth data [13]–[15]. However, most recent works use features that are not specifically designed for depth data.

Many authors have explored how to control a virtual environment with hands (*i.e.* PC desktop, 3D model). In this direction, Soutschek *et al.* [13] propose a user interface for the navigation through 3D datasets using a Time-of-Flight (TOF) camera. They perform a polar crop of the hand over a distance threshold to the centroid, and a subsequent NN classification into five hand gestures. With a similar objective, Van den Berg and Van Gool [16] improve their work in [17] by combining RGB and depth to construct classification vectors. Their alphabet consists of four gestures that enable selecting, rotating, panning and zooming of a 3D model on a screen. Hackenberg *et al.* [2] estimate hand pose by identifying palm and finger candidates, after a pixel-wise classification into tips and pipes. The final hand structure is obtained with optical flow techniques. Ren *et al.* [14] segment the hand under some restrictive assumptions and adapt the Earth Movers Distance to a finger signature, finding the NN according to this metric.

Other works have focused on finger-spelling using the American Sign Language (ASL). While still being an alphabet, the ASL contains 26 gestures and their accurate classification becomes a challenging task. Kollorz *et al.* [15] obtain a fast NN classification using simple feature projection on two axis, which they apply to the first 12 letters of the ASL. Uebersax *et al.* [18] perform an iterative hand segmentation by optimizing the center, orientation and size of the hand. They smartly aggregate three classifiers that take shape and orientation into account. Pugeault and Bowden [8] propose a multi-resolution Gabor filtering of the hand patch to train a Random Forest classifier. In their work, they provide a complete dataset of 24 American Sign Language (ASL) gestures captured with the Kinect sensor, with both color and depth information available. Their dataset contains patches roughly centered at the hand centroid.

Fewer works have tackled the fingertip localization problem. In [2], fingertips are detected but not labeled, as well as in [3] where also the palm and fingers orientation are estimated. Both approaches exploit geometrical features to detect fingertips on the hand point cloud. The body part classification approach proposed by Shotton *et al.* in [6] is applied to hand parts by Keskin *et al.* [4], obtaining full hand poses at the expense of a costly training. Recently, Oikonomidis *et al.* [5] formulate the hand pose recovery problem as an optimization approach, measuring the discrepancy between a model and the observed hand. Full hand pose is provided (including fingertips), requiring initialization at a known initial pose. On the other hand, their cost function relies on color information, reducing the performance to controlled scenarios.

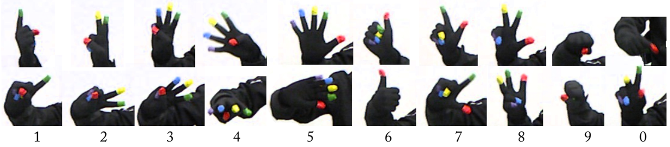

Fig. 1. Sample of the annotated gestures in the ColorTip dataset. Two examples per gesture are shown (columns). These examples are extracted from a *Set B* sequence, with a high intra-gesture variation. Note the rotations and translations. Label 0 corresponds to *no gesture* (*i.e.* other gestures, transitions).

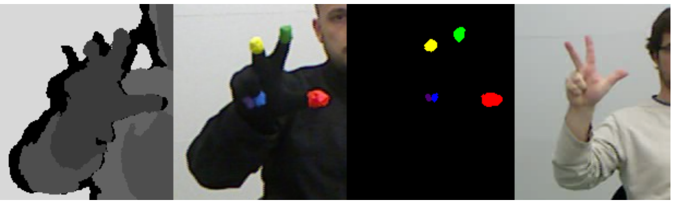

Fig. 2. Snapshot of the ColorTip dataset content. From left to right: depth image, color image (remark the colored glove), segmented fingertips (colors are directly finger labels, and centroids are finger positions) and a similar gesture in a test sequence.

III. The ColorTip Dataset

A. Description

ColorTip [10] is a public dataset for hand gesture recognition and fingertip localization captured with Kinect the sensor, which consists of a set of recordings and annotations with a two-fold objective. To provide a benchmark against which further research works may be assessed. But also, to enable novel interactive applications involving hand gesturing and fingertip localization.

In order to ease experimental setups, the ColorTip dataset is divided into folders according to:

- **Subject:** N subjects performing gestures like those shown in Fig. 1, ensuring intra user variability. Four of them are untrained users, which learned how to perform the gestures with a single and short explanation.
- **Challenge:** We consider that a given gesture may vary in orientation and translation. Therefore, raising 4 fingers is assumed as gesture number 4, but also moving these 4 fingers towards the camera, side views and hand rotations (Fig. 1). The amount of intra-gesture variability determines how challenging a given sequence is. The *Set A* sequences contain limited intra-gesture variation, which mainly consists in hand rotations on the vertical plane. On the other hand, the *Set B* sequences contain a higher intra-gesture variability, with free rotations and finger movement (as shown in Fig. 1).

In total, ColorTip contains a set of (7 subjects $\times$ 2 challenges) $= 14$ sequences of between 600 and 2000 frames each.

B. Annotations

Inspired by the work of Wang and Popović [19], a black glove with colored fingertips is used to capture the training sequences (see Fig. 2). In this way, we obtain a dataset together with a fingertip annotation in a single footage without requiring expensive motion capture systems, like those used in [4], [6]. Furthermore, one can easily record additional data to update the dataset. Actual fingertip locations are obtained by first segmenting the Kinect color images with a color-based Binary Partition Tree [20] (see Fig. 2) and then computing the region centroids. Color labels have an associated numerical label l.

Hand gestures are manually annotated among the 1-9 gestures, plus an extra label 0 for those frames with an unknown gesture. Also, a hand location annotation in image coordinates is provided.

IV. The Oriented Radial Distribution feature

We propose to characterize hands using the Oriented Radial Distribution feature, extending our previous work in [7]. ORD is a feature for the detection of end-effectors on depth camera captures. Such captures are considered as 3D point clouds which represent a sampling of the 3D surfaces in a scene.

ORD characterizes point clouds with an oriented 2D disk which is divided into sectors. The orientation of the disk is given by the surface normals on the point cloud. The average radius of the inlying points in each sector is used as a measure of the *curvature* and *extremeness* of the different parts of the point cloud. In Fig. 3, the bold crosses indicate the average radius of a sector's points, while the $\frac{1}{\sqrt{2}}\rho$ central circle indicates the value of such radius for flat zones. Thus, the more far away the crosses are located from this circle, the more *extremal* a point is. In the example, the point (a) belongs to a flat zone, while (b) belongs to an extremal zone.

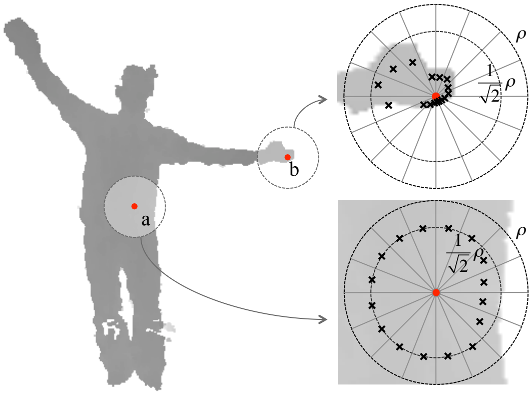

Fig. 3. ORD calculation, extracted from [7]. In (a), which corresponds to a flat zone, a low value of ORD is obtained, since the crosses barely differ from the central circle. On the other hand, a high ORD value is obtained in (b), most of the crosses being far from the central circle.

Parameter ρ is called ORD *scale*, and allows selecting the size of the extrema to be found. Selecting a scale of about the hand size ($\rho \approx 12\,cm$) will result in high ORD values at extrema of a similar scale. We propose to exploit this multi-resolution feature of ORD for the characterization of hands and fingers, by selecting the appropriate scales. In [7], the possibility of parameterizing ORD with a given radius was mentioned. However, since all the experiments were focused on detecting extremities, only a single scale was used.

ORD gives a representation of a depth-captured object, highlighting its end-effectors and curved parts (Fig. 4.c), but also characterizes flat zones with low values. We propose to use the global description ability of ORD to represent hand gestures for further classification; and to exploit the ORD multi-resolution extrema detection to locally detect hands and fingers, by choosing appropriate ρ scales (Fig. 4.b,c).

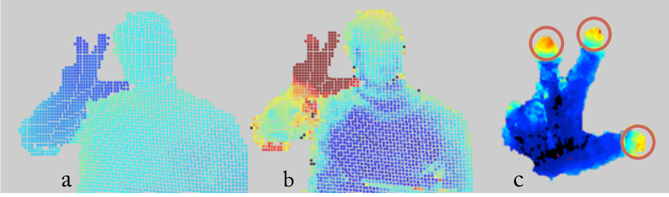

Fig. 4. Example of the use of ORD. From left to right, depth input data, ORD at *hand scale* on the whole body point cloud and ORD at *finger scale* on the segmented hand point cloud. In the latter, ORD maxima are marked with circles.

Contrarily to other features or descriptors that are defined on the image domain, the ORD scale is defined in the 3D space and thus ORD responses at different scales correspond to the actual size of objects, and not to the apparent size. This allows us to use a strong prior information on the multi-scale analysis, provided that we know the approximate sizes of body parts. Our approach efficiently exploits these ORD descriptor properties to jointly address three different tasks (Fig. 5):

- Hand detection and segmentation, with a simple threshold on the hand scale ORD as done in [7].
- Characterization global hand pose for hand gesture recognition (Section V-B)
- Characterization local hand parts for fingertip localization (Section V-C)

V. Fingertip Locations Conditioned on Hand Gesture Classification

A. *Method Overview*

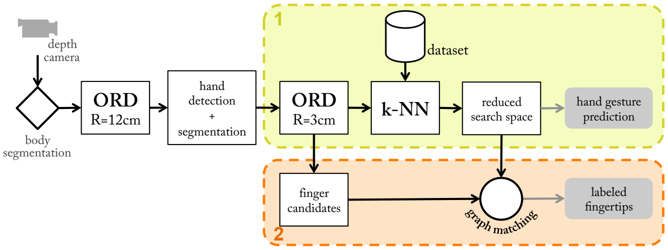

Fig. 5. General scheme of the proposed method. Fingertip locations are obtained (2) through an intermediate step, where the hand gesture is obtained as auxiliary variable (1).

The scheme in Fig. 5 summarizes the main blocks involved in the proposed method. In a preliminary step, we perform a *body segmentation* by means of background subtraction with depth data. Then, we detect and segment the hand by using the ORD at *hand scale* (in this work $R = 12$ cm).

Next, a two-step approach is proposed. We compute the ORD at *finger scale* ($R = 3$ cm) on a small patch containing the segmented hand, thus obtaining high ORD responses at fingertips and eventually at knuckles (see Fig. 4). The objective of this *finger scale* ORD is two-fold:

1) On the one hand, we use the ORD values to select the most likely hand poses (gestures) by computing distances between feature vectors, obtaining a subspace of likely hands from the ColorTip (Section V-B). We note this step as *gesture recognition*.
2) On the other hand, higher ORD responses are used as sparse fingertip candidates, and serve us to infer fingertip locations (*fingertip localization*) conditioned on the previously selected subspace. A structured inference framework is proposed, formulated as a graph matching problem (Section V-C).

The whole framework is based on the ORD feature data. We consider ORD a strong enough representation for this task, which, in addition, may be fast computed enabling real-time applications.

We introduce some notation hereafter, describing some of the variables handled in the proposed method. Training patches $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_i, \ldots, \mathbf{z}_N\}$ are squared patches of different sizes containing depth data of the segmented hand. We start by computing the ORD$(\mathbf{z}_i)$ at *finger scale* on each training patch. Then, we resample into a regular grid of $m \times m$ blocks to characterize the training patches (Fig. 6). Each block gets the mean ORD value of the pixels inside it, obtaining a set of m^2-dimensional feature vectors $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N\}$. Besides, let $\mathbf{r}_i \in \mathbb{R}^{2\times 5}$ denote the ground truth fingertip locations (in pixel coordinates) corresponding to the i-th training sample, denoting $\mathbf{r}_i[m] \in \mathbb{R}^2$ with $m = 1, \ldots, 5$ each fingertip location (used in Section V-C). Additionally, let y_i be gesture labels. Then, training templates are defined as $\mathbf{h}_i = \{\mathbf{x}_i, \mathbf{r}_i, y_i\}$, and the complete training dataset $\mathcal{H}$.

Given a test patch $\mathbf{z}$, the objective is to locate fingertip positions in it, that is $p(\mathbf{r}|\mathbf{z})$. Remark that our method is solely based on ORD information, thus one may replace $\mathbf{z}$ by ORD$(\mathbf{z})$.

We propose to break the problem of obtaining fingertips from data $p(\mathbf{r}|\mathbf{z})$ into two more tractable problems, that can be efficiently solved. In order to do so, we introduce the hand gesture y as auxiliary variable. By doing so, the problem of inferring fingertip locations from data can be posed as:

$$p(\mathbf{r}|\mathbf{z}) = \sum_{y} p(\mathbf{r}|y, \mathbf{z}) \cdot p(y|\mathbf{z}) \tag{1}$$

However, the marginalization of gestures implies a time consuming summation. Since real-time is a requirement, we approximate the problem in Equation (1) by firstly maximizing $p(y|\mathbf{z})$, obtaining the best candidate $\hat{\mathbf{h}} \in \mathcal{H}$. Secondly, we infer fingertip locations from the best template obtained after this maximization. The problem results as posed in Equation (2):

$$p(\mathbf{r}|\mathbf{z}) \approx p(\mathbf{r}|\hat{\mathbf{h}}, \mathbf{z}) \text{ with } \left\{\hat{\mathbf{h}} \in \mathcal{H} \mid \hat{y} = \underset{y}{\operatorname{argmax}}\{p(y|\mathbf{z})\}\right\} \tag{2}$$

We propose to solve the gesture recognition problem $p(y|\mathbf{z})$ using a k-Nearest Neighbors (k-NN) classifier. k-NN techniques are strongly sensitive to the data nature. Thus, an inappropriate feature selection could lead to a bad k-NN classification. Choosing a k-NN classifier helps to test the suitability of the ORD feature, as well as providing a fast classification taking advantage of a *kd-tree* [21] structure.

Concerning the fingertip localization problem $p(\mathbf{r}|y, \mathbf{z})$, we propose to solve it using a graph matching algorithm with a structure-based cost on edges. Such step is conditioned on the search space subspace obtained from the $p(y|\mathbf{z})$ problem.

B. *Hand Gesture Recognition*

In pattern recognition problems, the accuracy of a method ultimately depends on the distance metrics on the feature space, i.e., whether classes in the feature space appear separate enough to learn a robust classification rule. In this work, we propose a feature space solely based on the ORD descriptor. Feature vectors obtained by computing the ORD descriptor on the input data provide a representation of salient regions of the hand. In other words, ORD-feature vectors of hand poses can be seen as distribution of important parts of the hand and even interpreted as where the knuckles and fingers lay within the patch. For that reason, an ORD-based feature space is a suitable space for matching hand poses.

We choose to use a k-NN classifier for pose and gesture recognition. In this way, we show that even by simple matching techniques, the ORD feature space is adequate for hand analysis.

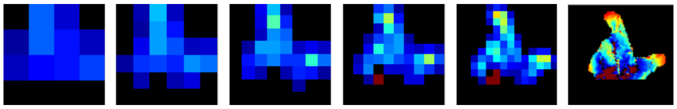

Fig. 6. Examples of feature vectors at various m resampling values. From left to right, $m = \{4, 6, 8, 10, 14, \text{full ORD patch}\}$

To use a k-NN classifier on a large set of instances, we use a m^2-dimensional *kd-tree* [22] that efficiently organizes feature vectors, allowing fast NN queries. The L_2 norm is used in this work, after an empirical trade-off between speed and performance.

For a test patch, the k-NN search returns a set of k training templates $\mathcal{H}^k = \{\mathbf{h}_1, \dots, \mathbf{h}_j, \dots \mathbf{h}_k\}$ with associated distances to the test patch $\delta : \mathcal{H}^k \mapsto \mathbb{R}$. Let $\Phi_y(\mathcal{H}^k)$ be the distribution of gestures obtained from $\mathcal{H}^k$.

We note $\hat{\mathbf{h}}$ the k-NN best match by majority, as specified in Eq. (3). The obtention of $\hat{\mathbf{h}}$ is conditioned on the gesture which maximizes $\Phi_y(\mathcal{H}^k)$. Therefore, maximizing $\Phi_y(\mathcal{H}^k)$ solves the $p(y|\mathbf{x})$ problem posed in Equation (1). Remark that one may refer to 1-NN best match, which is the first nearest neighbor in the training dataset.

$$\hat{\mathbf{h}} = \mathbf{h}_j \in \mathcal{H}^k \quad | \quad j = \underset{j}{\operatorname{argmin}} \Big\{ \delta(\mathbf{h}_j) \mid y_j = \underset{y}{\operatorname{argmax}} \{ \Phi_y(\mathcal{H}^k) \} \Big\} \quad (3)$$

The k-NN search may deliver false detections, resulting in a noisy gesture recognition. We propose hereafter to apply human dynamics restrictions to smooth the result of Eq. (3).

1) Dynamically Constrained k-NN: In many cases, we are subject to analyze video sequences, which intrinsically have a temporal consistency over consecutive frames. Hand dynamics are smooth, hence we assume that hand gestures are not instantly changing, but are maintained during a minimal number of frames.

In order to exploit such video consistency, we propose to keep a trace of the last Q predicted gestures $\hat{Y}_Q = \{\hat{y}_{t-Q}, \dots, \hat{y}_{t-1}\}$, obtained from the gesture labels of $\mathcal{H}^Q = \{\hat{\mathbf{h}}_{t-Q}, \dots, \hat{\mathbf{h}}_{t-1}\}$. Let $\tilde{y}_Q = \underset{y}{\operatorname{argmax}}\{\Phi_y(\mathcal{H}^Q)\}$ be the statistical mode of the last gestures $\hat{Y}_Q$, and let $\hat{y}_t$ be the predicted gesture at time instant t, which is obtained as detailed in Equation (3).

Such approach helps smoothing gesture transitions, as well as de-noising intra-gesture false detections. In practice, we consider that a gesture will not change during time interval of less than $0.5\,s$ (15 frames at $30\,fps$), and we set $Q = 15$, $k = 50$ and $m \approx 12$ after experimental results.

C. Fingertip Localization

We address the problem of fingertip location by making use of the ORD descriptor in a structured inference framework. Maxima of the ORD of the input patch are likely to represent fingertip locations. However, as mentioned before, for some hand poses these maxima may correspond to other salient points of the hand. But, even if all the maxima correspond to finger locations, one should be able to classify which finger belongs to each maximum. Consequently, there is a need to exploit the global hand structure to overcome these issues.

Section 1 Dynamically Constrained k-NN search

```
Input:
  $\mathcal{H}^k = \{\mathbf{h}_1, \dots, \mathbf{h}_j, \dots \mathbf{h}_k\} =$ k-NN set at time $t$
  $\tilde{y}_Q =$ mode of the last predicted gestures $\hat{Y}_Q$
  $\hat{y}_t =$ predicted gesture at $t$ or $\underset{y}{\operatorname{argmax}}\{f_y\}$
Output: $\hat{\mathbf{h}} =$ best k-NN

if $\hat{y}_t = \tilde{y}_Q$ then
    $\hat{\mathbf{h}} =$ k-NN by majority (Eq. (3))
else
    if $\exists\, y_j \in \hat{Y}_Q \mid y_j = \tilde{y}_Q$ then
        $\hat{\mathbf{h}} = \mathbf{h}_j \in \mathcal{H}^k \mid j = \operatorname{argmin}\{\delta(\mathbf{h}_j) \mid y_j = \tilde{y}_Q\}$
    else
        $\hat{\mathbf{h}} =$ 1-NN
    end if
end if
```

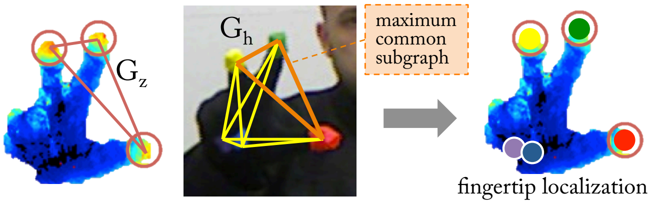

Fig. 7. Fingertip localization scheme. Fingertip locations are inferred from the ground-truth graph G_h by computing the Maximum Common Subgraph with respect to the test graph G_z.

Fingertip localization on test patches takes advantage of the pose recognition scheme presented in Section V-B. Let us recall that, in the training phase, we define templates $\mathbf{h}_i = \{\mathbf{x}_i, \mathbf{r}_i, y_i\}$ comprising the feature vectors, ground truth fingertip locations and gesture labels, respectively. Our method exploits the geometric structure of the ground truth fingertip locations $\hat{\mathbf{r}}$ of the best template match $\hat{\mathbf{h}}$ provided by the k-NN pose recognition block. The objective is to infer which ORD maxima of the test patch correspond to fingertip locations, and which are their finger classes. Let $G_h = (V_h, E_h)$ be a fully connected graph where vertices $v_h \in V_h$ correspond to the available fingertip coordinates in $\hat{\mathbf{h}}$, which we denote as $\mathbf{r}[v_h] \in \mathbb{R}^2$ (if a fingertip is not visible, such vertex is not considered). Let $G_z = (V_z, E_z)$ be the fully connected graph where vertices $v_z \in V_z$ correspond to the ORD maxima $\mathbf{s}$ of the test patch $\mathbf{z}_t$, namely $\mathbf{s}[v_z] \in \mathbb{R}^2$. We obtain a correspondence between vertices in G_h and vertices in G_z by computing the *maximum common subgraph* [23] (Fig. 7). This process consists in obtaining the graph G with the maximum number of vertices such that there exist subgraph isomorphisms[1] from G to G_h and from G to G_z. Note that in general there exists more than one maximum common subgraph. From the set of maximum common subgraphs we choose the one that best satisfies a geometric constraint defined on its edges. Let us denote $G_{mcs} = (V', E')$ a graph from the set of maximum common subgraphs of G_h and G_z, which involves the mappings $f_h : V' \mapsto V_h$ and $f_z : V' \mapsto V_z$. Then, for each edge $(u, v) \in E'$ we can obtain the vectors $\mathbf{e}_h = \mathbf{r}[f_h(u)] - \mathbf{r}[f_h(v)]$ and $\mathbf{e}_z = \mathbf{s}[f_z(u)] - \mathbf{s}[f_z(v)]$ which characterize geometrically the graphs G_h and G_z. We propose to select the maximum common subgraph that minimizes the cost:

$$\mathcal{C} = \sum_{(u,v)\in E'} \|\mathbf{e}_h - \mathbf{e}_z\| + 1 - \frac{\mathbf{e}_h \cdot \mathbf{e}_z}{\|\mathbf{e}_h\| \, \|\mathbf{e}_z\|} \qquad (4)$$

The measure in Eq. (4) combines a cost proportional to the difference of relative distances between fingertips with a cost that penalizes matchings with distinct relative orientation between fingertips. In this manner, we take account of the geometrical structure of the whole fingertips set, of both the test and template match, which allow matching even in case of misses or false fingertip detections.

ORD maxima are found by clustering pixels depending on their thresholded ($> t_f$) ORD values into, at most, 5 clusters of a given minimal size s_f. For clustering, connectivity between pixels is verified, but also depth connectivity, thus we are subject to work with 3D data. For this purpose, we use the 3D Euclidean clustering proposed by Rusu in [24]. Remark that t_f and s_f are parameters of the finger localization method. Summarizing, the method proceeds as follows:

1) The test feature vector $\mathbf{x}_j$ is processed by the k-NN pose gesture recognition block. As a result, we match a template $\hat{\mathbf{h}}$ and build the graph G_h using the ground truth finger coordinates r.
2) Coordinates of ORD maxima $\mathbf{s}$ are computed using the clustering method and the graph G_z is built.
3) The maximum common subgraph G that minimizes the cost $\mathcal{C}$ in Eq. (4) is obtained, which defines the fingertips matching between the test patch and the template match.
4) Missing fingers in the test patch with respect to the template match are copied from the latter according to the average displacement between both sets of fingertip coordinates.

VI. Experimental Results

A. ASL gesture results

The ASL dataset provided by [8] in [25] is used in the following experiment. Such dataset contains annotated hand patches of 5 subjects recorded with the Kinect camera, performing 24 ASL alphabet gestures. Accuracy is used as a measure to be able to compare with other reference methods.

As in [8], the dataset is randomly split into equally sized training and test subsets. Doing so is advantageous for a k-NN strategy, since the probability of having a consecutive frame (very similar) in the training subset is very high. Such effect is reflected in the *Random* column of Table I, achieving an accuracy of 98% against a 73% of [8] on the same dataset.

A *leave-one-subject-out-cross-validation* (LOSOCV) is also carried out. In that case, we achieve an accuracy of 74%, still 25 points higher than [8] (49%, results provided in [25]). Although methods are not directly comparable, since [18] results are obtained in a different dataset, our method achieves a similar performance with the advantage of not assuming that the hand is the closest connected component.

It should be remarked that both [8] and [18] are strictly gesture recognition methods. The proposed method uses the same ORD feature calculation to additionally provide fingertip localization.

Method	Random	LOSOCV
[18]*	0.88	0.76
[8]	0.73	0.49
Proposed	**0.98**	**0.74**

TABLE I
Comparative ASL hand gesture recognition average accuracy
Evaluated on a different dataset.

B. ColorTip Experimental Setup

The ColorTip dataset, consisting of a total of 14 sequences (Section III), is used for evaluation in the following experiments. We distinguish between *Set A* and *Set B* sequences in the results, given the considerable difference of intra-gesture variation.

Results are obtained considering a LOSOCV strategy. Thus, results for subject-i are obtained using as training dataset the remaining subjects' sequences. Remark that if we are considering the subject-i *Set A*, the sequence subject-i *Set B* is not used for training, and viceversa.

C. Gesture Recognition results

The suitability of using the ORD feature for gesture recognition is evaluated. With this purpose, we compare the results obtained with ORD against a benchmark of 3D features. Also, we provide a comparison with state-of-the-art gesture recognition methods using the challenging ASL alphabet.

1) ORD vs. Benchmark: A benchmark consisting of various 3D features is considered in order to evaluate the performance of ORD regarding the classification task. We note $\mathcal{P}$ the 3D point cloud of the segmented hand, and $\rho(p)$ the neighborhood of radius ρ of a point p. The proposed benchmark consists of:

- **Depth** Computed with respect to the average depth of $\mathcal{P}$.
- **Curvature** Computed as $\frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}$, where λ_i are the eigenvalues of the eigendecomposition of $\rho(p)$.
- **3DSC** 3D Shape Context, Frome *et al.* in [26].
- **VFH** Viewpoint Feature Histogram, Rusu *et al.* in [27].
- **SHOT** Signature of Histograms of OrienTations, Tombari *et al.* in [28]

[1] A graph isomorphism of graphs G and H is a bijection f between the vertex sets of G and H such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H.

The 3DSC and SHOT features provide pixel-wise histograms. As proposed in [29], to obtain scalar values per pixel, we compute the Kullback-Leibler divergence between each histogram and the average histogram. Then, the same $m \times m$ subsampling as in Section V-B is performed, obtaining m^2 sized feature vectors. The depth and curvature features provide a scalar values per pixel, and the VFH feature already delivers a feature vector of size 308, which is used untouched.

In order to compare ORD against the benchmark, a k-NN by majority classification is performed with every feature (see Eq. (3)). For this experiment, we use our 14 training sequences, with a LOSOCV strategy. We present in Fig. 8 the average F-Measure of the 14 tested sequences (about 18000 frames). The F-Measure is calculated as $\frac{2 \cdot P \cdot R}{P+R}$, where P = precision and R = recall. Results obtained with Dynamically Constrained (DC) k-NN classification using ORD (Section V-B1) are also included.

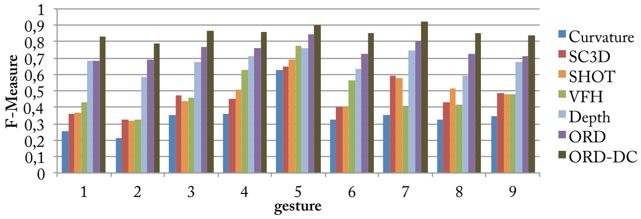

Fig. 8. Comparison between using a 3D feature benchmark and ORD. All the experiments are obtained with k-NN classification by majority.

The ORD feature outperforms the benchmark, with an average F-Measure of 0.75. The fact that ORD is focused on characterizing 3D surfaces (by adapting orientation locally) helps achieving such results, since $\mathcal{P}$ is indeed a 3D surface. The best features in the benchmark are *depth* with an average F-Measure of 0.67 and *VFH* with 0.50. The benchmark features do not take into account the 3D surface nature of $\mathcal{P}$, and analyze it as it was a 3D point cloud.

Note that the DC k-NN search proposed in Section V-B1 helps increasing the F-Measure from 0.75 to 0.86 by exploiting video temporal consistency of gestures.

2) Influence of the feature vector size: The dimension $m \times m$ of the feature vectors $\{\mathbf{x}_i\}$ has a noticeable impact on the hand gesture recognition results. In order to assess such effect, and with the objective of selecting an optimal value for m, we extract hand gesture recognition results for various values of m (Fig. 9). We recall that a feature vector consists of the resampling of an ORD patch to an $m \times m$ grid.

Experiments show that low $m \approx 4$ values lead to feature vectors which are not representative enough to distinguish between gestures. On the other hand, large $m > 14$ values lead to an overfitting problem, since feature vectors become too related to data (usually noisy). In such case, the predictive performance degrades. Thus, values of $m \approx 12$ provide the best results in terms of hand gesture recognition.

3) Influence of the dataset size: The size of training datasets may suppose the bottleneck of a classification system. Designing scalable methods is crucial, allowing further incorporation of new training data if required. Furthermore, memory access and capacity problems may also occur due to large datasets.

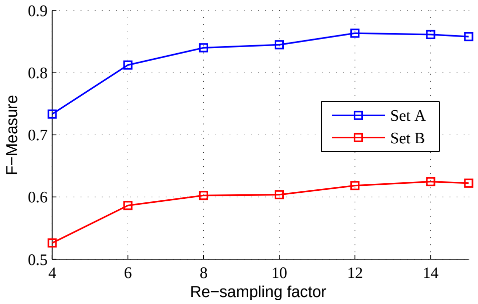

Fig. 9. Effect of the resampling factor m on the hand gesture classification. We observe how resamplings to $m \approx 12$ provide the best results.

We analyze in this experiment how the proposed method behaves with small training datasets. A basic clustering by Euclidean distance is performed to reduce the original training dataset $\mathcal{H}$, taking advantage of the already built *k-d tree*. More precisely, a template $\mathbf{h}_j \in \mathcal{H}$ is randomly selected, grouping all those templates $\mathbf{h}_i$ at a certain distance $\|\mathbf{x}_j - \mathbf{x}_i\| < D$ into a new average training template $\bar{\mathbf{h}}_j$. Such step is repeated until all the original templates are checked, obtaining the reduced dataset $\bar{\mathcal{H}} = \{\bar{\mathbf{h}}_j\}$. We note $F_{\%} = \frac{\bar{\mathcal{H}}}{\mathcal{H}}$ as reduction factor.

In Fig. 10 we present the F-Measure degradation as $\mathcal{H}$ is reduced, using a LOSOCV strategy. The original experiment at $(F_{\%} = 100\%)$ consists of an average of 15200 training templates.

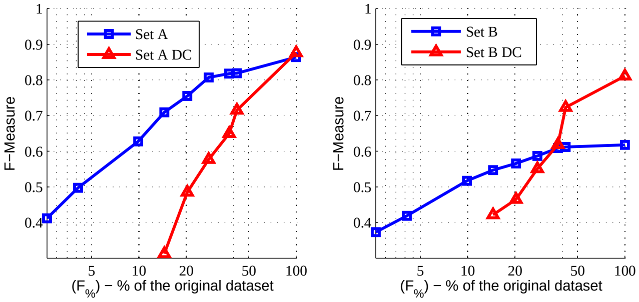

Fig. 10. F-Measure degradation for various reduction factors of the training dataset. Remark that the k-NN search degrades slower than k-NN DC. However, the latter performs better with the complete dataset, as already shown in Fig. 8. Such effect is more visible in the *Set B* sequence.

The proposed method successfully tolerates drastic reductions of the training dataset. Such scalable behavior allows reducing the training dataset until $F_{\%} = 20\%$ (3040 templates) with a degradation of less than 5%.

The k-NN DC search performs better with the complete dataset, even if in the case of *Set A* sequences such effect is barely visible given the good performance of the stand-alone k-NN. We remark that, in the case of *Set A*, the performance without DC is already close to the annotation error due to transitions between gestures. However, we note that k-NN DC degrades faster.

In the *Set B* case, at $F_{\%} \approx 35\%$ the stand-alone k-NN search already outperforms the DC version, since the number

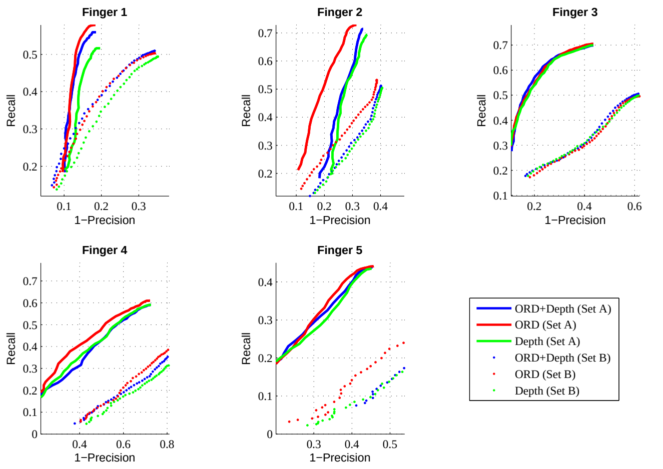

Fig. 11. Fingertip classification results using the RF baseline approach. 50 different detection thresholds are used. Note that RF using the stand-alone ORD values obtain the best results.

of erroneous gestures being smoothed grows.

In our case, a training template $\mathbf{h}_i = \{\mathbf{x}_i, \mathbf{r}_i, y_i\}$ occupies $12 \cdot 12 \cdot 4 + 10 \cdot 4 + 1 = 587$ *bytes*. Thus, at $F_{\%} = 20\%$, the reduced dataset only occupies about $3040 \cdot 587 \approx 1.78$ *Mb*. Scalability is achieved taking advantage of the robustness against drastic reductions of the dataset, allowing the incorporation of new training sequences at low memory cost.

D. Fingertip Localization results

We conduct several experiments to evaluate the ORD and the proposed framework in the fingertip localization task. First, we compare the proposed fingertip inference method (Section V), with a state-of-the-art fingertip detector based on Random Forests (RF). The RF method is also used to demonstrate the suitability of the ORD feature for hand analysis tasks. Then, we show the computational performance of the proposed method.

The fingertip evaluation protocol consists in a LOSOCV. We consider that a finger has been correctly localized if the estimated location and the ground-truth location are within a distance of 10 pixels.

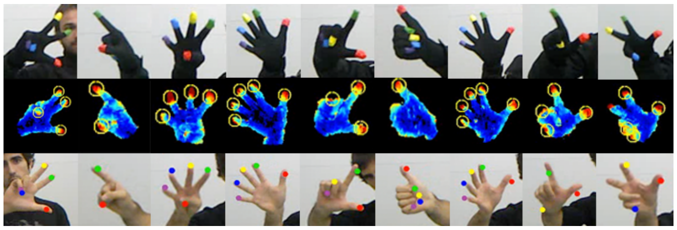

Fig. 12. Fingertip localization results (columns). The upper row contains the k-NN selected patch $\hat{\mathbf{h}}$ from the database, which intrinsically represents the recognized gesture. In the middle row, we show the ORD maxima to which fingers $\hat{\mathbf{r}}$ of $\hat{\mathbf{h}}$ are matched. The resulting fingertip localization on the testing hand is shown in the bottom row. Two erroneous examples are shown in the farthest right columns.

	RF(Depth)	**RF(ORD)**	**RF(ORD+Depth)**	**Ours**
finger 1	0.62	0.61	0.59	**0.67**
finger 2	0.64	**0.69**	0.64	0.66
finger 3	0.68	0.67	0.67	**0.68**
finger 4	0.46	0.49	0.46	**0.54**
finger 5	0.21	0.24	0.21	**0.54**
average	0.52	0.54	0.51	**0.62**

TABLE II
SET A SEQUENCES - COMPARATIVE FINGERTIP LOCALIZATION F-MEASURE.

	RF(Depth)	**RF(ORD)**	**RF(ORD+Depth)**	**Ours**
finger 1	0.53	0.51	0.52	**0.59**
finger 2	0.47	0.50	0.46	**0.57**
finger 3	0.42	0.41	0.42	**0.51**
finger 4	0.27	0.29	0.26	**0.34**
finger 5	0.13	0.17	0.14	**0.37**
average	0.37	0.38	0.36	**0.48**

TABLE III
SET B SEQUENCES - COMPARATIVE FINGERTIP LOCALIZATION F-MEASURE.

1) RF Fingertip Localization: In order to evaluate the proposed algorithm, we implement a fingertip localization method using Random Forests (RF) [30]. The RF localization method is based on the successful system for detecting body parts from range data proposed by Shotton et al. [6]. We use very similar depth-invariant features, but in addition to depth data, we include the ORD feature.

We employ RFs comprising 10 trees of maximum depth 15. Three baselines are trained: one using depth information exclusively, another using ORD exclusively and a baseline combining both features. The precision and recall performance of the RF approach is evaluated with 50 different detection thresholds (Fig. 11). The experiments reveal that RF trained with the stand-alone ORD values provide the best results (in red in Fig. 11), showing that ORD is also a suitable feature to locally describe parts of an object.

Our approach is evaluated with 3 different t_f and 8 different s_f parameters, obtaining the best results with $t_f = 0.3$ and $s_f = 0.8\,cm^2$. Some visual results are provided in Fig. 12.

Comparative results between our approach and the best RF baseline are presented in Table II (*Set A*) and Table III (*Set B*). The proposed method consistently outperforms all the RF baseline configurations. The main reason is the ability of our method to infer fingertip locations using structured inference given a template pose. First, hand pose matching allows to robustly locate fingertips under several hand rotations, which is the main limitation of the RF approach. Second, the global structure of the hand pose helps to robustly detect fingertips even when there is weak evidence of a finger location. In contrast, the RF approach requires each finger to have strong evidence (votes) in order to be robustly detected.

The RF baseline results also show the suitability of the ORD for the fingertip localization task, in terms of F-Measure. Best RF performance is achieved when binary tests exclusively use the ORD descriptor. Interestingly, the ORD contributes to a significant increase in the index finger localization (finger 2).

E. Computational Performance

The above experiments are carried out on an Intel Core2 Duo CPU E7400 @ 2.80GHz. To calculate the ORD feature, we have coded a parallel implementation on a NVIDIA GeForce GTX 295 GPU, performing about $70 - 140\times$ faster than the implementation in [7].Our approach performs in real-time, at a frame-rate of about $15 - 17\,fps$. A frame-rate of $16\,fps$ is achieved by [18]. Remark that our proposal delivers fingertip positions in addition to hand gestures. Moreover, a 176×144 camera is used in [18], with a smaller resolution than Kinect. Real-time is also attained by [8] for gesture recognition, using a state-of-the-art body tracker to detect hands.

VII. Conclusion and Future Work

In this paper, we have presented a joint method for gesture recognition and fingertip localization. To do so, we have proposed a two-fold exploitation of the ORD feature. Firstly, we utilize ORD to globally describe hand gestures for further classification purposes. Secondly, we take advantage of the multi-scale local description ability of ORD to robustly detect hands and fingertip locations.

More precisely, we have proposed a method to infer fingertip locations by including hand gesture as an auxiliary variable in the problem. Doing so, fingertips are obtained on a search space that is conditioned on the obtained hand gesture. Experiments showed that our method is robust, scalable and runs in real-time.

For gesture recognition, we conducted experiments on a publicly available ASL dataset and on our own hand gesture dataset. In both datasets, our method shows state-of-the-art performance, with the added value of providing fingertip positions. Furthermore, ORD has proven to be more effective than other 3D features for classification tasks

For fingertip localization, we compared our method with a state-of-the-art approach based on Random Forests. Our experiments show the superior performance of the proposed approach due to the ability to perform structured inference on robustly recognized hand poses.

A new dataset for hand gesture recognition and fingertip localization on depth data has been proposed, called ColorTip. In the near future, more sequences at various depth levels will be publicly available within the ColorTip dataset.

References

[1] Apple Inc, "Magic Trackpad," 2012.

[2] G. Hackenberg, R. McCall, and W. Broll, "Lightweight Palm and Finger Tracking for Real-Time 3D Gesture Control," in *VR*, no. March 2010. IEEE, 2011, pp. 19–26.

[3] MIT Finger Detection Demo, "http://www.ros.org/wiki/mit-ros-pkg/KinectDemos/FingerDetection," 2011.

[4] C. Keskin, F. Krac, Y. E. Kara, and L. Akarun, "Real Time Hand Pose Estimation using Depth Sensors," in *ICCV-CDC4CV*, 2011, pp. 1228–1234.

[5] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient Model-based 3D Tracking of Hand Articulations using Kinect," in *BMVC*, 2011.

[6] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-Time Human Pose Recognition in Parts from Single Depth Images," in *CVPR*, 2011, pp. 1297–1304.

[7] X. Suau, J. Ruiz-Hidalgo, and J. R. Casas, "Oriented Radial Distribution on Depth Data: Application to the Detection of End-Effectors," in *ICASSP*, 2012.

[8] N. Pugeault and R. Bowden, "Spelling It Out: RealTime ASL Fingerspelling Recognition," in *ICCV-CDC4CV*, 2011.

[9] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real Time Motion Capture Using a Single Time-Of-Flight Camera," in *CVPR*, 2010, pp. 755–762.

[10] ColorTip Dataset, "https://imatge.upc.edu/web/?q=res/colortip."

[11] A. Baak, M. Meinard, G. Bharaj, H.-p. Seidel, C. Theobalt, and M. P. I. Informatik, "A Data-Driven Approach for Real-Time Full Body Pose Reconstruction from a Depth Camera," in *ICCV*, 2011.

[12] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," *Image and Vision Computing*, 2011.

[13] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber, "3-D gesture-based scene navigation in medical imaging applications using Time-of-Flight cameras," in *CVPRW*, vol. 1-3. IEEE, 2008, pp. 1–6.

[14] Z. Ren, J. Yuan, and Z. Zhang, "Robust Hand Gesture Recognition Based on Finger- Earth Movers Distance with a Commodity Depth Camera," in *ACM-MM*, 2011, pp. 1093–1096.

[15] E. Kollorz, J. Penne, J. Hornegger, and A. Barke, "Gesture recognition with a Time-Of-Flight camera," *Intl. J. of Intelligent Syst. Tech. and Applications*, no. 3/4, p. 334, 2008.

[16] M. Van den Berg and L. Van Gool, "Combining RGB and ToF Cameras for Real-time 3D Hand Gesture Interaction," in *WACV*, 2011, pp. 66 – 72.

[17] M. Van Den Bergh, E. Koller-Meier, F. Bosche, and L. Van Gool, "Haarlet-based hand gesture recognition for 3D interaction," in *WACV*. IEEE, 2009, pp. 1–8.

[18] D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool, "Real-time Sign Language Letter and Word Recognition from Depth Data," in *ICCV-HCI*, 2011, pp. 1–8.

[19] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM Transactions on Graphics*, vol. 28, no. 3, p. 1, 2009.

[20] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval." *TIP*, vol. 9, no. 4, pp. 561–76, Jan. 2000.

[21] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *Trans. on Mathematic Software*, vol. 3, no. 3, pp. 209–226, 1977.

[22] M. Muja and L. G. David, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," in *VISAPP*, 2009.

[23] J. J. McGregor, "Backtrack search algorithms and the maximal common subgraph problem," *Software: Practice and Experience*, vol. 2, no. 1, pp. 23–34, 1982.

[24] R. B. Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments," Ph.D. dissertation, Technische Universitaet Muenchen, 2009.

[25] ASL Finger Spelling Dataset, "http://info.ee.surrey.ac.uk/Personal/N.Pugeault/."

[26] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *ECCV*, vol. 1, 2004, pp. 224–237.

[27] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *IROS*, 2010, pp. 2155–2162.

[28] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*, 2010, pp. 356–369.

[29] R. B. Rusu, A. Holzbach, M. Beetz, and G. Bradski, "Detecting and segmenting objects for mobile manipulation," in *S3DV-ICCV*, vol. 71, no. 6, 2009, pp. 47–54.

[30] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

Xavier Suau received a degree in Telecommunications Engineering at the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain in 2007. He also received a degree in Aeronautics Engineering at the Ecole Nationale de l'Aéronautique et de l'Espace (SUPAERO), Toulouse, France in 2007. In September 2007 he received a Master of Research in Automatics, Informatics and Decisional Systems by SUPAERO. His MSc research was carried out in the General Navigation Systems research department of THALES Avionics, Valence, France. During 2008 he joined THALES Avionics as full-time researcher in the Airbus A350-XWB Navigation System project. Since October 2009 he is a PhD student in the Image and Video Processing Group at UPC. He has taken part in the HESPERIA project developing algorithms to compensate illumination in foreground extraction applications. His current work is focused on exploiting range information for feature extraction and body pose estimation, in the framework of his PhD and also in the FP7 FascinatE project.

Marcel Alcoverro received his B.S.c. in Telecommunications in 2007 and Master degree in Computing in 2009 from the Technical University of Catalonia (UPC). During 2007 he worked as research assistant at the Equipes Traitement des Images et du Signal (ETIS) and the Institut delectronique et dinformatique Gaspard-Monge (IGM) in Paris, France, taking part in the EROS-3D project on artwork 3D databases. He is currently working with the Image Processing Group at UPC towards his Ph.D degree where he has been involved in projects of the Spanish Science and Technology System (VISION) and European FP projects (ACTIBIO, FASCINATE). His research interests include markerless motion capture, gesture recognition, 3D video processing and 3D graphics.

Adolfo López-Méndez received his B.S.c. (2007) and Master degree in Signal Theory and Communications (2009) from the Technical University of Catalonia (UPC), where he is currently working towards his Ph.D degree. His research interests include action and gesture recognition, markerless motion capture, 3D video processing and machine learning. Adolfo has been recently involved in the FP7 projects CHIL, ACTIBIO and FASCINATE.

Javier Ruiz-Hidalgo received a degree in Telecommunications Engineering at the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain in 1997. From 1998 to 1999, he developed an MSc by Research on the field of Computer Vision by the University of East Anglia (UEA) in Norwich, UK. During 1999 he joined the Image Processing Group at UPC working on image and video indexing in the context of the MPEG-7 standard where he obtained his Ph.D. thesis in 2006. Since 1999 he has been involved in various European Projects as a researcher from the Image Processing Group at UPC. During 1999 and 2000 he worked in the ACTS(AC308) DICEMAN project developing new descriptors and representations for image and video sequences. From 2001 to 2003 he is also involved in the IST/FET(2000-26467) project MASCOT developing an efficient compression scheme exploiting metadata information. Since 2006 he is the principal investigator in the HESPERIA (CENIT-2006) project involved in developing new image algorithms in security applications. Since 2001 he is an Associated Professor at the Universitat Politècnica de Catalunya. He is currently lecturing on the area of digital signal and systems and image processing. His current research interests include image segmentation, still image and sequence coding, compression and indexing.

Josep R. Casas is Associate Professor at the Department of Signal Theory and Communication, Technical University of Catalonia (UPC) in Barcelona. He graduated in Telecommunications Engineering in 1990 and received the PhD in 1996, both from UPC, where he is currently teaching Signals and Systems, Image Processing and Television Systems at the School of Telecommunications Engineering (Telecom BCN). He was visiting researcher at CSIRO Mathematics & Information Sciences in Canberra, Australia from 2000 to 2001. Josep R. Casas is Principal investigator of the project PROVEC (”Video Processing for Controlled Environments”) of the Spanish R&D&I Plan started in 2007, and has led or contributed to a number of industry-sponsored projects, projects of the Spanish Science and Technology System (VISION, HESPERIA) and European FP projects (ACTIBIO, SCHEMA, ADVISOR). In particular, he coordinated UPC contribution to CHIL (”Computers in the Human Interaction Loop”), an IP of the IST/EU 6th Framework Program in the strategic objective of Multimodal Interfaces, involving video, audio and natural language technologies. Josep R. Casas has authored or co-authored over 10 papers in international journals, 12 papers in LNCS, 50 contributions to conferences and 9 book chapters and a teaching book in the areas of video coding, analysis, indexing and image processing.