Human Limb Segmentation in Depth Maps based on Spatio-Temporal Graph Cuts Optimization

Antonio Hernández-Vela [a,b,*], Nadezhda Zlateva [c], Alexander Marinov [c], Miguel Reyes [a,b],
Petia Radeva [a,b], Dimo Dimov [c] and Sergio Escalera [a,b]

[a] *Computer Vision Center, Edifici O, Campus UAB, 08193 Bellaterra (Cerdanyola), Barcelona, Spain*
E-mail: {ahernandez, mreyes, petia, sescalera}@cvc.uab.cat
[b] *Dept. of Applied Mathematics and Analysis, Universitat de Barcelona, Gran Via 585, 08007 Barcelona, Spain*
[c] *Inst. of Information and Communication Technologies, BAS, Acad. G. Bonchev St., Block 2, Sofia 1113, Bulgaria*
E-mail: {zlateva, amarinov, dtdim}@iinf.bas.bg

Abstract. We present a framework for object segmentation using depth maps based on Random Forest and Graph-cuts theory, and apply it to the segmentation of human limbs. First, from a set of random depth features, Random Forest is used to infer a set of label probabilities for each data sample. This vector of probabilities is used as unary term in $\alpha - \beta$ swap Graph-cuts algorithm. Moreover, depth values of spatio-temporal neighboring data points are used as boundary potentials. Results on a new multi-label human depth data set show high performance in terms of segmentation overlapping of the novel methodology compared to classical approaches.

Keywords: Multi-modal vision processing, Random Forest, Graph-Cuts, Multi-label segmentation, Human body segmentation.

1. Introduction

Human motion capture is an essential acquisition technology with many applications in computer vision. However, detecting humans in images or videos is a challenging problem due to the high variety of possible configurations of the scenario, such as changes in the point of view, illumination conditions, and background complexity. An extensive research on this topic reveals that there are many recent methodologies addressing this problem [11,12,24,10]. Most of these works focus on the extraction and analysis of visual features. These methods have made a breakthrough in the treatment of human motion capture, achieving high performance despite the occasional similarities between the foreground and the background in the case of changes in light or viewpoint. In order to treat human pose recovery in uncontrolled scenarios, an early work used range images for object recognition or modeling [23]. This approach achieved a straighforward solution to the problem of intensity and view changes in RGB images through the representation of 3D structures. The progress and spread of this method came slowly since data acquisition devices were expensive and bulky, with cumbersome communication interfaces when conducting experiments. Recently, Microsoft has launched the Kinect, a cheap multisensor device based on structured light technology, capable of capturing visual depth information (RGBD technology, from Red, Green, Blue, and Depth, respectively). The device is so compact and portable that it can easily be installed in any environment to analyze scenarios where humans are present. Before Kinect, in the last decade, researchers have also used different methodologies and techniques for constructing 3D structures, such as stereoscopic images [13,29]. How-

[*] Corresponding author.

ever, in this case the problems of different lighting conditions and calibration still exist. Some of the research has also focused on the use of time-of-flight range cameras (TOF) for human parts detection and pose estimation [15,21,28], combining depth and RGB data [19].

Following the high popularity of Kinect and its depth capturing abilities, there exists a strong research interest for improving the current methods for human pose and hand gesture recognition. While this could be achieved by inter-frame feature tracking and matching against predefined gesture models, there are scenarios where a robust segmentation of the hand and arm regions are needed, e.g. for observing upper limb anomalies or distinguishing between finger configurations while performing a gesture. In that respect, depth information appears quite handy by reducing ambiguities due to illumination, colour, and texture diversity. Many researchers have obtained their first results in the field of human motion capture using this technology. In particular, Shotton et al. [25] present one of the greatest advances in the extraction of the human body pose from depth images, an approach that also forms the core of the Kinect human recognition framework. The method is based on inferring pixel label probabilities through Random Forest (RF), using mean shift to estimate human joints, and representing the body in skeletal form. Other recent work uses the skeletal model in conjunction with computer vision techniques to detect complex poses in situations where there are many interacting actors [20].

Currently, there exists a steady stream of updates and tools that provide robustness and applicability to the device. In December 2010, OpenNI [5] and PrimeSense [6] released their own Kinect open source drivers and motion tracking middleware (called NITE [3]) for PCs running Windows (7, Vista, and XP), Ubuntu and MacOSX. FAAST (Flexible Action and Articulated Skeleton Toolkit) is a middleware developed at the University of Southern California (USC) Institute for Creative Technologies that aims to facilitate the integration of full-body control within virtual reality applications and video games when using OpenNI-compliant depth sensors and drivers [2,27]. In June 2011, Microsoft released a non-commercial Kinect Software Development Kit (SDK) for Windows that includes Windows 7-compatible PC drivers for the Kinect device [7]. Microsoft's SDK allows developers to build Kinect enabled applications in Microsoft Visual Studio 2010 using C++, C# or Visual Basic. Microsoft is planning to release a commercial version of the Kinect for Windows SDK with support for more advanced device functionalities. There is also a third set of Kinect drivers for Windows, Mac and Linux PCs by the OpenKinect (libFree- Nect) open source project [4]. Code Laboratories CL NUI Platform offers a signed driver and SDK for multiple Kinect devices on Windows XP, Vista, and 7 [1].

In this paper we present a framework for object segmentation using depth maps based on RF and Graph-cuts theory (GC) and apply it to the segmentation of human limbs. The use of GC theory has recently been applied to the problem of image segmentation, obtaining successful results [9,17,18]. RF is used to infer a set of probabilities for each data sample, each one indicating the probability of a pixel to belong to a particular label. Then, this vector of probabilities is used as unary term in the $\alpha - \beta$ swap GC algorithm. Moreover, depth of neighbor data points in space and time are used as boundary potentials. As a result, we obtain an accurate segmentation of depth images based on the defined energy terms. Moreover, as long as we have a priori likelihoods representing target appearance, the presented method is generic enough to be applicable in any other object segmentation scenario. Our method is evaluated on a 3D data set designed in our lab, obtaining higher segmentation accuracy compared to standard segmentation approaches.

The rest of the paper is organized as follows: Section 2 presents the novel segmentation framework in depth images based on RF and GC theory. Section 3 presents a quantitative and qualitative evaluation of the methodology on a new multi-label depth video data set of human poses. Finally, Section 4 concludes the paper.

2. Method

The depth-image based approach suggested in [25] interprets the complex pose estimation task as an object classification problem by evaluating each depth pixel affiliation with a body part label, using respective Probability Distribution Functions (PDF). The pose recognition phase is addressed by re-projecting the pixel classification results and inferring the 3D positions of several skeletal joints using the RF and mean-shift algorithms. The work of [25] shows a number of achievements and improvements over previous work, most notably the growing of a randomized decision forest classifier of T decision trees applied on simple and computationally efficient depth features.

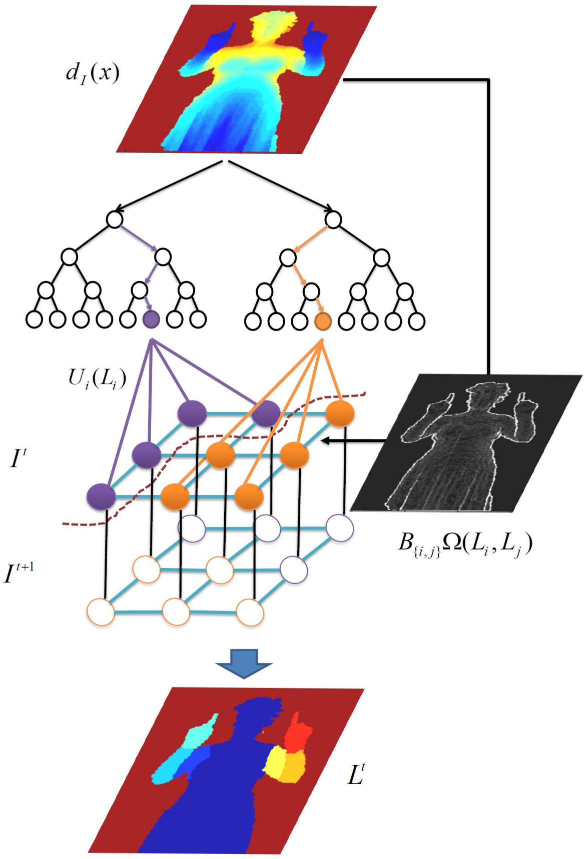


Fig. 1. Pipeline of the presented method, including the input depth information, Random Forest, Graph-cuts, and the final segmentation result.

Our goal is to extend the work of [25] and combine it with a general segmentation optimization procedure to define a globally optimum segmentation of objects in depth images. As a case study, we segment pixels belonging to the following seven body parts[1]: LU/LW/RU/RW for arms, (from Left, Right, Upper and loWer, respectively), LH/RH for hands, and the torso. The pipeline of the segmentation framework is illustrated in Fig. 1.

2.1. *Random Forest*

Considering a priori segmented human body from the background in a training set of depth images, the procedure for growing a randomized decision tree t is formulated over the same definition of a depth comparison feature as defined in [25]:

$$f_\theta(I, \mathbf{x}) = \mathbf{d_I}\left(\mathbf{x} + \frac{\mathbf{u}}{\mathbf{d_I(x)}}\right) - \mathbf{d_I}\left(\mathbf{x} + \frac{\mathbf{v}}{\mathbf{d_I(x)}}\right), \tag{1}$$

where $d_I(\mathbf{x})$ is the depth at pixel $\mathbf{x}$ in image I, I is considered subset of the Euclidean space E^2, $\theta = (\mathbf{u}, \mathbf{v})$, and $\mathbf{u}, \mathbf{v} \in \mathbb{R}^\mathbf{2}$ is a pair of offsets, the normalization of which ensures depth invariance. Thus, each θ determines two new pixels relative to $\mathbf{x}$, the depth difference of which accounts for the value of $f_\theta(I, \mathbf{x})$. Each tree consists of split and leaf nodes (the root is also a split node), as depicted in the upper part of Fig. 1. The training procedure of a given tree t over a unique set of ground truth images (avoid sharing images among trees), runs through the following steps:

1. Define a set Φ of node splitting criteria $\phi = (\theta, \tau)$, through the random selection of $\theta = (\mathbf{u}, \mathbf{v})$, and $\tau, \tau \subset \mathbb{R}$ (a set of splitting thresholds for each θ), with both θ and τ lying within some predefined range limits. After training, each split node will be assigned with its optimal ϕ value from Φ.
2. Define a set Q of training examples $Q = \{(I, \mathbf{x}) | \mathbf{x} \in \mathbf{I}\}$ over the entire set of training images for the tree, where I stands for an image, $\mathbf{x}$ is a randomly selected pixel in I, and the number of pixels $\mathbf{x}$ per image is fixed. Estimate the PDF of Q over the whole set of labels C (in our case $|C| = 7$):

$$P_Q(c) = \frac{h_Q(c)}{|Q|}, c \in C, \tag{2}$$

   where $h_Q(c)$ is the histogram of the examples from Q associated with the label $c \in C$. Each example from Q enters the root node, thus ensuring optimal training of the tree t.
3. At the currently being processed node (starting from the root), split the (sub)set Q, entering this node into two subsets Q_L and Q_R obeying Eq. (1):

$$\begin{aligned} &Q_L(\phi) = \{(I, \mathbf{x}) \,|\, f_\theta\,(I, \mathbf{x}) < \tau\}, \phi = (\theta, \tau), \\ &Q_R(\phi) = Q \setminus Q_{left}, \end{aligned} \tag{3}$$

[1] Note that the method can be applied to segment any number of labels of any object contained in a depth image.

and estimate the PDF of Q_L, $P_{Q_L}(c)$, as defined in Eq. (2). Compute the PDF of Q_R, which may be speeded up by the following formulae:

$$P_{Q_R}(c) = \frac{|Q|}{|Q_R|} P_Q(c) - \frac{|Q_L|}{|Q_R|} P_{Q_L}(c), \quad (4)$$

$$Q_R = Q_R(\phi), Q_L = Q_L(\phi), \quad (5)$$

$$c \in C. \quad (6)$$

4. Estimate the best splitting criterion ϕ^* for the current node, so that the information gain $G_Q(\phi^*)$ of partitioning set Q entering the node into left and right subsets to be maximum:

$$G_Q(\phi) = H(Q) - \frac{|Q_L(\phi)|}{|Q|} H(Q_L(\phi)) \quad (7)$$

$$- \frac{|Q_R(\phi)|}{|Q|} H(Q_R(\phi)), \quad (8)$$

$$\phi = (\theta, \tau) \in \Phi, \quad (9)$$

where $H(Q) = -\sum_{c \in C} P_Q(c) \ln(P_Q(c))$ represents Shannon's entropy for the input (sub)set Q and its splits (Q_L and Q_R) over the set of labels C. It is more or less obvious that $G_Q(\phi) > 0$, $\phi \in \Phi$, but it is difficult to make a more analytical statement for the behaviour of $G_Q(\phi)$. That is why we also use the full search approach to evaluate ϕ^*:

$$\phi^* = \arg\max_{\phi \in \Phi} G_Q(\phi). \quad (10)$$

5. Recursively repeat step 3 and 4 over $Q_L(\phi^*)$ and $Q_R(\phi^*)$ for the left and right node children respectively until some preset stop conditions are met: the tree reaches maximum depth; the information gain or the number of pixels in the node falls below a minimum. The node where the stop condition occurred is treated as a leaf node, where, instead of ϕ^*, the respective PDF for the subset Q reaching the node is stored (see Eq. (2)).

Once trained, such a randomized tree serves as a per pixel classifier for a test depth image. Each image pixel for recognition, i.e. an example $(I, \mathbf{x})$ is run through the tree, starting from the root and ending at a leaf node, taking a path that depends solely on the inequality $f_\theta(I, \mathbf{x}) < \tau$, using the splitting criterion $\phi = (\theta, \tau)$ stored at the current tree node. The pixel acquires the PDF kept at the reached leaf node. Because of the random factor when growing the tree, different trees have different predictions for the pixels of the same image. It cannot be stated that one tree is a better single classifier than another one since each tree is fitted to its training set. But an ensemble of trees, which form a random forest T, is expected to increase the predictive power of the classifier. Therefore, the inferred pixel probability distribution within the forest is estimated by averaging the PDFs over all trees in the forest as follows:

$$P(c|I, \mathbf{x}) = \frac{1}{|T|} \sum_{t \in T} P_t(c|I, \mathbf{x}), c \in C, \quad (11)$$

where $P_t(c|I, \mathbf{x})$ is the PDF stored at the leaf, reached by the pixel for classification $(I, \mathbf{x})$ and traced through the tree t, $t \in T$. Assuming the trees in the forest T are fairly balanced, the time complexity of classifying an image is $\mathcal{O}(|T| \cdot N_P \cdot L_{max})$, where N_P is the total number of pixels $(I, \mathbf{x})$ from the image and L_{max} is the averaged maximum depth level over the trees of T.

The randomized tree growing process suggested by Shotton et al. [25] involves two levels of randomness: in choosing the training images and in the random definition of the node splitting criteria. This ensures minimum correlation among the trees in the forest. Unlike Breiman's classic Random Forest algorithm [26], which chooses the best split candidate (criterion) among a small subset of all possible candidates, the presented split candidate selection procedure greedily explores all possible choices in order to guarantee the most efficient split at the current node. The after-effects are two: the most informative features are filtered down and pushed onto the tree; similar pixels have better chances of falling within the same descendant nodes. The estimated time complexity of building a randomized decision tree under the above conditions is $\mathcal{O}(|\Phi| \cdot |Q| \cdot L_{max})$.

We apply the RF methodology of [25], as described above, in the following two use cases: for rough detection of the main body parts, and for detailed segmentation of the fingers of the hands (to eventually be applied for sign/cued languages recognition problems).

2.2. Graph-cuts framework

GC [9] is an energy minimization framework which has been considerably applied in image segmentation –both binary and multi-label–, with highly successful results. In this work, we extend the GC theory to be used in depth images and optimize the results obtained from the RF approach in order to deal with automatic spatio-temporal multi-label segmentation.

Given $I = \{I^1, ..., I^s, ..., I^S\}$ the set of frames of the video sequence, and $\mathcal{X} = (\mathbf{x_1}, ..., \mathbf{x_i}, ..., \mathbf{x}_{|\mathcal{P}|})$ the set of pixels of I, let us define $\mathcal{P} = (1, ..., i, ...|\mathcal{P}|)$ the set of indexes of I; $\mathcal{N}$ the set of unordered pairs $\{i, j\}$ of neighboring pixels in space and time, under a defined neighborhood system –typically 6- or 26-connectivity–, and $L = (L_1, ..., L_i, ..., L_{|\mathcal{P}|})$ a vector whose components L_i specify the labels assigned to pixels $i \in \mathcal{P}$. This framework defines an energy function $E(L)$ that combines local and contextual information, and whose minimum value corresponds to the optimal solution of the problem –in our case, the optimal segmentation:

$$E(L) = U(L) + \lambda B(L). \tag{12}$$

The first term of the energy function is called the "unary potential". This potential encodes the local likelihood of the data by assigning individual penalties to each pixel for each one of the defined labels:

$$U(L) = \sum_{i \in \mathcal{P}} U_i(L_i). \tag{13}$$

The second term or "boundary potential" encodes contextual information by introducing penalties to each pair of neighboring pixels as follows:

$$B(L) = \sum_{\{i,j\} \in \mathcal{N}} B_{\{i,j\}} \; \Omega(L_i, L_j), \tag{14}$$

where $\Omega(L_i, L_j)$ is a function that introduces prior costs between each possible pair of neighboring labels. Finally, $\lambda \in \mathbb{R}^+$ is a weight that specifies the relative importance of the boundary term against the unary term.

Once the energy function is defined, a graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ is built following the neighborhood system used in the boundary potential $B(L)$. Since the Graph-cuts framework is defined for N-dimensional graphs, and we are working with video sequences –which can be seen as 3-D volumes–, we can extend the graph topology from 2-D to 3-D, and segment more than just one frame at a time. From a practical point of view, and considering that computer memory resources are limited, we adopt a sliding-window approach. More specifically, we define a fixed size volume window V like the one depicted in Fig. 2. This new graph topology introduces a new set of inter-frame connections between nodes, in addition to the existing intra-frame connections from the previous frame-by-frame approach. The values associated with these edges of the graph are computed with the same $B_{\{i,j\}}$ function as in the case of intra-frame edges.

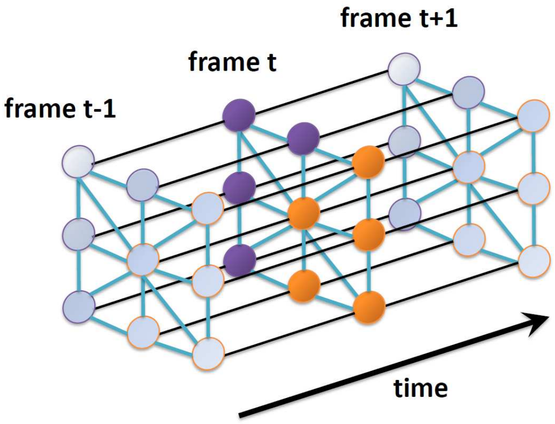


Fig. 2. Graph topology introducing temporal coherence.

The sliding-window approach starts segmenting the first $|V|$ frames, and covers all the video sequence volume, with a one-frame stride. This means that all the frames except the first and the last one are segmented at least twice and $|V|$ times at most. In order to select the final hypothesis for each frame, we use the energy value resulting from the minimization algorithm at each execution. Therefore, the execution with the lowest energy value is the one we trust as the best hypothesis.

Once the graph is built, the energy function is transferred to it. In the case of binary segmentation, i.e. $L_i \in \{0, 1\}, \forall i \in \mathcal{P}$, the min-cut algorithm [9] finds the minimum cut of this graph –which corresponds to the minimum energy– and thus, the optimal segmentation. When $L_i \in \{0, ..., N_L\}, N_L > 1$, two main algorithms can be applied in order to find not the minimum energy, but a suboptimal approximation of it: $\alpha - \beta$ swap and α-expansion [8]. While the first one is less restrictive and can be applied in a broader range of energy functions, the second one has been proven to obtain better results, as long as the energy function fulfills some conditions [8]. In the case of $\alpha - \beta$ swap the boundary term $B_{\{i,j\}}$ must be *semi-metric*, which means that the conditions in Eq. (15) and (16) must be

Table 1

Weights of edges in $\mathcal{E}$.

edge	weight (cost)	for
t_i^{α}	$U_i(\alpha) + \sum_{\substack{j\in\mathcal{N}_i \\ L_j\notin\{\alpha,\beta\}}} B(\alpha, L_j)$	$L_i \in \{\alpha,\beta\}$
t_i^{β}	$U_i(\beta) + \sum_{\substack{j\in\mathcal{N}_i \\ L_j\notin\{\alpha,\beta\}}} B(\beta, L_j)$	$L_i \in \{\alpha,\beta\}$
$e_{\{i,j\}}$	$B(\alpha,\beta)$	$\substack{\{i,j\}\in\mathcal{N} \\ L_i,L_j\in\{\alpha,\beta\}}$

fulfilled:

$$B(L_i, L_j) = B(L_j, L_i) \geq 0 \tag{15}$$

$$B(L_i, L_j) = 0 \leftrightarrow L_i = L_j \tag{16}$$

$$B(L_i, L_j) \leq B(L_i, L_n) + B(L_n, L_j), \tag{17}$$

for any $L_i, L_j, L_n \in L$, being $B(L_i, L_j) = B_{\{i,j\}}$ $\Omega(L_i, L_j)$. Additionally, if we want to apply α- expansion, the condition in Eq. (17) must also be fulfilled. In that case, the boundary term $B_{\{i,j\}}$ is said to be *metric*.

In our case, Eq. (17) is not true for all nodes in $\mathcal{G}$, and so, we use $\alpha{-}\beta$ swap in our segmentation methodology for depth maps. This way, the set of nodes $\mathcal{V}$ contains a node for each pixel in I, plus two terminal nodes: α and β. Similarly, $\mathcal{E}$ is composed by two kinds of edges: terminal links t_i^{α} and t_i^{β}, and neighbor links $e_{\{i,j\}}$. The values assigned to the edges of $\mathcal{G}$ are then assigned following Table 1. Regarding time complexity, both algorithms α-expansion and $\alpha - \beta$ swap run in $\mathcal{O}(|L|^2 \cdot |\mathcal{P}|)$.

The following subsections define the specific energy function potentials that we designed for our problem.

Unary potential

The unary potential encodes the local likelihood for each pixel to belong to each one of the labels L_i of our problem. In our case, we have used the log-likelihood of the probabilities returned by the RF for the computation of the unary potential:

$$U_i(L_i) = -\ln(P\,(c|I, x)), \tag{18}$$

obtaining a unary cost potential for each class c_i – corresponding to label L_i in GC. This step is shown at the top of Figure 1, where the output probabilities of the leafs of the RF trees are used to compute the unary potentials $U_i(L_i)$ at the input edges of the GC graph.

Boundary potential

In the case of the boundary potential, we use the following formulation:

$$B_{\{i,j\}} = \frac{1}{\mathsf{dist}(i,j)} e^{-\beta \cdot H(\mathbf{x_i},\mathbf{x_j})}, \tag{19}$$

where $\beta = \left(2\langle(d_I\,(\mathbf{x_i}) - d_I\,(\mathbf{x_j}))^2\rangle\right)^{-1}$ and $\mathsf{dist}(i, j)$ computes the Euclidean distance between the cartesian coordinates of pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. In a similar way, the $H(\mathbf{x_i}, \mathbf{x_j})$ function computes the Euclidean distance between certain features of pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. The simpler case just uses depth information, thus $H(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{d_I}\,(\mathbf{x_i}) - \mathbf{d_I}\,(\mathbf{x_j})$, but in the experimental section we also test the use of RGB and depth information joined as a 4-dimensional feature vector.

Finally, we defined two different $\Omega(L_i, L_j)$ functions in order to introduce some prior costs between different labels. On one hand, we considered the trivial case where all different labels have the same cost:

$$\Omega_1(L_i, L_j) = \begin{cases} 0 \text{ for } L_i = L_j \\ 1 \text{ for } L_i \neq L_j. \end{cases} \tag{20}$$

On the other hand, we introduced some spatial coherence between the different labels, taking into account the kinematic constraints of the human body limbs:

$$\Omega_2(L_i, L_j) = \begin{cases} 0 & \text{for} & L_i = L_j \\ 10 & \text{for} & L_i = \text{LU}, L_j = \text{RU} \\ & & L_i = \text{LH}, L_j = \text{RH} \\ 5 & \text{for} & L_i = \text{LW}, L_j = \text{RH} \\ & & L_i = \text{RW}, L_j = \text{LH} \\ 1 \text{ otherwise}\,. & & \end{cases} \tag{21}$$

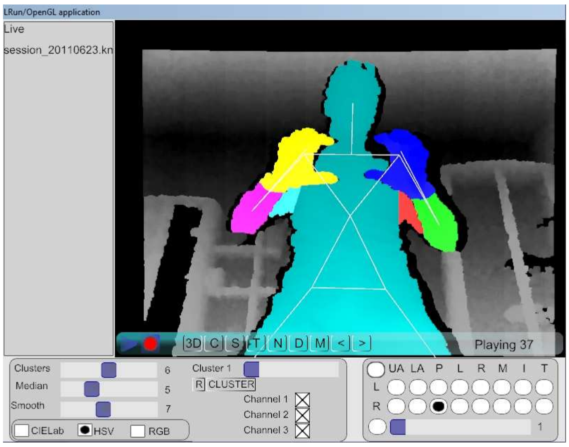
With this definition of the inter-label costs, we are making it difficult for the optimization algorithm to find a segmentation in which there exists a frontier between the right and left upper-arms, right and left hands, or in the lower measure, between left hand and right lower-arm, and vice-versa. Therefore, we are assuming that poses in which the two hands are touching are not probable[2].

3. Experiments and results

This section starts with a brief description of the considered data and the different methods, parameters, and validation protocol of the evaluation.

Data: For the purposes of gathering ground truth data, we defined a new data set of several sessions

[2] This label coherence cost should be estimated for each particular problem domain. In our particular data set of poses, the values of 1, 5, and 10 were experimentally computed.



Clusters 6
Median 5
Smooth 7
□ CIELab ● HSV □ RGB
Channel 1 ☒
Channel 2 ☒
Channel 3 ☒

Fig. 3. Interface for semi-automatic ground-truth generation.

where the actors are performing different gestures with their hands in front of the Kinect camera – only the upper body is considered. See Fig. 4 for some pose samples. Each frame is composed by one 24 bit RGB image of size 640x480 pixels, one 12 bit depth buffer of the same dimension, and a skeletal graph describing relevant joints of the upper human body. In order to label every pixel we created a special editing tool to facilitate labelling in a semi-supervised manner. Each frame is accompanied with label buffer of the same dimension as the captured images. The label buffer is automatically initialized through a rough label estimation algorithm. The pixels bounded by the cylinders between the enclosing joints of the shoulder to elbow are labelled as upper arm (LU/RU). By analogy the pixels inside the cylinder between the elbow and the joint of the hand are labelled as lower arm (LW,RW). The palm is labelled by the pixels bounded by a sphere centered in the joint of the hand (LH,RH). The RGB, depth, and skeletal data are directly obtained via the OpenNI library [5]. Finally each frame is manually edited to correct the roughly estimated labels by the initialization algorithm. The whole ground truth used in our experiments is created from capturing 2 actors in 3 sessions gathering 500 frames in total (15 fps). It should be noted that after the manual editing there still exist around 1% of false positive labels due to editor mistakes. An example of the developed interface for semi-automatic ground-truth generation is shown in Figure 3[3].

We also made an extra experiment for finger segmentation defining 6 labels per hand - one label for each finger and one for the palm. For gathering ground truth data from the fingers, we applied another initialization algorithm using coloured gloves, with each finger being painted with a different colour. Finally manual editing is still necessary due to the high level of false positive errors. 63 frames are generated and used in the experiment.

Methods and validation: In the first place, we analyze the results obtained directly using the probabilities returned by the RF approach. The RF algorithm used for the experiments computation has been implemented following the description of Shotton et al. [25]. In the same way, inspired by the reported test parameters and accuracy results in [25], our experiments rest on the following setup: we perform a 5-fold cross-validation over the available 500 frames by training random forest of $T = 3$ trees, therefore 130 unique training images per tree, with 1000 uniformly distributed pixels per image. We limit the maximum depth level L_{max} for all trees to 20, and use 100 candidate offset pairs θ, and 20 thresholds τ per θ to build the splitting criteria Φ. The remaining 100 images form the test set. Carrying a randomized test trial, we analyze the effect of the choice of test parameters on the classification accuracy and compare the results with another set of features: a mixture of the original depth features $f_\theta(I, \mathbf{x})$ from Eq. (1) and new features $g_\theta(I, \mathbf{x})$ based on the depth gradient:

$$g_\theta\left(I, \mathbf{x}\right) = \measuredangle\left(\triangledown I\left(\mathbf{x} + \frac{\mathbf{u}}{\mathbf{d_I}\left(\mathbf{x}\right)}\right), \triangledown I\left(\mathbf{x} + \frac{\mathbf{v}}{\mathbf{d_I}\left(\mathbf{x}\right)}\right)\right), \tag{22}$$

where $\triangledown I\left(\mathbf{x}\right)$ is the gradient of depth $d_I(\mathbf{x})$ at pixel $\mathbf{x}$. In fact, the new feature $g_\theta(I, \mathbf{x})$ represents the angle between the two gradient vectors at offsets $\mathbf{u}$ and $\mathbf{v}$ from $\mathbf{x}$.

In the second place, we compare our proposed segmentation approach with the state-of-the-art Random Walks (RWalks) image segmentation algorithm [16], both applied to the probability maps returned by the execution of the RF method. Since RWalks is designed to segment still images, no temporal coherence is taken into account in this approach. Furthermore, besides the probability maps, RWalks also receives some user-designed seeds, since it is semi-automatic. However, in order to perform a fair comparison between this method and our proposal, which is fully-automatic, we need to automatize the seed-selection process. For this task, we select the seeds for each label as the pixels with greatest probability value. When applying GC,

[3] The data set is public at http://www.cvc.uab.cat/~ahernandez/data.html

the λ parameter was set to 50 for all the performed experiments, the nodes of the graph are 10-connected – 8 spatial neighbors + 2 temporal neighbors–, and the size of the sliding window is set to $|V| = 5$. In order to achieve a more appropriate comparison of the results, we perform an additional GC experiment. It consists of removing the temporal coherence, i.e., segmenting each frame of the sequence independently, using a 2-D lattice graph topology. In this frame-by-frame approach, the α-expansion algorithm is used, since the boundary term is *metric* with this new topology. Moreover, in this second experiment we also compare the use of different pixel information for the computation of the boundary term. Apart from depth information alone, we also test using RGB information only, as in the standard GrabCut algorithm from [22], as well as RGB and depth together. For this last approach, we normalize the depth information in the range $[0...255]$, and concatenate it with the RGB information, resulting in a 4-dimensional RGBD vector per pixel. Finally, we also apply the Friedman test [14] in order to look for statistical significance of the performed experiments.

3.1. Random forest results

Table 2 shows the estimated average classification accuracy for each of the considered labels. The most likely label predicted for a pixel is chosen to be the one that corresponds to the maximum of the inferred RF probabilities for that pixel. Without claiming exhaustiveness of our experiments, the results from Table 2 allow us to make the following analysis: The upper limit O_{max} for the module of $\mathbf{u}$ and $\mathbf{v}$ offsets has the greatest impact on the accuracy results at the hands regions, which have the smallest area in our body part definition. Doubling O_{max} leads to an increase in the accuracy of about 20% for the hands and about 6% for the other body parts. In other words, O_{max} increases the feature diversity and the global ability to represent spatial detail. The number of candidate offset pairs θ would not have such a tremendous impact on the accuracy as the O_{max} parameter, though a higher number of θ candidates would help in identifying the most discriminative features. A decrease of the number of candidates from 100 to 80 features drops the hands accuracy with $1 - 3\%$.

We also tested the impact of L_{max}, the depth level limit of the decision trees. Trimming the trees to level 15 has a very little impact, showing an improvement of 0.1% on the average accuracy. The latter may weakly be attributed to better classification at the lower arm regions. Trimming to depth level 10 shows a 4% decrease in the accuracy at the hands. Our analysis indicates that we may be witnessing slight overfitting at tree depth level of 20 due to the small amount of training images. Our final test includes comparison over combination of both features f_θ and g_θ of Eq. (1) and Eq. (22). Since the depth data provided by Kinect is noisy, we apply a Gaussian smoothing filter before calculating the image gradients and the g_θ features. We chose the gradient feature since it complements the relations of depth features with information about the orientation of local surfaces. However, in our test we did not find significant improvement in the performance results of the RF approach.

In order to show the generalization capability of the proposed approach, we carried out an extra case study, stressing on the segmentation of the finger regions. For this test we only considered a manual annotated set of 63 depth image frames without including temporal coherence. The results applying the same validation as in the previous case show the best performance for the following setup: 1 tree of depth 15, 500 pixels per image, 100 candidate offset pairs θ, 20 candidate thresholds τ, and $O_{max} = 45$. The estimated average per class accuracy was 58.5%, mostly due to the small number of training images. Fig. 5 displays a couple of test images comparing the ground truth and the inferred labels for the fingers and hands. Reviewing the classification results from both use cases, the body parts and finger regions, we observe that some of the errors appear due to left/right uncertainty. Nevertheless, the results are promising, showing the generalization ability of the presented approach for general multi-class labelling in depth images.

3.2. Graph-cuts results

The results we obtained when applying GC over the probabilities returned by the RF are detailed in Table 3. We can see how these results improved the labelling obtained by the RF approach, and also the one obtained in the frame-by-frame approach. Moreover, all the GC approaches shown in Table 3 outperform the results obtained using the RWalks segmentation algorithm in most of the body parts. If we take a closer look at the measurements, we can see that we obtain the best results when using only depth information for the computation of the boundary potential. In our case study, adding RGB to the depth information reduces the generalization of the boundary potential. In Fig. 4 we can see some qualitative results of the segmentations.

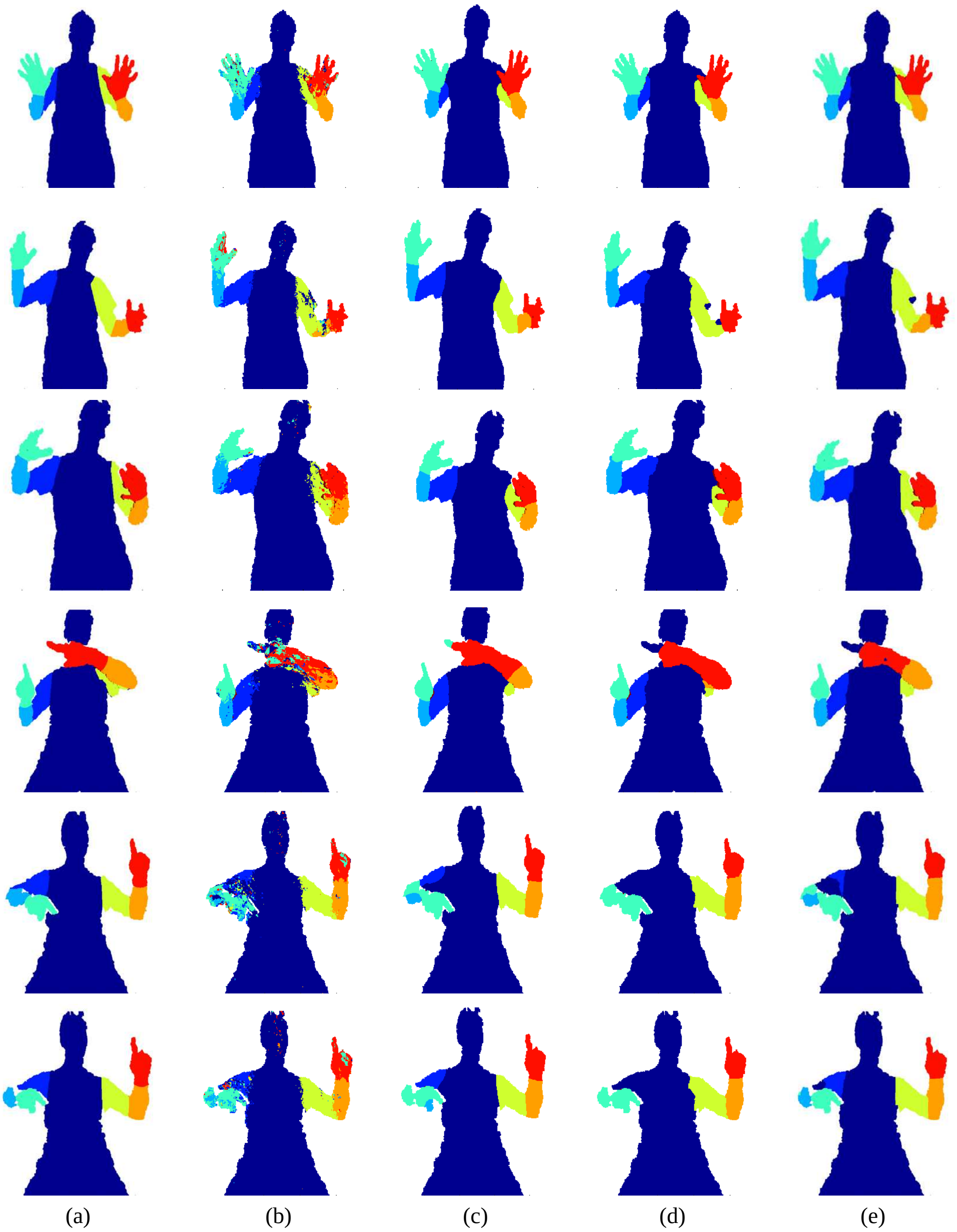


Fig. 4. Qualitative results; Ground Truth (a), RF inferred results (b), RWalks results (c), frame-by-frame GC results (d), and Temporally-coherent GC results (e).

Table 2

Average per class accuracy in % calculated over the test samples in a 5-fold cross validation. f_θ represents features of the depth comparison type from Eq. (1), while g_θ - the gradient comparison feature from Eq. (22). O_{max} is the upper limit of the $\mathbf{u}$ and $\mathbf{v}$ offsets, and L_{max} stands for the maximal depth level of the decision trees

	Torso	LU arm	LW arm	L hand	RU arm	RW arm	R hand	Avg.
100 f_θ, O_{max} = 30, L_{max} = 20	92.90	73.29	71.42	57.75	74.25	76.26	59.38	72.18
100 f_θ, O_{max} = 60, L_{max} = 20	94.17	79.83	77.69	**77.10**	81.04	82.65	80.17	81.81
80 f_θ, O_{max} = 60, L_{max} = 20	**94.22**	79.08	76.46	74.19	81.24	83.26	79.05	81.07
60 f_θ, O_{max} = 60, L_{max} = 20	94.09	78.86	75.86	73.49	79.43	82.60	78.08	80.34
100 f_θ, O_{max} = 60, L_{max} = 15	94.06	79.81	78.69	76.59	81.18	83.10	**80.23**	**81.95**
100 f_θ, O_{max} = 60, L_{max} = 10	91.83	**81.47**	**78.98**	72.30	**83.00**	**83.74**	76.85	81.17
60 f_θ + 20 g_θ, O_{max} = 60, L_{max} = 20	94.04	77.73	74.93	71.97	77.62	81.22	76.64	79.17

Table 3

Average per class accuracy in % obtained when applying the different GC approaches –TC: Temporally coherent, Fbf: Frame-by-Frame– , and the best results from the RF probabilities [25] and the RWalks segmentation algorithm [16], in the first and second rows, respectively.

	Torso	LU arm	LW arm	L hand	RU arm	RW arm	R hand	Avg. per class
RF results	94.06	**79.81**	78.69	76.59	81.18	83.10	80.23	81.95
RWalks results	**99.05**	72.17	81.04	86.98	73.27	88.48	91.68	84.67
TC, Depth, $\Omega_2\left(L_i, L_j\right)$	98.44	78.93	**84.38**	88.32	**82.57**	**88.85**	93.86	**87.91**
Fbf, Depth, $\Omega_1\left(L_i, L_j\right)$	98.86	75.05	82.87	91.45	77.57	87.35	93.96	86.73
Fbf, Depth, $\Omega_2\left(L_i, L_j\right)$	98.86	75.03	83.36	**92.41**	77.54	87.67	**94.20**	87.01
Fbf, RGB+Depth, $\Omega_1\left(L_i, L_j\right)$	99.02	72.02	81.86	90.29	76.56	86.84	92.14	85.53
Fbf, RGB+Depth, $\Omega_2\left(L_i, L_j\right)$	99.02	72.03	81.95	91.19	76.53	87.12	92.12	85.71

Another interesting result is the influence of the prior costs given by the different $\Omega\left(L_i, L_j\right)$ functions. Clearly, when introducing spatial coherence with $\Omega_2\left(L_i, L_j\right)$, we obtain better results, especially in the segmentation of the hands, which are the parts with more confusion among all. Fig. 6 shows a qualitative example of both approaches.

A more detailed analysis of the results from the temporally-coherent approach reveals that the highest improvement is obtained in the case of the upper part of the limbs. In contrast, the results related to both the left and right hands are slightly worse than the frame-by-frame approach. However, hands are the most moving body parts in the video sequences, and the time lapse between one frame and the next one can be too large, inducing the introduction of some noise.

Taking a look at the qualitative results in Fig. 4, one can first see how the spatial coherence introduced by the basic frame-by-frame GC approach –Fig. 4 (d)– allows to recover more consistent regions than the ones obtained with just the RF probabilities, in such a way that each limb is represented by just one blob. Moreover, when introducing temporal coherence –Fig. 4 (e)–, the classification of certain labels like the ones corresponding to the arms is more accurate compared to the results obtained without temporal coherence. The RWalks algorithm –Fig. 4 (c)– obtains accurate segmentations when the RF probabilities have low noise, but it fails in the opposite case, though in the shown cases it seems to perform better than the frame-by-frame GC approach. Furthermore, RWalks is prone to confuse the labels between the right and left body limbs, since no label consistency is enforced.

Finally, we use the Friedman test [14] to show that the results are not affected by randomness. For this purpose, we compute the ranks of each segmentation strategy in Table 3 independently for each segmentation label –and also for the average. We define the computation of the ranks for a certain label as one “experiment”. More specifically, the rankings are obtained estimating each relative rank r_i^j for each label i and each segmentation strategy j, and computing the mean ranking R for each strategy as $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j$ with $N = |L| + 1$, where $|L|$ is the total number of possible labels. The Friedman statistic value is then computed as follows:

$$X_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]. \quad (23)$$

In our case, with $k = 7$ segmentation strategies to compare, $N = 8$ different experiments, and ranks $R = [5.63, 4.88, 2.5, 3, 2.75, 4.75, 4.5]$ in the row order of Table 3,$X_F^2 = 15.48$. Since this value is undesirable conservative, Iman and Davenport proposed a corrected statistic:

$$F_F = \frac{(N-1)X_F^2}{N(k+1) - X_F^2}. \quad (24)$$

Applying this correction we obtain $F_F = 3.38$. With seven strategies and eight experiments, F_F is distributed according to the F distribution with six and 42 degrees of freedom. The critical value of $F(6, 42)$ for 0.05 is 2.23. As the value of F_F is higher than 2.23 we can reject the null hypothesis, and thus, looking at the best mean performance in Table 3, we can conclude that the spatio-temporal GC proposal is the best choice from the presented experiments.

In the second experiment, labelling pixels from hands –in a frame-by-frame fashion, we achieve an average per class accuracy of 70.9%, which supposes even a greater improvement than in the case of human limbs. Fig. 5 shows some qualitative results of the GC approach, where we can appreciate that regions are more consistent and better defined than in the case of just using RF probabilities. It is worth mentioning that for this experiment we used $\Omega_1(L_i, L_j)$ as the cost function between labels, and yet we obtained consistent results.

4. Conclusion

We proposed a generic framework for object segmentation using depth maps based on Random Forest and Graph-cuts theory in order to benefit from the use of spatial and temporal coherence, and applied it to the segmentation of human limbs. Random Forest estimated the probability of each depth sample point to belong to a set of possible object labels, while Graph-cuts was used to optimize, both spatially and temporally the RF probabilities. Results on two novel data sets showed high performance segmenting several body parts in depth images compared to state-of-the-art approaches.

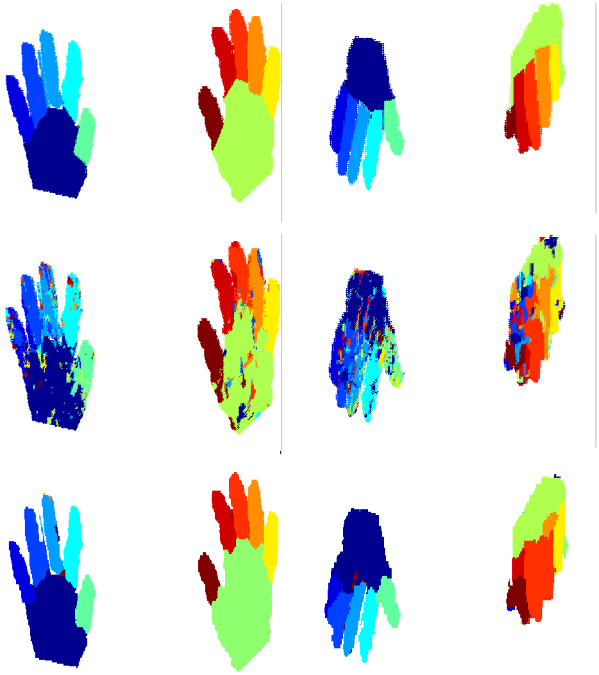


Fig. 5. Results from RF classification in the case of hands. First row shows the ground-truth for two examples. Second row shows the RF classification results. Third row shows the final α-expansion GC segmentation results.

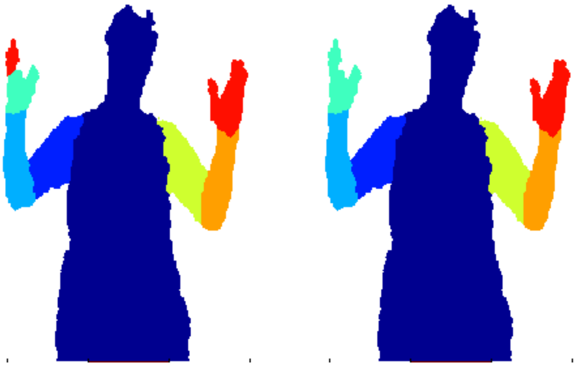


Fig. 6. Comparison of results without (a) and with (b) spatially-consistent labels.

As future work, we plan to increase the available data for improving pixel label inference, and to consider different multi-label object problems from depth maps. We also plan to explore other ways to deal with the temporal coherence, by using different graph topologies and different energy potential formulation, as well as to use the proposed method for real smart environment applications.

Acknowledgements

This work has been partially supported by the following projects: TIN2009-14404-C02, CONSOLIDER-INGENIO CSD 2007-00018, contract DO-02-275/2008 of the National Science Fund at the Bulgarian Ministry of Education, Youth & Science, and BG051PO001-3.3.04 /13 by the Operative Programme "Human Resources Development" of the European Social Fund. The work of Antonio is supported by an FPU fellowship from the Spanish government.

References

[1] Code laboratories cl nui platform - kinect driver/sdk. http://codelaboratories.com/nui/.

[2] Flexible action and articulated skeleton toolkit (faast). http://projects.ict.usc.edu/mxr/faast/.

[3] Nite middleware. http://www.primesense.com/?p=515.

[4] Openkinect (libfreenect). http://openkinect.org/, .

[5] Openni. http://www.openni.org, .

[6] Primesensor. http://www.primesense.com/?p=514.

[7] Kinect for windows sdk from microsoft research. http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/.

[8] Y. Boykov, O.Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:1222–1239, November 2001. ISSN 0162-8828.

[9] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal on Computer Vision*, 70:109–131, 2006. ISSN 0920-5691.

[10] G. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A. Real time system for robust 3d voxel reconstruction of human motions. 2:714–720, 2000. IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island,(USA),.

[11] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. volume 2, pages 886–893, 2005.

[12] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, pages 7–13, 2006.

[13] T. Darrell, G. Gordon, J. Woodfill, and M. Harville. Integrated person tracking using stereo, color, and pattern detection. *IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara*, pages 601 –608, 1998.

[14] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006. ISSN 1532-4435.

[15] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 755–762, 2010.

[16] Leo Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, November 2006. ISSN 0162-8828.

[17] A. Hernandez, M. Reyes, S. Escalera, and P. Radeva. Spatio-temporal grabcut human segmentation for face and pose recovery. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 33 –40, june 2010. doi: 10.1109/CVPRW.2010.5543824.

[18] L. Igual, J.C. Soliva, A Hernandez-Vela, S Escalera, X Jimenez, O. Vilarroya, and P. Radeva. A fully-automatic caudate nucleus segmentation of brain mri: Application in volumetric analysis of pediatric attention-deficit/hyperactivity disorder. 10(105), december 2011. doi: 10.1186/1475-925X-10-105.

[19] HP. Jain and A. Subramanian. Real-time upper-body human pose estimation using a depth camera. *HP Technical Reports*, 1(190), 2010.

[20] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and Theobalt C. Markerless motion capture of interacting characters using multi-view image segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 14(1):1249–1256, 2011.

[21] J. Rodgers, D. Anguelov, H.-C. Pang, and D. Koller. Object pose detection in range scan data. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2445 – 2452, 2006.

[22] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, pages 309–314, 2004.

[23] B. Sabata, F. Arman, and J. K. Aggarwal. Segmentation of 3d range images using pyramidal data structures. *CVGIP: Image Understanding*, 57(3):373–387, 1993.

[24] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis. Human detection using partial least squares. pages 24–31, 2009. International Conference on Computer Vision.

[25] 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. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[26] Leo Breiman Statistics and Leo Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[27] Evan Suma, Belinda Lange, Albert Rizzo, David M. Krum, and Mark Bolas. FAAST: the flexible action and articulated skeleton toolkit. In *IEEE Virtual Reality*, pages 245–246, Singapore, March 2011.

[28] B. Dariush Y. Zhu and K. Fujimura. Controlled human pose estimation from depth image streams. *Computer Vision and Pattern Recognition Workshop on TOF Computer Vision*, pages 1–8, 2008.

[29] HD. Yang and S.W. Lee. Reconstruction of 3d human body pose from stereo image sequences based on top-down learning. *Pattern Recognition*, 40(11):3120–3131, 2007.