HUMAN-ROBOT INTERACTION WITH DEPTH-BASED GESTURE RECOGNITION

Gabriele Pozzato, Stefano Michieletto, Emanuele Menegatti  
IAS-Lab  
University of Padova  
Padova, Italy  
Email: pozzato.g@gmail.com, {micheletto, emg}@dei.unipd.it

Fabio Dominio, Giulio Marin, Ludovico Minto  
Simone Milani, Pietro Zanuttigh  
Multimedia Technology and Telecommunication Lab  
University of Padova  
Padova, Italy  
Email: {dominiol,mariniu,mintolud, simone.milani, zanuttigh}@dei.unipd.it

Abstract—Human robot interaction is a very heterogeneous research field and it is attracting a growing interest. A key building block for a proper interaction between humans and robots is the automatic recognition and interpretation of gestures performed by the user. Consumer depth cameras (like MS Kinect) have made possible an accurate and reliable interpretation of human gestures. In this paper a novel framework for gesture-based human-robot interaction is proposed. Both hand gestures and full-body gestures are recognized through the use of depth information, and a human-robot interaction scheme based on these gestures is proposed. In order to assess the feasibility of the proposed scheme, the paper presents a simple application based on the well-known rock-scissors-paper game.

Keywords—Human-Robot Interaction, Gesture Recognition, Depth, Kinect, SVM

I. INTRODUCTION

Human-Robot Interaction (HRI) is a field of investigation that is recently gaining particular interest, involving different research areas from engineering to psychology [1]. The development of robotics, artificial intelligence and human-computer interaction systems has brought to the attention of researchers new relevant issues, such as understanding how people perceive the interaction with robots [2], how robots should look like [3] and in what manners they could be useful [4].

In this work we focused on the communication interface between humans and robots. The aim is to apply novel approaches for gesture recognition based on the analysis of depth information in order to allow users to naturally interact with robots [5]. This is a significant and sometimes undervalued perspective for human-robot interaction.

Hand gesture recognition using vision-based approaches is an intriguing topic that has attracted a large interest in the last years [6]. It can be applied in many different fields including robotics and in particular human-robot interaction. A human-to-machine communication can be established by using a predefined set of poses or gestures recognized by a vision system. It is possible to associate a precise meaning (or a command the robot will perform) to each sign. This permits making the interplay with robots more intuitive and enhancing the human-robot interaction up to the level of a human-human one. In this case also the robot plays a different role. During social interactions it must have an active part in order to really engage the human user. For example, recognizing body language, a robot can understand the mood of the human user adapting its behavior to it [7].

Until a few years ago, most approaches were based on videos acquired by standard cameras [6], [8]; however, relying on a simple 2D representation is a great limitation since that representation is not always sufficient to capture the complex movements and inter-occlusions of the hand shape.

The recent introduction of low cost consumer depth cameras, e.g., Time-Of-Flight (ToF) cameras and MS Kinect™ [9], has opened the way to a new family of algorithms that exploit the depth information for hand gesture recognition. The improved accuracy in modelling three-dimensional body poses has permitted removing the need for physical devices in human-robot interaction such as gloves, keyboards or other controllers.

HRI system based on depth cameras apply machine-learning techniques to a set of relevant features extracted from the depth data. The approach in [10] computes a descriptor from silhouettes and cell occupancy features; this is then fed to a classifier based on action graphs. Both the approaches in [11] and [12] use volumetric shape descriptors processed by a classifier based on Support Vector Machines (SVM). Another possibility is to exploit the histograms of the distance of hand edge points from the hand center in order to recognize the gestures [13], [14], [15]. Different types of features can also be combined together [16].

The paper is organized in the following way. Sections [17] presents the proposed hand gesture recognition strategy, while Section [18] describes the body-pose detection algorithm. These tools are integrated in a human-robot interaction scheme, which is described in Section [19] A case study referring to a simple rock-paper-scissors game is presented in Section [20] Finally Section [21] draws the conclusions.

II. HAND GESTURE RECOGNITION FROM DEPTH DATA

Fig. [1] shows a general overview of the employed hand gesture recognition approach. The approach is based on the method proposed in [16]. In the first step, the hand is extracted
from the depth acquired by the Kinect. Then, a set of relevant features is extracted from the hand shape. Finally a multi-class Support Vector Machine classifier is applied to the extracted features in order to recognize the performed gesture.

A. Hand Recognition

In the first step the hand is extracted from the color and depth data provided by the Kinect. Calibration information is used to obtain a set of 3D points \(X_i\) from the depth data. The analysis starts by extracting the point in the depth map closest to the user \((X_c)\). The points that have a depth value close to the one of \(X_c\) and a 3D distance from it within a threshold of about 20\,cm are extracted and used as a first estimate of the candidate hand region. A further check on hand color and size is then performed to avoid to recognize possible objects positioned at the same distance as the hand. At this point the highest density region is found by applying a Gaussian filter with a large standard deviation to the mask representing the extracted points. Finally, starting from the highest density point, a circle is fitted on the hand mask such that its area roughly approximates the area corresponding to the palm (see Fig. 2). Principal Component Analysis (PCA) is then applied to the detected points in order to obtain an estimate of the hand orientation. Finally the hand points are divided into the palm region (set \(P\), i.e., the points inside the circle), the fingers (set \(F\), i.e., the points outside the circle in the direction of the axis pointing to the fingertips found by the PCA), and the wrist region (set \(W\), i.e., the points outside the circle in the opposite direction w.r.t. the fingertips).

B. Feature Extraction

Two different types of features are computed from data extracted in the previous step. The first set of features consists of a histogram of the distances of the hand points from the hand center. Basically, the center of the circle is used as reference point and an histogram is built, as described in [11], by considering for each angular direction the maximum of the point distances from the center, i.e.:

\[
L(\theta_q) = \max_{X_i \in \mathcal{I}(\theta_q)} \|X_i\|,
\]

where \(\mathcal{I}(\theta_q)\) is the angular sector of the hand corresponding to the direction \(\theta_q\) and \(\|X_i\|\) is the distance between point \(X_i\) and the hand center. The computed histogram is compared with a set of reference histograms \(L_g^d(\theta)\), one for each gesture \(g\). The maximum of the correlation between the current histogram \(L(\theta_q)\) and a shifted version of the reference histogram \(L_g^d(\theta)\) is used in order to compute the precise hand orientation and refine the results of the PCA. A set of angular regions \(I(\theta_{g,j})\) associated to each finger \(j = 1, \ldots, 5\) in the gesture \(g\) is defined on the histogram, and the maximum is selected as feature value inside each region, normalized by the middle finger’s length:

\[
f_{g,j}^d = \max_{I(\theta_{g,j})} \frac{L_g^d(\theta)}{L_{max}}
\]

where \(g = 1, \ldots, G\) for an alphabet of \(G\) gestures. Note that the computation is performed for each of the candidate gesture, thus obtaining a different feature value for each of the candidate gestures.

The second feature set is based on the curvature of the hand contour. This descriptor is based on a multi-scale integral operator [17], [18] and is computed as described in [16] and [15]. The algorithm takes as input the edges of the palm and fingers regions and the binary mask representing the hand region on the depth map. A set of circular masks with increasing radius centered on each edge sample is built (we used \(S = 25\) masks with radius varying from 0.5 cm to 5 cm, notice that the radius corresponds to the scale level at which the computation is performed).

The ratio between the number of samples falling inside the hand shape and the size of the mask is computed for each circular mask. The values of the ratios represent the local curvature of the edge and range from 0 for a convex section of the edge to 1 for a concave one, with 0.5 corresponding to a straight edge. The \([0,1]\) interval is then quantized into \(N\) bins and the feature values \(f_{g,s}^e\) are computed by counting the number of edge samples having a curvature value inside bin \(b\) at scale level \(s\).
C. Hand Gesture Classification

The feature extraction approach provides two feature vectors, each one describing relevant properties of the hand samples. In order to recognize the performed gestures, the two vectors are concatenated and sent to a multi-class Support Vector Machine classifier. The target is to assign the feature vectors to \( G \) classes (with \( G = 3 \) for the sample rock-scissors-paper game of the experimental results) corresponding to the various gestures of the considered database. A multi-class SVM classifier based on the one-against-one approach has been used, i.e., a set of \( G(G − 1)/2 \) binary SVM classifiers are used to test each class against each other and each output is chosen as a vote for a certain gesture. The gesture with the maximum number of votes is selected as the output of the classifier. In particular we used the SVM implementation in the LIBSVM package [19], together with a non-linear Gaussian Radial Basis Function (RBF) kernel tuned by means of grid search and cross-validation on a sample training set.

III. BODY GESTURE RECOGNITION FROM SKELETAL TRACKING

In this work skeleton tracking capabilities come from a software development kit released by OpenNI. In particular, the skeletal tracking algorithm is implemented in a freeware middleware called NITE built on top of the OpenNI SDK.

NITE uses the information provided by a RGB-D sensor to estimate the position of several joints of the human body at a frame rate of 30 fps that is a good trade-off between speed and precision. Differently from approaches based only on RGB images, these kind of sensors based also on depth data, allow the tracker to be more robust with respect to illumination changes.

The skeleton information provided by the NITE middleware consists of \( N = 15 \) joints from head to foot. Each joint is described by its position (a point in 3D space) and orientation (using quaternions). On these data, we perform two kinds of normalization: firstly we scale the joints positions in order to normalize the skeleton size, thus achieving invariance among different people, then we normalize every feature to zero mean and unit variance. Starting from the normalized data, we extract three kinds of descriptors: a first skeleton descriptor (\( d_P \)) is made of the set of joints positions concatenated one to each other, the second one (\( d_O \)) contains the normalized joints orientations, finally, we considered also the concatenation of both positions and orientations (\( d_{TOT} \)) of each joint:

\[
d_P = [x_1 \ y_1 \ z_1 \ \ldots \ x_N \ y_N \ z_N],
\]

\[
d_O = [q_1^1 \ q_2^1 \ q_3^1 \ q_4^1 \ \ldots \ q_N^2 \ q_N^3 \ q_N^4],
\]

\[
d_{TOT} = [d_P^1 \ d_O^1 \ \ldots \ d_P^N \ d_O^N].
\]

These descriptors have been provided as features for the classification process. A Gaussian Mixture Model (GMM)-based classifier has been used in order to compare the set of predefined body gestures to the ones performed by the user. The described technique could be extended by considering a sequence of descriptors equally spaced in time in order to apply the resulting features to human activity recognition [20], [21] and broaden the interaction capabilities of our framework.

IV. ROBOT CONTROL AND SYSTEM INTEGRATION

The information provided by both hand and skeletal gesture recognition systems are analyzed in order to obtain a proper robot motion. Robot behavior really depends on the application, but the complete framework can be separated in different layers in order to easily extend the basic system to new tasks. We created three independent layers: gesture recognition, decision making, robot controller. These layers are connected one to each other by means of a very diffuse robotic framework: Robot Operating System.

Robot Operating System (ROS) [22] is an open-source, meta-operating system that provides, together with standard services peculiar to an operating system (e.g., hardware abstraction, low-level device control, message-passing between processes, package management), tools and libraries useful for typical robotics applications (e.g., navigation, motion planning, image and 3D data processing). The primary goal of ROS is to support the reuse of code in robotics research and, therefore, it presents a simplified but lightweight design with clean functional interfaces.

During a human-robot interaction task, the three layers play different roles. The gesture recognition layer deals with the acquisition of information from a camera in order to recognize gestures performed by users. The decision making layer is on a higher level of abstraction. Information is processed by ROS and classified by an AI algorithm in order to make the robot take its decision. Finally, the robot controller layer is responsible of motion. This layer physically modifies the behavior of the robot sending commands to the servomotors. Even though they are strictly independent also in what concerns implementation, the three levels can interact sending or receiving messages through ROS. This means that we can use different robots, AI algorithms or vision systems by maintaining the same interface.

V. CASE STUDY: A SIMPLE ROCK-PAPER-SCISSORS GAME

In order to test the developed framework a very simple even popular game has been used, namely rock-paper-scissors. Users are asked to play against a robotic opponent on several rounds. At each round, both the players (human and robot) have to perform one of the 3 gestures depicted in Fig. 1. The winner is chosen according to the well-known rules of the game. Game settings and moves can be selected by human users without any remote controls or keyboards: only the
The proposed approach has achieved an average accuracy of 99% with the robot. The performances are evaluated for each person without any cross-validation approach. In order to evaluate its effectiveness we asked 14 different people to perform the 3 gestures 10 times each.

Notice that the approach of Section [II] has been developed for more complex settings with a larger number of gestures, has very good performances in this sample application. In order to evaluate its effectiveness we asked to 14 different people to perform the 3 gestures 10 times each for a total of $14 \times 10 \times 3 = 420$ different executions of the gestures. Then we performed a set of tests, each time training the system on 13 people and leaving out one person for the testing. In this way (similar to the cross-validation approach) the performances are evaluated for each person without any training from gestures performed by the same tested user, as it will happen in a real setting where a new user starts interacting with the robot. The proposed approach has achieved an average accuracy of 99.28% with this testing protocol.

VI. CONCLUSIONS

In this paper a framework for gesture-based human-robot interaction has been proposed. Two different approaches for the exploitation of depth data from low-cost cameras have been proposed for both the recognition of hand and full-body gestures. Depth-based gesture recognition schemes allow a more natural interaction with the robot and a sample application based on the simple rock-scissor-paper game has been presented. Further research will be devoted to the application of the proposed framework in more complex scenarios.

ACKNOWLEDGEMENTS

Thanks to the Seed project Robotic 3D video (R3D) of the Department of Information Engineering and to the University project 3D Touch-less Interaction of the University of Padova for supporting this work.

REFERENCES


