HAL
open science

Fingers gestures early-recognition with a unified framework for RGB or depth camera

Sotiris Manitsaris, Apostolos Tsagaris, Alina Glushkova, Fabien Moutarde, Frédéric Bevilacqua

To cite this version:

Sotiris Manitsaris, Apostolos Tsagaris, Alina Glushkova, Fabien Moutarde, Frédéric Bevilacqua. Fingers gestures early-recognition with a unified framework for RGB or depth camera. 3rd International Symposium on Movement and Computing (MOCO'2016), Jul 2016, Thessalonique, Greece. pp.26, 10.1145/2948910.2948947 . hal-01425895

HAL Id: hal-01425895

https://hal.science/hal-01425895v1

Submitted on 9 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fingers gestures early-recognition with a unified framework for RGB or depth camera

Sotiris Manitsaris
MINES ParisTech, Robotics Lab, 60 Boulevard Saint-Michel, 75272 Paris, France
sotiris.manitsaris@mines-paristech.fr

Apostolos Tsagaris
University of Macedonia, Multimedia Technology and Computer Graphics Lab, 156 Egnatia Street, GR-54006 Thessaloniki, Greece
tsagaris@uom.edu.gr

Alina Glushkova
University of Macedonia, Multimedia Technology and Computer Graphics Lab, 156 Egnatia Street, GR-54006 Thessaloniki, Greece
alina.glushkova@uom.edu.gr

Fabien Moutarde
MINES ParisTech, Robotics Lab, 60 Boulevard Saint-Michel, 75272 Paris, France
fabien.moutarde@mines-paristech.fr

Frédéric Bevilacqua
IRCAM, Real-Time Musical Interactions Team, 1, place Igor-Stravinsky, 75004 Paris, France
frederic.bevilacqua@ircam.fr

ABSTRACT
This paper presents a unified framework computer vision approach for finger gesture early recognition and interaction that can be applied on sequences of either RGB or depth images without any supervised skeleton extraction. Either RGB or time-of-flight cameras can be used to capture finger motions. The hand detection is based on a skin color model for color images or distance slicing for depth images. A unique hand model is used for the finger detection and identification. Static (fingerings) and dynamic (sequence and/or combination of fingerings) patterns can be early-recognized based on one-shot learning approach using a modified Hidden Markov Models approach. The recognition accuracy is evaluated in two different applications: musical and robotic interaction. In the first case standardized basic piano-like finger gestures (ascending/descending scales, ascending/descending arpeggio) are used to evaluate the performance of the system. In the second case, both standardized and user-defined gestures (driving, waypoints etc.) are recognized and used to interactively control an automated guided vehicle.

Keywords
Finger motion patterns; early recognition; unified framework; computer vision; standardized interaction; user-defined interaction

INTRODUCTION
An important recent trend in Human-Computer Interaction (HCI) is the evolution towards more natural and/or dematerialized interaction: touch screens and gesture recognition tend to replace keyboard and mouse. Communication patterns that are used in the everyday life of human beings can be a basis to develop novel natural communication modalities between humans and computers. An understanding of the ways a human being communicates and interacts with his environment requires, among others, knowledge of non-verbal behavior produced by gestures that are involved in various activities. Hands and fingers constitute one of the most frequently used parts of the human body and their use for HCI is of a very high interest and potential. However, an interaction based on the use of hands and fingers constitutes a challenging issue since fingers are the body part with the highest number of degrees of freedom. In particular, finger-based interaction might require computing systems to capture, model, and early recognize finger motion patterns with various hand poses.

This paper presents a unified computer vision framework for fingers gestures early recognition and interaction applied to either RGB or depth image sequences. It aims to recognize finger motion patterns for hand with a semi-closed palm while fingers are simultaneously moving. The framework can be used either with RGB or depth cameras in order to recognize the finger gestural intentions of the user while he is interacting with systems. Therefore, the framework can provide solution with low-cost sensor (e.g. webcam) for real-time applications when the lighting conditions in the room are well-controlled, while it can also provide access to more precise information with more expansive sensors such as time-of-flight (ToF) depth cameras. This approach is evaluated with both standardised

motion patterns that are already used in Musical Interaction (MI), such as piano-like finger gestures, but also with user-defined motion patterns that can be used for the Human-Robot Interaction (HRI), such as the control of Automated Guided Vehicles. The machine-learning and pattern recognition is based on a hybrid approach combining HMMs-DTW, (Hidden Markov Models (HMMs) and Dynamic Time Warping), that enables early recognition and prediction using one-shot learning. Both computer vision approaches are fully compared since the evaluation is done within the same context based on common criteria.

STATE OF THE ART

Motion Capture

To capture hand motions, RGB low cost cameras can be used. Image processing techniques are applied to the image sequences. After the hand's skin is detected, different filters are applied in each image to segment the scene [1]. The most important limitations of RGB sensors is that they are subject to occlusions and are highly impacted by illumination variation. To reduce the latter problem, processing for hand gesture recognition is sometimes performed on infrared vision using external infrared light source, like in the example of U. Solanki [2]. However, this does not totally suppress illumination sensitivity, it doesn't give any information about the third dimension and the quality of the image also depends on the nature of the objects on the scene. Another possibility is to use depth cameras, as most of them are based on specific illumination systems using near-infrared light, and can be less perturbed by external light conditions. Kinect is the most popular low-cost depth sensor based on structured light, a technique where a known pattern is projected onto the surface to be analyzed, then estimating the corresponding depth by triangulation [3]. Despite the presence of a "near mode" implemented in the Kinect SDK software, this sensor is rather adapted to capture the whole body movements at a range of 1,2 to 3,5 meters and it is less precise for hand gestures. The PMD Cam Board Nano Time-of-Flight (ToF) depth camera has a much smaller effective range of 5cm to 50 cm and its depth images are of a very high precision. It uses a laminar modulated infrared light to measure the distance for each camera-pixel at the same time [4], [5]. The only way to obtain absolute robustness to light variations and totally avoid occlusion problem is to capture XYZ rotations of body and hand segments with inertial sensors integrating gyroscopes, accelerometers and magnetometers. Inertial gloves may provide an occlusion-independent hand and finger tracking with a high precision. However, wearing inertial sensors on body, hands, or even worse on fingers, is much too intrusive in many applications.

Leveraging on this, we propose a unified framework for finger gesture early recognition supporting both RGB and depth cameras. The use of low cost RGB cameras renders the framework available to the large public supporting off-the-shelf sensors. For cases where lighting conditions may vary considerably, depth cameras can provide image sequences of a high quality. The possibility to use an RGB camera equipped with infrared LEDs and a filter has been also studied. The disadvantage of this kind of sensor is that even with a high-resolution camera and the appropriate lens, the image quality can be good even beyond the 30 cm but the important information of the hand color is lost. Finally, inertial sensors have not been used since they remain cumbersome to use for finger movements

Hand segmentation and gesture descriptors

In order to extract gesture descriptors from the image, a hand segmentation phase is needed. One of the most widely used techniques is the thresholding. It divides pixels of the image belonging to the background and those potentially belonging to the hand. It uses a metric value, creating thus a binary mapping of the scene modeling both groups: foreground and background [6]. Thresholding is mostly used with depth sensors. Regarding RGB cameras, skin color models are used to detect hand regions on the image by finding pixels candidates with the highest probabilities for belonging to the hand [7]. However, this method however is light sensitive.

Different gesture descriptors can be used for recognition of hand and fingers gestures. They can be divided into 3 categories: appearance-based, model-based and feature-based. Among the most important appearance-based approaches we can find the Shotton et al. approach [8] where the body's joints 3D positions can be predicted in real-time from a single depth image of a Kinect XBOX depth camera. The pose estimation problem is transformed into a per-pixel classification issue that is solved with machine-learning techniques, with a large dataset of images for training. This method is applied to detect body joints and is less adapted for hands detection. Keskin et al. [9] proposed a method for hand pose estimation using the approach of object recognition by parts after the training of a Random Decision Forest with synthetic images. Depth images from Kinect are used for the recognition of sign language hand postures. Another hand skeletal model for depth images from PMD Cam Board Nano, applied to capture music-like finger gestures has been proposed by Dapogny et al. [10]. This work is also based on a Random Decision Forest learning technique, on a classification model from reduced training dataset as well as on a method for spatial aggregation of the classification results.

In this paper, the feature-based approach is applied for fingertip detection using either RGB or depth images. No skeletal model is required for the recognition of finger gestures. A comparison on a recognition level is provided in evaluation section.

Machine-learning for gesture recognition

In order to recognize the gestures, machine-learning is a crucial phase of the process. Several methods have been proposed for gesture recognition as reviewed by [11] and [12], such as Hidden Markov Models (HMMs) and Dynamic Time Warping (DTW). HMMs are stochastic

models that have been widely applied in gesture recognition. As exposed by Rabiner [13], HMMs compute the likelihoods between a given gesture and a pre-defined set of gestures. Each gesture is represented by a HMM characteristics (structure and probabilities). Nevertheless, HMMs are generally independent from time, which practically means that the gesture can be recognized independently from the gesture speed. DTW is another standard method where each template is associated to a state sequence. DTW allows for template matching, where time sequences of the gesture are warped to the reference time sequences [14]. In other words DTW permits to synchronize different sequences and to identify the minimal feature distance between them [15]. One of the main limitations of this method is that it cannot be efficiently used in real-time applications [16]. Caramiaux et al provide also one technique for an early recognition system which is evaluated on a 2D onscreen pen gestures and in a user study involving 3D free space gestures [26].

Gesture interaction

Different types of gesture interactions have been defined, like standardized, adapted and user-defined interaction. This typology refers to different types of gestures that can be used to interact with the computer. In the first category the user should execute predefined gestures referring in most cases to a precise know-how (e.g. pianistic gestures etc.). According to Bobiller-Chaumon et al. [17], this interaction is the easiest to implement since it is widely accessible and accepted by the majority of users. However, it may become necessary to personalize the interaction and the gestures used. User-defined interaction is the most flexible one, where the gestures can be introduced directly by the user in a free way, in order to facilitate their memorization. This type of interaction is frequently used for people with cognitive or motion impairments [18], or in creative art where numerous movements are required for the creative process.

Two different interaction modes have been used to evaluate our system: a) the standardized interaction based both on piano-like gestures in musical interaction, which is easily accepted by the users and b) the user-defined interaction, which is based on the combination of fingerings and applied in robotic interaction in order to provide more flexibility to the user and give him the possibility to propose his/her own gesture vocabulary. As described in the section 7, all the gestures are performed while the hand palm is in a natural semi-closed posture.

METHODOLOGY OVERVIEW

The proposed computer vision methodology is based on feature extraction and takes as input either RGB or depth image sequences that capture motions for every finger individually when the hand palm is semi-closed.

The camera, RGB or time-of-flight, should be installed in front of the hand so as to clearly capture finger motions. The first step is the background subtraction and the detection of the hand silhouette on the image. For this, two options are proposed depending on the type of the camera: a) skin color model for RGB images; b) distance slicing for depth images. The skin color model aims to detect Regions of Interests (RoI) that are similar to the color of the human skin. The distance slicing eliminates the information being out of a user-defined range, which depends on the specificities of the sensor and the needs of the application.

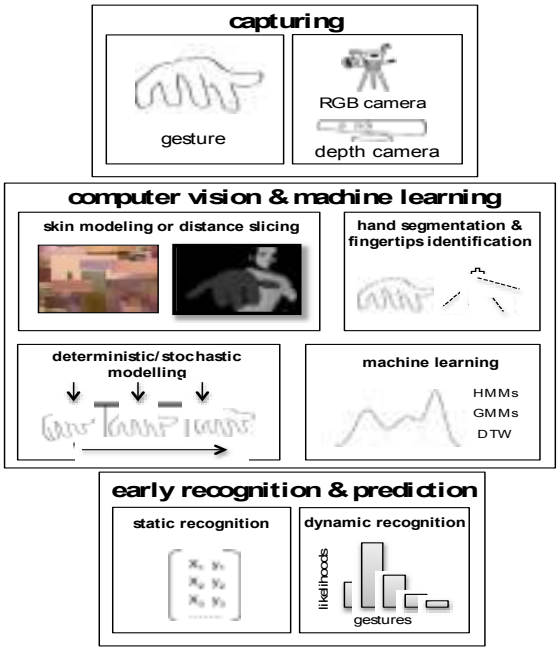

Figure 1. Methodology overview

In both cases, a binary mask is the output while the remaining noise should be reduced and the hand should be further segmented using mathematical morphology methods. The Canny algorithm is applied in order to segment the hand silhouette and extract its contour. The pixels that determine the hand contour are potentially considered as candidates for being the fingertips. Based on the geometric properties of the hand pose, Euclidean distances between the centroid and the contour pixels are calculated. By calculating the local maxima on the distances, only five distances remain at the end and their corresponding pixels are used for the identification of the fingertips as well as the localization of each finger. Coordinates of fingertips in 2D together with finger distance are extracted.

The next step concerns the machine-learning and recognition that is based on time series of features that describe the motions of all the fingers. A combined HMM and DTW approach is developed, permitting to early recognize and predict the gestures. The method allows for parallel one-shot learning. It can be used for mapping between different modalities (i.e. gesture/sound or gesture/waypoints).

HAND DETECTION

Skin color modeling of RGB image for hand detection

A number of studies that has been conducted in the recent past show that variations in skin color between different genders or ethnicities are due more to the difference in brightness (*luminance*) rather than the color itself (*chrominance*) [19], [25]. Consequently, the color information can be maintained even if the brightness information is removed. Finally, this low dimension information requires less computation power computing power to detect skin areas on the image.

So, the skin model is developed in four steps:

a. The skin sampling: A Pianist's Image Library (PIL) has been created based on 200 photos including both skin and nails from people of different ethnicities and genders in various resolutions and lighting conditions in RGB format. Samples of pixels P_i of fingers skin and nail pixels $p_j$have been chosen from the PIL: $P_i(p_j) = [R_j, G_j, B_j]^T$.

b. The specification of the Region of Interest (RoI): $P_i$samples have been imported into the $RoI^{RGB}[m, n]$, which is an image file containing only p_j pixels.

c. The RoI normalisation: The normalisation of the RoI^{RGB} (3D) is converted from chrominance and luminance components into only chrominance components, and its depiction on the normalized rg color space (2d). So, $\forall p_j^{RoI} \in RoI^{RGB}$there is a space conversion function [1]:

$$N: RoI^{RGB} \rightarrow RoI^{rg}, N([R_j, G_j, B_j]^T) = [r_j, g_j]^T \qquad (1)$$

that creates the RoI^{rg}, with the help of the following formulas:

$$r = \frac{R}{R+G+B} \quad \text{and} \quad g = \frac{R}{R+G+B} \qquad (2)$$

d. The skin model definition: The RoI^{rg} is defined as a rectangle in the rg chrominance space where the ranges $r_{range} = [r_{min}, r_{max}]$ and $g_{range} = [g_{min}, g_{max}]$determine the skin model. The skin model is not perfect since the two ranges don't fully describe the dermal information. For example a pixel may be considered as a dermal pixel even if it is not and vice versa.

Based on the skin model, we can easily detect dermal regions in an RGB image. Thus, a binary mask $B = \{b_i\}_{i\in\mathbb{N}}$ is created for each frame of the sequence $F = \{f_i\}_{i\in\mathbb{N}}$, defining regions that contain skin information, as shown in figure 2. Note that the nail color is sufficiently similar to skin color, so that hand segmentation obtained with chrominance includes also the nails (except for very small parts that are more whitish, mostly on half-moons). Therefore the obtained segmented hand always contains also the fingertips, despite the fact that in pianistic semi-closed palm hand pose fingertips are seen on nail side.

Distance slicing of depth image for hand detection

The distance slicing approach is adopted in case of depth sensors. The PMD CamBoard Nano 3D Camera has been used. It is based on single-path ToF imaging that provides an image resolution of 165x120 resolution and its working principle is the measurement of the phase difference between the emitted, the modulated and the received signal. Four measures $I_{i\in[1,4]}$ are recorded at different phase offsets of 90o in four discrete times $\tau_{i\in[1,4]}$ in order to estimate the 3D structure of the scene. Then the arc tangent formula is used to calculate the difference between the phases:

$$\varphi = \arctan\left(\frac{C(\tau_4) - C(\tau_2)}{C(\tau_3) - C(\tau_1)}\right) \qquad (3)$$

where C is the auto-correlation function for all the $I_{i\in[1,4]}$. The fact that these measures are carried out successively allows robust measurement of the phase shift even if some artifacts appear in case of fast motions.

Once the depth map with grey value image data is delivered, the distance between the sensor and the object is derived. This distance is represented as bit-planes and its slicing consists on cutting the appropriate bit-planes in order to isolate the hand from the background. Our depth-slicing is performed with distance boundaries that are constant and determined during an initialization phase: at t=0, the minimum distance Z min on the depth image is computed, and assumed to be on the hand; in order to make the latter assumption more reliable, the user is asked at start-up to place his/her hand in the center of the image and at normal operating distance, so that estimation of Z_{min} can be done only around center and surely on hand. Then the depth slice is defined as [Z_{min} - hS, Z_{min} + fL + hS], where hS is the typical hand size and fL is the typical forearm length; the rather large slice boundary ensures that hand is always totally included in it (provided that it is not placed closer to the camera than at start-up time), and due to capture set-up there is normally no other object within the slice. There is also the option for the distance slice boundary to be manually defined by the user depending on the application. According to the technical specifications of the PMD camera but also to several experiments, the optimal range for the camera is between 25 and 50 cm. So, the maximum slice should not exceed the 25 cm.

After the distance slicing, a binary mask $B = \{b_i\}_{i\in\mathbb{N}}$is created for each frame of the sequence defining the region of interest (Figure 2).

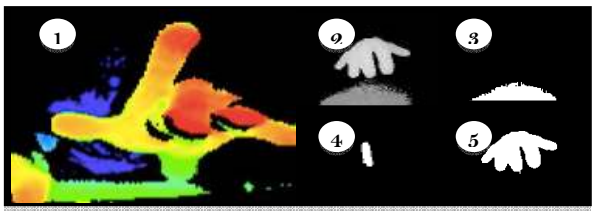

Figure 2.1.Artifacts on fingertips when fingers move very fast; 2. Initial depth map;3 & 4. Bit-planes with a small distance from camera; 4. Bit-plane including the whole hand silhouette

HAND SEGMENTATION AND FINGERTIPS LOCALISATION

Taking into account the specific hand pose with semi-closed palm, hand segmentation is needed to face the problem of hand detection as one mass. It includes simplification of the image and two-components hand decomposition.

Each b_i is composed by two components: the noise n_i and the hand h_i with $b_i = n_i + h_i$. An Alternating Sequential Filtering (ASF) is applied to each b_i, filtering it by n=10 iterations of the close and open binary operators, so: $ASF_n = h_i, n = 10$. Thus, the noise is reduced and the hand silhouette is better depicted on this image, even if the hand is still not perfectly extracted [1].

The extraction of the hand silhouette is based on the application of different filtering techniques, including Open, Min and Gauss filters [1]. Thus, a set S_i of five filters is applied on the $h_i$component of each frame:

$$S_i = \{Open_i, Min_i^1, Gauss_i^1, Min_i^2, Gauss_i^2\}_{i\in[1,4]} \quad (3)$$

As a final stage of the extraction of the hand contour $h_c(h_{cx}, h_{cy})$, Canny Filter is chosen. Canny's algorithm is not just a method of edge detection. By having extracted the hand contour, the centroid $(c_x, c_y)$is calculated, for every frame. Detection of fingertips is done by searching local maxima of centroïd-to-border distance on segmented hand mask. This way a large number of non-realistic fingertip pixel candidates are excluded. The Euclidean distances are calculated only between the pixels of $h_c(h_{cx}, h_{cy})$, whose h_{cy} becomes minimum for each h_{cx}, and the $c(c_x, 0)$. Then, local maxima are detected and they are saved with their positions. If fingers are well-separated, each fingertip should be among those local maxima. Even if fingers are instead connected one along each other, since fingertips are normally convex, there is usually at least a (possibly shallow) local maximum on each tip. In case of more than five local maxima, the one with the smallest value is eliminated until only five remain.

Scale and rotation invariance techniques are integrated into the system. The skin coverage criterion has been used for RGB cameras in order the recognition to be reliable and render the system invariant to the distance [20]. For depth images we consider that the hand is always within the range of 25 and 50 cms. Nevertheless, if the number of candidate pixels to belong to the hand is below a given threshold, the system considers that there is no hand on the image and the coordinates of the fingertips and the centroid become 0.

GESTURE MODELING AND RECOGNITION

Gesture descriptors and modeling

In piano playing, gestures can be distinguished thanks to kinematic parameters of finger motions (gesture descriptors). A set of gesture musical descriptors corresponds to a set of image features, which determines the feature vectors (observations) $\mathbf{O_k}$on an image sequence, where **k** the number of the images. Consequently, the feature vector is extracted describing (a) the differences of the coordinates between the fingers and the centroid, (b) the abscissa of the fingers and (c) differences between the abscissas of adjacent fingers (Table 1). More precisely, for the feature vector $\mathbf{O_i}$, the following 14 features are extracted:

$$O_i = \{o_i^1, \dots, o_i^{14}\}_{i\in[1,k]} \quad (4)$$

Features extracted from the images			
o_1^i	t_x (thumb)	o_8^i	$c_y - r_y$
o_2^i	$c_y - t_y$	o_9^i	p_x (pinky)
o_3^i	i_x (index)	o_{10}^i	$c_y - p_y$
o_4^i	$c_y - i_y$	o_{11}^i	$t_x - i_x$
o_5^i	m_x (middle)	o_{12}^i	$i_x - m_x$
o_6^i	$c_y - m_y$	o_{13}^i	$m_x - r_x$
o_7^i	r_x (ring)	o_{14}^i	$r_x - p_x$

Table 1. Features used as gesture descriptors

In order to address the problem of the self-occlusion for fingers and predict their positions, a five-class classifier is used; each of them corresponds to the (x,y) position of a finger on the image plane. Initially, each observation is classified in one of the five classes using a minimum distance criterion. In case that more than one observation have been attributed to a class, the observation with the highest likelihood is chosen (i.e. the nearest to the class centre), while the other observations are discarded as false detections. If no observation has been attributed to a class (i.e. in case of self-occlusion), then the centre of the class is considered as the position of the missing finger. Otherwise, the position of the fingertip is determined by the coordinates of the current observation. The centre of the class is updated considering the last N positions of the finger according to the following equation:

$$P_{cc} = \frac{1}{N+1}\sum_{t=0}^{N} P_{t-i} \quad (2)$$

where, P_{cc} is the updated class centre, N is the number of previous observations and P_{t-i} is the position of the finger in N previous time instances. In our experiments, described in section "Experimental Results", N has a relatively small value (N=3).

Static recognition

Static recognition concerns the detection of fingerings in each single frame and relies on the definition of the feature vectors containing the differences of the coordinates between the fingers and the centroid. It is implemented by determining the threshold of the key press for each finger. It is applied as the displacement of the fingering, in terms of pixels, comparing to the rest position of the hand. The definition of the threshold is based on experimental results and it is strongly related to the application.

Dynamic recognition

Dynamic recognition corresponds to temporal successions of fingerings, it is applied on a sequence of images and it is

based on stochastic modeling. The proposed framework uses a hybrid machine learning engine that has been proposed by Bevilacqua [21], [22] allowing to use an one-shot procedure such as in DTW, while using a full probabilistic framework as formalized in HMM.

In this hybrid method, each data sample is associated with a state of a left to right Markov model, with only self, next and skip transitions allowed. As the template is regularly sampled in time, it is straightforward to show that these transition probabilities must be constant over the whole Markov chain (see [22] for a discussion about the different choices for setting these probabilities). The decoding is then efficiently performed using the forward procedure [13] instead of the classic Viterbi. The forward procedure allows for estimating the likelihood values at each step of the decoding computation. Therefore, we get an likelihood estimation at the same rate than the data sampling rate, with a maximum delay of one sample. This is thus in contrast to the Viterbi decoding that implies a delay typically of the duration of the gesture. In this case, it is possible to obtain a likelihood value as soon as the gesture starts, which accuracy will increase over time (as shown in [23]). Comparatively, to obtain early results, this approach is more efficient than standard HMM using the Viterbi procedure or DTW that would require to repetitively run the algorithm on a sliding window. Our hybrid method is thus especially adapted for real-time cases of interactive scenarios where the system latency must be minimized. All the 14 features described in the Table 1 can be used for the training of the models.

EXPERIMENTS

The described methodology has been applied to two different experimentation sets. The first concerns musical interaction and more precisely piano-like finger gestures recognition based on standardized gestural interaction performed on a surface. The second experimentation set is referring to robotic user-defined interaction. Finger gestures with semi-closed palm have been registered performed in space.

The resolution of RGB images is 640x480 and the frame rate used was 20 fps. Concerning depth images the resolution has been defined to 165x120 and the frame rate also to 20 fps. In both cases the cameras were installed at 15-20 cm in front of the hand. For the recordings with the PMD CamBoard Nano 3D camera, a specific black tissue has been used to easily isolate the background from the hand. The average duration of these gestures is small, from 3seconds for a fingering to 7 seconds approximately for an arpeggio or fingerings combination.

Musical Interaction

Four fundamental piano-like musical gestures have been recorded with both RGB and depth cameras: G_1^M corresponds to the ascending scale, G_2^M to the descending, G_3^M is the ascending arpeggio and G_4^M is the descending. Arpeggios are non-adjacent notes, corresponding to non-adjacent single keystrokes and their global gesture relies on four keystrokes.

Robotics Interaction

In the context of robotic interaction, two different types of interaction can be considered. The first type is the standardized interaction where the fingering detection (static finger motion pattern recognition) with semi-closed palm has been used as discrete commands. After the completion of the fingering and its detection, the command is transmitted to the AGV (Automated Guided Vehicle). The second mode of interaction is user-defined. As soon as the gesture is recognized, a set of commands is transmitted to the AGV. More precisely, the user can choose his/her own finger gestures and thus create his/her own gesture vocabulary in order to early activate (even with few data as input) specific set of commands per gesture.

The main issue to address is the selection of the finger gesture vocabulary. The goal of this vocabulary is to select finger gestures that are mapped afterwards to commands for the AGV. More precisely, a finger gesture vocabulary is defined as a set of correspondences between verbal commands or expressions of finger gestures. Within the context of this research, a large number of users proposed their own gestures to create the gesture vocabulary and to interact with the AGV. Based on basic ergonomic principles, such as ease of use, intuitiveness, learnability, memorability, weariness and recognition accuracy, four gestures have chosen because they fulfill all the above criteria. However, the user can still use his/her own gestures.

EVALUATION

Fingertips localization and fingerings detection

Two databases have been created for testing the fingertip and fingering detection. The first database has 28 RGB 20 fps videos with 286 fingerings while the second has 79 depth 20 fps videos with 458 fingerings from 5 users.

The first step our evaluating our system is to calculate the error of position estimation of the fingertips based on the ground-truth. It is based on the comparison between manual extractions of the fingertip positions and the outputs of the algorithm. The average fingertips localization error using RGB images is ~ 2 mm on X axis (perpendicular to fingers), and ~ 3-5 mm on Y axis (along finger length). The fingertips localization error is much lower when using ToF camera: ~ 0.5mm on X axis, and ~1mm on Y axis. Since semi-closed palm has been used, the motion of the fingers is more important on Y than on X and thus the error on Y is bigger. The error on Y can vary depending on the motion of the finger. That means, it remains very small when the finger is not in motion while it has its peak when the user performs a fingering. However, the fingertip is well tracked even during the fingering with both types of images.

Regarding evaluation of our fingerings static recognition, we computed the output we obtain on all static images of single fingerings included in our databases. Of course, some fingerings are not detected at all. This is due to the particular performance style of the user but also to a very

strict fingering threshold, which has been defined at 10 pixels. The recall (percentage of fingerings correctly recognized) remains high in both cases, while the depth camera gives better results for all the fingertips. The fingerings with the lowest accuracy are the thumb and the pinky. The thumb is the finger with the most particular physiology, which has the possibility to do wider movements and thus to be very easily self-occluded. The pinky can also be occluded by other fingers especially when the hand has a small tilt from the camera. Additionally, the line "NO" refers to the images where there is no fingering at all (rest position). False fingerings are detected only for the middle since sometimes the fingertip may beslightly offset compared to the rest position. This error is more related to the user than to a false detection of the system. Nevertheless, the recall of the fingering detection for thumb and pinky is above 80,6% for RGB and 87,6 for depth images, thanks to the five-class classifier that has been described in 6.1. Compared with fingertips position errors can conclude that these errors are low enough to have no significant impact

Standardized gestures in musical interaction

In order to evaluate the recognition accuracy of the system, we used the "Jackknife resampling" as a cross-validation method: each example for each type of gesture is taken in turn as training prototype for its own class (remember we use one-shot learning with only one example per class). In our particular use of Jackknife resampling method, it is very similar to a "leave all-but-one out" cross-validation, as pointed in [27].

Our dataset contains 10 observations of 4 standardized gesturesG_i^Minspired from the piano-playing. These 40 observations have been used to create discrete learning and testing databases per iteration. In iteration, one dataset is left out for one-shot learning of the modelM_i^Mper gesture$G_i^M$and the rest of datasets is used for training. So, 90 queries were given to eachM_i^M, or 360 in total. The metrics used to evaluate the system are Precision and Recall. For RGB images, the total Precision is 86,8% and the Recall is 86,4%. For depth images, the results are better since the Precision of the system is 90,4% and its Recall 90%. It can be easily observed that in both cases all the false recognitions are observed between G_1^M- G_3^M and G_2^M - G_4^M. This is due to the fact that the pairs of gestures are very similar since they are both of gestures ascending or descending.

The difference in the recognition accuracy between RGB and depth image sequences is not very important in terms of Precision and Recall. Nevertheless, since the datasets of the RGB image sequences have been recorded in controlled lighting conditions while those of depth image sequences in various lighting conditions, thus the approach based on the depth camera is likely to be more robust in real situation.

User-defined gestures for robotic interaction

The final phase of the evaluation of the unified framework for the finger gesture recognition consists in testing recognition of the gestures $G_{i\in[1,4]}^R$ that have been used for the user-defined interaction. Once the gesture is early recognized a set of commands is continuously streamed to the AGV.

The dataset used for the evaluation contains 7 observations of 4 user-defined gestures G_i^R. 60 queries were given to each M_i^R, or 240 in total. The Precision is 86,95% and the Recall is 84,5%.

The results are similar to those of the musical gestures. The RGB images give satisfying results for controlled lighting conditions while depth images give much better results for various conditions. With regards to the RGB images, there is some important confusion when the hand is rotated since the shadows have an impact to the quality of the image. Confusions are also observed between i.e. G_1^R - G_3^R or G_1^R - G_4^R, since the inverse rotation of the images is done before the fingering detection. So, currently there is no specific feature about the hand rotation for the learning of the HMMs. Of course, in cases where the hand is rotated in the air and 3 simultaneous fingerings are performed while the fingering F1 is not clearly performed, the system outputs some errors. This is also due to the fact that it is somehow difficult to perform clear fingerings with a rotated semi-closed palm. The majority of these conclusions are also valid for the depth images, where the confusions between gestures are much less important. The very positive point of the depth images is the fact that the intention of the user is captured very early: the M_i^R gives a clear maximum likelihood starting from 3 seconds after gesture beginning, while using the RGB images the M_i^R gives correct

CONCLUSIONS

Conscious of the need for more natural gestural interaction based on hand and fingers, a unified computer vision framework for simultaneous fingers gestures early recognition and interaction with semi-closed palm without any skeleton extraction has been proposed. Depending on the context and the application, the framework can work with either low or medium cost sensors, which are RGB and time-of-flight depth cameras. Early recognition and prediction of the gestural intention of the user can be used to interact with the computer, either with standardized or user-defined gesture vocabulary. Based on common evaluation criteria, a number of experimental comparisons has been conducted in order to test the position estimation error of the fingertips, the fingering detection, the standardized finger gesture recognition in musical interaction and the user-defined gestural interaction for automated guided vehicles. The evaluation are quite satisfying for both types of sensors: for depth images the fingertip position error is less than 1 mm; furthermore, the average precision and recall for standardized and user-defined gestures are 80% and 86% for RGB and 90,4% and 90% for depth images respectively. As a future work, the combination of synchronized RGB and depth images as well as the integration of the ongoing hand skeleton extraction research is planned. With regards to the applications, a finger gesture control of drones as well as a comparison of different technique in sound interaction using finger gestures is also a medium-term goal.

ACKNOWLEDGMENTS

A part of the research project is implemented within the framework of the Action «Supporting Postdoctoral Researchers» of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

REFERENCES

1. Manitsaris S., 2010. *Computer vision for gesture recognition: gesture analysis and stochastic modeling in music interaction,* University of Macedonia, Greece.
2. Solanki, U.V, Desai, N.H.,2011. Hand gesture based remote control for home appliances: *Handmote, Information and Communication Technologies (WICT)*, World Congress.
3. Agraval R., Srikant R., 2000, Privacy-preserving data mining. *ACM Sigmod Record* , 29(2) :439–450.
4. Buxbaum B., 2002. Optische Laufzeitmessung und CDMA auf Basis der PMD-Technologie mittels phasenvariabler *PN-Modulation*, Schaker, Aachen.
5. Wiedemann, M., Sauer, M., Driewer, F., & Schilling, K., 2008. Analysis and characterization of the PMD camera for application in mobile robotics. In *Proceedings of the 17th IFAC World Congress*, 6-11.
6. Abdallah, Manel Ben. 2013. Different Techniques of Hand Segmentation in the Real Time. *IJCAIT 2.1* (2013): 45-49.
7. Phung, Son Lam, Abdesselam Bouzerdoum, and D. Chai Sr. 2005. Skin segmentation using color pixel classification: analysis and comparison. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 27.1 (2005): 148-154.
8. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., & Moore, R. ,2013. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1), 116-124.
9. Keskin, C., Kıraç, F., Kara, Y. E., & Akarun, L. 2013. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, Springer London, 119-137.
10. Dapogny, A., De Charette, R., Manitsaris, S., Moutarde, F., & Glushkova, A., 2013. Towards a Hand Skeletal Model for Depth Images Applied to Capture Music-like Finger Gestures. In *proc. Of International Symposium on Computer Music Multidisciplinary Research*
11. Turaga, P., Chellappa, R., Subrahmanian, V., And Udrea, O. 2008. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology*, IEEE Transactions on 18, 11, 1473–1488.
12. Mitra, S. and Acharya, T. 2007. Gesture Recognition: A Survey. Systems, Man, and Cybern, Part C: *Applications and Reviews*, IEEE Trans. on 37, 3, 311–324.
13. Rabiner L.R, 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, 257 -286.
14. Celebi, S., Aydin, A. S., Temiz, T. T., & Arici, T., 2013. Gesture Recognition Using Skeleton Data with Weighted Dynamic Time Warping. *Computer Vision Theory and Applications*. Visapp.
15. Ten Holt, G.A., Reinders, M.J.T., & Hendriks, E.A.,2007. Multi-dimensional dynamic time warping for gesture recognition. In 13*th annual conference of the Advanced School for Computing and Imaging* (Vol. 19).
16. Boukir, S. & Chenevière, F. 2004. Conception d'un système de reconnaissance de gestes dansés. *Traitement du signal*, 21(3), 195-203.
17. Bobillier-Chaumon M.E., Carvallo S., Tarpin-Bernard F., Vacherand-Revel J., 2005. To adapt or standardize the human-computer interactions?, *Revue d'Interaction Homme-Machine*, vol. 6, no. 2, 91–129.
18. Jégo, J. F., Paljic, A., & Fuchs, P., 2013. User-defined gestural interaction: A study on gesture memorization. In *3D User Interfaces (3DUI),* IEEE Symposium, 7-10.
19. Yang, J., Lu, W., & Waibel, A. 1997. Skin-color modeling and adaptation. *Lecture Notes in Computer Science*, Springer.
20. Tsagaris A., Manitsaris S., Dimitropoulos K., Manitsaris A. 2011. Intelligent invariance techniques for music gesture recognition based on skin modelling, *12th IEEE International Symposium on Computational Intelligence and Informatics* (CINTI 2011), 21-22/11/2011, Budapest, Hungary
21. Bevilacqua, F., Guédy, F., Schnell, N., Fléty E. and Leroy N., 2007. Wireless sensor interface and gesture-follower for music pedagogy. In *Proceedings of the ICNIME*, New York, USA, 124-129.
22. Bevilacqua, F., Zamborlin, B., Sypniewski, A., Schnell, N., Guédy, F. and Rasamimanana, N., 2010. *Continuous realtime gesture following and recognition*, LNAI 5934, 73–84.
23. Bruno Zamborlin, Frederic Bevilacqua, Marco Gillies, and Mark D'inverno. 2014. Fluid gesture interaction design: Applications of continuous recognition for the design of modern gestural interfaces. *ACM Trans. Interact. Intell. Syst*. 3, 4, Article 22, 30 pages.
24. Priddy, K. L., & Keller, P. E. 2005. Artificial neural networks:an introduction(Vol. 68). SPIE Press, 101-103.
25. Tsagaris, A., & Manitsaris, S., 2013. Colour space comparison for skin detection in finger gesture recognition. *International Journal of Advances in Engineering & Technology*, 6(4).
26. Caramiaux, Montecchio, Tanaka, Bevilacqua. Adaptive Gesture Recognition with Variation Estimation for Interactive Systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4 (4), 18. 2014

View publication stats