HAL
open science

A Low-Cost Natural User Interaction Based on a Camera Hand-Gestures Recognizer

Mohamed-Ikbel Boulabiar, Thomas Burger, Franck Poirier, Gilles Coppin

To cite this version:

Mohamed-Ikbel Boulabiar, Thomas Burger, Franck Poirier, Gilles Coppin. A Low-Cost Natural User Interaction Based on a Camera Hand-Gestures Recognizer. HCI International 2011, 2011, orlando (FL), United States. pp.214-221. hal-00816596

HAL Id: hal-00816596

https://hal.science/hal-00816596v1

Submitted on 22 Apr 2013

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.

A Low-cost Natural User Interaction Based On A Camera Hand-Gestures Recognizer

Mohamed-Ikbel Boulabiar[1], Thomas Burger[2] Franck Poirier[3], and Gilles Coppin[1]

[1] LAB-STICC, Telecom-Bretagne, France,
boulabiar@gmail.com, gilles.coppin@telecom-bretagne.eu,
[2] LAB-STICC, University of Bretagne-Sud, France thomas.burger@univ-ubs.fr
[3] VALORIA, University of Bretagne-Sud, France franck.poirier@univ-ubs.fr

Abstract. The search for new simplified interaction techniques is mainly motivated by the improvements of the communication with interactive devices. In this paper, we present an interactive TVs module capable of recognizing human gestures through the PS3Eye low-cost camera. We recognize gestures by the tracking of human skin blobs and analyzing the corresponding movements. It provides means to control a TV in an ubiquitous computing environment. We also present a new free gestures icons library created to allow easy representation and diagramming.

Keywords: natural gesture interaction, low-cost gesture recognition, interactive TV broadcast, ubiquitous computing

1 Introduction

HCI research focuses more attention than before on enhancing the user experience regarding human-display interaction. This area of research focus on making interaction more natural, so that it is not necessary to click on buttons and to touch screens. To do so, automatic gesture recognition is an interesting paradigm [3], [13], [6]. However, in everyday life, for interaction with home devices having embedded computational power, such as interactive TVs, one does not make benefit from these new interaction paradigms yet.

We present in this paper an approach that allows the detection and the interpretation of gestures of a TV spectator from a simple and low cost camera. this work takes place in the context of the French FUI[4] Project RevTV, the aim of which is to add new interaction techniques, sos that telespectators take activaly part to the TV broadcast. The final objective of this project is to control the animations of an avatar that should be inserted within a TV show beside the presenter or actor. By now, the major scenario that we rely on, corresponds to some educational game where a pupil controls her or his avatar which interacts in a broadcast program led by a "real" TV animator. During such kind of scenarios,

[4] Fond Unique Interministeriel

commands interaction like pointing, selecting and moving, as well as natural gesture animation, are required.

The paper is organized as following. Section 2 presents the study held on gestures semantics, sources, and taxonomies and include the presentation of the free icons library created and used. Section 3 shows the technical details of handling the camera input for gestures recognition. Section 4 shows the modes where the gestural information generated are used with some screenshots of the running application. Section 5 answers the usability and naturelness question of the use of gestures specially in an ubiquitous environment. Finally section 6 discusses the future of the work and the possible integration of other multimodal modules.

2 Gesture Semantics

Symbolic gestures, such as emblems, play an important role in human communication as they can fully take the place of words. These gestures are processed by the same area of the human brain as the spoken language [17]. Hence, these gesures do not come alone, and they are often combined with speech, or with cognitive activities, as they are classically used to add more precision on the details of a talk.

We have extracted some semantics of such type of gestures, by the analysis of the movements of a user explaining a story, and by searching for a relation between gestures and the part of the story being told [9]. After extraction, most of gestures semantics can be divided into two parts for their possible future use as:

Animation Gestures used to animate a virtual avatar.
Command Gestures used to launch a specific predefined action.

2.1 Gestures taxonomies

A rapid look the possible gestures human can perform [16], lead to the conclusion that the amount of possibilities is tremendous. Nonetheless, the gestures can be classified according to various criteria not only to simplify their recognition, but also to allow their reproducing on an avatar to putting tags on them for future analysis[10]. Here, we consider a taxonomy based on the context of the gestures as well as on their types: According to McNail [9], it is possible and useful to distinguish between these gestures. Here follows a possible taxonomy :

Gesticulation is a motion that embodies a meaning reliable to the accompanying speech. It is made chiefly with the arms and hands but is not restricted to these body parts.

Speech-linked gestures are parts of sentences themselves, the gesture completes the sentence structure.

Emblems are conventionalized signs, such as thumbs-up or the ring (first finger and thumb tips touching, other fingers extended) for "OK."

Pantomime is dumb show, a gesture or sequence of gestures conveying a narrative line, with a story to tell, produced without speech.

In our work, we focus on Emblems for Command Gestures, and on Gesticulations for Animation Gestures.

2.2 Gestures Library

In order to identify relevant gestures we have analyzed some talk-shows, and we have extracted the most frequent gestures being performed. According to our scenario of use in the context of interactive TV (RevTV project), we need some gesture to trigger commands, as well as other gestures to implement natural interaction with the avatar. We have created a vector based gestures icons to represent them. These 4 icons, represented in figure 1, are part of the **ishara** library [1]. They represent the gestures supported by the software of our application. We provide a movement mode, a multitouch mode, a command mode, a hand fingers recognition, and, finally, a cursor movement mode (see Section 5 for detailled explanation of these modes).

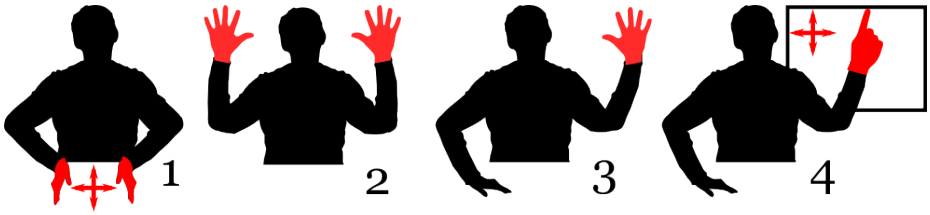

Fig. 1. Family of supported gestures represented using the free gestures icons library from [1]

These five basic gestures modes are already supported in our system as a first step. They will be extended depending on the context fulfilling the requirements of our scenario.

3 Technical Work

Figure 2 is a description of the pipeline of the modules used to extract gestural information. Each part is described in details in the next subsections.

3.1 Camera Input

We have chosen a low-cost camera that can be integrated with other components in the future. The PS3Eye camera matched our expectations with a 640x480 resolution combined with a 60 frames-per-second capability. This camera costs about 40$, which is a reasonable price for a potential market deployment.

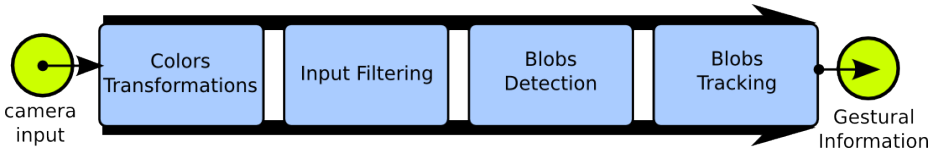

Fig. 2. Pipeline of the work components showing the routing and transformation of the camera information until getting the gestural information ready to use in the scenario

3.2 Colors Transformations

The video processing which depend on the OpenCV library [4] is based on several modules, such as illustrated in figure 2. First, a basic but adaptive skin color segmentation of a large interval of human skin varieties is performed [2]. We also perform histogram equalization technique to better enlarge the colors to all the space available. Then we transform the color space from BGR, which come from the camera to YCrCb, as it is the most adapted one to skin segmentation. Other color spaces like HSV or CIE-Lab can be used for skin color segmentation, but the transformation from BGR either takes more time (the transformation we use is linear) or needs more parameters in the next step [2].

Space	BGR	HSV	YCrCb	CIE Lab
Param.	R>95, G>40, B>20, Max{R,G,B}-Min{R,G,B}<15 abs(R-G)>15, R>G, R>B	H[0.4 .. 0.7] S[0.15 .. 0.75] V[0.35 .. 0.95]	Cb[77 .. 127] Cr[133 .. 173]	C[0 .. 65] I[0 .. 14]

3.3 Input Filtering

During this step, a 5x5 opening morphological kernel [15] is applied to smooth the results. Practically, it cleans the image from isolated pixels and holes inside skin zones.

3.4 Blobs Detection

Each blob is labeled in linear time using the contour tracing technique with the linear-time component-labeling algorithm [5], which requires 1 to 4 passes to recognize all blobs with their bounding contour. We identify the hands and the head by a calibration focused on their first position in the screen. We identify them and add a tag to their respective blobs.

3.5 Blobs Tracking

After having the hands and head blobs identified, an efficient tracking method based on appearance model is used [14]. Let us note that it handles occlusion problems. We have used the cvblob implementation [8] for that algorithm which

Fig. 3. A representation of different human skin color variations

stores the measures of bounding boxes distance and provides tracks instead of just blobs.

Once the matrix of distance is defined, the following loops are executed to handle blobs changes:

- A loop to detect inactive tracks: those tracks with no blobs near.
- A loop to detect and create new tracks: those blobs without a track near.
- A loop to assign blobs to tracks. It makes clusters with blobs that are close and assign them to a track. In this step some tracks could merge in one.
- A last loop which check all inactive tracks to delete the old ones.

To better handle blobs, the bounding path of the blobs is simplified to a set of few points when needed. This is used to identify picks for the counter mode (see Section 5). Finally, a phase of calibration is triggered when the logic of tracking is lost or damaged.

4 Gestural information and Recognition modes

The output of the video processing is the input of the recognition module, which is based on the track location on the screen. Four different modes are defined according to the hand locations. The switch between them is based on the following grammar:

The movement mode: it is activated when two hands are close to each other, so that the user selects between 4 directions. This mode can be used as input for games to select between the directions.

The multi-touch mode: where we consider the similarities between 2D gestures in a tactile multi-touch context and gestures in the 3D space. To do so, we consider the hands blobs in a manner similar to that of two finger tips in the input of a multi-touch device. Based on that, we can recognize well known multi-touch gestures, such as Drag, Pinch/Zoom and Rotate with a consideration of a interaction centroid[7] but applied in our case.

The counter mode: counts the number of fingers in a preselected hand (the left and right hand are automatically discriminated) by counting the peaks in the simplified contour of the blob. This mode can again be used in games for kids.

The mouse mode: it is used to control a cursor to select something in an arbitrary place on the screen. With this mode, we only get the input from a sub zone in the screen and map it to the full screen. We use it in a similar way to a computer touchpad. The user can validate clicks using a tempo on another zone.

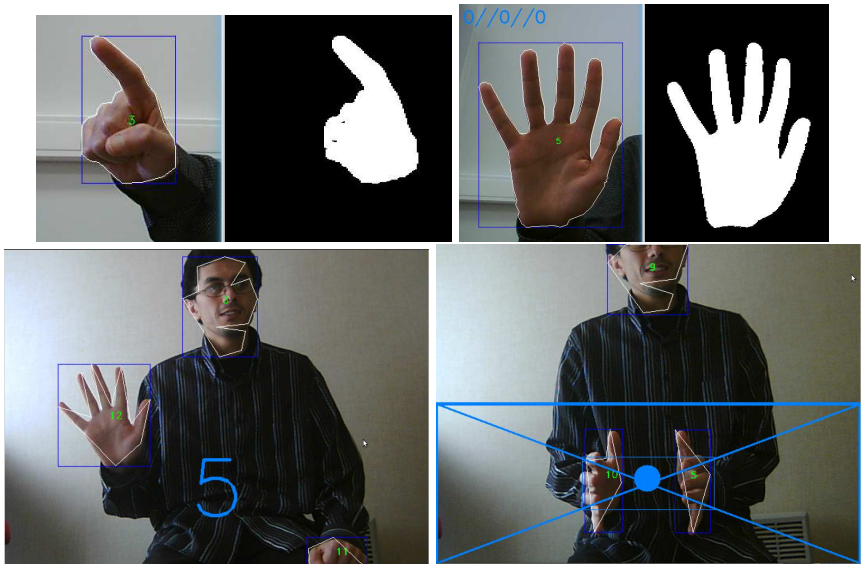

Fig. 4. A sub-list of recognition modes showing pointing, number of hand fingers counting and the movement mode

5 Usability and Naturalness in an ubiquitous environment

The usability of our system should be compared to other systems in an ubiquitous environment. In our case, we have a TV instead of a PC. The gestures in such

environment, and specially for children game scenario, are made for a short period of time, and doesn't require a good usability, even if we can recognize gestures in real-time.

The Naturalness of gestures is supported by the analysis of those to be supported and the preference of gesticulations. Gesticulations are the most common gestures, so they inherit from this their ability to be produced with less complication.

Emblems gestures which englobe commands, are chosen to be produced in more difficult positions to allow easy discrimination from gesticulations. Even with these extractions, the naturalness aspect of the gestural interaction itself are still objected. [11]

Spatial gestural interactions always lack from the feedback for the interaction. This affects also the usability because the user can no more be guided in space as he was with surface movement or simple physical joysticks.

6 System integration of this interaction and future work

Human brain Broca's area, identified as the core of the brain's language system, are not in fact committed to language processing, but may function as a modality-independent semiotic system that plays a broader role in human communication, linking meaning with symbols whether these are words, gestures, images, sounds, or objects [17]. This natural handling of multimodal communications gives an argument to build recognizer taking part of most of these modalities specially to remove ambiguous gesture meaning decision cases.

Our system only supports gestural interaction at the moment but the integration of other multimodal means are possible to fine tune the input and give the user better feedback. These possible evolution of the system are possible:

Facial Recognition : our system don't take care of the facial expressions recogntions. A future support in this area can be added for more reactiveness in pupil's games.

Voice Recognition : To speed up commands handling, we can use short voice keywords either to move from one mode to another, or to select object.

Haptic feedback : A special wearable vest can be used to allow more reactiveness with the user. But this area still lacking innovation because the haptic feedback is by area and not continue.

Conclusion

In this paper we have shown a proof-of-concept mechanism to recognize hand gestures then use them to control a TV or to take actions in a Game scenario for pupils. We have also provided a free library which can be used to represent gestures in other scenarios and other studies.

The evolution of our system could be possible in the direction of a multimodal environment in the case where we can be aware of the limits and myths [12]. But

this evolution is a natural choice as new devices are getting more computational power and the ubiquitous environment is becoming a reality.

References

1. ishara vector based and open gestures icons (2011), https://github.com/boulabiar/ishara
2. Askar, S., Kondratyuk, Y., Elazouzi, K., Kauff, P., Schreer, O.: Vision-based skin-colour segmentation of moving hands for real-time applications. In: Visual Media Production, 2004. (CVMP). 1st European Conference on. pp. 79–85 (march 2004)
3. Bolt, R.A.: Put-that-there: Voice and gesture at the graphics interface. In: Proceedings of the 7th annual conference on Computer graphics and interactive techniques. pp. 262–270. SIGGRAPH '80, ACM, New York, NY, USA (1980)
4. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
5. Chang, F., Chen, C.J., Lu, C.J.: A linear-time component-labeling algorithm using contour tracing technique. Comput. Vis. Image Underst. 93(2), 206–220 (2004)
6. Derpanis, K.: A review of vision-based hand gestures. Tech. rep. (2004)
7. Gorg, M.T., Cebulla, M., Garzon, S.R.: A framework for abstract representation and recognition of gestures in multi-touch applications. In: ACHI '10: Proceedings of the 2010 Third International Conference on Advances in Computer-Human Interactions. pp. 143–147. IEEE Computer Society, Washington, DC, USA (2010)
8. Lin, C.C.: cvblob, http://cvblob.googlecode.com
9. McNeill, D.: Gesture and Thought. University of Chicago Press (2005)
10. Neff, M., Kipp, M., Albrecht, I., Seidel, H.P.: Gesture modeling and animation based on a probabilistic re-creation of speaker style. ACM Trans. Graph. 27(1), 1–24 (2008)
11. Norman, D.A.: The way i see it: Natural user interfaces are not natural. interactions 17(3), 6–10 (2010)
12. Oviatt, S.: Ten myths of multimodal interaction (1999)
13. Pavlovic, V.I., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 677–695 (1997)
14. Senior, A., Hampapur, A., Tian, Y.L., Brown, L., Pankanti, S., Bolle, R.: Appearance models for occlusion handling. Image and Vision Computing 24(11), 1233–1243 (2006)
15. Soille, P.: Morphological Image Analysis: Principles and Applications. Springer (2004)
16. Streeck, J.: Gesturecraft. John Benjamins Publishing Company (2009)
17. Xu, J., Gannon, P., Emmorey, K., Smith, J., Braun, A.: Symbolic gestures and spoken language are processed by a common neural system. Proc Natl Acad Sci U S A 106(49), 20664–9 (2009)