2007 International Symposium on Field-Programmable Custom Computing Machines

Hand-based Interface for Augmented Reality

F. Javier Toledo-Moreo, J. Javier Martínez-Álvarez J. Manuel Ferrández-Vicente
Dpto. Electrónica, Tecnología de Computadoras y Proyectos, Univ. Politécnica de Cartagena
Antiguo Cuartel de Antiguones, Pl. Hospital, 1, 30202 Cartagena Spain
javier.toledo@upct.es

1. Introduction

Augmented reality (AR) is a highly interdisciplinary field which has received increasing attention since late 90s. Basically, it consists of a combination of the real scene viewed by a user and a computer generated image, running in real time. So, AR allows the user to see the real world supplemented, in general, with some information considered as useful, enhancing the users perception and knowledge of the environment. Benefits of reconfigurable hardware for AR have been explored by Luk et al. [4]. However, the wide majority of AR systems have been based so far on PCs or workstations.

In this paper, a hand-based interface for mobile AR applications is described. It detects the user hand with a pointing gesture in images from a camera placed on a head-mounted display worn by the user, and it returns the position in the image where the tip of the index finger is pointing at. A similar system is proposed in [5], but our approach is based on skin color, without the need of glove or colored marks. The hand-based interface is aimed for performing pointing and command selection in a platform for developing FPGA-based embedded video processing systems [1]. This is a hardware/software platform which acquires video in standard analog formats, digitizes and stores it, and makes feasible the interaction with the user and runtime customization of processing algorithms thanks to a user interface which allows choosing options, configuring parameters, etc. The whole platform, including the hand-based interface herein described, is intended to build an FPGA-based aid for people affected by a visual disorder known as tunnel vision [2].

2. Color-based skin recognition

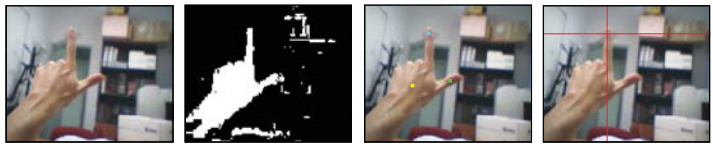
Human skin color has proven to be a useful cue in applications related to face and hands detection and tracking. The color feature is pixel based and therefore it allows fast processing. Besides, its orientation and size invariance confer high robustness on geometric variations of the skin-colored pattern. When building a skin color classifier, two main problems must be addressed: the choice of the most suitable colorspace and the modelling of the skin color distribution.

The transformation of the image data into another colorspace is aimed at achieving invariance to skin tones and lighting conditions. In this work, the following colorspaces have been evaluated: RGB, normalized RGB, YCbCr (601 standard), HSV, YUV, YIQ and TSL.

To model skin color, two different statistical solutions have been adopted: one based on explicitly defined rules and the other on a look-up table derived from the histograms. On the one hand, we have analyzed the three 2D histograms of each selected colorspace and defined explicitly the boundaries of the skin cluster through a number of rules, each one expresed by means of a line equation. With these rules, which define a closed area in the 2D histogram, a pixel is classified as skin if its color components values are inside the corresponding area, otherwise it is labelled as non-skin. A bias allows making wider or narrower this area, and thus achieving different tradeoffs in the skin pixels correctly classified/non-skin pixels misclassified (SC/NSF) ratio. On the other hand, histograms of training data have been used to build a Skin Probability Map (SPM) in a colorspace. An SPM is a look up table which assigns a color its probability of being skin. A pixel is classified as skin if the probability associated to its color in the SPM satisfies a threshold. Different SC/NSF ratios can be achieved by modifying the threshold.

Receiver Operating Characteristic (ROC) curves have been used to evaluate the performance of both solution in each colorspace. Therefrom, we have concluded to use rules-based classifiers built on the 2D histogram of IQ components from YIQ colorspace, and on the 2D histogram of UV components from YUV colorspace, together with a SPM in RGB. To merge the output of each classifier, logic `AND` and `OR` functions have been evaluated. The `AND` of the outputs yields a better NSF percentage at the expense of the SC percentage, whereas the `OR` leads to the contrary effect. ROC curves have also been used to determine the values of parameters, thresholds and logic combinations that imply the optimum set of SC/NSF ratios.

0-7695-2940-2/07 $25.00 © 2007 IEEE
DOI 10.1109/FCCM.2007.24



a) b) c) d)

Figure 1. Debugging output: a) original image from the camera; b) skin segmented image; c) coordinates of the maximum of each convolution; d) place where the user hand is pointing at.

Xilinx System Generator has been the tool used for implementing the rules-based and SPM-based classifiers, and the colorspace transformations. The whole classifier occupies 2059 LUTs, 802 flip-flops, 1192 slices and 30 BlockRAMs in a XC2V4000 FPGA.

3. Hand gesture recognition

Once the image has been segmented the next processing task is to look for the pointing gesture, shown in Fig. 1a. The solution adopted in this work consists of convoluting the binary image from the skin classifier (Fig. 1b) with three different templates: one representing the forefinger, another the thumb and the third the palm. This modularity makes easier the addition of new functionality to the system through the recognition of more gestures. In the templates, the value 1 is associated to skin, and -1 to background. Due to the large size of the hand, the image is zoomed out by a factor of 5. This limits the size of the largest template to 30×30 pixels. The convolution with the templates have been designed in System Generator using distributed arithmetic. It occupies 7280 LUTs, 4626 flip-flops, 5364 slices and 29 BlockRAMs (the templates share the BlockRAMs where the sequential stream of pixels is stored).

Each convolutional module sends to the MicroBlaze soft processor, through the OPB bus, its maximum value and its coordinates on the image (marks in Fig. 1c). A software algorithm running in MicroBlaze decides that a hand with the wanted gesture is present when the maximum of each convolution reaches a treshold and their relative positions satisfy some constraints derived from training data. Then, the algorithm returns the position of the forefinger (where the red lines crosses in Fig. 1d). Otherwise, it reports that no pointing hand is detected. The Fig. 2 depicts the block diagram of the overall system. It can process 640×480 pixel images at more than 190 frames per second with a latency of one frame.

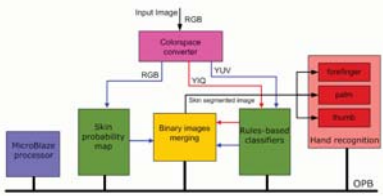


Figure 2. Block diagram.

With training and evaluation purposes, two different video database with white people hands under changing illumination conditions in very different backgrounds have been collected. The images also contain skin-like colored objects like wooden objects or cardboard. Some images of example are available at [3].

The goodness of the gesture recognition relies upon the skin classification: if it classifies correctly the pixels the hand pointing pose is easily detected when it is present. The classifier achieves good performance ratios around 90% on SC and 10% on NSF. However, results get worse on either highly saturated or shadowed skin, where its color changes dramatically. To improve the results in these situations, an algorithm for dynamically adapting the skin classification has been developed to be executed on MicroBlaze. It tunes the biases and the thresholds of each skin classifier and the merging of their binary output images to their suitable values in order to achieve the optimum SC/NSF ratio, in function of the number of pixels classified as skin in the image, the maximum value and the coordinates of each convolution, and the detection or not of the pointing hand pose.

Acknowledgement: This research has been funded by MTyAS of Spain, IMSERSO RETVIS 150/06.

References

[1] F.J. Toledo, J. Martínez, and J. Ferrández. FPGA-based platform for image and video processing embedded systems. In *Proc. 3rd Southern Conf. on Programmable Logic*, 2007.

[2] F.J. Toledo, J. Martínez, F. Garrigós, and J. Ferrández. FPGA implementation of augmented reality application for visually impaired people. In *Proc. Int. Conf. Field Programmable Logic and Applications (FPL)*, pages 723–724, 2005.

[3] http://wsdetcp.upct.es/Personal/JToledo/Skin/images.

[4] W. Luk, T.Lee, J.Rice, and P. Cheung. Reconfigurable computing for augmented reality. In *Proc. IEEE Symp. Field-Programmable Custom Computing Machines*, pages 136–145, 1999.

[5] W. Piekarski, R. Smith, G. Wigley, B. Thomas, and D. Kearney. Mobile hand tracking using FPGAs for low powered augmented reality. In *Proc. 8th IEEE Int. Symp. on Wearable Computers, ISWC04*, pages 190–191, 2004.