Extending human-computer interaction by using computer vision and colour recognition

Talmai B. de Oliveira, Leizer Schnitman, Fabíola G. P. Greve
talmai@ufba.br, leizer@ufba.br, fabiola@im.ufba.br
Pós-Graduação em Mecatrônica
Universidade Federal da Bahia
40.210-630 - Salvador - BA - BRASIL

J.A.M.Felippe de Souza
felippe@dem.ubi.pt
Dept of Electromechanical Engineering
UBI - University Beira Interior
6201-001 - Covilhã - PORTUGAL

***Abstract*— Computer vision techniques supply us with promising human-computer interaction methods by analysing and recognising human movements. The process of detection and tracking human body parts is one of the main steps necessary to reach a robust and precise recognition. Nevertheless, this task is rather difficult, specially when the response from that interaction is required to be in real time. This work provides a study as well as an example of colour recognition using a web cam as computer vision for human-computer interaction.**

***Keywords*— Computer Vision, Human-Computer Interaction, Colour Recognition, Web Cam**

I. Introduction

Research related to human-computer interaction has had a huge impact on software used to date. Terms such as “windows”, “menus”, “mouse”, “icons”, “click”, and “double-click” are not strange and have already become part of every computer-user vocabulary nowadays. Myers [1] says that “almost all modern applications present resemblance in their interface and utilization”. However, due to the growing tendency in computer power and to new forms of interaction and data exhibition, he concludes that “a point has been reached where it will not be possible to continue to use a *WIMP* (*Window, Icons, Menus and Pointing devices*) interface model to interact with a computer”.

This is because, besides limiting the speed, the ease of use and the overall experience of the user, this interface model forces the user to adapt himself to the computer and not the opposite. Following efficiency and functionality, most users agree that the most important factor for an applications success is how easy it is to use it [2]. Researches inspired on human-human interactions [3] have been one of the options for an alternative interaction technique. Voice recognition, gesture and motion detection are just a few examples of this. The greatest motivation behind these researches is the fact that human beings easily grasp this kind of interaction method and intuitively communicate using them. This implies that the existing techniques of human-computer interaction (HCI) will need to change in order to allow the effective use of the information available as technology evolves.

One choice available to reach a satisfactory and efficient HCI is to recognize and track parts of the human body [4]. The use of sensory gloves [5] have been suggested to achieve this. They allow for greater and more precise extraction of movement and hand characteristics, and are free of problems related to occlusion, although there have also been cases of magnetic interference [4]. This is one of the reasons why alternative, noninvasive visual sensors which can analyse and detect movement have been studied. But this inherently has its own set of problems, such as light interference and tracking of complex movements.

This work provides a study of colour recognition using a web cam as computer vision for HCI as well as an example that uses this model of human-computer interaction, where movements of the mouse are controlled using a web cam.

The remaining sections are organized as follows: Section II discusses some related work. Section III describes the process of colour recognition and computer vision. Section IV, V and VI show how this is done and also provides an example. A summary of the challenges make up the final section.

II. Related Work

Bolt [3] describes one of the first designs of HCI by using voice and gestures, that allowed the user to manipulate geometric figures on the screen. It is curious how many system since then have not offered much more than that, even thought technology has evolved. Price [6] implemented a vision system on a humanoid robot using an modified HSV colour space, that would filter the desired characteristics in order to track in real-time multiple objects of varied colours.

Koike [7] demonstrated how gestures could be used to integrate multimedia information with traditional printed material. In his proposal, the reader points at a picture printed on a book and the system initializes the appropriate resource; being it a web page, an interactive simulation or a video. By using an infrared camera, segmentation problems were solved. In another related work, Koike [8] extends his research to provide an interface capable of

This work is supported by CNPq - Brazil

using two hands to draw geometric figures. It is shown that within fifteen minutes of training, users would be able to perform tasks with their hands quicker than the traditional *WIMP* approach. Olivers' [9] system is based on blobs to determine contours and to recognize certain body parts and positions inside these contours. Blobs are groups of pixels related by some threshold value. By using a monitoring center he can also predict possible trajectory of movements.

III. General Overview

The process in which the image is captured and analised is divided into four steps: (a) image extraction, (b) image segmentation and recognition, (c) quadrant identification and (d) movement detection. The first step is self-explanatory. The next two steps consist of image segmentation, background subtraction and quadrant division.

The outcome of these steps is the recognition of where the selected colour is in the image, and the grouping of these points to determine where there exists the greatest concentration of the colour. Lastly, the sequence of frames are studied in order to obtain a sequence of movements.

A. *Real-time Video Extraction*

In this first step, the image that is captured by the web cam is extracted. Using Vídeo4Linux [10], it is possible to control the web cam to produce a single stream of data direct to memory. The web cam used was the `Logitech QuickCam Express`. This is a CCD[1] web cam that provides an image in the *RGB* colour space, where each colour is represented by three numbers that represent each of the components: red, green and blue (see Fig 2). We chose to study colour recognition without reducing the size of the image. Since the image is distributed in an array of $352x288$ pixels, for each cycle of image extraction we have to analise 101.376 pixels.

B. *Segmentation and Recognition*

The second step is where the image is segmented. Algorithms in charge of processing the image and detecting colours are used. They divide the image into quadrants and detect what it is background and what it is not. These algorithms will be explained in section IV.

C. *Identified Quadrants Gathered*

The third step deals with the storage of all the identified quadrants during a certain time slice to detect a sequence of movements. Once formalized, this information becomes available to other applications.

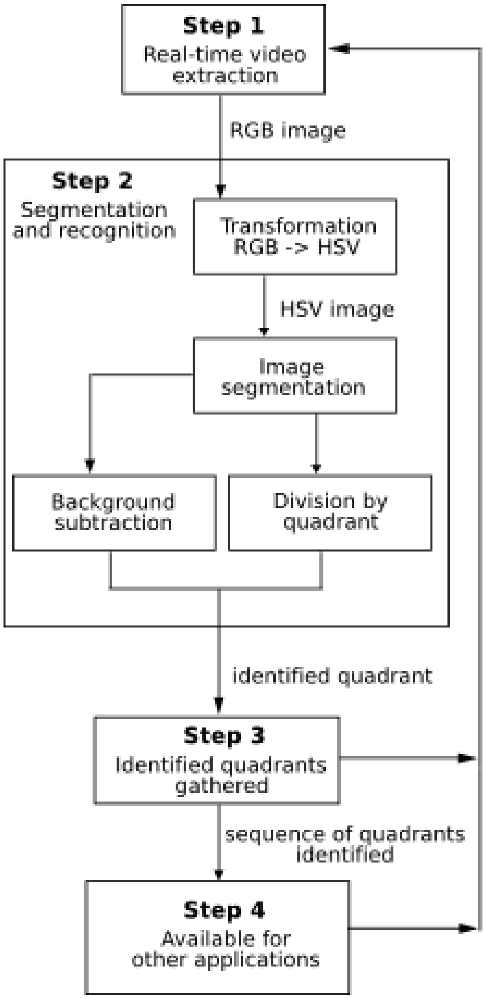

Step 1
Real-time video extraction
RGB image
Step 2
Segmentation and recognition
Transformation RGB -> HSV
HSV image
Image segmentation
Background subtraction
Division by quadrant
identified quadrant
Step 3
Identified quadrants gathered
sequence of quadrants identified
Step 4
Available for other applications

Fig. 1. Functional steps.

D. *Available for Other Applications*

The results of an application developed for HCI using the information captured by the web cam are presented. The application is the movement of the mouse using the web cam. In this case, when comparing to current *WIMP* interface model, it is shown how it is so much more natural to have a HCI using hand movements.

The steps which are presented and used in this work, to recognize colours and identify movements, can be seen in Fig 1.

IV. Image Segmentation and Recognition

The image captured is distributed in an array. Each element of this array is related to a single pixel of the image. The data contained in each pixel is structured in an RGB colour space. Each colour is represented by three numbers that represent each of the components: red, green and blue.

Once captured, it is necessary to segment the image. Image segmentation consists of grouping pixels with similar

[1] (*Charge-Coupled Device*) A technology that uses semi-conductors that are light sensitive. Each CCD chip consists of an array of cells, where each individual cell is activated by an electrical charge prior to exposition to light.

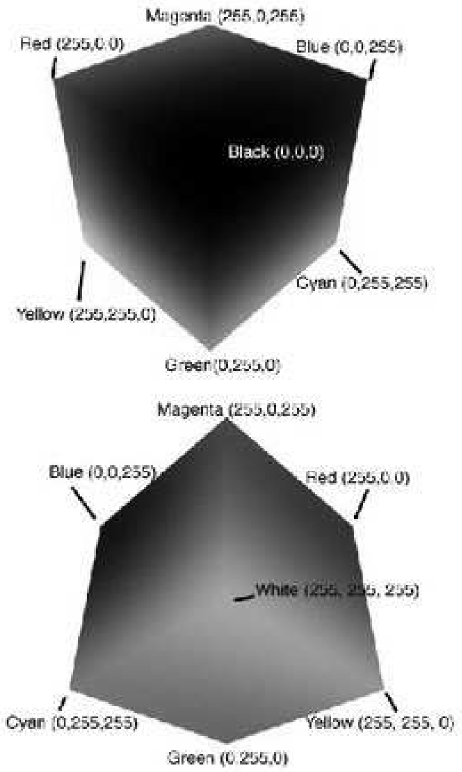


Fig. 2. The RGB cube.

characteristics. This task, see Wang [13], is one of the most difficult to perform and it has not had a unique model or solution yet.

According to [13] & [14], one can classify the algorithms of segmentation by:

- histogram;
- clustering;
- border detection;
- fuzzy logic based and
- artificial neural network based.

The techniques based on histogram identify, in the colour space, one or more peaks and their surroundings so that each pixel can be classified. The border detection techniques - very similar to what human beings use to identify objects - is a process by which pixels in the border of objects are located and enhanced, increasing contrast between border and background. This process detects variation in the luminosity values of an image.

Fuzzy logic based segmentation analyses the pertinence functions for each pixel and for all the fuzzy blocks defined. Fixed blocks of pixels are obtained by the process of defuzzyfication and then subdivided in maximum connected regions. Artificial neural network based segmentation on the other hand proposes the training of a network capable of correctly classifying pixels.

Clustering techniques identify homogeneous point in the colour space (such as RGB, HSV, etc...) and mark each group as a different region, as long as these points are within some predefined threshold. In this work, the colour of each pixel is used to group them into different regions. Because of its speed and low cost, these techniques were preferred.

A. *Simplifications and Limitations*

Due to the real-time requirements of the design, the algorithms used were simplified, maintaining only the fundamental aspects to reduce processor usage and time of detection.

Colours belonging to the objects that are detected by the camera are highly dependent on the environment's conditions of illumination. In this sense, not only it is required very good illumination, but it has also to be consistent during the time frame of capture and detection [12]. Additionally, it is important to note that, if the environment's conditions change from the moment of calibration to the actual capture moment, the whole process of recognition and tracking can (an probably will) fail [11]. So, when these techniques of HCI are applied, there will always be a possibility of failure and uncertainty, requiring from the user a constant monitoring and correct interpretation of what is happening [11].

The authors were capable of developing a frame-by-frame segmentation that can detect colour and movement in spite of the automatic exposure correction (*AEC*)[2]. The solution provided is robust due to the transformation of the colour space from *RGB* to *HSV*.

B. *Transformation to the HSV Colour Space*

RGB is a colour space broadly used, but *HSV* (see Fig 3) with its components hue, saturation and value is preferred because of its speed in the process of segmentation. The hue determines the specification of the intrinsic colour, saturation describes how pure the colour is and value the brightness of the colour.

Conceptually, the colour space *HSV* is seen as a cone. Viewed from its circular side, the hues are represented by the angle of each colour. Red is designated as 0°, whereas yellow, for example, is 60°. Saturation is represented by the distance to the center of the circle. Highly saturated colours are placed on the borders of the cone while colours that are gray (which have no saturation at all) are placed in the center. The value is determined by the vertical position inside the cone. The vertex of the cone contains no brightness at all, therefore all the colours are black. And on the base are placed colours with great brightness.

The algorithm used for the transformation is defined as follows:

[2] The *AEC* is a characteristic of CCD cameras that cant be disabled on the majority of the web cams. The *AEC*, optimizes the brightness of the image for human perception by normalizing the captured light. When an object is placed very close to the web cam, due to its small size, the total amount of light is reduced. The *AEC* then increases the brightness of every pixel to maintain the intensity of light constant.

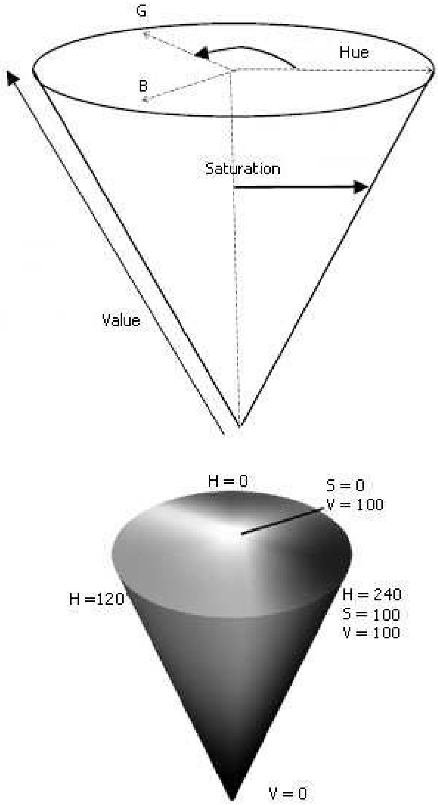


Fig. 3. HSV Cone

$$V = max(R, G, B) \tag{1}$$

$$S = \begin{cases} \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)}, \max(R,G,B) \neq 0 \\ 0, \max(R,G,B) = 0 \end{cases} \tag{2}$$

The original transformation of hue, suggested by Foley [15], required that the calculations of arcsine and square root were known. To reduce the processor usage it is preferred an equivalent transformation suggested by Cardani [16]. Depending on the largest value of the *RGB* components, with this transformation one of the three expressions is chosen, and then, if the hue is negative, a simple addition is performed.

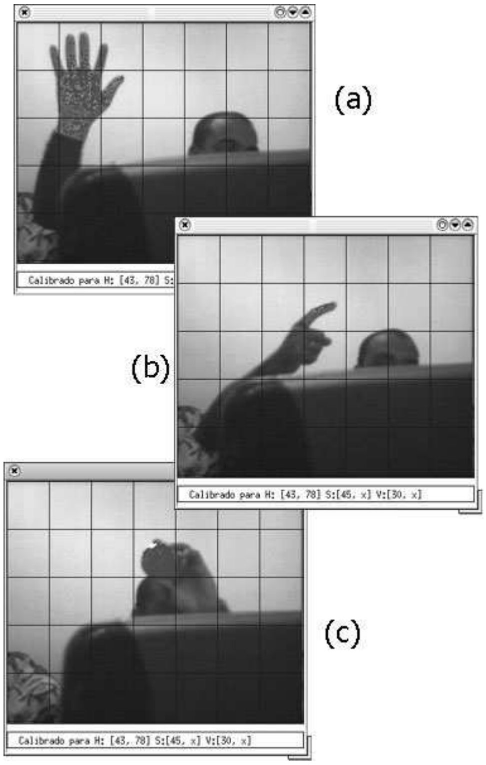
Hue Algorithm [15]:

$$H = acos\frac{0.5((R-G)+(R-B))}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}} \tag{3}$$

Hue Algorithm [16]:

$$H = 60 * \begin{cases} \frac{(G-B)}{MAX-MIN}, MAX = R \\ 2.0 + \frac{(B-R)}{MAX-MIN}, MAX = G \\ 4.0 + \frac{(R-G)}{MAX-MIN}, MAX = B \end{cases} \tag{4}$$

$$(H < 0) \ ? \ H = H + 360 \tag{5}$$



Calibrado para H: [43, 78] S:

Calibrado para H: [43, 78] S:[45, x] V:[30, x]

Calibrado para H: [43, 78] S:[45, x] V:[30, x]

Fig. 4. Image segmentation using 5x7 quadrants. (a) hand using yellow glove, missing a finger (b) hand using only a yellow wrapper around the finger (c) yellow ball

where *MAX = max(R, G, B)* and *MIN = min(R, G, B)*.

C. Colour Detection and Background Subtraction

After transformation to the *HSV* colour space, the segmentation process continues by analising every pixel of the image and seeing if they are of the colour which is being searched for. The image is also divided into quadrants and the numbers of detected pixels inside each of them are counted. By counting the number of pixels it is not difficult to determine where there is a greater concentration of points and to determine the most representative quadrant.

Finally, the actual frame is compared with the previous one, where the points that have not changed are identified and marked as belonging to the background. Ivanov [17] and Javed [18] agree that the process of background subtraction is interesting since it speeds up the efficiency of the algorithms related to computer vision. The two main existing approaches are:

1) The use of statistical properties of the background during a certain time period. It assumes that the background is totally static in all aspects, such as geometry, illumination, etc...
2) Assume that the background is almost static, changing very slowly and infrequently. In this case the person (or object) is the one in movement.

In most cases, it is acceptable to suppose that the background is static and the objects being analised move. So, to reduce time and effort, there is no need to have to analyse points that clearly belong to the background during an extended period. The following approach is then used to detected the background:

$$diff = \begin{cases} 0, |(last\_point - previous\_point)| < \epsilon \\ 1,\ otherwise \end{cases} \quad (6)$$

Where ϵ is an acceptable tolerance threshold used to compensate for possible interferences, abrupt changes in the illumination and errors during the segmentation process. We can see in the Fig 5 the result of the background detection algorithm.

In the moment that the most representative quadrant is identified (the one with the greatest number of points detected), it is verified if, in the same quadrant, there does not exist a large number of points belonging to the background. If so, then there is a good chance that the quadrant in question has an object with of the colour which is being tracked, but that belongs to the background. In this case, another quadrant must be choosen.

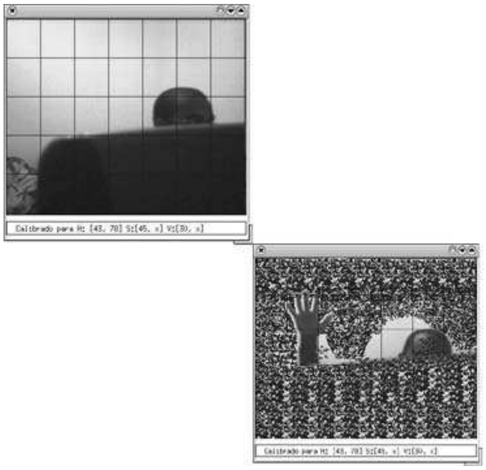
A simplified algorithm is shown below:

```
$buffer\_RGB = capture\_video\_image();$
$buffer\_HSV = tranform\_RBG\_HSV(buffer\_RGB);$
$while(checking\_all\_points\_buffer\_HSV)$
|  $current\_quadrant = detect\_quadrant(current\_point);$
|  $if(current\_point = colour)$
|  |  $increment\ num\_points[current\_quadrant];$
|  $end\_if$
|  $bg\_point = last\_frame(current\_point);$
|  $if(current\_point \in bg\_point)$
|  |  $increment\ num\_points\_previous[current\_quadrant];$
|  $end\_if$
$end\_while$
$ident\_quadrant = analise\_quadrants();$
$if(ident\_quadrante\ has\ more\ num\_points\_previous)$
|  $re\_analise\_quadrants\_detected();$
$end\_if$
```

In Fig 4 it is shown that the algorithm is capable of detecting point in the image of the selected colour, as well as dividing the image into quadrants. In the Fig 6, the hand is in constant movement. But, the algorithm has a certain internal threshold making it capable of identifying a good portion of the glove.

V. Sequence of Movements

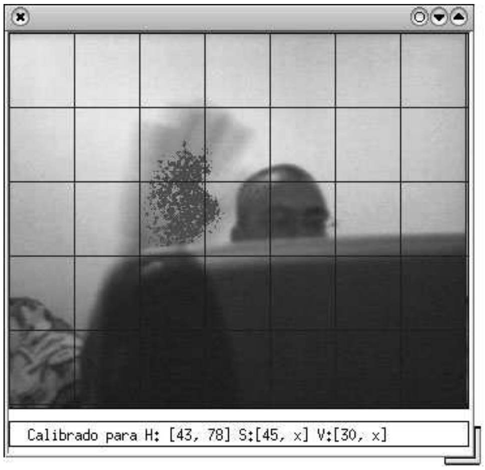
After capturing a sequence of movements, an array of identified quadrants is offered to any possible computer application. In Fig 7 the sequence of quadrants drawn on the screen for viewing can be seen. The web cam captured in the first image an elliptical movement and in the second a small square.



Calibrado para H: [43, 78] S:[45, x] V:[30, x]

Calibrado para H: [43, 78] S:[45, x] V:[30, x]

Fig. 5. Background Subtraction: Comparison between two captured frames



Calibrado para H: [43, 78] S:[45, x] V:[30, x]

Fig. 6. Image Segmentation using an yellow glove while the hand is in movement.

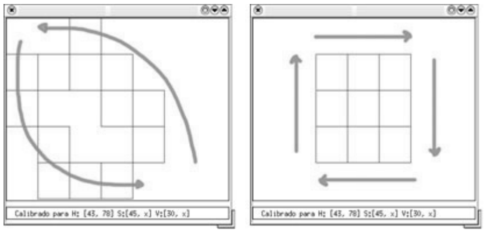
VI. Results: Mouse Movement

To move the mouse, it is necessary to dimension the quadrants to an $3x3$ matrix. One can see in Fig 8, if the quadrant identified is (0, 0), then the mouse will move diagonally to the upper left corner. The central quadrant stops all mouse movements.

Although capable of capturing, detecting and treating movement in real-time, there still exists a need of greater precision of the movements.

For now, the mouse moves a little for every cycle of detection, and since the image was limited to nine quadrants, there was no way to change the speed of the mouse. By changing the matrix to $5x5$, it was then possible to have different speed values for the mouse: slower, if closer to the centre; and faster, if not.

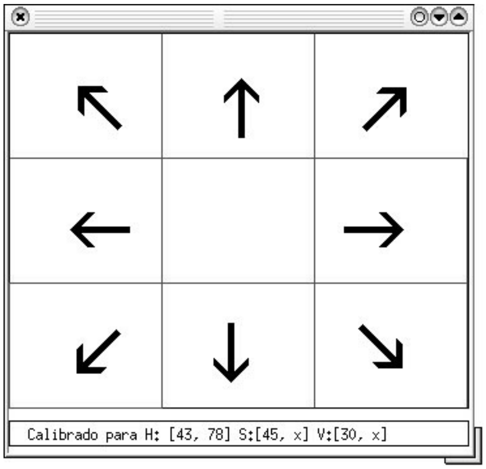
Yet, it is necessary to mouse click occasionally since for now, the mouse only *moves*.



Calibrado para H: [43, 78] S:[45, x] V:[30, x]

Calibrado para H: [43, 78] S:[45, x] V:[30, x]

Fig. 7. Captured image detection. The red squares are the quadrants identified by the algorithm. The green arrows where placed by the authors to show the movement performed.



Calibrado para H: [43, 78] S:[45, x] V:[30, x]

Fig. 8. Identification of the direction of the mouse movements that depends on which quadrant is chosen.

VII. Conclusions

Computer vision techniques supply us with promising human-computer interaction methods by analysing and recognising human movements. The process of detection and tracking human body parts is one of the main steps necessary to reach a robust and precise recognition. Nevertheless, this task is rather difficult, specially when the response from that interaction is required to be in real time.

This work focused primarily on the recognition of colours, demonstrating and applying many simple and efficient techniques for computer vision. It has also presented a practical application using this model of HCI.

The process of transformation of the colour space *RGB* to *HSV*, the segmentation, the background detection and the most representative quadrant chosen has been detailed. There are some topics where further developments are still possible, such as better calibration techniques, segmentation and methods on how to improve and reduce the interference of the light intensity of the environment.

References

[1] MYERS, B., HUDSON, S., PAUSCH, R., *Past, Present, and Future of User Interface Software Tools*. In: ACM Transactions on Computer-Human Interaction, Vol. 7, Number 1, March 2000, p. 3-28.

[2] BATINI, C., et al., *Visual Query Systems: A Taxonomy*. Visual Database System II, E. Knuth and L. M. Wegner, eds., Elsevier Science Publishers B.V. (North-Holland), 1992. p. 153-168.

[3] BOLT, R. A., *Put That There*. Computer Graphics. Vol 14, 1980, p. 262-270.

[4] WU, Y., HUANG, S., *Nonstationary Colour Tracking for Vision-Based Human-Computer Interaction*. In: IEEE Transaction on Neural Networks, Vol. 13, Number 4, July 2002. p.948-960.

[5] WEXELBLAT, A., *An Approach to Natural Gesture in Virtual Environments*. In: ACM Trans. On Computer-Human Interaction (TOCHI), Vol. 2, Number 3, September 1995, pp. 179-200.

[6] PRICE, A., TAYLOR, G., KLEEMAN, L., *Fast, robust color vision for the monash humanoid*. In: Australian Conference on Robotics and Automation, Melbourne, 30th of August to 1st of September, 2000, p. 141-146.

[7] KOIKE, H., SATO, Y., KOBAYASHI, Y., *Integrating Paper and Digital Information on EnhancedDesk: A Method for Realtime Finger Tracking on an Augmented Desk System*. In: ACM Transactions on Computer-Human Interaction, Vol. 8, Number 4, December 2001, p. 307-322.

[8] KOIKE, H., et al., *Two-handed Drawing on Augmented Desk*. CHI 2002, 20-25 of April, 2002.

[9] OLIVER, N., ROSARIO, B., PENTLAND, A., *A Bayesian Computer Vision System for Modeling Human Interaction*. Vision and Modeling Media Laboratory, MIT. 1999.

[10] Vídeo for Linux Resources. Available at: <http://www.exploits.org/v4l/>. Accessed on: 1st of August, 2004.

[11] MARTINKAUPPI, B., *Face colour under varying illumination - analysis and applications*. Department of Electrical and Information Engineering, University of Oulu.

[12] SENGUPTA, K., BING, W., KUMAR, P., *Computer Vision Games Using A Cheap (<100$) Webcam*. Department of Electrical and Computer Engineeering, Singapore, 2000.

[13] WANG, H., SUTER, D., *Color Image Segmentation Using Global Information and Local Homogeneity*. In: Proceeding of the VIIth Digital Image Computing: Techniques and Applications, Sydney, 10-12 of December, 2003.

[14] CECHINEL, C., *Técnicas de Segmentação de Imagens a Cores. Seminário Visão Computacional - CPGCC/UFSC - 2000.2*. Available at: <http://www.inf.ufsc.br/ visao/2000/Cores/>. Accessed on: 25th of August, 2004.

[15] FOLEY JS, FEINER, SK, e HUGHES, JF, *Computer Graphichs Principles and Practice: Second Edition in C*. Addison-Wesley, New York, 1996.

[16] CARDANI, Darrin, *Adventures in HSV Space*. 2000, p. 1-10.

[17] IVANOV, Y., BOBICK, A., LIU, J., *Fast Lighting Independent Background Subtraction*. MIT Media Laboratory. 2nd of February, 2001.

[18] JAVED, O., SHAFIQUE, K., SHAH, M., *A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information*. Computer Vision Lab, School of Electrical Engineering and Computer Science, University of Central Florida. 2002