Development of a Virtual Input Device Using Stereoscopic Computer Vision to Control a Vehicle in a Racing Game

Thiago Ribeiro de Azeredo
Universidade Candido Mendes
Campos dos Goytacazes, Brazil
E-mail: thiagoribeiro@gmail.com

Italo de Oliveira Matias
Universidade Candido Mendes
Campos dos Goytacazes, Brazil
E-mail: italo@ucam-campos.br

Weverson Machado de Oliveira
Universidade Candido Mendes
Campos dos Goytacazes, Brazil
E-mail: weversonmachado@yahoo.com.br

***Abstract*—Nowadays, in the technological world is not hard to see the growing concern about the interaction between users and electronic devices, where the dynamism and the usability have been a decisive factor in the design of new projects and where new developments with ability to increase the experience of the users are in focus. In this light, a system was developed being capable of performing the interaction between the user and the computer using cameras as input devices. It was specifically designed to allow a car in a racing game be controlled only by the user's hand movements without use of markers, where it is possible to turn left, to turn right, accelerate and to brake only moving the hands. For this, haar classifiers, stereoscopic vision and video card programming were used. A racing game was also created to perform tests and validate the proposal. All work was done on Linux environment using C++ with OpenCV , Ogre3D, ODE and CUDA libraries. The system called "virtual wheel" proved satisfactory, having good quality and speed of response, even on a home computer.**

Keywords-stereoscopic vision; human-computer interface; racing game; computer vision.

I. INTRODUCTION

The electronics are increasingly present in our day by day and how to interact with them is a crucial factor for their use to be increasingly simple, practical and functional. The desire to facilitate the use of devices has contributed to a significant increase in agility and dynamism to the user, making it easier to perform tasks. Another important factor is the aid digital inclusion of those who, regardless of reason, have difficulty dealing with keyboards, mice and joysticks.

It is easy to see that computing has evolved dramatically in recent years, where the devices have become increasingly faster and smaller operating systems and seeking to deliver more features and flexibility for the user. But usability has always been restricted to the keyboard and mouse that despite undergoing improvements, they had no substitutes. Today, new forms of access are being explored. The new touch screens, and tilt position sensors, voice commands and computer vision are drawing the attention of the industry in several areas such as video games, smartphones, tablets, automobile industry, among others.

Although this theme is not new, the time consumed by the algorithms was an impediment to its use in an application that needs answers almost immediately. Today, some techniques have evolved into the performance, progress has been made in processing power of home computers and video cards have a programmable interface idle low complexity for mathematical computations. These facts combined with the low cost of good quality cameras contributed to the popularity of this line of research, although some are still complicating factors for the techniques of computer vision and therefore compromise the quality of results, such as the background unstable lighting conditions. These benefits can be seen at [1], that presents a technique for fast objects recognizing in images, [2] and [3] shows techniques for obtaining the distance of objects creating disparity maps.

Being a computer environment for experimentation with new forms of interaction, this study aims to create a computer system capable of converting movements performed with the hands free in front of cameras, control actions in a car. Namely: Accelerate, brake, turn left and turn right. This process must occur without markings or devices in the hands of the user, and have to be quick enough to get smooth movements. This system is called a virtual steering wheel.

The computer vision techniques employed are even more interesting is observed that serve as basis for several important tasks, such as aid for the visually impaired locomotion, control of robots and automatons touchless control interface [4] (very useful for environments such as hospitals).

Section 2 will describe the steps to prepare the project, as the images capture , background substract, hands detection, obtaining the hands depth. In the end, the performance analysis. In Section 3 the virtual racing simulator creation as a test environment will be explained, and the proposal used to integrate it into the project will be described. Section 4 will present conclusions and suggestions for future work.

II. BUILDING THE "VIRTUAL WHEEL"

As previously reported, the paper proposes the development of a system capable of capturing images generated by cameras and, online, treat them and turn them into motion. Specifically in this case was chosen to identify the hand's closed fist and its conversion into motion, controlling the car with four functions: turn left, turn right, accelerate and brake.

 ISBN: 978-1-61208-202-8

All these functions with the degree of sensitivity adjusted, allowing a precise control.

In order to allow this project, the "virtual wheel" was created using the OpenCV library [5] (Open Source Computer Vision) and CUDA [6]. OpenCV is cross platform and consists of more than 2500 already optimized algorithms for capturing, creating, editing, processing and obtaining information from images. Since CUDA is a library developed by *NVIDIA* company that allows the development of algorithms for mathematical calculation can be run on video cards.

A. Image Capture

To make capturing the images you must first make a selection of cameras. This factor is vital to the process due to several variables that relate to this device, such as level of distortion of the lens opening angle lens, focusing system, quality shutter, shutter speed, among others.

For this work the camera chosen was the ps3 eye (Fig. 1). Created by *Sony* for the play station 3, was born with the goal of combining quality with performance. This camera has a lens opening 56 degrees, a microprocessor capable of sending images without compression, allowing for better utilization and capacity to generate 60 frames per second capture frames of 640x480 pixels and 120 frames per second at 320x240 pixels. Its speed was the primary factor for the choice, because the response time of the system must be transparent to the user. Thus, two cameras were purchased to be used in conjunction with this system.

Figure 1. Image of Sony PS3 Eye

The image size used in this study was to 640x480 pixels, in order to obtain large amounts of information to work.

At each step an image of each camera is captured and converted into two other images in grayscale. One with the average color and the other containing only the red channel. Each image will have a utility in the following steps.

B. Background Substract

The subtraction of the background consists of making the algorithm learn what was already on the scene and what is new. In this case, this process is done so that the whole environment is eliminated and only the user is recognized. This is done to eliminate irrelevant information for the process.

The process of background subtraction used is quite simple. Using only the red channel of the camera, the user should avoid being exposed during the first 100 frames, at which time the catch is made. The average of the images from each camera is generated separately, taking then the average color of each pixel of the same to the left and right image. These images are used as average of comparison, where the color of each pixel of a new image is compared in the same coordinate with the image of the average so that if the variance between the color exceeding a predetermined threshold, this new color is considered something new on the scene. So is created a mask for the image.

C. Hands Detections

Object detection systems use feature detection algorithms, there are several techniques that can be employed for this task, but the detector created by Vioja-Jones has been widely used recently [1]. The Viola-Jones is able to detect objects with precision, high accuracy rate, low false positive rate and low computational cost. The algorithm consists of 3 parts: The first is the representation of the image in a feature space based on the Haar filter. The second is the assembly of a classifier based on Boosting able to select the most relevant features. The third part concerns the cascade combine classifiers to ensure good performance and processing speed (Fig. 2).

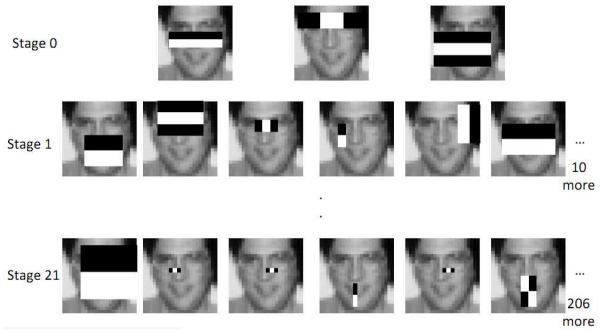

Figure 2. Example identifying characteristics using the algorithm of Viola Jones removing [1]

Both the speed of recognition as the quality of the final result of the Haar like are affected by factors such as:

- Number of images used in training;
- Images size;
- Number of features used for recognition;

So these parameters must be adjusted to achieve a good quality without losing performance. This work requires the effort to make several attempts to reach the goal, which in this case occurred six times.

The training process starts with getting the images. In many areas of science you can get them in stock photos public which are widely used in research projects. This is useful because whith it is possible for example, compare

 ISBN: 978-1-61208-202-8

the quality of work using the same images. But there was no success in finding images containing fists stock photos, thereby obtaining such images was made from photographs.

There was developed a small program with the function of capturing a sequence of frames of the cameras and converting each individual image files. During this process, the program recorded 250 frames from each camera, then completing 500 paintings. This little program was used twice to take pictures of the user on different days, then generating 1000 photographs. Of these 1000, 306 were randomly selected to be part of the training process, of which 578 examples were generated for left and right hands, some samples were discarded because of poor quality. Fig. 3 examples a photo used and Fig. 4 shows some hands selected.

In the quest for improved quality of recognition, the equalization of each image in the training and recognition was used.

These hands were then placed for training over 94 background images. These images were selected to ensure a diversity of environments and consist of rooms, textures and people. It is worth noting that all photographs of persons used in this step became an issue in order to remove his hands. Among these images, 29 were obtained using the camera Nikon Coolpix S4000, 5 were obtained from the camera ps3 eye, while the other 59 were obtained from images free databases.

Figure 3. Example of photography used for training

In generating the final version, the training process took about 8 hours, where 42 iterations were performed for a total of seven features were identified to obtain a good quality results.

After obtaining the location of both hands on the image, the center point of each is identified and used for the discovery of the angle which they form, as showed on Fig. 5.

This angle is then adjusted to suit the sensitivity of the system and then is used to rotate the wheel.

To turn possible to visually evaluate the behavior of the system, an ellipse was displayed on the screen with the second function of rotating the angle generated.

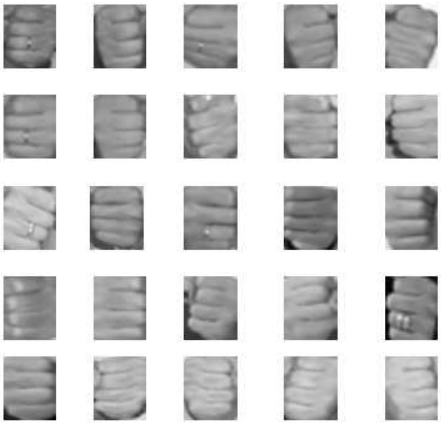

Figure 4. Example of hands identified

Figure 5. Proof of identification of the angle between the hands

D. Obtain the deph of hands

The term comes from stereoscopic two Greek words that represent "vision" and "solid" and the origin of this knowledge area can be rescued at least the year 1838.

In general, the depth information is obtained from a system for processing stereoscopic (Fig. 6) three main problems: calibration correlation and reconstruction. In the calibration seeks to determine the parameters which describe the acquisition system used. The correspondence problem is to determine which element in the captured image from a point of view corresponds to a given element in the captured image from another point of view. In turn, by rebuilding seeks to retrieve information from depth based on parameters obtained in the calibration step and pairs of corresponding points obtained in the step of matching.

For this study, we used the algorithm proposed by [3] that is capable of generating a disparity map quality and extremely fast using a GPU.

The GPU (Graphics Processing Unit) is responsible for all calculations to generate graphics on electronic devices, especially three-dimensional environments. Initially created to offload the CPU, allowing streamline tasks such as texture mapping and rendering polygons, received more functionality over time, such as rotation and translation of vertices and

 ISBN: 978-1-61208-202-8

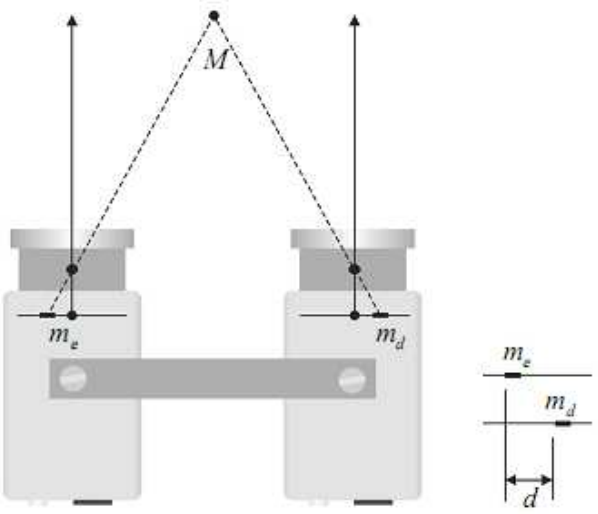

Figure 6. Simplified model of stereoscopic vision([2])

now support lighting calculations and processing of vertices, and direct manipulation of pixels. These features are designed to grow with the gaming industry, not only computers but also videogames, working with similar architecture.

As you can see in Fig. 7, the architecture of the central processing unit (CPU) includes communication with many other structures and has a throughput small compared with the GPU. The difference becomes large when compared to the speed of memory access.

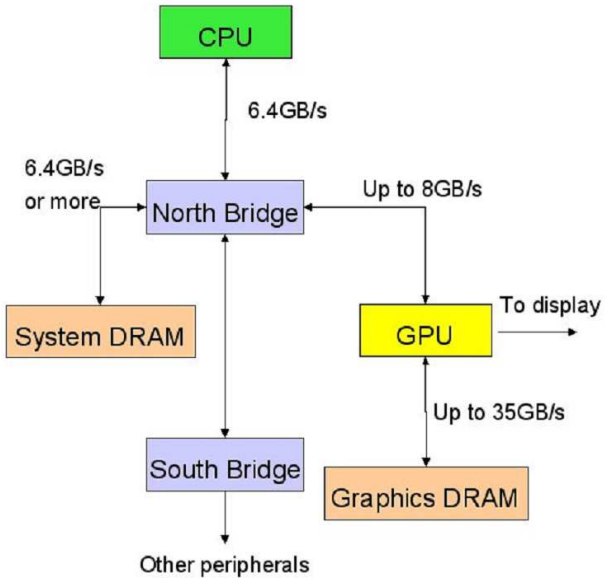

Figure 7. Defining the outer structure of the CPU and GPU capabilities with maximum data transfer. [7]

The Table 1 compares two of the most modern and powerful devices found in Brazil today, and hence its high price. You can clearly see that the *NVIDIA* GTX580 is actually better prepared for the scenario of mathematical calculations that the processor Intel core i7 3930k.

Table I
COMPARISON OF CALCULATION POWER AND PRICE BETWEEN CPU AND GPU. INFORMATION FROM [8], [9], [10], [11], [12]

	Intel Core i7-3930K	*NVIDIA* GTX580
Kernels	6 (12 threads)	512
GFLOPS	105	1.581
Memory Speed	21 GB/s	192.4 GB/s
Cache	12 MB	768 KB
Frequency	3.2 Ghz	772 Mhz
Price (Brazilian Real)	R$ 2.399	R$ 1.899

Knowing that calculations images are nothing more than multidimensional arrays with numeric values and that the procedures for rotation, translation, transformation and lighting are just mathematical equations is easy to understand as it was thought the creation of the CUDA (Compute Unified Device Architecture). This, in turn, was created in 2007 by *NVIDIA* and became a powerful platform to perform mathematical calculations in parallel directly to the video card, using high-level language. Unavailable until now, where it was necessary to program in assembly. Process that began around 1998.

The technique of [3] was compared with the literature to assess the quality of their results as seen in Fig. 8. This technique was then used in the captured images to provide the map of disparities. With prior knowledge of the location of the player's hands, together with the removal of the background it was possible to determine the distance of the hands using the average of the colors within the bounding box, each hand. This value is then framed between -1 and 1, where numbers greater then 0 means speed increases and smaller then 0 means speed decreases. The variation occurs in relation to its proximity to either 1 or -1. Thus, the farther the hands are the camera, the lower the speed and the closer, the greater the speed.

E. Performance

At the end of development, the system was tested and it was possible to obtain the execution time. At this point, each loop step took an average of 128ms to run.

Table 2 shows the time of each step separately and in the order they are executed. Looking for increase the speed of execution, an analysis was made in the steps used in order to locate those who consume reasonable time and that can run in parallel with others, in this way could make this time was absorbed without affecting the system structure.

It was identified that the detection step of hands would be a good candidate for this change. This was then ported to a

 ISBN: 978-1-61208-202-8

Figure 8. Comparison of disparity maps. a) original image b) expected result; c) result from[13]; d) result from [14]; e) result from [15] f) algorithm from[3] used on this work.

Table II
TIME SPENT BY EACH PROCEDURE

Step name	Time(ms)
Get new frames from cameras	40
Images preprocess	10
Hands detection	37
Disparity calculation	40
Move calculation	1

thread that runs in parallel whereas the main runs to obtain the disparity map, and then its time taken up by the main process steps.

Thus has a gain of 37ms, lowering the average time of each iteration to 91ms on average, generating 11 frames per second. This change has backpack fluidity for the result, but it was observed that there is another step where the same strategy could be applied, the image capture cameras. The capture process alone consumes 40ms, since each camera needs 20ms to return the image. With this further modification of a gain medium in 16ms, then lowering the average of the iteration to 75ms, which represents 13.3 frames per second. Compared to the initial time of 128ms with 7.8 frames per second, was obtained a gain of 41% and a degree of fluidity pleasant.

III. THE GAME

To evaluate the quality of results generated by the virtual wheel, a computational experiment was developed. The virtual wheel was integrated with the virtual racing car created by [16], [17], where its gameplay could be checked in a subjective way, after the user experience should be taken into account. The Fig. 9 examples the experiment views.

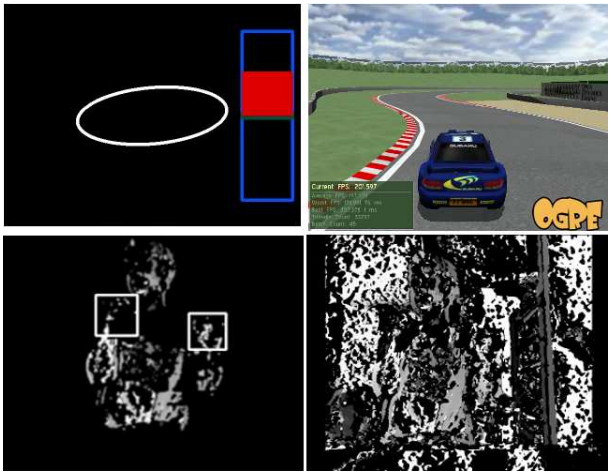

Figure 9. Representation of four views (a, b, c, d). a) represents the steering angle on the ellipse (inverted relative to the image) and acceleration based on the distance of the hands for the cameras at the bar; b) presents the response of the car in the game to the command; c) Represents the disparity map with background removal and hands found; d) Represents the disparity map of the moment.

The virtual scenario used here was developed in 2005, in Windows, in C + +, using Ogre3D [18] and ODE [19] engines. Since the virtual wheel was developed in Linux and libraries Ogre3D and ODE suffered several updates over the past six years, so it was necessary to translate the project to linux environment and update it to use the latest versions of libs applied. In this previous work, the whole project was used except the autopilot, not be relevant to the project scope.

As the project was running architected to allow the inclusion of new controls for the vehicle, a new control was implemented to integrate with the virtual wheel. This was developed using the network protocol UDP (User Data Protocol) which fits better in the proposal to be extremely fast due to the failure to implement controls and safeguards in existing TCP (Transmission Control Protocol). The use of network protocol for integration also allows new environments to integrate virtual driving future.

On average, 82% of the cycles was the detection of two hands, which during the game guarantees a quality experience satisfactory. To maintain control of the vehicle frame in which at least one hand is not located in the last value of the detection is sent as well as the last value of the acceleration and braking. As in a common environment

 ISBN: 978-1-61208-202-8

lighting is not constant, small variations occur in the map of disparities any time by changing the values of acceleration, but are not observed during the game.

IV. Conclusion and Future Work

The algorithms used to develop this work demonstrate adequate for this task. The viola-jones algorithm requires an effort to match the quality and performance using trial and error. The algorithm [3] proved to be extremely fast to generate the disparity map, although its quality is not fitted with other techniques from the literature, it is enough to reach the proposed objective. The results in uncontrolled environments demonstrate that the algorithm is quite robust. It is also possible to treat the result to minimize noise, improving the final quality.

The 13 frames per second achieved were sufficient to pass the feeling of immediate response to the player, allowing a good gameplay and spending confidence.

The atmosphere can be adapted to new features such as control of robot arm, moving the mouse, among others.

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 511–518, 2001.

[2] M. E. Stivanello, E. S. Leal, N. PALLUAT, and M. R. STEMMER, "Desenvolvimento de uma biblioteca para sistemas de visao estereoscopica para robotica movel," *VIII Conferencia Internacional de Aplicacoes Industriais*, 2008.

[3] D. Gallup, J. Frahm, and J. Stan, "Real-time local stereo using cuda," *NVIDIA Research*.

[4] J. P. Wachs, H. I. Stern, Y. Edan, M. Gillam, J. Handler, C. Feied, and M. Smith, "A gesture-based tool for sterile browsing of radiology images," *Journal of the American Medical Informatics Association*, vol. 15, pp. 321–323, 2008.

[5] (2012, Fev.) Opencv. [Online]. Available: http://opencv.willowgarage.com/

[6] (2012, Fev.) Cuda. [Online]. Available: http://www.nvidia.com/cuda

[7] A. Cuno and J. R. M. Vianna. (2011, Dec.) UFRJ. [Online]. Available: http://www.lcg.ufrj.br/Cursos/GPUProg/gpuintro

[8] (2011, Dec.). [Online]. Available: http://www.waz.com.br/produtos/101319

[9] (2011, Dec.) Techpowerup. [Online]. Available: http://www.techpowerup.com

[10] (2011, Dec.) Intel. [Online]. Available: http://ark.intel.com/products/63697/Intel-Core-i7-3930K-Processor

[11] (2011, Dec.) Hexus. [Online]. Available: http://hexus.net/tech/reviews/graphics/29509-gigabyte-geforce-gtx-550-ti-oc-graphics-card-review/

[12] (2011, Dec.). [Online]. Available: http://www.waz.com.br/produtos/101385

[13] R. Yang and M. Pollefeys, "Multi-resolution real-time stereo on commodity graphics hardware," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, p. 211, 2003.

[14] R. Yang, M. Pollefeys, and S. Li, "Improved real-time stereo on commodity graphics hardware," *Computer Vision and Pattern Recognition Workshop*, 2004.

[15] J. C. Kim, K. M. Lee, B. T. Choi, and S. U. Lee, "A dense stereo matching using two-pass dynamic programming with generalized ground control points," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 1075–1082, 2005.

[16] T. R. Azeredo, P. S. VIEIRA, A. A. S. NETO, and I. O. MATIAS, *Inteligencia Artificial aplicada a tomada de decisao em jogos eletronicos.*, Bacharel em Ciencia da Computacao, Universidade Candido Mendes Monografia, Aug. 2006.

[17] M. R. Junior, R. C. B. P. Gomes, S. F. P. P. Judice, and I. O. Matias, *Simulacao computacional aplicada a jogos eletronicos*, Bacharel em Ciencia da Computacao, Universidade Candido Mendes Monografia, Aug. 2006.

[18] (2012, Fev.) Ogre3d. [Online]. Available: http://www.ogre3d.org

[19] (2012, Fev.) Ode. [Online]. Available: http://www.ode.org

 ISBN: 978-1-61208-202-8