Skin Segmentation Framework for Head Mounted Displays and First Person Game Styles

Heverton de Melo Sarah
Media Lab - UFF

Alexandre Ribeiro Silva Jnior
Instituto Federal do Tringulo Mineiro Campus Uberaba , IFTM

Esteban Clua
Media Lab - UFF

Abstract

A new generation of Head Mounted Display (HMD) devices is becoming a trend in the game industry. Using these devices, the player can drastically increase his immersion and interaction inside the game. For first-person style games, this presence can be improved by capturing the player's real body, including his hands and arms in the virtual environment. In this paper we introduce a framework that, based on real-time image segmentation techniques, suitable for Graphic Processing Unit (GPU) architecture, extracts the skin pixels from real-time video streams attached near the user's eye. The resulting images present good segmentation, but there are some frames that have failures in the segmentation process. These images are available for a game engine, in order to composite with 3D real-time rendered images.

Keywords:: Virtual Presence, Virtual Reality, Self Presence, Skin Segmentation, Head Mounted Display, First Person Game

Author's Contact:

{hmelo,esteban}@ic.uff.br
alexandre@iftm.edu.br

1 Introduction

The Virtual Reality eld has received great attention in the media. Much of this attraction is due to new tech devices with the objective to entertain, such as Oculus Rift, a low-cost virtual reality Head Mounted Display (HMD) for gaming immersion. With this device, the player can feel he is inside the game. If the game is in first-person, the player sees the game as the character of the game sees it. This already permits good immersion into the game, but it would be better if the player, while looking at himself, it could see his real arm, for example, in the virtual world.

The immersion a player can feel when playing is related to the three kinds of presence a player can experience: *spacial presence*, *social presence* and *self presence*. Spacial presence is related to being physically located or to interact in a virtual environment. The social presence is how the user experiences the social interaction in the virtual environment. The feeling of being present inside the virtual world, in a way that the user can see himself in this world, is called self presence [Bracken and Skalski 2010].

According to [Hedberg 2010]], image segmentation is a field of image analysis, where the main idea is to differentiate objects of an image. Therefore, we can detect the objects in an image and use them for whatever we want, like rendering only an object detached of the background.

This work uses image segmentation to capture skin pixels from a webcam to create a framework that extracts the user's arms from the webcam frames. The output of this framework can be rendered in an HMD, creating an increased immersive experience inside a virtual world.

1.1 Motivation and Contribution

This work aims to create a framework for segmenting players' images captured by a camera attached to a virtual reality headset, to create a more immersive experience in a virtual world.

The method for image segmentation has to work in high speed to not decrease the perception of real-time of the player actions inside the game. To solve this problem, this paper extends the functionalities developed in [Lattari et al. 2011], that segment images or video frames with the help of GPUs, to segment also frames from a webcam captured in real time.

The organization of this work is as follows: Section 2 presents related work. Section 3 explains skin image segmentation and presents the segmentation method applied in this work. Section 4 discusses GPU computing and the programing model of computing that was used. Section 5 presents the framework for presence on virtual reality developed. Section 6 presents the preliminary results of this research. Finally, Section 7 presents the conclusions of this work.

2 Related Work

There are two kinds of work that use images of the user's real body to superimpose images into a virtual environment, with the aim being to increase the feeling of immersion, the feeling of being there. The first one, which is more related with this work, has an HMD with one or two cameras connected near the users eyes; the other is a CAVE [Cruz-Neira et al. 1993] environment.

One of the first works with HMD [Metzger 1993] used a helmet system equipped with a micro-miniature camera and CRT or LCD display, and earphones to create a virtual reality system. The objective was to build a vehicle simulator where the driver would wear the HMD and could see the virtual world with objects of the real world, like his arms and hands. Video keying was used to subtract the real image captured by the cameras. To make this, the external environment had to be in the same color, like blue. This characteristic limits the environment to only one color, restricting the scope of the application.

Another application of HMD is in manipulating haptic devices. The work of [Yokokohji et al. 1996]] made a system where What you can see is what you can feel. That is, when the user's hand touches the haptic device, it is, at the same time, touching the virtual object. The image segmentation technique used in this system is also the video keying. The purpose of this work is to segment the image captured by two cameras, at the user's eyes position, to superpose the PUMA robotic device's image that is usually visualized in applications like this. Due to using the video keying segmentation technique, the whole environment has to be in a unique color; in this case, blue.

In this study [Bruder et al. 2009], besides segmenting the user's skin, the whole body is used to represent the user in the virtual world. The process of capturing the images is divided in two phases: a training phase and a classification task. In the first phase, the user is asked to move the hand, in front of the cameras attached to the helmet device, to put the hands in squares that it sees on the screen. This is applied to take the hand's color as input. To segment the body, when the user looks at his body, the system segments the ground (that is in a unique color) and subtracts the ground; the rest is the body. The fact that the system needs to be trained to be available for use is something that makes the user lose time when the application of this system is for entertainment.

In teleoperation manipulation, virtual reality headsets can be used to increase the sensation of being in a remote place. The work by [Saraiji et al. 2013] accomplishes it by superimposing the slave's vision with a 3D model that has textures of the user's skin. These textures are the skin images captured by the TELESAR V [Saraiji et al. 2012]. The fixed length size 3D model represents the user's hands and arms and it was used as a reference for masking. In this system, the user has to wear a jacket for tracking the shoulder movements and gloves for tracking the finger movements. The process of generating the mask was made in GPU and the rendering in a GPU render target. The method applied in this work needs another tracking devices besides the two cameras, and the 3D model needs to be modified if another person wants to use the system, which decreases flexibility.

Another application where the user sees his body in a virtual environment is the so-called CAVES. In [Gross et al. 2003] an immersible environment for virtual collaboration and design is studied. It uses several cameras to make live streams of the user, and three rectangular projection screens of glass panels containing crystal layers. These panels can be switched from an opaque state, which can be used for projection, to a transparent state, allowing cameras to look through the walls. The video streams are used to realize a background subtraction and silhouette extraction, resulting in the computation of the 3D video representation of the user in real time. With these silhouettes, the video representation is taken from visual hulls. This application doesn't need an HMD for rendering the virtual world, just 3D glasses to see the 3D images that are rendered on the walls. The images appear only on the walls; when the user looks down he sees the real ground. This kind of application needs a large amount of apparatus to create the virtual experience.

The [Petit et al. 2009] approach of CAVE gives users the opportunity to see objects in their hands: that is, by occlusion-free co-located interactions. All the processes for segmentation are realized in a PC cluster to make possible the real-time execution. The user has to wear an HMD tracked with an infrared positioning system. This technique also creates textures (from images of the user) for a 3D model. This texture is the result of mixing photometric data from the closest cameras to the user's viewpoint.

The most common approach of representing the human movement in a computer is by motion capture devices that translate movements with sensors attached to the user's body. This technique is applied in [Dobbins et al. 2014] to take the user's movements to move a 3D model that represents this user. This work aims to provide a virtual environment to training for one or more people. It consists of HMDs, a CAVE, spherical cameras, a motion capture system and a 3D laser scanner. The skin segmentation process is not present in this approach; that is, the user sees a 3D model that represents his body, but not with his skin textures, like the other approaches.

Other work [Spanlang et al. 2010] also has motion capture and reproduces a virtual body representing the user, but not his real body. The HMD is applied to see the virtual world. The system has representation of tactile sensation on the user's body when it touches something in the virtual world. This sensation is achieved by vibrators mounted on small boards placed on suit regions. A collision actuator map on GPU is used to map collisions and vibrators.

[Kalra et al. 1998] studied a method to model and animate believable and realistic humans (expressions and animations) in a virtual environment that was applied in a case study where two people from different locations played a tennis game. This application also uses an HMD to render the virtual world, techniques to create the user's virtual body representation in real-time, but not the user's real image, and motion trackers to identify the user's movement.

ARQuake [Thomas et al. 2000] is a study that converted a desktop first-person application into an outdoor/indoor mobile augmented reality application. The player wears an HMD, a gun and plays walking around an area, and what it sees is the area inside the game. This application does not superimpose the user's image onto the game's avatar.

3 Skin image segmentation

According to [Hedberg 2010]], image object differentiation can be done according to pixel intensity, varying between 0 and 255. The subtracting of objects from the background of an image can be made by the following main approaches:

- Threshold-based techniques: The image is compared according to a threshold value. If the pixel intensity differentiates a lot from this value, the pixel does not belong to the image first plane.
- Edge-based techniques: An edge is a set of pixels connected that has a common characteristic, as a connection between pixels with the same level of intensity, and can be differentiated with the intensity gradient.
- Region-based techniques: Consists in sorting an image in regions. At first, the image limits and discontinuities are found. The pixels inside a region are compared with a connectivity model. Then, some kind of region growing is applied, making smaller regions to merge into larger ones.

The image segmentation technique [Lattari et al. 2011] used in this work treats skin colors as a subset of the RGB color spectrum and models the segmentation as minimum cost graph-cut problem.

To approach this, a pixel labeling process is defined where P is a set of pixels and L is a set of labels. This process assigns a label l from the set L to a site $p \in P$. The solution of this problem can be represented as a characteristic function X.

The technique defines each element $p \in P$ a single image pixel. The set L has two elements: O (object region) and B (representing the background).

The characteristic function can be computed from the following probabilistic distribution:

$$X(p) = \begin{cases} 1, p \in O \\ \\ 0, p \in P \end{cases}$$

Where p is 1 when it is from the *Object Set* and it is 0 when it is from the *Background Set*. The characteristic function can be found by minimizing an object function. The most used one is the Gibbs Energy [Greig et al. 1989].

In [Lattari et al. 2011], the elements of the energy function are defined based on the image pixels so that it can be used to specify the costs of every edge on a directed graph. The energy terms gather skin and non-skin information from a database of images. Therefore, it is not necessary to mark the pixels that are background or object in the image that is going to be segmented - the process is automatic.

The process of graph-cut is realized by the Push-Relabel method, which is suitable for parallelization on GPU.

A large amount of works that also use the Gibbs Energy to minimize the Graph Cut is in literature. In [Boykov and Jolly 2001], a technique is developed for general purpose interactive segmentation of N-dimensional images where the user has to mark some pixels that represent object or background as a initial step for the segmentation process. The pixels can be marked after the initial segmentation without the need to process from the beginning.

Because of segmenting a large measure of pixels can be expensive, some works have been trying to minimize this cost. [Boykov and Kolmogorov 2004] developed a comparison of the efficiency of min-cut/max-flow algorithms inside the computer vision scope, and also their own technique, that tries to approach faster processing. Their method worked several times faster than the other ones.

After the popularization of the use of GPU for general purpose, it became a good tool to reach faster image segmentation. Some works [Vineet and Narayanan 2008] started to study the application of GPU, that also implements the Push-Relabel algorithm to graph cuts, as the segmentation method applied in this paper.

4 GPU Computing

A GPU is well designed for applications that have the following characteristics [Owens et al. 2008]:

- Computational requirements are large. Well-used to real-time rendering.
- Paralelism is substantial. Large amount of computational units for processing tasks.

- Throughput is more important than latency. GPU prioritizes throughput over latency.

General-purpose computation can also be made on GPU. The field where applications which are not graphics and are processed in these GPUs is called General-Purpose Computing on GPU (GPGPU).

CUDA$^{\text{TM}}$ [Nvidia 2008] is a general-purpose parallel computing platform and programming model created by NVIDIA.

With CUDA$^{\text{TM}}$, it is possible to create multithreaded programs partitioned into blocks of threads, each thread executing independently from each other. Hence, if a GPU has more multiprocessors than another, the first one will execute programs in less time than the second with fewer multiprocessors [Nvidia 2008]. This programming model was used in this work to manipulate the data that are calculated inside GPU.

5 Framework for presence on Virtual Reality

The framework is an extension of the work by [Lattari et al. 2011], a method to segment skin images or videos. This technique can segment video frames, generating a new video with those segmented images as frames of this one.

In this work, the technique was extended to segment frames of a webcam. The result of this segmentation can be used to superimpose images that are rendered in an HMD like Oculus Rift.

Each frame from the webcam is sent to a method similar to the one that segments video. This is made by OpenCV [Bradski 2000], with a function that controls streams of the main webcam in the system. Inside the method, energy terms are constructed based on the webcam frames; after that, the graph cut operation is executed with Push-Relabel algorithm. The result of this process is frames with only the pixels that have colors similar to the ones that represent human skin in the database, and each frame is rendered in a system window of the computer screen. The energy function and graph-cut calculations are executed using the CUDA$^{\text{TM}}$kernel to perform inside the GPU.

After the segmentation process, the frame can be used to superimpose the images that are rendered in an HMD, creating the illusion for the user of seeing his own body inside the headset, but this part is not finished yet.

Figure 1 illustrates the framework, where the Extention parts in the picture are the contribution of this work to the segmentation technique. The framework consists of an input, the segmentation method and the output. The input is conceived by the old kind of input (an image or a video), that existed in [Lattari et al. 2011] and the new one that was developed in our work (webcam frames) and added in [Lattari et al. 2011]'s technique. Each webcam frame is segmented by the method. The result of the segmentation can be the old one (an image or video) and the new output that was added in this work (image frames), that can be used to render in an HMD or on video screen.

6 Preliminary Results

The CPU used is an Intel®Core$^{\text{TM}}$i7-3820 with 3.60 GHz and the GPU is an NVIDIA Tesla K20. The webcam used to capture the images is a Microsoft®LifeCam HD-5000. The NVIDIA CUDA Toolkit v5.0 was used to develop the CUDA kernels. OpenCV 3.0 was used to get the webcam frames and show the resulting frames on screen.

The results have showed that the frames-per-second (FPS) still have to improve to be applied in virtual reality. Oculus Rift, for example, needs 60 FPS in order to not disturb the sense of immersion and the actual results do not achieve this goal yet. One of the changes that could increase the FPS quality could be rendering the segmented images from the GPU with OpenGL. This way, the image would not need to return to the host.

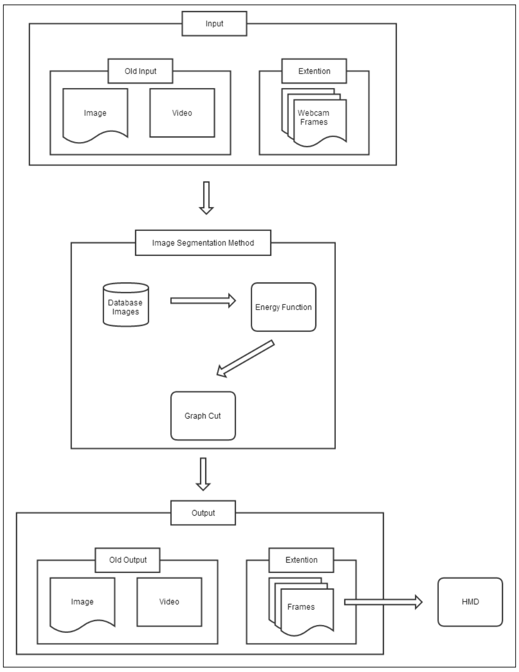

Figure 1: *Framework Diagram*

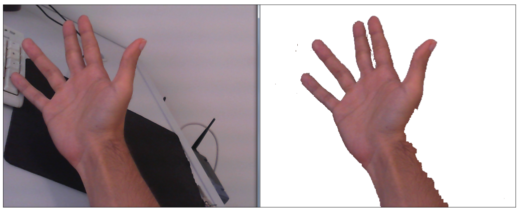

Figure 2: *Good Segmentation*

Another problem that was identified is some frames do not have a good segmentation, presenting failures. Figure 2 represents a good segmentation and Figure 3 represents a frame with failures in the segmentation process. The reason for this could be that the segmentation method used depends on the ambient illumination. So, parts of an image that are in shade, and that are not represented in the database, could be treated as background.

Figure 4 represents the amount of time, in seconds, takes to calculate the energy and graph cut to each frame. The Y axis show the amount of time and the X axis, the frame number.

7 Conclusion

It was possible to finish a framework that extracts skin images captured from a webcam to segment it. Currently, it can be rendered on video screen, but there is the need to improve the segmentation results and the frame rate, in order to achieve better images that can be rendered in an HMD without decreasing immersion.

OpenGL buffers can be used to generate a better sending rate of images to be processed on GPU. The frames can be stored in an

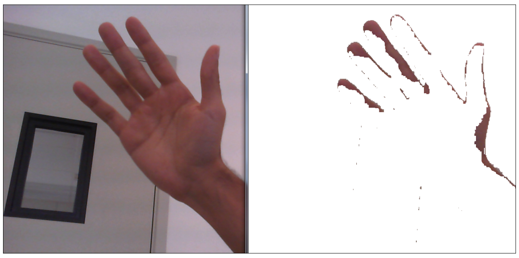

Figure 3: *Segmentation with Failures*

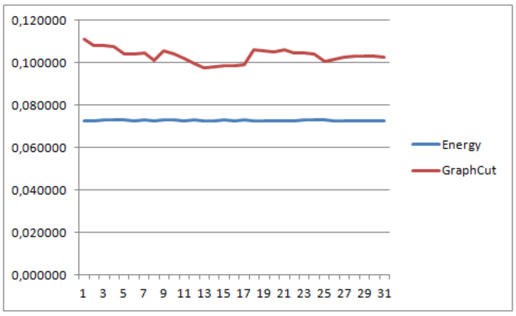

Figure 4: *Energy and Graph-Cut time distribution (in seconds)*

OpenGL buffer. This way, the buffer can be accessed both by GPU or CPU, excluding the need to make copies of the frames.

The final part of the framework, the rendering in an HMD, is still in development. Looking to the future, this final phase must be developed and the input improved using two cameras to capture images near the user's eyes, simulating the player's eyes.

References

BOYKOV, Y., AND JOLLY, M.-P. 2001. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, 105–112 vol.1.

BOYKOV, Y., AND KOLMOGOROV, V. 2004. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 26*, 9 (Sept), 1124–1137.

BRACKEN, C. C., AND SKALSKI, P. 2010. *Immersed in Media: Telepresence in Everyday Life*. Routledge, July.

BRADSKI, G. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

BRUDER, G., STEINICKE, F., ROTHAUS, K., AND HINRICHS, K. 2009. Enhancing presence in head-mounted display environments by visual body feedback using head-mounted cameras. In *International Conference on CyberWorlds, 2009. CW '09*, 43–50.

CRUZ-NEIRA, C., SANDIN, D. J., AND DEFANTI, T. A. 1993. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '93, 135–142.

DOBBINS, M. K., RONDOT, P., SHONE, E. D., YOKELL, M. R., ABSHIRE, K. J., SR, A. R. H., LOVELL, S., AND BARRON, M. K., 2014. Portable immersive environment using motion capture and head mounted display, Jan. U.S. Classification 345/633; International Classification G09G5/00; Cooperative Classification G06F3/011.

GREIG, D., PORTEUS, B., AND SEHEULT, H. 1989. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B 51*, 2, 271–279.

GROSS, M., WRMLIN, S., NAEF, M., LAMBORAY, E., SPAGNO, C., KUNZ, A., KOLLER-MEIER, E., SVOBODA, T., VAN GOOL, L., LANG, S., STREHLKE, K., MOERE, A. V., AND STAADT, O. 2003. Blue-c: A spatially immersive display and 3D video portal for telepresence. In *ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, SIGGRAPH '03, 819827.

HEDBERG, H., 2010. A survey of various image segmentation techniques.

KALRA, P., MAGNENAT-THALMANN, N., MOCCOZET, L., SANNIER, G., AUBEL, A., AND THALMANN, D. 1998. Real-time animation of realistic virtual humans. *Computer Graphics and Applications, IEEE 18*, 5 (Sep), 42–56.

LATTARI, L., MONTENEGRO, A., CONCI, A., CLUA, E., MOTA, V., VIEIRA, M. B., AND LIZARRAGA, G. 2011. Using graph cuts in GPUs for color based human skin segmentation. *Integr. Comput.-Aided Eng. 18*, 1 (Jan.), 4159.

METZGER, P. 1993. Adding reality to the virtual. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, 7–13.

NVIDIA, C., 2008. Programming guide.

OWENS, J., HOUSTON, M., LUEBKE, D., GREEN, S., STONE, J., AND PHILLIPS, J. 2008. Gpu computing. *Proceedings of the IEEE 96*, 5 (May), 879–899.

PETIT, B., LESAGE, J.-D., BOYER, E., AND RAFFIN, B. 2009. Virtualization gate. In *ACM SIGGRAPH 2009 Emerging Technologies*, ACM, New York, NY, USA, SIGGRAPH '09, 26:126:1.

SARAIJI, M., FERNANDO, C., FURUKAWA, M., MINAMIZAWA, K., AND TACHI, S. 2012. Virtual telesar - designing and implementation of a modular based immersive virtual telexistence platform. In *2012 IEEE/SICE International Symposium on System Integration (SII)*, 595–598.

SARAIJI, M., FERNANDO, C., FURUKAWA, M., MINARNIZAWA, K., AND TACHI, S. 2013. Real-time egocentric superimposition of operator's own body on telexistence avatar in virtual environment. In *2013 23rd International Conference on Artificial Reality and Telexistence (ICAT)*, 35–39.

SPANLANG, B., NORMAND, J.-M., GIANNOPOULOS, E., AND SLATER, M. 2010. A first person avatar system with haptic feedback. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, ACM, New York, NY, USA, VRST '10, 4750.

THOMAS, B., CLOSE, B., DONOGHUE, J., SQUIRES, J., DE BONDI, P., MORRIS, M., AND PIEKARSKI, W. 2000. Arquake: an outdoor/indoor augmented reality first person application. In *Wearable Computers, The Fourth International Symposium on*, 139–146.

VINEET, V., AND NARAYANAN, P. J. 2008. Cuda cuts: Fast graph cuts on the gpu. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, 1–8.

YOKOKOHJI, Y., HOLLIS, R., AND KANADE, T. 1996. What you can see is what you can feel-development of a visual/haptic interface to virtual environment. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, 46–53, 265.