Computer Vision based Machine Interaction

Carlos H. Morimoto, Flavio L. Coutinho, Jefferson R. da Silva, Silvia E. Ghirotti, and Thiago T. Santos
Computer Science Department
University of São Paulo
http://latin.ime.usp.br

***Abstract*—This paper introduces the Laboratory of Technologies for Interaction (LaTIn) and briefly describes its current main projects. The main focus of LaTIn has been developing new ways of human-machine interaction using computer vision techniques. The projects are categorized according to the distance between the human user and the machine being operated. For close distances, appropriate for interaction with desktop computers for example, we have developed eye-gaze based interfaces. For mid range distances, we have built hand and body gestures interfaces that are appropriate for virtual and augmented reality settings and, for large distances, we have developed novel multiple people tracking techniques that have been used for surveillance and monitoring applications.**

***Index Terms*—Computer Vision, Human Computer Interaction, Augmented Reality.**

I. Introduction

THE Laboratory of Technologies for Interaction (LaTIn) was created in 1999 in the Computer Science Department of the Institute of Mathematics and Statistics of the University of São Paulo, Brazil, to research non conventional interfaces based on computer vision techniques. Non conventional interfaces are not restricted to traditional graphical user interfaces, and can rely on other modes of interaction such as eye gaze and body gestures.

From 2000 to 2005, LaTIn was funded by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) and IBM. Its primary research focus was to develop eye-aware interfaces and eye gaze enhanced applications.

Because cameras and computers suited for real-time image processing were very expensive in the late 90's (they have dropped considerably since then), one of the main goals of our project was to develop new interactive technologies based on low-cost off-the-shelf components. For example, the cost of commercial eye-tracking systems were above US$20,000.00 (and still are in 2011). An eye tracking system, or simply eye-tracker, is a device that allows the computer to estimate the point-of-regard of the user on the computer screen [1].

In [2] we describe a low cost eye tracking system using components under US$200.00. Besides improving the state of the art of eye-tracking technology, our group has also contributed to the development of gaze enhanced applications, such as MAGIC Pointing [3] and Context Switching [4]. Section II describes the main current projects related to eye-tracking.

To track the eye, the camera has to be placed close to the user and a long and narrow field-of-view lens must be used to capture good eye images. Using shorter focal lenses allows the camera to capture images with a wider field-of-view. But independently from the lens used, we can categorize the research projects being conducted at LaTIn according to the distance from the camera to the object of interest. Within short distances, we can use eyes and faces for interaction. For mid-range distance, we can use the whole body and, for long distances, more than one person can interact with the system. Sections II, III, and IV describe current LaTIn projects under these scenarios and Section V concludes the paper.

II. Short distance

A remote eye gaze tracker, or simply a remote eye-tracker (RET), is a device that allows the computer to estimate the location at the monitor where the user is looking at [5]. Most RET use external near infra red (NIR) to enhance the image quality and facilitate eye tracking. NIR is invisible to the human eye, and therefore does not distract the user. Commercial RET are very expensive devices that have about 1^o accuracy (most claim 0.5^o in 2011).

We have developed, in collaboration with IBM, a low cost remote eye gaze tracker described in [2] that can be used for gaze based interaction within short distances. Though some assembly is required, our RET use off-the-shelf components that cost less than US$200.00.

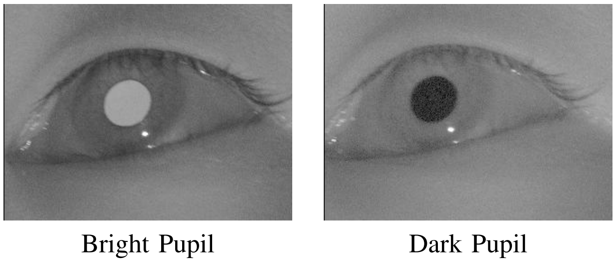


Fig. 1. Bright and dark pupil images generated by the dual light source eye tracker.

Our RET uses two light sources, synchronized with the camera interlaced fields. An on-axis light source is placed around the optical axis of the eye and produces a bright pupil image, as shown in Fig. 1. An off-axis light source, placed farther away from the optical axis, produces a dark pupil image. This bright/dark pupil effect is well known as the red eye seen in flash photographs. By exploring this phenomena, the system can detect the pupil by computing the difference between the bright and dark pupil images.

Observe in Fig. 1 that the corneal reflection (glint) can be easily detected as well. If we assume that the eye is

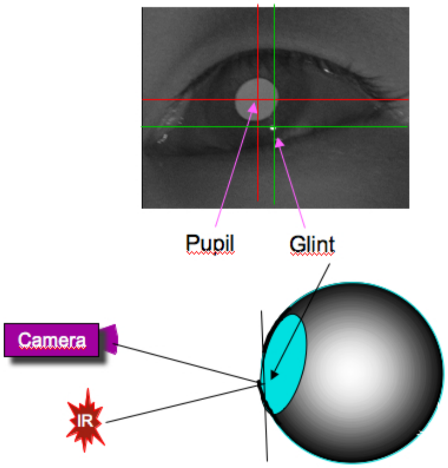


Fig. 2. Pupil-glint eye tracker.

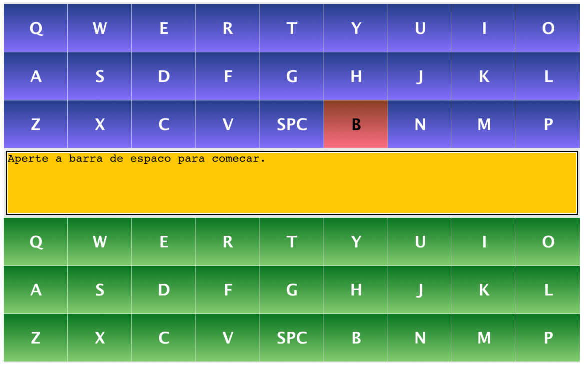
spherical, then the pupil-glint vector can be used to map the eye orientation to computer screen positions using a calibration procedure, as seen in Fig. 2. During calibration, the user looks at a sequence of control points shown on the screen, and presses a key. A typical calibration function uses nine screen points and pupil-glint vector correspondences to compute a second order polinomial interpolation [6].

After calibration, the computer is able to estimate the gaze position as screen positions. Though reasonably accurate (approximately 1^o) the calibration is not able to compensate for large head motion (this is also true for the very expensive commercial systems).

To improve the usability of the system by reducing the need for frequent calibration and allow wider head motion, we have also proposed the use of a spherical mirror model in [7], [8] to compute the gaze direction. We have also proposed the use of more complex geometric models in [9], [10]. Though these methods are more complex to build, they are more accurate and robust to head motion.

Eye gaze trackers are fundamental to eye-movement research and are being used for usability tests and in computer interfaces for people with motor disabilities. Because we use our eyes to capture information of the world around us, we believe that overloading the eye to control computer applications is not appropriate. Therefore, instead of using the eye as another pointing device, we have proposed the use of eye-trackers as a complementary information channel or to rely on natural eye movements. Examples of such applications are the MAGIC Pointing [3] and the Context Switching paradigm for gaze based interaction [4].

The idea behind the MAGIC (Manual And Gaze Input Cascaded) Pointing [3] is to combine the speed of eye movements with the accuracy of the hand. If you have used a mouse before, a considerable time for pointing the cursor might be wasted first locating the cursor and then dragging the mouse to the desired position. MAGIC Pointing instantly warps the cursor to the position where the user is looking at when she touches the mouse. Therefore, it instantly "locates" the cursor. Because the user in general looks at the desired target before touching the mouse, the cursor "magically" appears near the target, within the 1^o resolution of the RET. Therefore, the user only has to drag the mouse for a short distance.



Q W E R T Y U I O
A S D F G H J K L
Z X C V SPC B N M P

Aperte a barra de espaco para comecar.

Q W E R T Y U I O
A S D F G H J K L
Z X C V SPC B N M P

Fig. 3. Virtual keyboard using the context switching paradigm [4].

The Context Switching Gaze based interaction paradigm proposes a more natural mapping of eye movements to basic interface actions. For example for eye typing, a virtual keyboard is presented to the user. The keys can be pointed by gazing at the desired key, but most selection mechanisms suggested in the literature [11] present some drawbacks, such as the Midas touch (i.e., all keys are typed by just looking at them), or the user can never rest her/his eyes.

Natural eye-movements can be classified into saccades (rapid eye movements) and fixations. Most eye-typing solutions only use fixations. For example, one can use dwell-time to control the typing speed, i.e., if the user fixates her/his eyes for longer than the dwell time, the key is selected. If the user does not want to type the gazed key, s/he must change his/her gaze before the dwell time.

Context switching is a new paradigm that we have suggested in [4]. The idea is to use fixations for pointing and saccades for selection. Fig. 3 shows how the paradigm can be applied to eye-typing. Note that there are actually two keyboards, a top and a bottom one. Each keyboard defines a context. The user can freely explore each context, without being concerned about the Midas touch problem. To select one key, the user just have to look at the desired key and "switch" to the other context. When the system detects that the context have changed, the last observed key in the previous context is typed on the text area placed between the two contexts.

We are current extending the paradigm to be applied beyond simple typing applications.

III. Mid-range distance

Before gesture interfaces became popular in game platforms such as the Nintendo Wii and Microsoft Xbox, our group investigated computer vision solutions to gesture based interaction for virtual and augmented reality environments.

In [12] we use a single camera to track the hands relative to the head in 2D. The hands and head are first detected using a skin color segmentation algorithm, and tracked using Kalman filters. A finite state machine was used to model gestures to point, select and drag widgets in a graphical interface. Other modes of operation were developed to manipulate 3D objects and for navigation in 3D environments.

A more complex system was described in [13]. Using a pair of stereo cameras, the hands and head of multiple people were tracked in 3D. Real-time performance was achieved by segmenting and tracking only skin color blobs in 3D. 3D tracking allows for more natural gestures for navigation and manipulation of virtual 3D objects.

IV. Long distance

Many video applications, such as TV broadcasting, surveillance, and monitoring, use several sparse cameras to capture the scene from different angles. In [14], [15] we describe a system for 3D free view video interaction that allows the observation of people acting within the 3D volume captured by the sparse cameras.

First multiple people are detected by combining the information from background subtraction of the multiple sparse cameras. Assuming that every person seen by the cameras is walking on a flat surface (the floor plane), the video from each camera can be projected onto the floor plane using a homography. The homography constraint basically states that, because a person does not belong to the floor plane, the projection of the image of a person on the floor plane will coincide for every camera only where the person has contact with the floor, that is, only at their feet positions. Fig. 4 shows the results of background subtraction over the original images of 3 cameras. We have developed a technique that accumulates the evidence from the multiple cameras to detect each person [16], where local maxima correspond to the location of each person, as seen in Fig. 5. Once each person is detected, they are tracked using an appearance model.

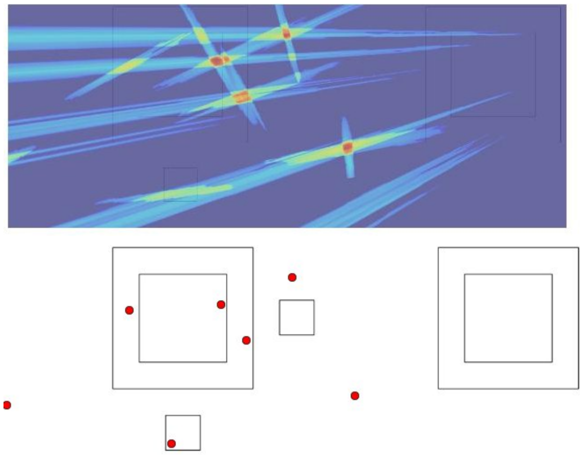


Fig. 5. Detection results for multiple people detection using the homography constraint. The top image show the accumulation result, and the bottom image shows the location of the detected people from the computation of local maxima [16].

A. Free View Video

When several cameras are used to monitor a scene, such as in surveillance applications and the broadcasting of sports events, it is possible to combine the visual information provided from the several cameras to estimate a model of the scene. Once this model is provided, one can generate a virtual view of the scene (one that does not correspond to any of the real cameras). As this can be computed for every video frame, we can generate a virtual video stream, called free view video.

To generate a free view video stream, we will assume that a previous planar patch scene model has been previously built, as seen in Fig. 6, and that the people segmentation and tracking results are also available. We also assume that the calibration information from a camera is available and can be used to project the video on the scene model.

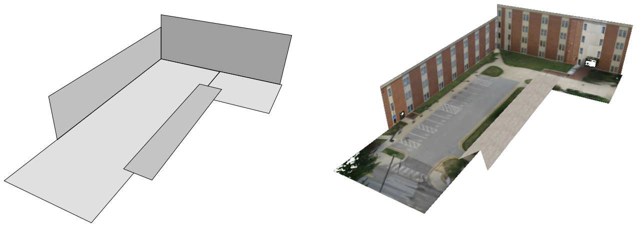


Fig. 6. Example of a planar patch scene model [15].

Once segmentation has been computed, free view video can be rendered in real time. The steps of the free view video generation can be seen in Fig. 7. From top left to bottom right, the first image shows the model as seen by the virtual camera after the user specifies a position and orientation.

The second image shows the video from the camera closest to the user selected view. Observe, as expected, that objects not belonging to the floor plane are not correctly projected by the homography between the camera image and the floor plane.

To avoid this distortion, moving objects are removed from the virtual video stream by subtracting the foreground detected from background segmentation, as shown in the third image of Fig. 7. Observe that, because the cameras have different color properties due to hardware and illumination conditions, the color information from the cameras are different between each pair of cameras and between each camera and the scene model.

Non-linear optimization was used to calibrate the color of each video to the model [15]. A color correction table for each camera is pre-computed and used to render a color correct image as seen in the fourth image of Fig. 7.

Finally, the last two images of Fig. 7 show virtual views with superposed perspective corrected foreground. Because we know the location of each person from tracking, a planar billboard oriented towards the virtual camera is placed in the correct location, and the foreground texture removed from the original images are projected onto the billboards, given the user better 3D experience.

V. Conclusion

This paper describes the Laboratory of Technologies for Interaction (LaTIn), established in 1999 in the Department of Computer Science of the Institute of Mathematics and Statistics of the University of São Paulo. LaTIn was created to research novel interaction technologies, particularly those using computer vision techniques.

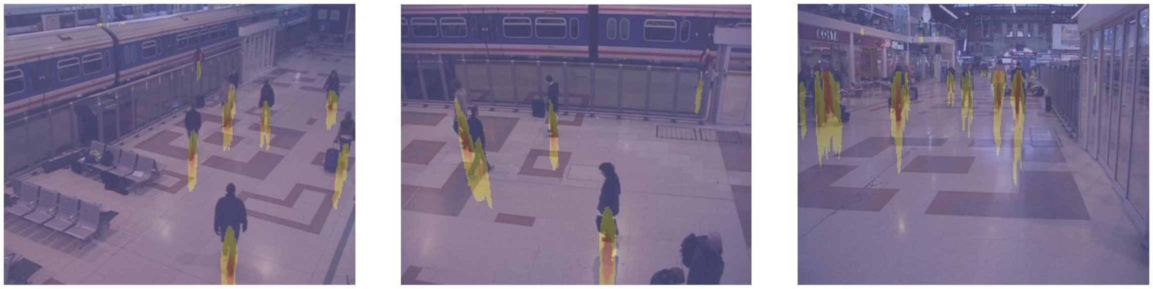


Fig. 4. Results of background subtraction superposed on top of the original images of 3 cameras [16].

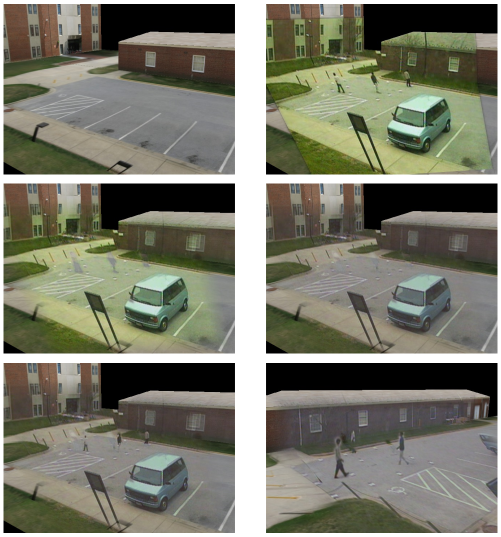


Fig. 7. Steps for generating free view video from sparse cameras [15].

The history of LaTIn has been associated with the development of state of the art remote eye trackers (RET) and gaze-aware applications. Recent contributions include the use of novel geometric models to improve the robustness of RET to head motion, and a novel gaze interaction paradigm called context switching that is more comfortable and efficient than current eye interaction paradigms.

Some current projects include the use of computer vision techniques to smart environments. We have developed a sparse multi camera people detection and tracking using the homography constraint. The main advantage of the method is the robustness to occlusion. We have used this method to create a free-view-video system, that allows the user to visualize events in 3D from a set of sparse cameras.

We invite interested students and researchers to find more information about our lab at http://latin.ime.usp.br, and contact us using the information available at the LaTIn's home page.

ACKNOWLEDGEMENTS

The authors would like to thank IBM, FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) for their financial support to projects described in this paper.

REFERENCES

[1] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, March 2010.

[2] C. Morimoto, D. Koons, A. Amir, and M. Flickner, "Pupil detection and tracking using multiple light sources," *Image and Vision Computing*, vol. 18, no. 4, pp. 331–336, March 2000.

[3] S. Zhai, C. Morimoto, and S. Ihde, "Manual and gaze input cascaded (magic) pointing," in *Proc. ACM SIGCHI - Human Factors in Computing Systems Conference*, Pittsburgh, PA, May 1999, pp. 246–253.

[4] C. Morimoto and A. Amir, "Context switching for fast key selection in text entry applications," in *Proceedings of the ACM Symposium on Eye Tracking Research & Applications - ETRA 2010*, Austin, TX, 2010, pp. 271–274.

[5] A. T. Duchowski, "A breadth-first survey of eye-tracking applications." *Behavior research methods instruments computers a journal of the Psychonomic Society Inc*, vol. 34, no. 4, pp. 455–470, 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12564550

[6] C. Morimoto and M. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[7] C. Morimoto, A. Amir, M. Flickner, and S. Zhai, "Detecting eye position and gaze from a single camera and 2 light sources," in *Proc. of the ICPR 2002: 16th International Conference on Pattern Recognition*, vol. VI, Quebec, CA, August 2002, pp. 314–317.

[8] C. Morimoto, A.Amir, and M. Flickner, "Free head motion eye gaze tracking without calibration," in *Proc. of CHI 2002 Extended Abstracts on Human Factors in Computing Systems*, Minneapolis, Minnesota, 2002, pp. 586–587.

[9] F. Coutinho and C. H. Morimoto, "Free head motion eye gaze tracking using a single camera and multiple light sources," in *In: Proc. of the Sibgrapi 2006 - Simpsio Brasileiro de Computao Grfica e Processamento de Imagens*, Manaus, AM, Brazil, 2006.

[10] ——, "A depth compensation method for cross-ratio based eye tracking," in *Proceedings of the ACM Symposium on Eye Tracking Research & Applications - ETRA 2010*, Austin, TX, 2010, pp. 137–140.

[11] P. Majaranta, "Text entry by eye gaze," Ph.D. dissertation, University of Tampere, Department of Computer Science, Tampere, Finland, August 2009.

[12] M. Cabral, C. Morimoto, and M. Zuffo, "On the usability of gesture interfaces in virtual reality environments," in *CLIHC'05, Conferencia Latinoamericana de Interacion Humano-Computadora*, Cuernava, Mxico, October 2005.

[13] S. Ghirotti and C. H. Morimoto, "Um sistema de interação baseado em gestos manuais tridimensionais para ambientes virtuais," in *Anais do 9o Simpósio de Fatores Humanos em Sistemas Computacionais - IHC 2010*, Belo Horizonte, MG, 2010.

[14] J. R. da Silva, T. T. Santos, and C. H. Morimoto, "Real time novel view scene rendering from multiple sparse videos," in *Anais do XII Simpsio de Realidade Virtual - SVR 2010*. Natal, RN: SBC, 2010, pp. 184–193.

[15] ——, "Automatic camera control in virtual environments augmented using multiple sparse videos." *Computers & Graphics*, vol. 35, no. 2, pp. 412–421, 2011.

[16] T. Thiago T. Santos and C. Morimoto, "Multiple camera people detection and tracking using support integration," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 47–55, 2011.