Camera Based Face Tracking for Enhancing Surgical Teamwork Training with Non-Verbal Communication

S. Marks[1], J. Windsor[2], B. Wünsche[1]

[1]Department of Computer Science, The University of Auckland, New Zealand.
[2]Department of Surgery, The University of Auckland, New Zealand.
Email: smar189@aucklanduni.ac.nz

Abstract

In recent years, the increased use of simulation for the training of surgical skills has improved the medical curriculum and the overall patient outcome.

Advances in hardware and simulation techniques have resulted in many commercial applications for training technical skills. However, most of these simulators are extremely expensive and do not consider non-technical skills like teamwork and communication. This is a major drawback since recent research suggests that a a large percentage of mistakes in clinical settings are due to communication problems. In addition training teamwork can also improve the efficiency of a surgical team and as such reduce costs and workload.

We present an inexpensive camera-based system for the acquisition of aspects of non-verbal communication of users participating in virtual environment-based teamwork simulations. This data can be used for the enhancement of virtual-environment-based simulations to increase the realism and effectiveness of team communication.

Keywords: non-verbal communication, face tracking, virtual environment

1 Introduction

In 1999, the Committee on Quality of Health Care in America released a report estimating that each year, 98,000 people die because of medical errors occurring in hospitals [1]. This report caused massive changes in the medical education system, including the increased use of medical simulation for training and assessment. As a result, the market now offers a variety of simulators, ranging from inexpensive bench models made of rubber for training suturing and other basic technical skills to expensive mannequins that can be used for the training of whole emergency response teams.

With increasing realism through rising computational and graphical power, virtual environment (VE) based tools have also secured their position among medical simulators [2]. Commercially available products offer training opportunities in areas like endoscopic procedures [3], obstetrics, cardiology or other related fields.

But these simulators only serve the purpose of training a single student. They do not take into account the fact that surgery and other medical procedures are always performed as a team and thus, among the technical skills, require non-technical skills like communication, teamwork, leadership, and decision making [4]. Research indicates that failure in communication occurs in 30% of team exchanges and that one third of these failures results in situations putting the patient's life at risk [5]. To increase patient safety, training and improvement of communication among team members has to be an important aspect of clinical simulation.

2 Related Work

Few tools exist for addressing non-technical skills. 3DiTeams [7] (see figure 1), for example, simulates a military operating room, including all necessary devices, medical instruments, and the patient. All members of the team are represented in the VE by avatars and can interact with each other, the devices and the patient. Each user operates at an individual client computer that is connected to a central server by network. This server receives all actions and movements of the users from the clients, synchronises the state of every user on every client and runs the physical simulation of

978-1-4244-2582-2/08/$25.00

objects and medical simulation of the patient's physiology. Verbal communication is possible by microphones and headsets or speakers.

The simulator is built using a game engine which provides a well-tested, stable, and high-performance software framework for realistic graphics, animation, sound, physics, and networking support for multi-user scenarios [8].

Nevertheless, communication and interaction within VEs is inefficient and unnatural if non-verbal communication aspects are not taken into account [9, 10]. Communication without gaze direction and head orientation suffers from a decrease of 50% in deictic references to persons, like "you" or "him/her" [11]. This reduces the efficiency of communication because the lack of non-verbal communication channels has to be compensated by other channels, for example by replacing deictic references to objects and persons by explicitly saying their names and positions (see figure 2).

By introducing head orientation and gaze direction, users can simply look at objects or other users instead of referring to them by voice (see figure 3).

Other capabilities of modern game engines allow for even more non-verbal communication channels that can be used to increase the bandwidth. Detailed animation systems in the avatar's face models enable the rendering of finely controlled facial expressions like joy, anger, fear (see Figure 4). Partial animations that can be applied independently could be used for the control of hand or body gestures like pointing, turning, or shrugging shoulders.

Instead of using explicit user interface controls for changing facial expression, triggering gestures, or controlling gaze direction [12], we propose the use of a camera to extract the necessary data in real-time. This has several advantages. Manual control of the aspects of non-verbal communication

Figure 1: 3DiTeams – A Virtual Environment Simulation for the training of teamwork and cooperation in a military emergency room (Source: [6])

Figure 2: Without gaze direction and head orientation, communication has to use unnatural verbal reference to objects or people.

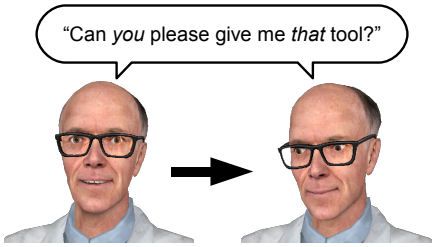

Figure 3: With gaze direction and head orientation, the verbal reference to objects and people is not necessary any more.

Figure 4: Changing an avatar's facial expression to represent emotions like joy, anger, fear.

is only possible if the user is aware of them, which is not necessarily the case. The camera requires no control from the user and can capture conscious and unconscious elements. In addition, the user can completely concentrate on the simulation content instead of getting distracted by the additional controls. A second advantage is the temporal immediacy of the captured data. Momentarily raised eyebrows during an emphasis in a spoken sentence can be perceived by all others users at the same time. If the optical clue would follow the verbal clue with a delay, for example, when using manual control, it would degrade or even counteract the purpose of the gesture.

3 Methodology

The overall design of our system is depicted in figure 5. A webcam, mounted close to the monitor, captures a video stream of the user participating in the simulation. A user monitor application detects the user's face in the video and calculates parameters like head orientation, gaze direction, and facial expression. Via an extension, called plug-in, the application provides this data to the simulation client, in this case a game engine. The client sends

this information to the simulation server together with other data, for example, mouse movement, or pressed keys. The server receives the data from all clients, applies it to the simulation and in turn synchronises all clients with the updated information, including the aspects of non-verbal communication. The clients receive these updates and display it in form of changed positions of objects, a changed patient state, or changed gaze directions and facial expressions of all avatars.

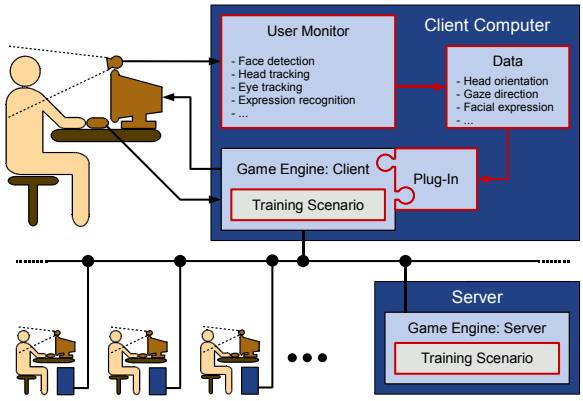

Figure 5: The functional blocks of our framework. Red borders indicate the parts that have been developed by us.

3.1 DataModel

Based on [13], we have derived a list of aspects of communication and interaction in real and virtual environments. Table 1 lists the components of this model, their occurrence in reality, and the technical feasibility in VEs, given the current state of computer technology. Not all aspects of the real world can be implemented in the virtual world, for example, olfactics. Also, not all aspects are equally important and we have to weigh the benefit of adding an aspect against the technological challenge in acquiring the data.

Depending on the choice of the simulation platform we are connecting our user monitor application to, several aspects of the data model are already implemented. A modern game engine usually enables verbal communication by support of headsets and microphones, spatial behaviour by the movement and orientation of avatars, physical appearance by a selection of avatars that can be adjusted to the user's needs, and environment by physically simulated objects that can be grabbed or moved. Data input for these aspects comes from the keyboard or mouse movement, or configuration menus, and does not have to be provided by our user monitor application.

Our framework adds occulesics, kinesics, and facial expression by defining an interface between the application and the simulation that exchanges data about the gaze direction, the head orientation and the facial expression of the user.

3.2 Image Acquisition and Processing

Our system utilises an inexpensive camera, for example a webcam, to capture the user's face. The video stream is handled by the user monitor application that allows modular processing of each frame. The modularity enables us to plug in a variety of pre-processing filters to measure their influence on the performance of different feature tracking algorithms. So far, we have implemented the following modules:

3.2.1 Colour space conversion

According to [14], normalised colour spaces allow the use of simpler models for the distinction between skin colour and non-skin colour pixels for face detection. It also reduces the computational complexity of other modules, for example the lighting compensation. The implemented module converts each pixel colour value from RGB (R, G, B) to normalised RGB space (r, g, b) by applying

$$\begin{aligned} Sum &= R + G + B \\ r = \frac{R}{Sum},\ g &= \frac{G}{Sum},\ b = \frac{B}{Sum}. \end{aligned} \tag{1}$$

3.2.2 Lighting correction

To compensate the colour changes caused by different lighting, we apply an improved Grey World algorithm, described in [15].

Working in normalised RGB colour space, an adaptive mean grey value C_{std} of all m pixels is calculated as

$$\begin{aligned} C_{std} &= \frac{\sum_{1}^{m}(max(r,g,b) + min(r,g,b))}{2n} \\ n &= m - \sum_{1}^{m}(r = g = b = 0) \end{aligned} \tag{2}$$

with m being the total number of pixels and n being the total number of non-black pixels to avoid overcompensation of dark images. The scaling factors S_c for each colour channel $c \in$ (R,G,B) are calculated as

$$\begin{aligned} r_{avg} &= \frac{\sum_{1}^{m} r}{n}, \quad g_{avg} = \frac{\sum_{1}^{m} g}{n}, \quad b_{avg} = \frac{\sum_{1}^{m} b}{n} \\ S_R &= \frac{C_{std}}{r_{avg}}, \quad S_G = \frac{C_{std}}{g_{avg}}, \quad S_B = \frac{C_{std}}{b_{avg}}. \end{aligned} \tag{3}$$

Component	Reality	VE
Occulesics		
- gaze direction, duration, focus	+	+
Language		
- text based chat	-	+
- sound effects	-	+
- speech	+	+
- paralanguage (e.g., voice, tone, pitch, speed)	+	+
Facial expression		
- Facial Action Coding System	+	+
- emotion (e.g., anger, disgust, fear, happiness, sadness, surprise)	+	+
Spatial Behaviour		
- orientation, proximity, distance, position	+	+
Kinesics		
- body movement, posture, head movement, gestures	+	+
Physical appearance		
- face and skin, hair, physique, clothes, adornment, equipment	+	+
Physical contact/Haptics	+	-
Olfactics (scent, odour)	+	-
Environment		
- artefacts (e.g., using, modifying, exchanging)	+	+

Table 1: Aspects of communication and interaction and their appearance in reality ("+": used naturally, "-": could be used, but unnatural) and technical feasibility in virtual environments ("+": available using standard equipment, "-": requires special hardware and software).

3.2.3 Fast normalised cross correlation

For the detection of features in the video stream, we implemented a fast normalised cross correlation module [16]. A pre-calculated sum table of the image M, called integral image, accelerates the process of finding a template image N. In combination with the reduction of the template image N to a sum of k rectangular basis functions, the computational complexity can be reduced from $O(M_x \cdot M_y \cdot N_x \cdot N_y)$ to $O(M_x \cdot M_y \cdot k)$. Further reduction of the computation time is achieved by reducing the Region of Interest (ROI) for each template (see the coloured regions in figure 7).

During the initial calibration process, we are creating a snapshot of the user's face and define rectangular regions of the eyes, the nose tip and the mouth as templates for the feature detection module.

We will further investigate the possibility of removing the need for a calibration process by applying advanced face detection methods as described in [17].

3.2.4 Extraction of Non-Verbal Communication Parameters

Using the input of the cross correlation module, we are able to calculate a rough estimate of the spatial orientation of the head. This estimate can be refined by applying advanced mathematical models like Kalman filtering, applying constraints that take into account the symmetry of a face, and additionally tracked features.

If the head orientation is known, the captured image of the face can be normalised and gaze direction and facial expression features can be extracted [18, 19].

The game engine we connected the user monitor application to (see section 3.3) supports facial animation in high detail. Based on the FACS (Facial Action Coding System) [20], a range of controllers enables fine grained control of the face of the avatar. But not all simulation engines might support facial animation in such detail. In that case, it might be necessary to interpret the acquired data and to create an abstract representation of the facial expression, like, for example, "happy", "angry". Our data model takes this interpretation into account.

3.3 Interfacing with the Simulator Client

When the user monitor application has processed a camera frame and calculated a set of data, it passes this set on to the simulation client, using a plug-in. The plug-in is an addition to the simulation engine that translates the data model into a specific representation that considers the graphical and other capabilities of the engine.

We connected the user monitor application to the Source Engine [21], a modern, versatile, commercial game engine with advanced graphical features, allowing us to make full use of the potential of our data model.

Figure 6 shows the data flow from the user monitor application to each client connected to the simulation. The client receives the data and sends it, together with the input from the mouse and the keyboard, in form of a compressed "User Input" data packet to the server. Each server entity of a user's avatar gathers the input from its specific client entity. Then the simulation of the movement of the avatars, the interaction with the simulated physical environment and other simulation modules, for example for the patients biological parameters, are executing one simulation time step. Afterwards, networked variables automatically synchronise the state of the simulation on each client. The clients then display the updated state of the VE.

The implementation of the plug-in does not require extensive changes to the original source code of the game engine. Especially the support for networked variables simplifies the process of adding information to a simulation object or a player's avatar.

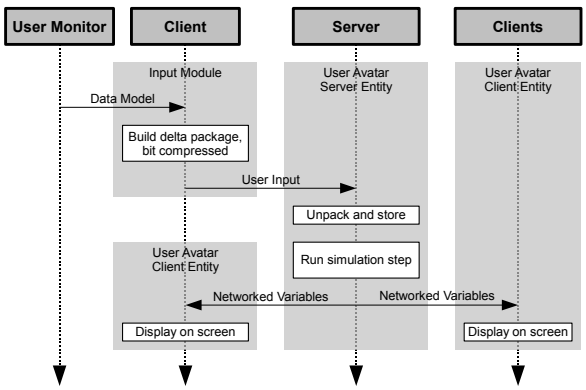

Figure 6: Data exchange between clients and the server of the simulation based on the Source Engine.

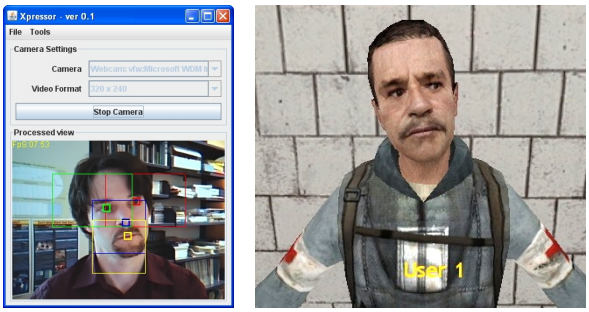

Figure 7: Screenshot of the video capture of the user participating in the VE simulation. The head tilt and eye gaze are transferred onto the avatar representing the user in the simulation.

4 Results

Figure 7 depicts our user monitor application connected to a simple test simulation. In the left image, the large coloured rectangles in the camera picture show the ROIs of the feature extractor, and the small squares symbolise detected features, like eyes, nose, and mouth. The information about head tilt and gaze direction is transferred to the simulation. The right screenshot shows how other users participating in the simulation can see the head tilt and gaze direction projected on the avatar.

Our data model is universal and independent of the VE it is connected to. If support for detailed facial expression is provided, for example, if the VE implements FACS or the MPEG-4 standard, the corresponding parameters of the data model can be used with only minor adaptations. This is the case for the majority of modern game engines, for example, the Source Engine [21] or the Unreal Engine 3 [22]. If the support is not as fine, the *interpreted* data model parameters (see section 3.2.4) can be used for an alternative way of displaying the emotional state of the user, for example by changing the texture of the avatar's face. This flexibility of our data model enables the connection to a wide range of VEs and game engines.

An example for limited support of facial expressions is Second Life [23]. This VE has gained much popularity in the last years and is also increasingly used for teamwork training [24, p. 96]. Second Life does not provide such a fine grained control over the avatar's facial animation and is limited to displaying pre-defined animations. In this case, the interpreted parameters of our data model can be used to display basic emotions in form of short animations.

5 Conclusion

We have presented a framework that allows the enrichment of VE based teamwork simulations by non-verbal communication. The necessary data is captured in real-time by an inexpensive webcam. Our framework is flexible, extendible and independent of the used simulation engine.

We have received positive feedback from medical professionals and developers of teamwork simulations in Second Life about the use and the potential of our application and will perform a more detailed user study in the future.

6 Future Work

In cooperation with researchers and educators from the medical school, we are going to design surgical training scenarios to be used with our application. Furthermore, our user monitor application and the data model is suitable for teamwork training applications beyond the field of medicine and surgery. For this reason, we are also in cooperation with developers for emergency response teamwork training in Second Life, giving us the opportunity to collect valuable information with their simulation scenarios.

Additionally, the stability of the facial recognition and feature tracking algorithms will be subject to further investigation. Several facial recognition algorithms require an extensive training phase that we would like to eliminate or at least hide as much as possible from the end user. Also, we will examine how to overcome difficulties in the image processing caused by non-ideal lighting, users wearing glasses, or other circumstances.

Another goal is the integration of our application with the Unreal Engine that is increasingly used for simulations [6, 25]. The recent version 3 of this engine is capable of displaying very realistic facial animation and human skin.

References

[1] L. T. Kohn, J. M. Corrigan, and M. S. Donaldson, Eds., *To Err is Human: Building a Safer Health System*. Washington, DC, USA: National Academy Press, Nov. 1999. [Online]. Available: http://www.nap.edu/catalog.php?record_id=9728

[2] D. J. Scott, J. C. Cendan, C. M. Pugh, R. M. Minter, G. L. Dunnington, and R. A. Kozar, "The Changing Face of Surgical Education: Simulation as the New Paradigm," *Journal of Surgical Research*, vol. 147, no. 2, pp. 189–193, Jun. 2008.

[3] S. Undre and A. Darzi, "Laparoscopy Simulators," *Journal of Endourology*, vol. 21, no. 3, pp. 274–279, Mar. 2007.

[4] S. Yule, R. Flin, S. Paterson-Brown, and N. Maran, "Non-technical skills for surgeons in the operating room: A review of the literature," *Surgery*, vol. 139, no. 2, pp. 140–149, Feb. 2006.

[5] L. Lingard, S. Espin, S. Whyte, G. Regehr, G. R. Baker, R. Reznick, J. Bohnen, B. Orser, D. Doran, and E. Grober, "Communication failures in the operating room: An observational classification of recurrent types and effects," *Quality & Safety in Health Care*, vol. 13, no. 5, pp. 330–334, Mar. 2004.

[6] Virtual Heroes. (2008) Virtual Heroes Inc – Serious Games and Andvanced Learning Systems. [Online]. Available: http://www.virtualheroes.com

[7] J. Taekman, N. Segall, E. Hobbs, and M. Wright, "3DiTeams – Healthcare Team Training in a Virtual Environment," *Anesthesiology*, vol. 107, no. A2145, p. A2145, Oct. 2007.

[8] S. Marks, J. Windsor, and B. Wünsche, "Evaluation of Game Engines for Simulated Surgical Training," in *GRAPHITE '07: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*. New York, NY, USA: ACM, Dec. 2007, pp. 273–280.

[9] R. Vertegaal, R. Slagter, G. van der Veer, and A. Nijholt, "Eye gaze patterns in conversations: there is more to conversational agents than meets the eyes," in *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2001, pp. 301–308.

[10] M. Garau, M. Slater, S. Bee, and M. A. Sasse, "The impact of eye gaze on communication using humanoid avatars," in *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2001, pp. 309–316.

[11] R. Vertegaal, G. van der Veer, and H. Vons, "Effects of Gaze on Multiparty Mediated Communication," in *Graphics Interface*, 2000, pp. 95–102.

[12] M. Slater, J. Howell, A. Steed, D.-P. Pertaub, and M. Gaurau, "Acting in Virtual Reality," in *Proceedings of the Third International Conference on Collaborative Virtual Environments*, 2000, pp. 103–110.

[13] T. Manninen, "Rich Interaction Model for Game and Virtual Environment Design," Ph.D. dissertation, University of Oulu, Finland, 2004.

[14] J.-C. Terrillon, H. Fukamachi, S. Akamatsu, and M. N. Shirazi, "Comparative Performance of Different Chrominance Spaces for Color Segmentation and Detection of Human Faces in Complex Scene Images," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, Mar. 1999, pp. 54–61.

[15] L. Chen and C. Grecos, "A fast skin region detector for colour images," *IEE Conference Publications*, vol. 2005, no. CP509, pp. 195–201, Apr. 2005.

[16] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*, vol. 4387, no. 95, pp. 95–102, Mar. 2001.

[17] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[18] Z. Hammal, C. Massot, G. Bedoya, and A. Caplier, "Eyes Segmentation Applied to Gaze Direction and Vigilance Estimation," in *Pattern Recognition and Image Analysis*, ser. Lecture Notes in Computer Science. Heidelberg: Springer Berlin, 2005, vol. 3687/2005, pp. 236–246.

[19] J. Whitehill and C. W. Omlin, "Haar Features for FACS AU Recognition," *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 97–101, 2006.

[20] P. Ekman, W. V. Friesen, and J. C. Hager, *The Facial Action Coding System – Second Edition.* Weidenfeld & Nicolson, 2002.

[21] Valve Corporation. (2004) Valve Source Engine Features. [Online]. Available: http://www.valvesoftware.com/sourcelicense/enginefeatures.htm

[22] Epic Games. (2006) Unreal Engine 3. [Online]. Available: http://www.unrealtechnology.com/html/technology/ue30.shtml

[23] Linden Research, Inc. (2008) Second Life. [Online]. Available: http://secondlife.com/

[24] D. Livingstone and J. Kemp, Eds., *Second Life Education Workshop 2007*, Aug. 2007. [Online]. Available: http://www.simteach.com/slccedu07proceedings.pdf

[25] P. Kenny, A. Hartholt, J. Gratch, W. Swartout, D. Traum, S. Marsella, and D. Piepol, "Building Interactive Virtual Humans for Training Environments," in *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2007*, ser. 2007, no. 7105, 2007.