Perceptive three dimensional interface via stereo observation

D Padbury, R J McCrindle and H Wei

School of Systems Engineering, The University of Reading
Whiteknights, Reading, Berkshire, UNITED KINGDOM

siu02dp@reading.ac.uk, r.j.mccrindle@reading.ac.uk, h.wei@reading.ac.uk

www.api.reading.ac.uk

ABSTRACT

This paper describes an intuitive approach for interacting with a computer or computer-driven applications. Interaction is achieved by observing, through a stereo camera set-up, the motion of a user's hands. This motion is then translated into 3-dimensional (3-D) coordinates to enable interaction with either a traditional 2-dimensional (2-D) desktop or a novel 3-D user interface. The aim of this work is to provide an intuitive method of interaction to computer based applications for individuals whose condition might restrict their ability to use a standard keyboard/mouse.

1. INTRODUCTION

Computer systems are now capable of producing incredible 3-dimensional (3-D) graphics but most users still work with a 2-dimensional (2-D) user interface that has remained largely unchanged for over 20 years. The orginal concept for the desktop user interface, the oNLine System (NLS), was demonstrated by Dr Doug Engelbart in 1968 (Englebart 1968). For this system a device was created for 2-D interaction which later became known as the 'Mouse'.

Today, nearly forty years on, interaction with a computer still general occurs via a combination of the keyboard and mouse. Whilst this is an appropriate form of interaction for many users, some users do not find it an intuitive form of input and for some users with mobility restrictions (e.g. lack of motor control in the fingers or arthritic conditions) they provide an obstacle rather than aid to interacting with a computer or computer-driven environment. In these instances it would be far more productive if a user could interact with the computer in an alternative manner to achieve a range of simple or complex tasks. Following on from the success of the Interaction via Motion Observation (IMO) system (Foyle & McCrindle 2004) first reported at ICDVRAT 2004, we have again used motion observation as a means of interaction but this time have extended the interaction to enable translation of the 3-D positioning of a users hands/head into a way of interacting with a 2-D desktop or a 3-D environment. To achieve this, a stereo rather than single camera set-up has been used together with a more sophisticated model of motion observation. Again like IMO, the MOTH (Motion Observation Three-dimensional Hand-tracking) system does not require the user to wear any external devices.

2. SYSTEM OVERVIEW AND APPROACH

The MOTH system uses image processing techniques to capture the 3-D position and movement of a user's hands from two cameras calibrated for stereo vision. This information is then supplied to the visualisation component of the system which is displayed as a 2 or 3-dimensional interface to the user. This process is illustrated in Figure 1.

Two Philips TouchCam II Pro cameras were used for image capture due to their high image quality, high frame rate, relatively low price and the availability of open source libraries to support development (Lavrsen 2006). C/C++ was chosen as the development language as a key requirement of the system is to respond to the user's motion in real-time. OpenCV (Intel 2006), an open source vision SDK which includes functions for camera capture, image processing and a cross platform GUI (Graphical User Interface) library was also used.

2.1 Calibration and Processing

For stereo vision to be used the system requires an initial calibration phase to derive properties of the cameras, such as their focal length and their location in the world – these are known as the intrinsic and extrinsic parameters.

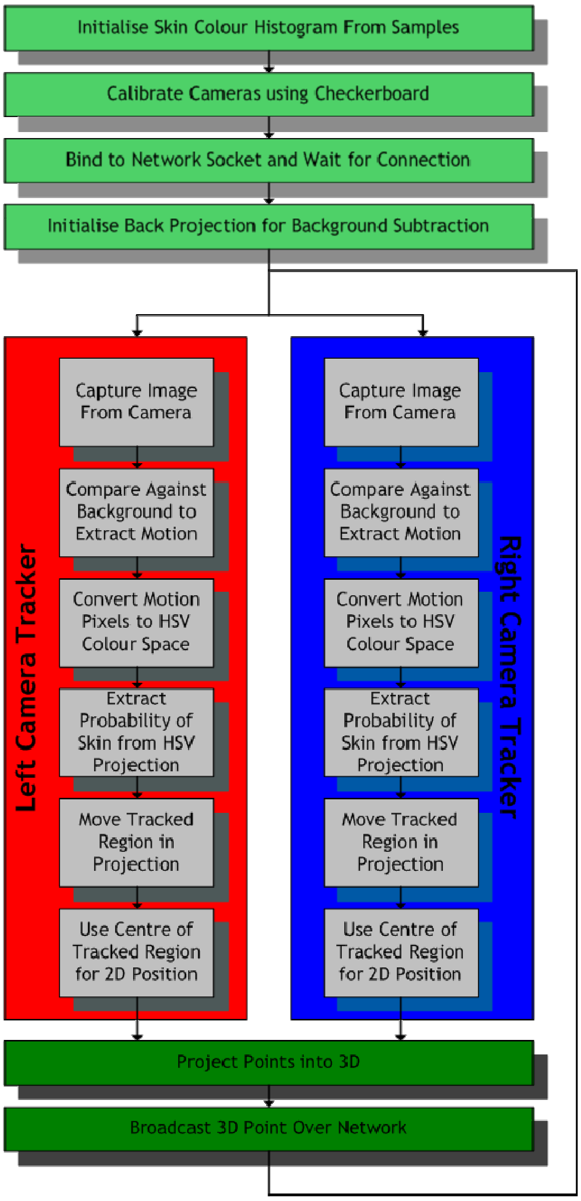
Initialise Skin Colour Histogram From Samples
Calibrate Cameras using Checkerboard
Bind to Network Socket and Wait for Connection
Initialise Back Projection for Background Subtraction
Left Camera Tracker
Right Camera Tracker
Capture Image From Camera
Compare Against Background to Extract Motion
Convert Motion Pixels to HSV Colour Space
Extract Probability of Skin from HSV Projection
Move Tracked Region in Projection
Use Centre of Tracked Region for 2D Position
Capture Image From Camera
Compare Against Background to Extract Motion
Convert Motion Pixels to HSV Colour Space
Extract Probability of Skin from HSV Projection
Move Tracked Region in Projection
Use Centre of Tracked Region for 2D Position
Project Points into 3D
Broadcast 3D Point Over Network

Figure 1. MOTH *System Diagram.*

Additionally, as both the computer vision tasks required for hand tracking and the three dimensional visualisation of the interface can be resource intensive, the system is spilt between two separate machines, one for the computer vision part of the system and the other for the interface. The two machines communicate over a TCP/IP channel which allows communication between different platforms to occur. In the developed system the computer vision sub-system is running on Fedora Linux whilst the interface visualisation sub-system is running on Microsoft Windows XP.

2.2 Colour Space

Computer images and graphics, and video signal transmission standards have defined many colour spaces with different properties. Some of them have been applied to the problem of skin colour modelling (Vezhnevets et al 2003). In this work, images captured from the web camera are delivered to the system in a 640 x 480 24-bit RGB (Red Green Blue) colour space but are then converted to the HSV (Hue Saturation Value) colour space. This is because whilst RGB colour values are easily corrupted by brightness, the HSV colour space separates the brightness from the actual colour (the hue). The resultant benefit of using HSV is that all people have the same skin pigment colour whatever the lightness or darkness of their skin (Bradski 1998). Brightness however can still corrupt the hue component if it is very high or very low and therefore to compensate for this all hue values with a corresponding brightness within 10% of the maximum and minimum were ignored. The value of 10% was derived from our experimental experience.

2.3 Skin Colour Segmentation

Nonparametric skin distribution modelling was used in the development of the MOTH system. The key idea of this method is to estimate skin colour distribution from the training data without having to derive an explicit model of the skin colour. This method is also referred as a SPM (Skin Probability Map) (Brand & Mason 2000, Gomez 2000). By using this method, a skin colour database was established as a training data. To improve accuracy, a skin colour histogram was built using a selection of sampled skin regions from images taken from a range of users. A simple program was developed to capture an image from the web camera, and by using a basic interface to enable the user to manually select regions of skin from the captured image. Figure 2 shows an example output from the skin colour selection application.

A database of the skin colour samples was developed during the project which was then used to build a two dimensional histogram of skin colour using the hue and saturation components of the gathered skin values from a number of individuals, see Fig. 3. To achieve this, the image was first converted from RGB to HSV and then each pixel checked for a saturation outside 10% of the maximum or minimum values to avoid brightness corruption. If the pixel colour matched these criteria it was then added to the histogram of hue values. An example of image sampling and the resultant histogram is shown in Figure 3.

After training, the histogram counts are normalised to convert histogram values to discrete probability distribution as:

$$P_{skin}(c) = \frac{skin[c]}{Norm} \qquad (1)$$

where *skin[c]* represents the value of the histogram bin corresponding to the skin colour vector *c*, and *Norm* is the maximum bin value present (Zarit et al. 1999). An inequality as expressed in (2) is used as a skin detection rule (Jones & Rehg 1999).

$$P_{skin}(c) \geq \Theta \qquad (2)$$

where Θ is a threshold defined in experiments.

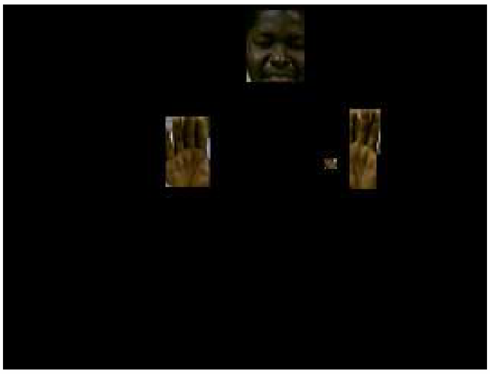

Figure 2. *Skin Colour Sample.*

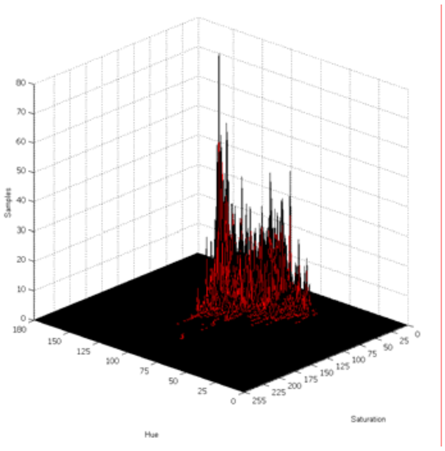

Figure 3. *Skin Histogram.*

Lighting conditions can significantly affect the results of the skin colour segmentation process. Currently all skin colour samples were captured in the same scene as the system was tested, reducing any differences in brightness. A scene with alternating lighting conditions such as windows being open/shut would make tracking significantly more difficult. An adaptive threshold technique to compensate for changes in scene brightness is currently being prototyped.

Subsequently when processing the received frame the hue component is extracted and each pixel's hue value looked up in the histogram. If it is more than a pre-determined threshold the pixel is considered to be skin coloured. This threshold is dependent on the lighting conditions of the scene and can be manually readjusted by the user during the calibration stage. The skin probability lookup process is extremely fast and produces good results, Fig. 4 shows a captured image and Fig. 5 shows which pixels of the image were considered skin coloured.

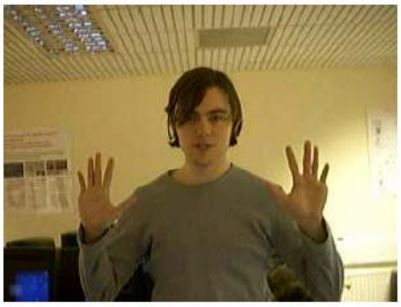

Figure 4. *Captured Image.*

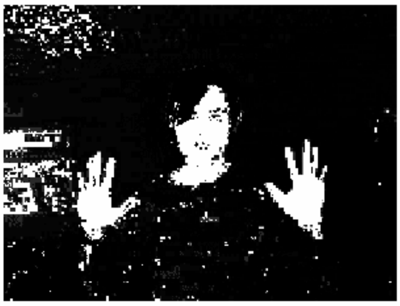

Figure 5. *Detected Skin.*

Although the skin colour detection generally works well, as demonstrated above it may also recognise parts of the background as being of skin colour which would be incorrect. This false-positive is expected to be improved when adaptive thresholds are in use. In our current system skin colour combined with motion detection prevents this from occurring.

2.4 Motion Detection

The MOTH system primarily works by capturing hand or head movement and therefore a method for calculating which parts of the image have moved has been used. In this approach the first frame captured is considered to be the background and is then subtracted from all subsequent frames to calculate the movement. This approach not only captures the motion but prevents aspects of the background being mistaken for 'skin'. The motion detection process is demonstrated in Figures 6–8: the captured image is shown in Fig. 6, the current background accumulator in Fig. 7 and the detected motion in Fig. 8.

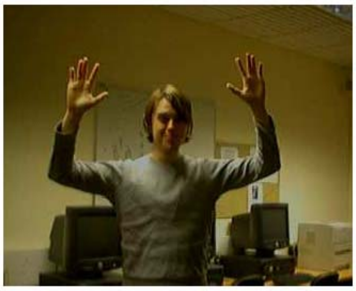

Figure 6. *Captured Image.*

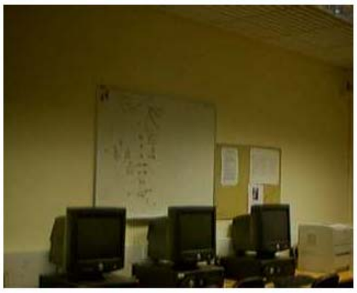

Figure 7. *Background.*

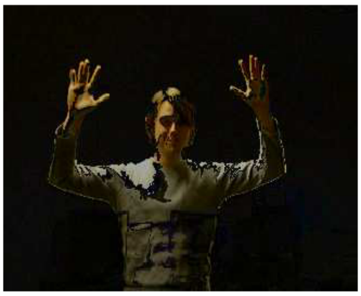

Figure 8. *Detected Motion.*

2.5 Object Recognition

The techniques for skin colour recognition and motion detection are used in combination to detect the motion of an individual's hands as shown below. Fig.9 shows the image captured from the camera, Fig. 10 shows the motion detected which clearly shows the area of the hands, Fig.11 shows which pixels it believes are skin and Fig.12 shows the combination of the skin pixels and the motion. Because the users head and the skin coloured poster were both stationary only the user's hands were picked up. This method now gives a suitable back projection method for use as a tracking algorithm.

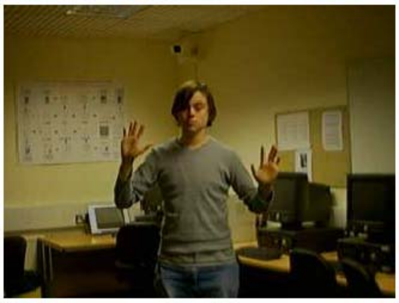

Figure 9. *Captured Image.*

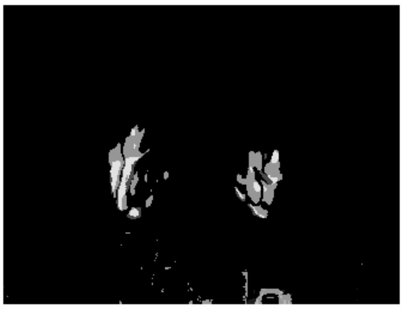

Figure 10. *Detected Motion.*

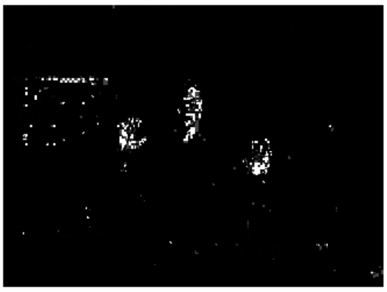

Figure 11. *Skin Colour.*

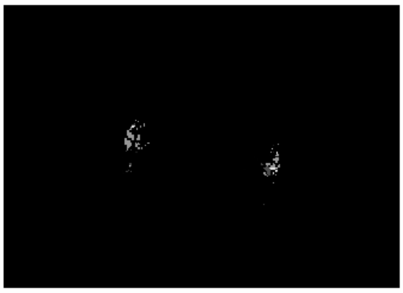

Figure 12. *Moving Skin Objects.*

The OpenCV implementation of the CAMshift algorithm with an adaptive search window size is used to find the centre of an object on the back projection (Bradski, 1984). This algorithm allows for tracking moving objects in the back projection between subsequent frames.

In this application three regions of the image are required to be tracked; the left hand, right hand and head. The position of these objects must be known before the CAMshift tracker can be started for each region. To find these initial positions three rectangles are drawn over the display from the cameras and the user is prompted to place their hand and hands in the boxes, see Fig. 13. After a short delay to allow the user to insert their hands into the regions, a CAMshift tracker is started for each region.

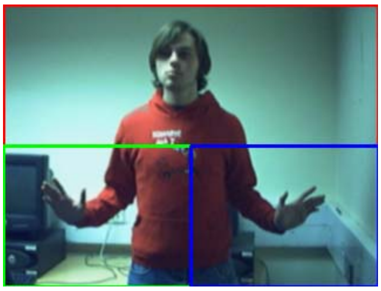

Figure 13. *Initialisation.*

The hand tracking algorithm performs very effectively, it is capable of processing 60 frames a second which provides a very smooth action even when the hands move very fast as shown in Figs. 14−17.

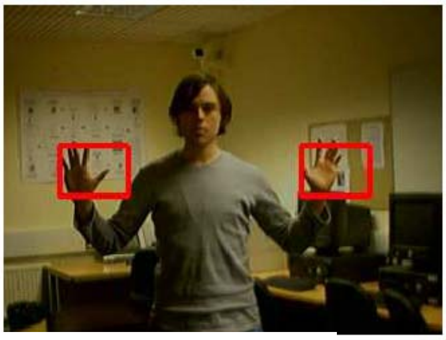

Figure 14. *Initialisation.*

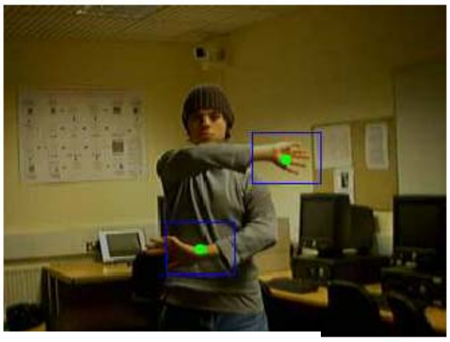

Figure 15. *Tracking 1.*

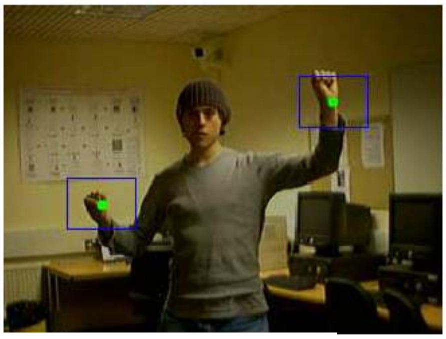

Figure 16. *Tracking 2.*

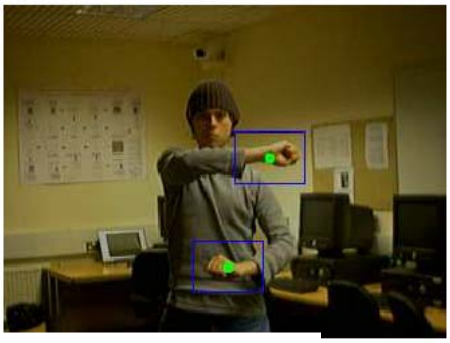

Figure 17. *Tracking 3.*

However, a potential problem which occasionally occurs, is when the object tracker loses the object it is tracking, for example, the user moves their head or hands outside of the image boundaries. A count is made of how many pixels of the tracked object are present inside the tracked region, when this decreases significantly the object is considered lost. In this even the initialisation rectangle for that object is replaced on the scene and after a short delay resumes tracking. This process is demonstrated in Fig. 18, Fig. 19, Fig. 20 and Fig. 21.

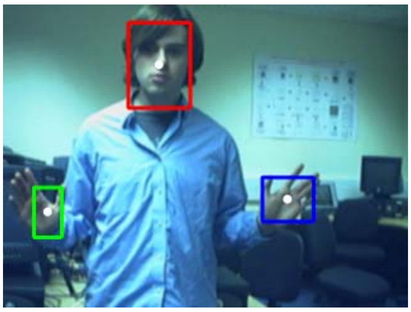

Figure 18. *Tracking Successfully.*

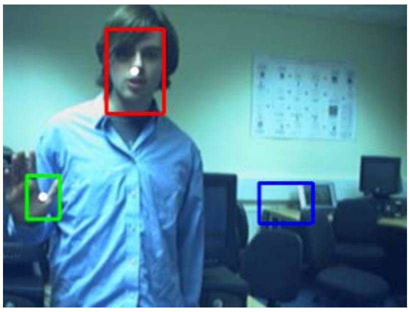

Figure 19. *Hand Removed.*

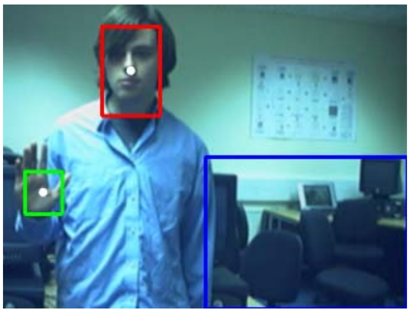

Figure 20. *Re-Initialisation.*

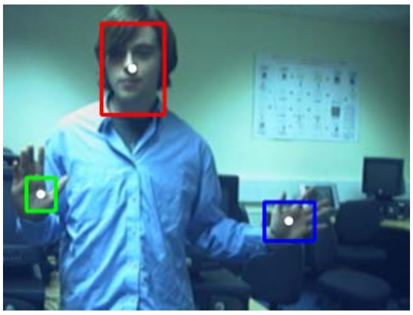

Figure 21. *Resumed Tracking.*

2.6 Stereo Vision

As stated earlier, the tracking processes are performed separately for each of the two cameras; this produces two dimensional positions for the hands and head for both cameras. The intrinsic and extrinsic parameters of the cameras are found in an initial calibration phase in which the user is required to display a checker board of known dimensions in front of both cameras. In this system the calibration procedures in OpenCV were used.

Corresponding two dimensional points in both images can be projected into a single three dimensional point in the world by finding the intersection of lines from the camera positions though the points in the image planes.

3. EXAMPLE APPLICATIONS

The above approach has been integrated into both a 2-dimensional desktop and 3-D environment. In the 2-D model the cameras can be used to detect movement of the left and right hands simultaneously. This enables more than one interaction with applications to occur at once, or to enable a simple exercise set to be developed which encourages movement of the hands in the left and right plane. An example interaction is demonstrated of the user moving both hands in a circular direction in Fig. 22; the position of the hands was tracked during motion which is shown in Fig. 23.

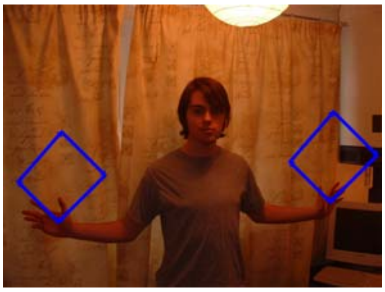

Figure 22. *Circular Hand Motion.*

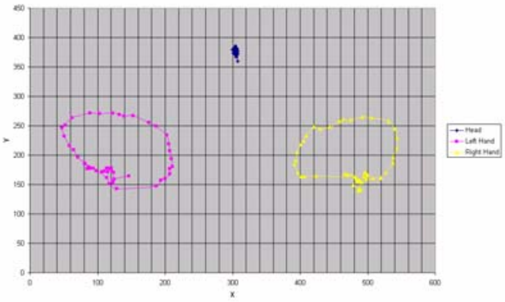

Figure 23. *Tracked Hand Motion Results.*

In the 3-D environment, movement is also encouraged in the forward and back plane, for example by manipulating a number of coloured spheres within the 3-D space, see Fig. 24. This can become a more entertaining exercise by placing a 3-D tube in the environment, such that the spheres can be controlled though the tube enabling simple basketball style games to be played. Fig. 24 shows the user manipulating the 3-D spheres.

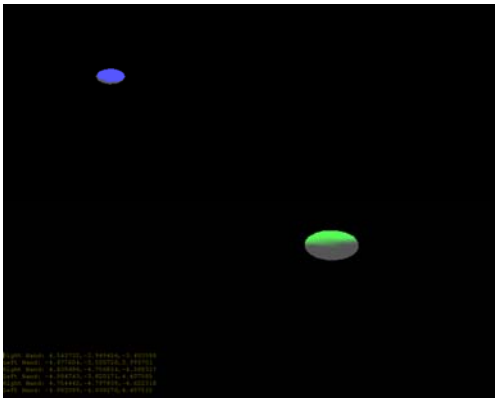

Figure 24. *3-D Ball Exercise.*

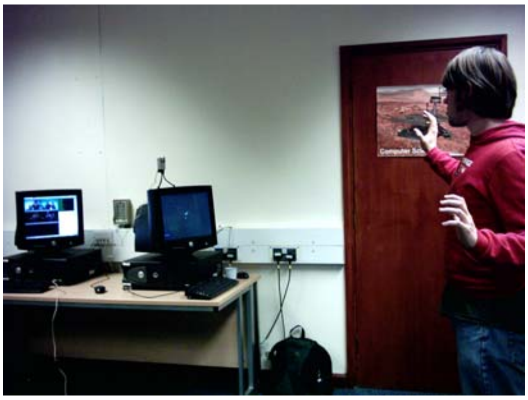

Figure 20. *System Use.*

4. FURTHER APPLICATIONS

There are a number of areas to which this technology and the MOTH system can be applied. These include the ability to interact with a traditional 2-dimensional computer desktop in a more productive manner and without the need to type or control the mouse; the ability to control a range of devices such as MP3 players, televisions or radios; and the ability to interact with objects in a 3-dimensional interface in order to develop spatial awareness or to coach or encourage a user to perform certain exercises for general health or rehabilitation purposes. A number of users have had the opportunity to use the MOTH system and results to date have been encouraging. The next stage is to conduct more formal trials with users who have limited fine motor control and to develop more fully the 3-dimensional aspects of the interface and corresponding exercises and games.

5. CONCLUSIONS

This paper has described a new way in which individuals, who do not possess the fine co-ordination required to use a mouse or keyboard, can interact with a computer or computer-driven environment. The MOTH system has been shown to effectively detect hand and/or head movements and translate these movements into inputs to a 2-D or 3-D interface. The interface may be populated with traditional computer desktop applications or may run applications which through a pattern of motion would encourage a user to undertake general or rehabilitation-oriented exercises, or to enable them to control their environment.

6. REFERENCES

G R Bradski (1998), *Computer Vision Face Tracking For Use in a Perceptual User Interface.*

J Brand and J Mason (2000), *A comparative Assessment of Three Approaches to Pixellevel Human Skin-detection*, Proc. of the 16th International Conference on Pattern Recognition, vol. 1, pp 1056-1059

D Engelbart (1968), *Demo of the Graphical User Interface*, http://sloan.stanford.edu/mousesite/1968Demo.html

M Foyle and R J McCrindle (2004), *Interaction Via Motion Observation (IMO).* ICDVRAT 2004.

G. Gomez, 2000, *On Selecting Colour Components for Skin Detection*, Proc. of the 16th International Conference on Pattern Recognition, vol. 2, pp 961-964

Intel (2006) *Open Source Computer Vision Library*, www.intel.com/technology/computing/opencv/index.htm.

M J Jones and J M Rehg (1999), *Statistical Colour Models with Application to Skin Detection*, Proc. of CVPR'99, Vol. 1, pp. 274 – 280.

K Lavrsen (2006), *PWC Documentation Project*, http://www.lavrsen.dk/twiki/bin/view/PWC/WebHome.

V Vezhnevets, V Sazonov and A Andreeva (2003), *A Survey on Pixel-Based Skin Color Detection Techniques,* Proc. Graphicon-2003, pp. 85-92, Moscow, Russia, September 2003.

B D Zarit, B J Super and F K H Quek (1999), *Comparison of Five Colour Models in Skin Pixel Classification*, Proc. of ICCV'99 International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, pp58-63.