2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

Human Computer Interaction for Virtual Haka Metamorphosis

Liyanage C De Silva James D Story Masood M Sujau

Information and Telecommunications Engineering
Institute of Information Sciences and Technology, Massey University
Private Bag 11222, Palmerston North, NEW ZEALAND.
L.DeSilva@massey.ac.nz

Abstract

In this paper we propose a platform at which any person can perform haka (War Dance) without any previous training. In the intended final system performance is done in front of a simple camera and a sensor system set up. The user will then be able to see his instant virtual haka metamorphosis in a giant screen in front of him. The proposal is to capture the participant's hand gestures, foot stamping and chanting words and exaggerate it to fit the actual haka performance and display as a graphical human character using a projector on a giant screen.

The implementation of this system involves Image Processing to identify human body movements and facial expression change. Computer graphics will be utilised to regenerate the exaggerated body movements and facial expressions. Emotional Speech Recognition and Synthesis will be used to create output voice of the performer with exaggerate emotional sounds. Here the body motion is captured via colour cues, which are placed on critical points of the human body. The colour cues are identified in 3D space through image processing and mapped to a controlled animation. The facial expression recognition study in this paper aims to classify 4 basic emotions (angry, sad, happy, and surprise). To determine the emotional state of a subject, a Feed-forward Multi-layered Perceptron is used. The present animation is of a very elementary nature and modification and improvement is need to make it look like a real haka metamorphosis

Keywords: Image processing, Colour Detection, Virtual Metamorphosis, Emotional Speech Recognition.

1 Introduction

Haka is another name for New Zealand war dance. It uses hand gestures, foot stamping, and chanting. Originally the haka or the War Dance is performed by Warriors before a battle, highlighting their strength and power and generally abusing the opposition. Recent days the haka is used in a number of situations such as performing before commencing a rugby game. Also it is performed at certain state functions, to welcome foreign dignitaries. However to perform haka the performer has to undergo a fair amount of practice to control his gestures and facial emotions.

Humans naturally communicate with one another through speech and body language. Yet when it comes to interacting with a computer system they are forced to use interfaces such as a keyboard, mouse, and screen. Obviously, this is not natural to humans and consequently in some cases it tends to make interaction difficult and/or time consuming. Examples of this would be in areas such as movies and animation, game play, and humanoid control just to list a few. Traditionally interfaces such as the keyboard and screen have been used due to the limited amount of processing power available, the inherently expensive alternative solutions, and the limited amount of research knowledge available. However, over recent years this has rapidly been changing and now many other options are becoming more and more economically viable. One of these options lies in the use of image processing.

Currently, the typical types of alternative interfaces, which allow human motion to be captured, are very expensive. Examples of this are the magnetic and RF based solutions used in the movie making industry. Here hundreds of sensors are placed on critical points of the body for tracking. Not only are these solutions normally bulky and therefore motion inhibiting but they also require high precision sensors which need to be setup in a dedicated environment i.e. it would not be suitable for home use. Image processing has recently been employed but once again these solutions have been relatively expensive and require professional expertise to operate. For example 18 high-speed cameras are commonly used [8] for Xbox animation sequences. In these systems there is no way to define which point is which i.e. all the points are the same; the points are captured and then linked at a later stage by an expert operator/animator. The aim of these expensive

systems is to provide a very high level of accuracy. In the case of many systems e.g. game play, this level of precision is not required.

Due to the recent acceleration in the use of digital camera equipment amongst the consumer market and the rapid increase in processing power per dollar the costs involved with developing an image processing system have become much more viable. Originally it was very difficult to produce a system based on image processing due to the costs involved with the camera equipment (CCD sensors, lenses etc) and the processing power required – a system would struggle to handle the sheer volume of data involved with processing an image. This of course has now changed with an average quality CCD camera costing only a few hundred dollars.

For this project it was decided that an image processing system should be developed to provide a computer interface that would calculate the location of critical points on the human in 3D space but instead focus on keeping the cost low and flexibility high – the precision of the system should not be the major objective. Also the facial expression changes were tracked using a camera focused to capture the face image in the full image view. Finally it was decided to target a system for use in simple animation.

2 System Overview

Before developing the system a number of methods were investigated for the tracking of critical points on a human Figure using image processing [3][4][5][6][7]. The problem with these systems is the fact they are either expensive (both computationally and monetary wise) or limited in their tracking ability.

For example, the methods described in [3] and [4] use properties of skin tone. Here only the hands and head are tracked and they must be wearing a long sleeved shirt. If more points such as the feet are to be tracked then this is not currently possible without severe limitations to the movement. The methods described in [5] and [6] have their own limitations with regard to the applications they can be applied to but they are also expensive and not really suitable for a home environment. The developed system used some of the background ideas from [5] and [6] combined with some restrictions.

Figure 1 shows the system block diagram. The heart of the algorithms operation lies in the image capture, image processing and output blocks. These will be discussed in more detail in the remainder of this paper.

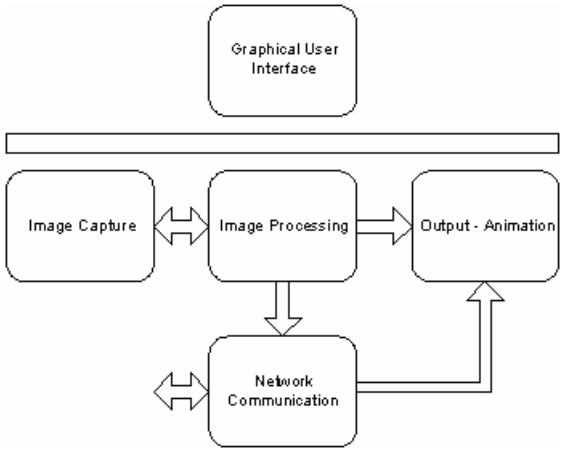

Figure 1: System Block Diagram.

First the frames are captured at a resolution of 320x240 in a 24 bit RGB format at 30 frames per second from an Ikegami ICD-835PAC color camera into a single channel PCI based frame grabber card. A resolution of 320x240 was used to ensure that there was enough resolution in the image to detect someone's hand at a significant distance from the camera and to provide a challenge to the processing system in terms of data volume (if it works with this data volume then it at lower resolutions the algorithm will easily execute in time but it may not have the resolution required for identifying the colour cues).

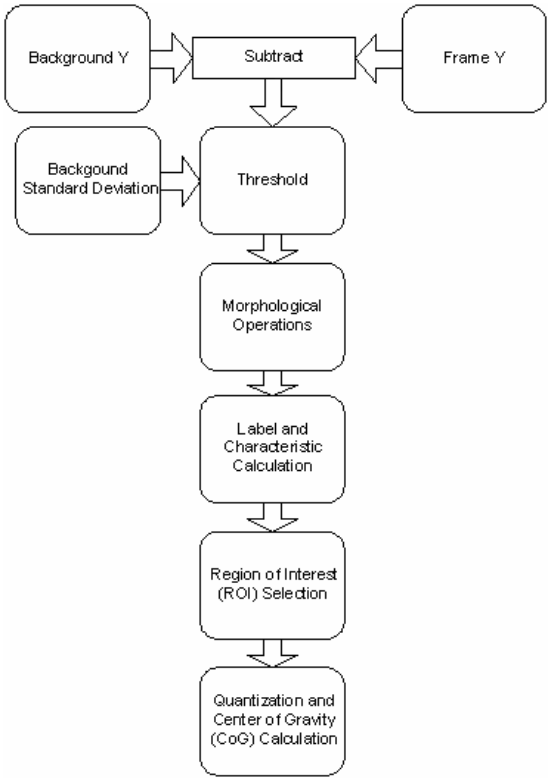

Figure 2: Image Processing Block Diagram.

3 Image Capture and Processing

The image processing carried out aimed at providing a high speed of execution and high memory efficiency. This was for two reasons: 1) The application needed to run in real time, and 2) The application needed to aim at being able to be ported to a restricted hardware platform at a later stage. Figure 2 shows a block diagram detailing the image processing stages that lead to the location of the critical points.

3.1 Image Capture

An Ikegami ICD-835PAC camera was used over the cheaper web camera solution due to the fact that the web cameras tested only offered a USB1.1 interface operating at a maximum data rate of 12Mb/s. For this application over 55Mb/s was needed for an uncompressed video stream. The image captured from a web camera was formed through interpolation and resulted in the captured image being reduced in color and having a slow response rate. This was unsuitable for the developed application. In future developments a system using cheaper cameras will need to be reinvestigated.

3.2 Background Subtraction and Binary Thresholding

The first stage in the actual image processing algorithm is background subtraction. The reasons for using background subtraction were: 1) It is simple and consequently fast, 2) The cameras were in a static location, and 3) The application environment allowed for initialization. Here subtraction was performed using the luminance component of the image. Since the image was in an RGB format the Y component was computed using a lookup table (LUT) that had been filled using (1).

$$Y = (299R + 587G + 114B)/1000 \quad (1)$$

The reason for using a LUT was to increase the speed of execution although a significant amount of extra memory is required in using this approach.

The background was computed by taking a 16 frame average when the subject to track was removed from the tracking area. The standard deviation for each of the background pixels was also computed and stored for the binary thresholding.

This thresholding technique was detailed in [1] and assumes that pixels naturally vary over time. The variation is assumed to follow a normal distribution and thus it can be assumed that 95% of the time the natural deviation will lie within 2 standard deviations of the mean. Any deviation outside this bound could be deemed as not natural i.e. something has changed. However, as detailed in [1] it can be found that the pixels do not follow a normal distribution, although this provides a useful starting point, so an offset is included.

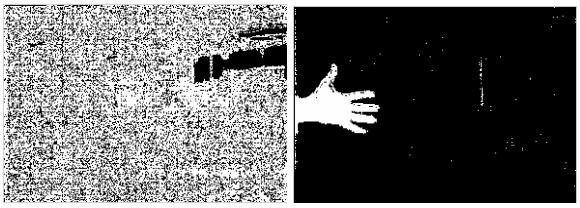

Figure 3: Changing the Offset from 0 to 5.

Figure 3 shows the results obtained when the offset is varied. Here an offset of 5 was found to be the best. When the offset was too large some of the areas needing to be detected would be missed.

3.3 Morphological Operations

As shown in Figure 3, with an offset of 5 there is still a significant amount of noise that needs to be removed before further processing can be carried out. In image processing this is commonly carried out through morphological operations. In this research a morphological opening, erosion followed by dilation, was used. The mask size was set to 3x3. Figure 4 shows the format of the erosion process:

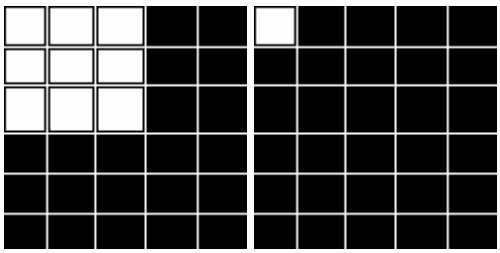

Figure 4: Morphological Erosion.

Here the output pixel is taken to be the top left pixel. When this is combined with the scan order of left to right, top to bottom it means that the result can be written back into the binary buffer being processed without it affecting any proceeding operations. This significantly improves the memory efficiency since no temporary result buffer is needed and it also improves execution speed. In terms of carrying out the actual calculation a pipeline was used. This ensures that only 3 memory accesses are required per shift of the mask rather than the full 9 memory accesses had a pipeline not been utilized.

3.4 Binary Blob Labeling and Characteristic Calculation

In order to find the characteristics of the blobs a labeling process is carried out. By giving a label to each of the binary pixels it is able to be determined which pixels belong to which blob and hence

properties such as the blob area, the blobs center of gravity (CoG) and the bounding box can be computed. These characteristics are used in the classification process leading to the selection of the region of interest (ROI) ready for the final processing stage.

In order to label the blobs a contour-tracing algorithm described in [2] is used. This algorithm is superior in performance to the many common 2 pass algorithms used since it requires only one pass through the image and requires no extra memory or large tables. The contour-tracing algorithm is not a true 1-pass algorithm since some pixels are visited more than once. These are contour pixels/branch pixels. A pure contour pixel is visited 1 (+1) times. In [2] it describes that under the situation shown below in Figure 5 the centre pixel may be visited up to 4 (+1) times. This situation is avoided in our case due to the 3x3 mask applied in the morphological opening i.e. this situation will never occur. In our proposal the most any pixel can be visited is 2 (+1) times as shown in Figure 6. Since the number of contour pixels is usually insignificant to the number of pixels in the image, and branch pixels are even lower in number, this is not an issue.

Figure 7 shows the tracing in operation and the corresponding labeled image. Due to the way the contour tracing algorithm is carried out it also means that characteristics such as the area, bounding box, and CoG for each of the blobs can be calculated in the same pass through the image with minimal extra computation.

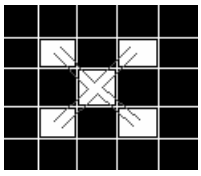

Figure 5: 4 (+1) Visits Avoided by 3x3 Opening.

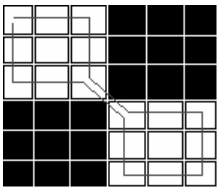

Figure 6: 2 (+1) Visits is the Worst Case.

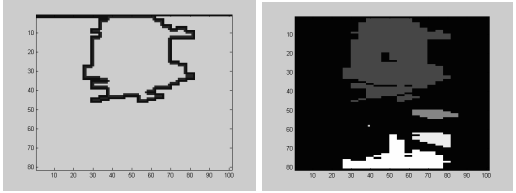

Figure 7: The Contour Tracing Algorithm and the Resulting Labeled Image.

3.5 Region of Interest Selection

Once the characteristics of each of the blobs have been calculated, it is then time to make a decision about which blob represents the person. If there is no blob greater than a set minimum size then it is assumed that no person is present to track and the processing terminates. If blobs exist with a size greater than a set minimum threshold then the largest of these is chosen as the region of interest (ROI). Once this has been established the other blobs have the opportunity to also join the region of interest and get relabeled based on the distance between the CoGs. The distance between the CoGs must be less than an adjustable radius, which is set by the area of the largest blob. This extra inclusion was based on the observation that often the main blob is the body – here the hands may or may not be joined to the main blob. However, the hands are one of the critical points and have an associated color cue. This region needs to be included to ensure that each color cue is successfully located otherwise the hand would be returned as not found. This is illustrated below in Figure 8.

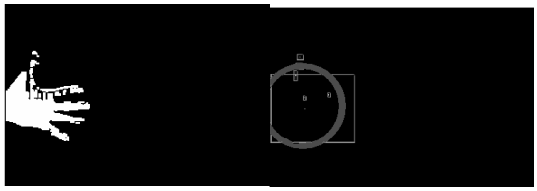

Figure 8: Binary blobs and Their Associated Bounding Boxes and CoGs Following Labeling.

Note the adjustable radius used for inclusion of offshoot blobs.

3.6 Color Quantization and Point Calculation

The ROI found above has an associated label and bounding box. The bounding box is used to reduce the search space and correspondingly result in an increased execution speed through the final processing stages. The binary blob for the ROI is masked back onto the original RGB image. Any pixels lying under a binary 1 are quantized; the pixels lying under a binary 0 are ignored. These RGB pixels are converted to chrominance (color difference) values, U and V, using the following formula:

$$U = [(-169R - 331G + 500B)/1000] + 128; \quad (2)$$

$$V = [(500R - 419G - 81B)/1000] + 128; \quad (3)$$

Here a LUT was not utilized due to the large memory requirements and the limited number of values actually indexed in the table. The U and V values are used since we are looking for each of the color cues i.e. we need to look at the color information – the luminance or gray level component is not important. Here the U and V

values are compared to a series of preset ranges that each color cue is given during initialization.

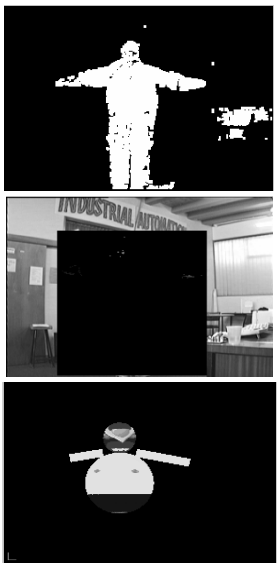

Figure 9: The Complete Processing Algorithm.

None of these ranges overlap in both U and V meaning that a quantized pixel can only belong to one color group. Any pixels whose U and V values lie outside all the preset ranges are discarded from any further processing. The CoG is calculated for each color during the quantization pass. This gives the location of the color cues. The results are shown below in Figure 11.

Here only 3 colors have been initially tracked – Red, Green, and Blue. The U and V values for the other colors to be added in the future are given in Table 1. As we can see the separation between all of the colors should be large enough that on addition of this extra color cues they will still be able to be uniquely identified.

Table 1: U and V values for the color cues.

Color	Mean U	Min U	Max U
Red	90	75	101
Green	44	32	57
Blue	144	137	150
Yellow	38	31	49
Light Blue	148	139	158
Orange	71	67	81

Color	Mean V	Min V	Max V
Red	226	214	238
Green	129	116	136
Blue	123	119	131
Yellow	194	183	210
Light Blue	102	94	111
Orange	235	225	241

4 3D Coordinate generation

The developed system locates the points in 3D space using two cameras placed orthogonally to one another. Here the system was not developed for providing accurate locations but rather on proving a concept. In future developments the inclusion of a calibration phase would be necessary. In the developed solution one camera is used to calculate each colors XY coordinate and the second camera is used to calculate each colors ZY coordinate. The bird's eye view of the setup is shown below in Figure 10.

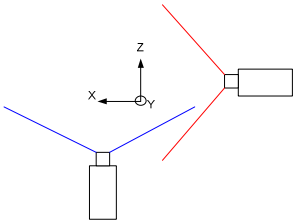

Figure 10: Birds eye view of camera setup.

In the case of the developed application, the controlled output was an animated character generated in OpenGL. Here the model purely mimicked the motion of the person being tracked. At later stages the model will become more complex through the use of additional color cues. Obviously this coordinate information could be used to control robots, game characters etc.

5 Facial Expression Recognition

The accuracy of the final feature extraction algorithm of the facial expression recognition algorithm was estimated at 78%. The test sample consisted of 50 different individuals from the CMU face database with varying skin colour and features. From those that failed, the majority were due to false eye plane and eye position detection. Subjects wearing earnings or having bushy eyebrows were the main cause for this.

The tracking algorithm was applied to the successful feature extraction results, i.e. a total of 37 video clips. A tracking accuracy of 94% was achieved. Figure 11 shows a series of frames depicting the tracking results for an individual.

Figure 11: Results of tracking

To obtain the overall system accuracy, a total of 25 different clips were analysed, making sure that they were not the same ones used for training the system. The overall system accuracy was estimated at 67%. Table 2 highlights the details of the trial.

Table 2: Classification results.

		Deduced Emotion					
		Happy	Angry	Surprise	Sad	Neutral	Accuracy
Actual Emotion	Happy	6	1	0	0	0	85.71%
	Angry	1	3	1	1	0	50.00%
	Surprise	0	0	6	0	0	100.00%
	Sad	1	2	0	2	1	33.33%
						Average Accuracy	**67.26%**

System accuracy rapidly drops when dealing with negative emotions such as sad at 33% and angry at 50%.

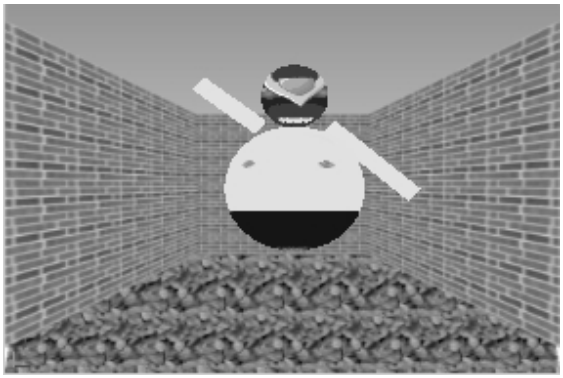

Figure 12: Controlled 3D character.

6 Performance

The system that was developed was written in Microsoft Visual C++ 6.0 and takes approximately 10-14ms to process a 320x240 24 bit RGB frame on a 32 bit AMD 1900+ with 512MB of RAM. At 30fps the CPU usage is approximately 35%. This machine is currently operating as the server and is also performing the animation. On the client (second) PC, which processes the frames from the second camera, the same processing time was experienced but here approximately 70% CPU usage was experienced. This PC was a 32 bit 550MHz Pentium III with 256MB of RAM. The client PC performed no animation.

As we can see the server has enough processing power available to analyze both of the cameras output. During development there were issues when the two capture devices where housed in the same machine. This lead to the client/server situation described above where the two computers shared the color cue coordinates over a TCP/IP connection in order to generate the 3D animation. In future developments this will need to be removed.

7 Conclusion

A system has been produced which successfully locates colour cues in 3D space in order to control an animated character. It operates in real time and proves that the concept of using image processing for animation using readily available consumer technology. The facial expression recognition system is currently work as a separate block but will be combined in the final system. The final expected system is shown in the Figure 13.

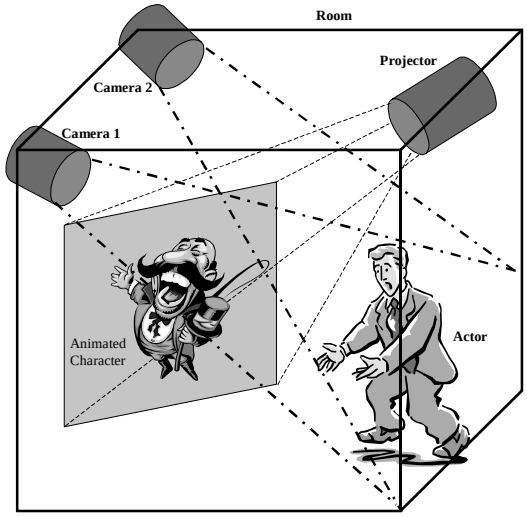

Figure 13: Final Expected System.

8 References

[1] Sangho Park and J.K. Aggarwal, *Head Segmentation and Head Orientation in 3D Space for Pose Estimation of Multiple People*, in Proceedings of 4th IEEE Southwest Symposium Image Analysis and Interpretation, 2-4 April 2000, pp. 192 – 196.

[2] Fu Chang, Chen-Jen Chen and Chi-Jen Lu, *A linear time component labeling algorithm using contour tracing technique*, in Proceedings of the Seventh International Conference on Document Analysis and Recognition, 3-6 Aug. 2003, pp. 741 – 745.

[3] Stan Birchfield, *Elliptical Head Tracking Using Intensity Gradients and Color Histograms*, in Procs. of 1998 IEEE Computer Society Conf. on Computer Vision and Patt. Recog., 23-25 June 1998, pp. 232 – 237.

[4] Olivier Bernier and Daniel Collobert, *Head and Hands 3D Tracking in Real Time by the EM algorithm*, in Procs. of IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 13 July 2001, pp. 75 – 81.

[5] Jun Ohya, Kazuyuki Ebihara, Jun Kurumisawa, and Ryohei Nakatsu, *Virtual Kabuki Theater: Towards the Realization of Human Metamorphosis Systems*, in Procs. of the 5th IEEE Int. Workshop on ,Robot and Human Comm., 11-14 Nov. 1996, pp. 416 – 421.

[6] Shoichiro Iwasawa, Jun Ohya, Kazuhiko Takahashi, Tatsumi Sakaguchi, Kazuyuki Ebihara, Shigeo Morishima, *Human Body Postures from Trinocular Camera Images*, in Procs. of the Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition,28-30 March 2000, pp. 326 – 331.

[7] R.C.K Hua, Liyanage C. De Silva and Prahlad Vadakkepat, *Detection and Tracking of Faces in Real-Time Environments*, in Procs. of The Int. Conf. on Imaging Science, Systems, and Technology, Las Vegas, USA, 24-27 June 2002.

[8] Xbox Game Animation (19,9,2004). http://www.xbox.com/enUS/nflfever2003/spotlight2.htm