iGesture: A Platform for Investigating Multimodal, Multimedia Gesture-based Interactions

Jonathon S. Hare, Maria Karam, Paul H. Lewis, m.c. schraefel
{jsh02r | amrk03r | phl | mc}@ecs.soton.ac.uk
Intelligence, Agents, Multimedia Group
Electronics and Computer Science, University of Southampton
Southampton, SO17 1BJ, United Kingdom

ABSTRACT

This paper introduces the iGesture platform for investigating multimodal gesture based interactions in multimedia contexts. iGesture is a low-cost, extensible system that uses visual recognition of hand movements to support gesture-based input. Computer vision techniques support gesture based interactions that are lightweight, with minimal interaction constraints. The system enables gestures to be carried out 'in the environment' at a distance from the camera, enabling multimodal interaction in a naturalistic, transparent manner in a ubiquitous computing environment. The iGesture system can also be rapidly scripted to enable gesture-based input with a wide variety of applications. In this paper we present the technology behind the iGesture software, and a performance evaluation of the gesture recognition subsystem. We also present two exemplar multimedia application contexts which we are using to explore ambient gesture-based interactions.

Categories and Subject Descriptors

I.5.4 [**Applications**]: Computer Vision; H.5.2 [**User Interfaces**]: Input devices and strategies; I.4.8 [**Scene Analysis**]: Motion, Tracking; J.5 [**Arts and Humanities**]: Music

General Terms

Algorithms, Human Factors, Experimentation, Reliability

Keywords

Gesture-based interaction, semaphoric gestures

1. INTRODUCTION

This paper presents iGesture, our extensible platform for research into gesture-based, off the desktop interactions. We call these interactions 'ambient gestures'.

MM'04, October 10-15, 2004, New York, New York, USA.

iGesture was initially developed to investigate the use of 'semaphoric' gestures [6] in one's environment to control multimedia applications, such as a digital jukebox [3]. Semaphoric gestures are those which represent signals or signs that can correspond to functional commands. To explore semaphoric, ambient gestures in more contexts, we extended the iGesture platform to support rapid deployment of gesture-based input for a broader range of applications.

Primarily, the iGesture software performs a number of computer vision tasks in order to extract information from the visual scene to determine if gestures are being performed. The platform has been deliberately designed to work with a standard PC and single webcam. iGesture supports a number of loosely constrained single and two-handed gestures, such as waving. These gestures can be combined into a large command vocabulary. Significantly, these gestures can be readily detected at dynamically variable distances from the camera. This use of lightweight gestures with variable position interaction is in contrast to previous visual gesture recognition work which has required either close proximity of the gestures to the camera [1] or fixed proximity to the camera for application control [2]. By enabling these naturalistic, 'in the environment' gestures, iGesture provides a platform to explore ambient gestures within a ubiquitous, multimodal interaction space.

Figure 1: Participant performing gestures using coloured markers during an evaluation session

In the first part of the paper, we describe the iGesture system, and provide results of an evaluation of the accuracy of the gesture detection. We then review two different applications in which we are currently using the platform to explore

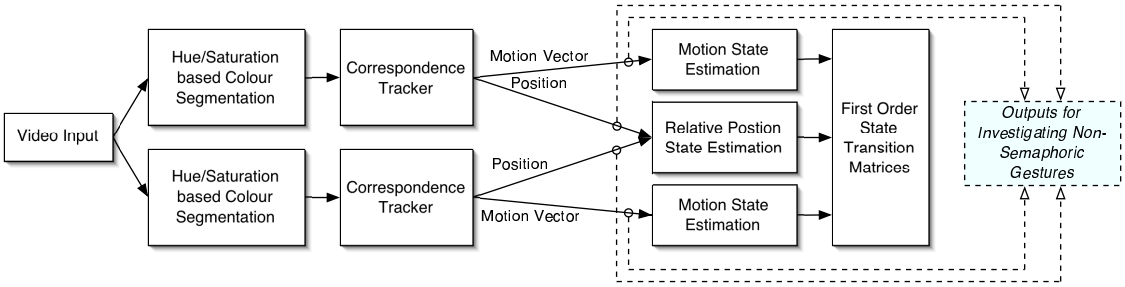

Figure 2: iGesture Video Processing Subsystem

ambient gestures. First, we look at the use of ambient gestures for traditional interactions, such as functional control of desktop multimedia applications, thus extending the interaction from the desktop into the environment. Second, we look at ambient gestures for creative applications, such as their use as MIDI controllers for ambient orchestration. We conclude with a brief look at future work.

2. iGesture

The iGesture software conceptually consists of two subsystems. The first is responsible for the near real-time video processing and state estimation. The second is responsible for recognising semaphoric gestures. In addition, the software allows information from the video processing stage to be extracted and used for investigating non-semaphoric gesture based interactions.

2.1 Video Processing

In order to enable near real-time processing and ensure robustness, the vision subsystem is kept relatively simple. The original design of the specifications called for a system that was able to recognise coarse scale semaphoric gestures performed with one or two hands. In order to simplify the problem domain in terms of the computer vision, it was decided that coloured markers would be used rather than attempting to track raw hand motion. The system performs processing on two *channels* in parallel. These *channels* were originally designed to correspond to the left and right hand of the user. Figure 2 illustrates the computation performed by the subsystem.

Within each channel, the input video is smoothed using a Gaussian filter and converted to a Hue, Saturation, and Value colour-space in order to de-couple intensity changes due to variations in lighting conditions. The colour marker corresponding to the channel is then segmented from the image by selecting pixels whose value (and 8-nearest-neighbours) have a similar hue and saturation to the marker. In order to reject noisy pixels, only the largest segmented region within a pre-determined upper and lower bound is kept for further processing. The tracking process involves comparing the shape and position of the segmented object in the current frame to the position and shape in the previous frame, and checking that the variation is within reasonable bounds. If the variation in position and shape is within the bounds defined, then a motion vector is easily created by comparing the objects position in the previous frame to its position in the current frame.

The final video processing stage involves transforming the motion vectors and marker positions into a state representation. In the current embodiment, there are 5 motion states per channel, and 4 position states which are shared across the two channels. The motion states represent whether the marker's motion between the previous and current frame was upward, downward, left, right, or stationary. The position states represent the relative position of the two markers (if both are present) and are: marker 1 above marker 2, marker 2 above marker 1, markers approximately level, and markers really close together. From this set of states it is possible to model the motion of markers over a given number of frames as a Markov chain and determine a set of state-transition matrices that represent the chain. This set of state-transition matrices can be used to describe a semaphoric gesture. The construction of such a matrix is illustrated in Figure 3.

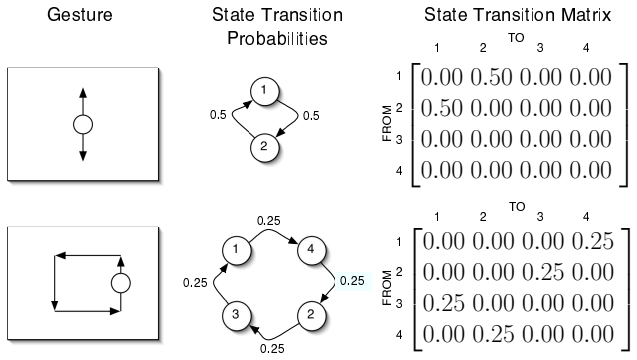

Figure 3: Example showing construction of state-transition matrices from simple semaphoric gestures

2.2 Semaphoric Gesture Recognition

As shown in the illustration in Figure 3, the state transition matrices for different semaphoric gestures tend to be quite unique. We use this fact to enable the system to recognise gestures by determining a distance relationship between state transition matrices. In the current implementation, a sum-of-absolute-difference measure is used:

$$D_{\mathrm{SAD}}(\mathbf{P}^{(1)}||\mathbf{P}^{(2)}) = \sum_{i=1}^{N}\sum_{j=1}^{N}|\mathbf{P}^{(1)}_{i,j} - \mathbf{P}^{(2)}_{i,j}|,$$

where $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ are $N \times N$ state transition probability matrices. Other distance metrics could also be used, such as the Kullback-Leibler distance, which has been successfully

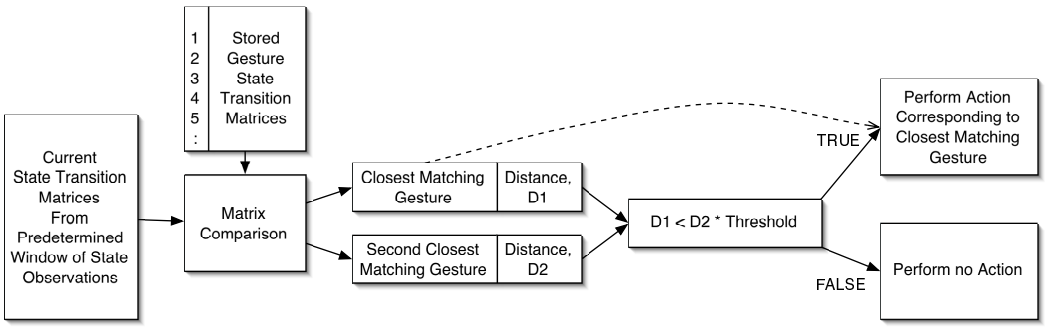

Figure 4: Diagram illustrating the semaphoric gesture recognition process

used for comparing state transition matrices of music from different composers [4].

The actual gesture recognition proceeds as follows. Given a set of semaphoric gestures, the system is trained, and records the three state transition matrices corresponding to each gesture, together with an *action* which is to be performed when the gesture is recognised. The *action* in this case is a shell command or script that is executed when the gesture is recognised. Once all the gestures have been trained, the system can be put into a monitoring mode where the input is continuously monitored for gestures. The system watches for gestures by considering state transitions in a continuously updating window. This window is updated every time a new frame of video is captured (at least 15 times per second), and the state transition matrices corresponding to the window are compared to each of the matrices of the stored gestures. An overall distance between each of the matrices representing the recorded gestures and the matrices representing the gesture in the current window is calculated by linear summation:

$$D_{\text{overall}}(\{\mathbf{P}^{(U)}\}||\{\mathbf{P}^{(G_n)}\}) = \sum_{i=1}^{3} D_{\text{SAD}}(\{\mathbf{P}^{(U)}\}_i||\{\mathbf{P}^{(G_n)}\}_i),$$

where $\{\mathbf{P}^{(U)}\}$ represents the set of three state transition matrices (left channel motion, right channel motion and relative position) from the unknown gesture in the window and $\{\mathbf{P}^{(G_n)}\}$ is the set of three matrices from the nth learnt gesture. Finally, a test is performed between the distances of the two gestures with the lowest distances to determine if the *action* corresponding to the gesture with the lowest distance should be performed. This is illustrated in Figure 4. If a gesture is recognised, the window of state observations is reset, and the system stops monitoring the input for a predetermined time, before re-commencing monitoring. This helps ensure that the action for a given gesture is only performed once. If no gesture is recognised, the system just continues updating the window of state observations until a gesture is recognised.

2.2.1 Evaluation

In order to check the accuracy of the semaphoric gesture recognition subsystem, we performed an evaluation in controlled conditions. The system was set up in a room with fixed lighting and the camera was positioned to cover as much area as possible. The fixed lighting conditions mimic the office environment in which the system is currently deployed. Future studies will look into the effect of lighting change on the recognition performance. The evaluation was designed to assess the accuracy rate of the system in terms of percentage of correctly recognised gestures, the percentage of false positives (gestures incorrectly recognised) and percentage of false negatives (gestures not recognised). A gender balanced group of 8 volunteers was assembled for the evaluation in order to assess the performance over a group of potential users. The evaluation was performed in two parts. First, the system was loaded with a set of pre-trained gestures. Second, the participants were asked to train the system themselves before the evaluation commenced. These two parts allowed us to evaluate the effect of user-trained versus pre-trained gestures on the recognition accuracy. Both parts of the evaluation consisted of two subparts. First, the subjects were asked to perform each of the gestures 5 times in a stationary position directly in the centre of the camera's field of view. In the second subpart, five different points in the room were pre-selected to cover the full visual field of the camera. The subjects were then asked to perform the gestures at each point, whilst facing the camera. The results of the evaluation showed no statistically significant intra- and inter-subject variability, so the results have been averaged and are shown in Table 1.

The results show that the system is capable of recognising semaphoric gestures with an accuracy of over 90% when the gesture is performed in the centre of the camera's field of view, and with a false positive rate of under 5%.

2.3 Extensibility

As the iGesture platform was designed to be extensible, we engineered the software to exploit much of the raw visual information from the tracking module. As Figure 2 shows, both the spatial position and motion vector of the marker used for each channel are available for use in exploring non-semaphoric gestures. As explained in more detail in Section 3.2, we have demonstrated this extensibility by allowing MIDI instruments to be controlled by using the relative spatial positions of the tracked markers and their velocity or acceleration.

3. GESTURE-BASED INTERACTIONS

Gestures are rapidly becoming a viable addition to multimodal interactions within ubiquitous computing environments [5]. However, the ability to use gestures in everyday computing applications has not received much attention in the literature. What the iGesture platform provides, is a means to implement gesture based interactions for a wide

Table 1: Averaged results from each part of the evaluation

	Average Correct	Average Incorrect	Average Missed
Pre-trained, Centred	93%	4%	3%
Pre-trained, Non-Centred	84%	5%	12%
Subject-trained, Centred	91%	3%	7%
Subject-trained, Non-Centred	87%	2%	11%

variety of software and hardware applications. In addition, it allows researchers to conduct extensive investigations and evaluations on the use of gestures in many different domains.

3.1 Extending Desktop Interactions

In ambient gestures, we investigated gestures 'in the environment' as a means of exploring the effect of controlling applications without interacting with the GUI. The work was based on semaphoric gestures that functionally control traditional desktop applications such as browsing and playing tracks from a digital jukebox. The interaction does not require visual attention from the user as audio is the primary feedback mechanism. With the ambient gestures interaction, we successfully provide a means of separating the background from the foreground applications to allow the user to have simultaneous control over multiple applications. This means that a user can control their applications while maintaining primary focus on tasks such as washing dishes or reading a book. Based on this work, we have extended the iGesture platform to allow us to design gesture based interactions for a diverse range of applications.

3.2 Creative Interaction

Beyond taking desktop multimedia applications into the environment, we also use iGesture to support several interaction styles that explore various mappings of gestures onto MIDI output. One style maps the gestures associated with the playing of an air guitar to the expected output. As the user moves their leading hand in a strumming motion, sounds are produced while the frequencies of the notes are controlled by the distance between the hands. A second style maps all detected movements of the hands to create the sounds, also using the distance between hands to control the notes. We chose these and other interaction styles, including a drumming gesture, to investigate the use of gestures in non-traditional applications exposing the users to a novel way to explore interaction in a ubiquitous, multimedia, multimodal environment.

4. FUTURE WORK

Future work involving ambient gestures includes conducting studies to investigate a range of issues with 'in the environment' interactions. We want to compare ambient gesture performance against other multimodal I/O in order to better understand the trade-offs between approaches in ubiquitous environments. The iGesture system also supports multiple channels. Thus, multiple users can participate in an appropriately configured application. This gives us a new way to investigate collaborative, concurrent interaction in multimodal, ubiquitous spaces. Future work involving the iGesture platform includes upgrades to provide more control of visual input and output features. This will allow for additional extensibility in terms of the gesture interactions we are able to implement and evaluate.

5. CONCLUSIONS

This paper has presented our extensible platform for researching ambient gestures for multimodal interactions using computer vision techniques. We have demonstrated that the platform provides a simple, flexible, cost-effective solution for designing and investigating gesture-based interactions in real-world, everyday scenarios for ubiquitous computing. The iGesture technology allows research into interaction techniques that would be intractable with other gesture recognition solutions. An evaluation of the recognition performance of the iGesture software, showed a recognition accuracy rate of above 90%.

Our current work allows us to explore gesture based user interactions independently of the underlying technology used to recognise and process the gestures. Our approach is to investigate a variety of interactions that use gestures to better understand the situations for which gestures are best suited in various multimodal and ubiquitous computing environments.

6. ACKNOWLEDGEMENTS

We are grateful to the EPSRC for funding this work. Special thanks go to Simon Goodall, Chris Bailey, Deborah Santa-Clara, Derya Sahin, Peter Appleby, Jendra Gosai, Wasara Rodhetbhai, Guillermo Power and Thanyalak Maneewatthana for helping with the evaluation. We would also like to thank the members of the IAM lab for their discussion.

7. REFERENCES

[1] R. Bowden, A. Zisserman, T. Kadir, and M. Brady. Vision based interpretation of natural sign languages. In *Exhibition at ICVS03, The 3rd Int. Conf. on Computer Vision Systems*, Graz, Austria, April 2003.

[2] G. Iannizzotto, M. Villari, and L. Vita. Hand tracking for human-computer interaction with graylevel visualglove: turning back to the simple way. In *Proc. of the 2001 workshop on Percetive user interfaces*, pages 1–7. ACM Press, 2001.

[3] M. Karam, J. Hare, m. c. schraefel, and P. Lewis. Ambient gestures. In *(Unpublished)*, 2004.

[4] Y.-W. Liu. Modeling music as markov chains - composer identification. Music 254 final report, Center for Computer Research in Music and Acoustics, Stanford University, 2002.

[5] S. Oviatt and P. Cohen. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Commun. ACM*, 43(3):45–53, 2000.

[6] F. Quek, D. McNeill, R. Bryll, S. Duncan, X.-F. Ma, C. Kirbas, K. E. McCullough, and R. Ansari. Multimodal human discourse: gesture and speech. *ACM Trans. Comput.-Hum. Interact.*, 9(3):171–193, 2002.