Phalanger: Controlling Music Software With Hand Movement Using A Computer Vision and Machine Learning Approach

Chris Kiefer
Department of Informatics
University of Sussex
Brighton, UK
c.kiefer@sussex.ac.uk

Nick Collins
Department of Informatics
University of Sussex
Brighton, UK
n.collins@sussex.ac.uk

Geraldine Fitzpatrick
Department of Informatics
University of Sussex
Brighton, UK
g.fitzpatrick@sussex.ac.uk

Abstract

Phalanger is a system which facilitates the control of music software with hand and finger motion, with the aim of creating a fluid style of interaction that promotes musicality. The system is purely video based, requires no wearables or accessories and uses affordable and accessible technology. It employs a neural network for background segmentation, a combination of imaging techniques for frame analysis, and a support vector machine (SVM) for recognition of hand positions. System evaluation showed the SVM to reliably differentiate between eight different classes. An initial formative user evaluation with ten musicians was carried out to help build a picture of how users responded to the system; this highlighted areas that need improvement and lent some insight into useful features for the next version.

1. Introduction

What musicians consider to be a musical instrument is something that varies continually with the arrival of new technologies. One expansion of this concept, compared to traditional acoustic instruments, is to consider a musical studio in itself as a musical instrument [15, 8]. Bertelsen et. al's [3] case study of two composers portrays in detail how this can occur in practice, observing how a musician can alternate between reflection on software as an object and software as an instrument, this happening rapidly and dynamically so as to blur the distinction between the two.

In considering the studio as a musical instrument, it would be ideal to be able to interact with it expressively and intuitively as one would with an acoustic instrument. As the studio becomes increasingly or completely condensed into the digital realm, for many the principal controllers for their tools are a mouse and keyboard, devices which cannot convey the subtlety and dynamism of musical interaction, and so limit musicality. This contributes to what Armstrong [2] labels the 'disconnect' between performers and digital instruments, describing a 'missing dimension' in musical experience due to the lack of potential for engaged and embodied interaction.

New technologies and research projects are looking at solving this problem of bringing the body into a deeper engagement with creative tools; multi-touch systems such as the Lemur [1] and tangible systems such as ReacTable [9] have shown considerable success. This paper describes Phalanger, a system that takes the approach of using computer vision techniques to facilitate control of music software with hand and finger motion.

2. Related Work

While many music related computer vision based projects, for example [11, 5], have focused on bodily gestures, this project focuses on smaller scale hand motion. A range of systems exist for finger and hand tracking using a variety of sensors and accessories; a small proportion of these, like Phalanger, rely purely on video data and use no accessories or wearables. These systems tend to employ a combination of computer vision analysis algorithms and machine learning techniques to extract information from a video source and translate it into control data. Zhou et. al's [16] Visual Mouse employs the Scale-invariant Feature Transform algorithm along with Principal Component Analysis to detect and track fingertips. Oka et. al. [12] use an infrared camera to track fingertips and fingertip gestures using a heuristic algorithm along with Hidden Markov Models. Premaratne and Nguyen [14] use moment invariants as input to a neural network to recognise hand positions for control of consumer electronics devices. A more advanced system has been built by Agarwal et. al [1], who use stereo cameras along with a SVM to detect fingertip location and to distinguish between touch and hover positions. Finally, in a musical context, Burns and Wanderley [4] have used the Circular Hough Transform to track a guitarist's fingers over frets. Phalanger combines features from these systems by focusing on broader scale hand motion as well as finger tip motion.

[1] http://www.jazzmutant.com

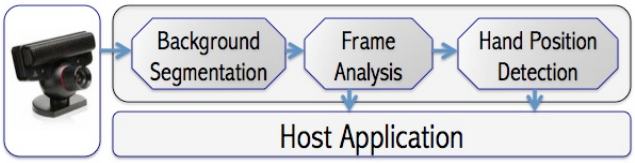

Figure 1. An Overview

3. Implementation

The use of hand movement to control software has strong potential for enhancing musicality in interaction. Hand movement is natural and direct, implying easy learnability. It can also convey subtle, complex and fluid movement, with a potential for virtuosity in interaction. The challenge is in how to detect and track the subtleties of movement in a robust and reliable way. Along with addressing this challenge, Phalanger had some more practical design aims. Firstly to design a system which would work without wearables such as markers or sensors on the hand, which can be inconvenient and may impede motion. Secondly, to design a system which would work on lower cost and accessible hardware. Phalanger was developed using the openFrameworks [2] C++ library, chosen for it's range of add-on libraries, cross-platform portability and speed. It also uses the openCV [3] computer vision library, along with Fann [4] and libSVM[6] for machine learning. The reference system here is a MacBook Pro 2.2GHz laptop. For the video source, good results were achieved with both a Sony DCR-TRV80E firewire camcorder and a low cost (£25) Sony PS3Eye USB camera at 320x240 resolution. The system works in three phases (see figure 1); firstly background segmentation, and then frame analysis with hand position recognition. It functions as a library which can be integrated within a host application.

3.1. Background Segmentation

The background segmentation process uses skin colour detection to separate the hand from the background. A neural network technique was chosen here, as opposed to employing a histogram method, so that the system would be dynamically configurable for each user's particular camera, room lighting and skin tone. Phalanger takes snapshots of the room without the user, and then of the front and back of the user's hand; these images are used as training examples for a back propagation network. Following from [10], the pixel values are converted from RGB to the YCbCr colour space; in this way the luminance value (Y) can be discarded, leaving the chrominance values as neural network inputs. This makes the algorithm more robust to lighting changes, and allows for a smaller network. The network architecture was determined experimentally, and consists of two input neurons, four hidden neurons with linear transfer functions and one output neuron with a sigmoid transfer function. In use, the trained network is run on every pixel of video data, separating the skin from the background as in figure 2.

[2] http://www.openframeworks.cc/
[3] http://sourceforge.net/projects/opencvlibrary/
[4] http://leenissen.dk/fann/

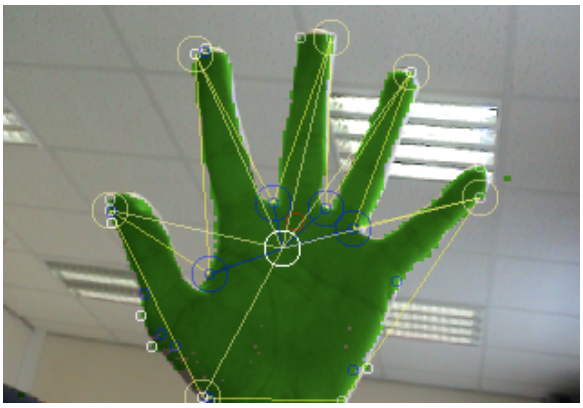

Figure 2. Frame Analysis.

Table 1. SVM inputs

6 furthest contour points from the centroid (normalised)
4 closest hull defects to the centroid (normalised)
angles between the far points
angles between the near points
first 6 Hu moments
the centroid (normalised)
height:width ratio of the hand blob rectangle

3.2. Frame Analysis

The next stage analyses the video data with imaging techniques. There's an extensive selection of algorithms available for image features analysis [7], the following were chosen with the aim of arriving at a fixed number of data points which could be passed to a machine learning process. The first stage is to take the grayscale output from the skin detector and smooth out the edges with erosion and dilation, before performing blob tracking to locate the rectangle enclosing the hand. An active contour algorithm is then used to find the hand shape, and this is simplified to an approximation with a reduced number of points. The contour is used to find the convex hull of the hand, from which the convexity defects can be derived. The approximated contour and convexity defects are sorted in relation to their distance from the blob centroid to obtain the six farthest points on the contour and the four nearest defects. Finally, the contour is analysed to find the first 6 Hu moment invariants. These outputs, summarised in table 1, are used both as inputs to the hand position recognition process and as tracking data for individual points on the hand.

3.3. Hand Position Detection

The hand position detection process is centred around a SVM, which observes the inputs from the frame analysis and attempts to predict the shape of the hand. Phalanger uses libSVM's *C_SVC* type SVM, with a radial basis function kernel configured with $C=2$ and $\gamma=0.2$. In training the system records observations, over a number of frames, of the

hand in one or more positions which represent a particular class. The amount of frames needed for training varies with the number of classes and the quality of the training data. In some cases Phalanger works reliably with 10 frames per class, but generally recording 100 or over yields better results. Under-training the system can result in undesirable jitter in the output of the SVM.

3.4. In Use

The system is embedded within a host application, which can use the combination of hand position class and frame analysis as control data. Based on data about hand shape, inferences can be made from the frame analysis about other parameters of motion. For example, if it is known that the hand is in a closed fist position with the index finger extended, it can be inferred that index finger tip position is the highest point in the contour.

4. Evaluation

There are two significant performance benchmarks: speed and recognition reliability. The speed of operation is currently between 14 and 18 frames per second. In terms of reliability, in testing the SVM could differentiate between up to 8 different hand positions with a cross-validation accuracy of over 96.5% when trained with 100 frames per class.

User experience was also tested. Phalanger is a work in progress so at this stage, informal 'formative' [13] evaluation seemed most appropriate, with the aim of getting initial feedback on which to base the next stages of development. Ten musicians were asked to try out Phalanger in three different scenarios (see figure 3), giving their feedback in semi-structured interviews which broadly focused on their experience of trying this style of interaction. The first scenario comprised a Tetris like sound game, where participants could knock falling blocks with their index finger to trigger different sounds depending on where the blocks landed, and also change hand position to hold the blocks from falling. The second application was an experiment in controlling sound with hand shape, the points and angles of a web drawn around the extremities of the hand directly controlling granular synthesis parameters. Lastly, the participants tried a sound mixing scenario where they could navigate across a set of virtual sliders by moving their hand in a parallel plane to the camera, and zoom in and out by moving their hand forwards and backwards. By changing to a grabbing position, they could change the level of the sliders. In all these scenarios, users controlled the software with their hand facing a camera pointing up from the table, their elbows resting on the table. The scenarios were designed to explore the range ways in which the system might be used in different musical contexts, testing mappings for discrete and continuous control, and both hand and fingertip motion. The interviews were analysed using a grounded theory approach.

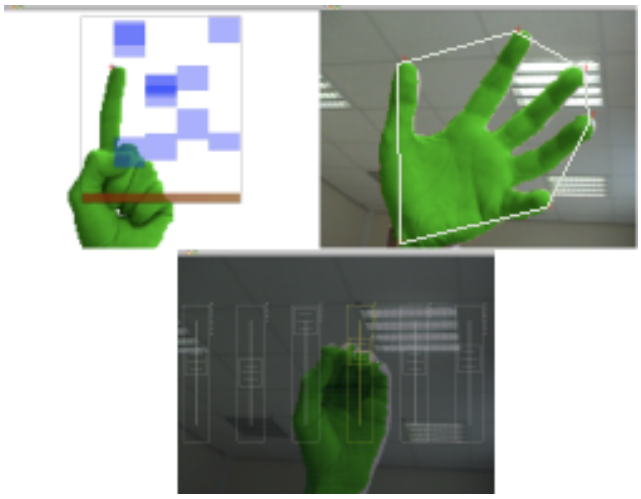

Figure 3. **Evaluation Scenarios. Clockwise from top-left: a Tetris style sound game, hand shape controlled granular synthesis, sound mixing**

4.1. Results

Responses fell broadly into four categories: control, feedback, ergonomics and learnability. In terms of control, reactions ranged from negative ('*I keep on moving things when I don't intend to*', '*it's a bit unpredictable*') to unsure ('*I felt in control to a certain extent, I don't think I could quite find the direct correlation*') to positive ('*it's got a really light kind of feel to it, I think I find it controllable*','*it's easy to control*'). Half of the interviewees felt that the system needed to be more responsive ('*it needs to be a bit faster somehow*', '*speed of response could be better*'). Precision was also an issue for some ('*fine control is difficult*'), leading to suggestions for creative uses which suited less precise control ('*I'd like to draw parametric EQ lines with hand movements whereas something like volume levels needs more precision*'). There was some comment about the mapping of hand motion to sound; one interviewee described how they would prefer discrete gestures to continuous control for navigating the mixer, another participant liked the way that hand motion in the granular synthesis scenario seemed to match the way the sound changed. As with the issue of controllability, there was a wide range of reactions about learnability. Some found the system difficult to adjust to ('*sometimes I get it and then sometimes I seem to lose what I'm doing*') while others picked it up successfully ('*it's taken me a little while to get used to it, I'm finding I can quite consistently get it to do what I want now*'). This was related to issues of familiarity with a style of interaction most had not tried before ('*the mouse is easier because I've used it before*', '*this is weird, I'm used to a mixing desk*'). There were some interesting results concerning visual feedback which again showed the range of preference between the participants. Phalanger shows the user's hand on the screen in three variations: just the hand, the hand with markers from

the frame analysis, and the markers on their own. All three modes were preferred by different participants, and in some cases their choice of visual feedback made a significant difference to their ability to control the software. In the granular synthesis scenario, one participant felt comfortable with the screen turned blank. Some also commented on physical feedback (‘*there's nothing to resist your movement, I don't know when I should stop my hand*’, ‘*there's nothing in terms of feedback, I'm not pushing anything*’). Finally, ergonomics was a strong theme. There were some comments about hand movement which didn't seem natural (‘*waving your finger like this isn't the most natural thing*’,‘*I find it harder to go that way left to right*’). Fatigue was also an issue, participants fingers, arms or shoulders tiring after a while of using the system.

5. Discussion

The evaluation results highlight some areas where improvements need to be made to the system. Responsiveness is a key issue; at the moment the program runs as a single-threaded process, upgrading this to run multi-threaded should provide improvement in speed. Another option is to run some operations using GPU stream processing with a system such as CUDA [5] . Increased speed should also allow an increase in video resolution and therefore increased control precision. The ergonomics results show that the hand position used in the evaluation could be tiring, a better solution would be to use an overhead camera with the hand resting on a horizontal surface. The results also show how some hand and finger movements can seem unnatural, something to be kept in mind when specifying control movements with this system. Of particular interest was the range of user reactions, some feeling instantly comfortable using the system and some finding it harder to use; this needs to be investigated in more detail.

6. Conclusion and Future Work

Phalanger has reached its initial objectives of providing a system through which users can control music software with hand motion. The evaluation results have highlighted some areas which need attention, given insights into new design features which would improve operation, and helped to build a picture of how musicians respond to this ‘digital’ style of interaction. The next phase of development is to implement these improvements and conduct a more detailed longitudinal evaluation of the system in comparison to mouse and keyboard operation of musical applications.

References

[1] Ankur Agarwal, Shahram Izadi, Manmohan Chandraker, and Andrew Blake. High precision multi-touch sensing on surfaces using overhead cameras. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer System*, 2007.

[2] Newton Armstrong. *An Enactive Approach to Digital Musical Instrument Design*. PhD thesis, Princeton University, 2006.

[3] Olav W. Bertelsen, Morten Breinbjerg, and Søren Pold. Instrumentness for creativity mediation, materiality & metonymy. In *C&&C '07: Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, pages 233–242, New York, NY, USA, 2007. ACM.

[4] Anne-Marie Burns and Marcelo M. Wanderley. Visual methods for the retrieval of guitarist fingering. In *NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression*, pages 196–199, Paris, France, France, 2006. IRCAM — Centre Pompidou.

[5] Antonio Camurri, Paolo Coletta, Giovanna Varni, and Simone Ghisio. Developing multimodal interactive systems with eyesweb xmi. In *NIME '07: Proceedings of the 7th international conference on New interfaces for musical expression*, pages 305–308, New York, NY, USA, 2007. ACM.

[6] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] David Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[8] Don Ihde. Technologies–musics–embodiments. *Janus Head*, 2007.

[9] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, New York, NY, USA, 2007. ACM.

[10] Aamer Mohamed, Ying Weng, Jianmin Jiang, and Stan Ipson. Face detection based neural networks using robust skin color segmentation. *Systems, Signals and Devices, 2008. IEEE SSD 2008. 5th International Multi-Conference on*, pages 1–5, July 2008.

[11] Kia Ng and Paolo Nesi. i-maestro framework and interactive multimedia tools for technology-enhanced learning and teaching for music. *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS '08. International Conference on*, pages 266–269, Nov. 2008.

[12] K. Oka, Y. Sato, and H. Koike. Real-time fingertip tracking and gesture recognition. *IEEE Computer Graphics and Applications*, 22(6):64–71, 2002.

[13] Jennifer Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design: Beyond Human Computer Interaction*. Wiley, 2002.

[14] P. Premaratne and Q. Nguyen. Consumer electronics control system based on hand gesture moment invariants. *Computer Vision, IET*, 1(1):35–41, March 2007.

[15] David Toop. Replicant: On dub. In *Audio Culture: Readings In Modern Music*. Continuum Books, 2002.

[16] Hailing Zhou, Lijun Xie, and Xuliang Fang. Visual mouse: Sift detection and pca recognition. In *International Conference on Computational Intelligence and Security*, 2007.

[5] `http://www.nvidia.com/cuda`