An Efficient Application of Gesture Recognition from a 2D Camera for Rehabilitation of Patients with Impaired Dexterity

G. Ushaw[1], E. Ziogas[1], J. Eyre[2] and G. Morgan[1]
[1]*School of Computing Science, Newcastle University, Newcastle upon Tyne, U.K.*
[2]*Department of Child Health, Royal Victoria Infirmary, Newcastle upon Tyne, U.K.*

Keywords: Human-machine Interfaces for Disabled Persons, Image Processing and Computer Vision.

Abstract: An efficient method for utilising a 2D camera to recognise hand gestures in 3D space is described. The work is presented within the context of a recuperation aid for younger children with impaired movement of the upper limbs on a standard Android tablet device. The hand movement recognition is achieved through attaching brightly coloured models to the child's fingers, providing easily trackable elements of the image. The application promotes repeated use of specific hand skills identified by the medical profession to stimulate and assess rehabilitation of patients with impaired upper limb dexterity.

1 INTRODUCTION

In this paper we describe an implementation which recognises specific finger movements with one or two hands, and we present results that show this solution is viable on typical tablet hardware. This is achieved through attaching brightly coloured models to the subject's fingers and tracking those bright colours as they move.

The work was developed within the context of an interactive storybook, consisting of a series of mini-games encouraging a child with impaired upper limb dexterity to perform specific hand movements. Medical practitioners have identified a framework of specific hand movements which are optimum for both assessment and intervention (Chien et al., 2009). The application achieves the twin goals of running on the kind of low power hardware typically found in a family home, and encouraging the child to practice the rehabilitative movements within the context of a fun game rather than a boring chore.

2 RELATED WORK

The detection of movement in a sequential set of images is of widespread interest and the solutions are well understood for a static camera image (Radke et al., 2005). Real time applications may require a rapid response to an image change. The image data itself consists of a two-dimensional array of pixels, each containing an n-dimensional vector of values corresponding to an intensity or colour at that position in the image; this recorded data will get very large very quickly in some applications.

The goal in general, is to identify a set of pixels in an image which differ significantly from one image to the next. These pixels are known as the change mask. The algorithms become more complex, and more application-specific, when attempting to classify the change masks within the semantics and context of a particular problem - this is often labelled change understanding. Change understanding requires the system to be able to reject unimportant changes, while identifying relevant significant changes - this usually involves some prior modelling of the kind of expected changes that can occur in the image (both significant, and for rejection). A variety of methods are used to filter out expected insignificant changes, typically taking into account lighting changes and camera movement. When the camera movement is small, the techniques for accounting for it, using low-dimensional spatial transformations, are well-understood (Zitova and Flusser, 2003). Discarding changes in the image caused by inconsistent lighting is also well-researched (Lillestrand, 1972). It is also commonplace to transform the images into a different intensity space before carrying out the motion detection algorithms.

Once the pre-processing is complete, the search for changes in the image can commence. The earli-

Ushaw G., Ziogas E., Eyre J. and Morgan G..
An Efficient Application of Gesture Recognition from a 2D Camera for Rehabilitation of Patients with Impaired Dexterity.
DOI: 10.5220/0004190103150318
In *Proceedings of the International Conference on Health Informatics* (HEALTHINF-2013), pages 315-318
ISBN: 978-989-8565-37-2

est technique was a straightforward summation of the number of pixels comprising the change mask, with a threshold value for what constitutes a significant change (Rosin, 2002). Various techniques have been employed to better define how the threshold is chosen, but this simple differencing approach is unlikely to provide as reliable results as later developments; in particular, it is sensitive to noise and lighting variations (Lillestrand, 1972). Further development has centred on significance and hypothesis testing, and on predictive models, both spatial and temporal.

2.1 Gesture Recognition

Gestures are ambiguous and incompletely specified, as they vary from one person to the next, and each time a particular person gesticulates. Consequently, the two main issues to resolve when recognising gestures are to identify specific elements of the gesture, and to have some prior knowledge of which gestures to search for. Gesture recognition is achieved by either attaching a sensor of some type to various parts of the body, or from interpreting the image from a camera. There is an inherent loss of information in interpreting the 2D image of a 3D space, and algorithms which address this can be computationally expensive.

Identifying a hand gesture involves determining the point in time when a gesture has started and ended, within a continuous movement stream from the hands, and then segmenting that time into recognizable movements or positions. This is not a trivial problem due to both the spatio-temporal variability involved, and the segmentation ambiguity of identifying specific elements of the gesture (Mitra and Acharya, 2007).

The use of Hidden Markov Models (HMM) has yielded good results in gesture recognition, as gestures consist of a set of discrete segments of movement or position (Yamato et al., 1992). Sign language recognition processes have been designed and implemented using HMM's (Starner and Pentland, 1996). In the cited implementation, the user wore coloured gloves, and the approach required extensive training sets; it successfully recognized around fifty words within a heavily constrained grammar set. The HMM approach has been further developed, splitting each gestures into a series of constituent ”visemes” (Bowden et al., 2004).

Gestures can also be modelled as ordered sequences of spatio-temporal states, leading to the use of a Finite State Machine (FSM) to detect them (Hong et al., 2000). In this approach each gesture is described as an ordered sequence of states, defined by the spatial clustering, and temporal alignment of the points of the hands or fingers. The states typically consist of a static start position, smooth motion to the end position, a static end position, and smooth motion back to the rest position. This approach is less suited to detecting motion in small children (who tend not to be static), and especially not those with impaired movement ability.

2.2 Low Power Device Considerations

The key to gesture recognition in a two-dimensional image is in identifying the parts of the image which are relevant to the gesture, and monitoring their contribution to the change mask. The quicker that the elements of the change mask that are unrelated to the gesture can be discarded, the more time the algorithms have to process the gesture data.

Further to this, the smaller the amount of data that is used to represent the change set, the faster the algorithms for analysing that change are likely to be. Most image capture methods used in gesture recognition retain some sense of the overall image, or sections of it, during analysis of the change set - for example, tracing the movement of an edge between a section of the image which is skin tone coloured, and a section which is not.

A significant saving in computing power is also made if the application has prior knowledge of which gesture(s) it is searching for. If the algorithms need to check for any gesture at any time, then this is drastically more computationally expensive than attempting to detect a specific gesture at a particular instant in time.

3 IMPLEMENTATION

The implementation which is described addresses the potentially large amount of processing power required in recognizing hand gestures in three ways.

Brightly coloured models are attached to the subject's fingertips. This means that the image processing software only needs to identify areas of specific, pre-determined colours in the real-time moving image. Further to this, the areas of specific colour are reduced to a single coordinate per frame within the two dimensional screen-space, which greatly speeds up the gesture recognition process.

Each gesture which must be identified has been designed to require tracking of no more than three fingertips. This reduces the amount of data tracked from frame to frame which again allows the algorithms to perform on the lower power target device.

The application is designed so that at any time it is only searching for one specific gesture.

3.1 The Colour Space

The aim of the colour processing algorithm is to identify a single point in the image for each colour that is being tracked. The application includes some configuration and calibration routines to ensure that the colours of the models on the fingers are identified and are sufficiently distinct from the rest of the image.

The Android device records the colour image in a NV21-encoded YC_rC_b format. The first step is to convert this to ARGB format, so it can be stored in an OpenCV IplImage, for further processing. It is then straightforward to convert to the desired CIE La^*b^* colour space via RGB. For each colour of interest, a binary image is constructed representing the presence of that colour at that pixel in the image.

Library functions are then utilised to dilate and erode the resulting binary image, for each colour of interest, and further library functions are used to identify the contours around the resultant shapes. If the number of pixels within a contour is higher than a threshold, then the points are averaged to give a gravity centre for that colour. Scaling the results according to the window coordinates gives one point per target colour in the 2D camera coordinate system.

3.2 The Gestures

The gestures are based on a framework of children's hand skills for assessment and intervention (Chien et al., 2009), including unimanual skills, individual finger movements, and bimanual gestures.

Bespoke algorithms have been developed to recognise the specific movements of the coloured finger tips for each type of gesture. The algorithms are based on interpreting the movement of gravity centres as they change from frame to frame in the colour image. A maximum of two gravity centres is required for each gesture.

The **Pinch-Grasp** move involves bringing together the forefinger and thumb. Two colours are tracked, and the pinch is identified when the distance between them reduces below a threshold value. A release is identified when the distance increases over a greater threshold, to ensure there is hysteresis in the algorithm.

The **Power-Grasp** move involves clenching the fist. Two colours are tracked (on thumb and little finger) and the grasp is identified when the distance between them reduces below a threshold value.

Supination and pronation involve rotating the wrist, so the palm goes from facing down to facing up, and back again. Two colours are tracked (on thumb and little finger), the vector between them is calculated and compared from one frame to the next.

Wrist flexion and extension involves rotating the wrist vertically, so the hand moves up and down. Two colours are tracked (on thumb and little finger), their relative position vertically is calculated. If they invert their relative vertical position, while both moving in the same vertical direction, then flexion/inflexion is detected.

As the underlying application of the work is in monitoring rehabilitation of patients with movement difficulties, the algorithms are designed to detect sequences of movement, rather than specific ”posed” hand shapes. In conventional movement detection algorithms, this would entail identifying change masks between successive images, and carrying out costly computation. In this implementation, the gesture recognition is based on the movement of a set of coordinates in two dimensional screen-space, identified as the gravity centres of the coloured regions of interest. This significantly speeds up the gesture recognition process enabling it to be implemented on the target lower power tablet devices.

4 RESULTS AND EVALUATION

The algorithms described in this paper have been successfully implemented within an Android application for tablets. The camera resolution of the minimum specification device was 176x144 pixels, recording images at around 15 frames per second.

The video in (Ziogas et al., 2012) shows the application of this technology within the interactive storytelling software. The brightly coloured finger models are constructed from Play-Doh, and can be any colour. In the video, after using the touch-screen to progress through the early stages, the Pinch-Grasp gesture is detected, and used to move a ribbon onto a bone. Wrist flexion and extension are then recognised in the section which opens the door. The final section shows the detection of supination and pronation, which is used to turn a tap in the story.

As the algorithms are designed for low performance devices, tests were carried out at a series of decreasing sample rates to assess the robustness of the solution. Each of the gestures was tested five times by the subject at the maximum sample rate of 30 frames per second. During each test the gesture was repeated ten times, and the number of positive identifications was logged. The information resulting from colour segmentation was recorded for every frame of the test and subsequently subjected to down-sampling to investigate how many of the ten gestures could be identified as the sampling rate reduces. For each of the four gestures the number of positive identifications in each of the test samples were averaged and the results for two of them are depicted in Figures 1 and 2.

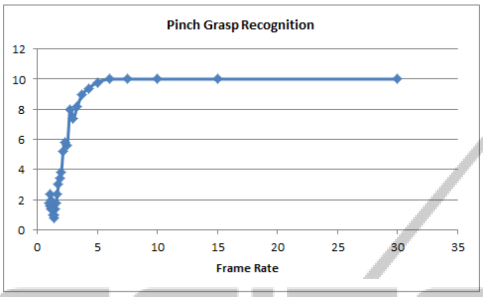

Figure 1: Average pinch grasp recognition.

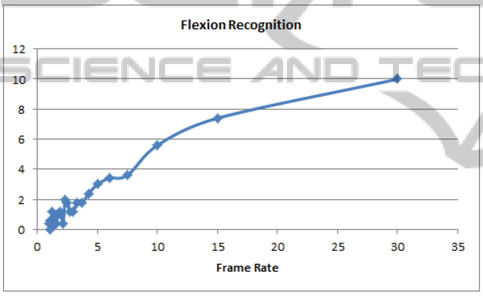

Figure 2: Average flexion recognition.

The algorithms for detecting three of the four gestures perform very well, down to low sample rates of around four frames per second (pinch grasp, power grasp and pronation/suppination). The performance of the flexion/extension gesture recognition algorithm tails off much more linearly. These results have an interesting ramification on the application as a whole as they suggest that the camera and sampling algorithms can be updated considerably less often than the game logic and rendering algorithms, which will lead to a more immersive experience for the child.

5 CONCLUSIONS

Recognition of hand gestures is possible on a low powered device such as an Android tablet. The two key steps which have been taken to mitigate the potentially computationally expensive nature of gesture recognition are to reduce the detection requirements to brightly coloured finger models, and to require that the application has advance notice of which specific gesture to search for.

The use of coloured models allows the early stage of the image detection software to reduce the problem to that of tracking points within the two-dimensional screen-space. The gesture recognition can be constructed around straightforward algorithms for analysing the movement of these points, greatly reducing the computational overhead compared to analysing change masks within full images. This increase in efficiency comes at the cost of accuracy; the specific application of this technology requires us to know *a priori* which gesture we are searching for.

The test results show that the algorithms are very robust to lower frame-rates for most of the gestures. Indeed the results strongly suggest that the limiting factor for the interactive storybook application on low power devices is the presentation of the game itself, as the gesture recognition algorithms continue to perform satisfactorily at a much lower sample rate than the frame-rate required for an immersive experience.

REFERENCES

Bowden, R., Windridge, D., Kadir, T., Zisserman, A., and Bradyi, M. (2004). A linguistic feature vector for the visual interpretation of sign language. *Proc. 8th Eur. Conf. Comput. Vis.*

Chien, C. W., Brown, T., and McDonaldi, R. (2009). A framework of children's hand skills for assessment and intervention. *Child care health and development.*, (35):873–884.

Hong, P., Turk, M., and Huangi, T. (2000). Gesture modeling and recognition using finite state machinesl. *Proc. 4th IEEE Int. Conf. Autom. Face Gesture recogn.*

Lillestrand, R. (1972). Techniques for change detection. *IEEE Trans. Comput*, 21(7):654–659.

Mitra, S. and Acharya, T. (2007). Gesture recognition: a survey. *IEEE Transactions on Systems, Man and Cybernetics*, 37(3):311–324.

Radke, R. J., Andra, S., Al-Kofah, O., and Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):394–307.

Rosin, P. (2002). Thresholding for change detection. *Comput. Vis. Image Understanding*, 86(2):79–95.

Starner, T. and Pentland, A. (1996). Real-time american sign language recognition from video using hidden markov models.

Yamato, J., Ohya, J., and Ishii, K. (1992). Recognizing human action in time sequential images using hidden markov model. *Proc. IEEE Int. Conf. Comput. Vis. Pattern recogn.*, pages 379–385.

Ziogas, E., Ushaw, G., Eyre, J., and Morgan, G. (2012). http://homepages.cs.ncl.ac.uk/2010-11/games/tyney/videos/.

Zitova, B. and Flusser, J. (2003). Image registration methods: a survey. *Image Vis. Comput*, 21:9771000.