Mixed Reality Kinect Mirror Box for Stroke Rehabilitation

Craig Robertson
Dept. of Elec. Comp. Eng.
University of Auckland
Auckland, New Zealand
crob139@aucklanduni.ac.nz

Liam Vink
Dept. of Elec. Comp. Eng.
University of Auckland
Auckland, New Zealand
liamvink@gmail.com

Holger Regenbrecht
Dept. of Information Science
University of Otago
Dunedin, New Zealand
holger.regenbrecht@otago.ac.nz

Christof Lutteroth
Dept. of Computer Science
University of Auckland
Auckland, New Zealand
lutteroth@cs.auckland.ac.nz

Burkhard C. Wünsche
Dept. of Computer Science
University of Auckland
Auckland, New Zealand
burkhard@cs.auckland.ac.nz

ABSTRACT

Mirror Box therapy is used for rehabilitating stroke patients who have lost control of a hand. The therapy uses mirrors to give patients the illusion that movements of their healthy hand are movements of their impaired hand. The goal of this stimulation is to enable the healthy part of the brain to learn to control the impaired hand. In this paper we present a mirror therapy tool developed using the Kinect motion sensor. We evaluate five different finger tracking SDKs using a controlled environment resembling a clinical setup. The strengths and weaknesses of each SDK are discussed and compared in a trade-off matrix which attempts to quantify the individual performances. We present some algorithmic improvements to hand and finger tracking, explain how this information is used to control a skeletally animated hand, and present a simple game to make exercises more enjoyable. We evaluate our application and identify shortcomings of the presented technique.

Categories and Subject Descriptors

I.3.6 [**Methodology and Techniques**]: Interaction techniques; I.3.8 [**Computer Graphics**]: Applications; K.8 [**Personal Computing**]: Games

Keywords

Kinect, RGB-D camera, finger tracking, stroke rehabilitation

1. INTRODUCTION

A stroke is a traumatic event that results in the death of a localised area of brain cells. This can have significant long term effects for victims which often include hemiparesis, the weakness of muscles on one side of the body [11]. Patients undergo extensive rehabilitation in order to regain some control of the affected limbs. One such technique used for rehabilitation is mirror therapy, in which the patient places their affected limb on one side of a mirror and their unaffected limb on the other side. The patient then performs certain exercises with the unaffected limb while looking into the mirror from the same side. The intention is that the patient will perceive the limb shown in the mirror as their affected limb and "trick" the brain into believing that it is, in fact, the stroke affected limb performing the movements. Mirror box therapy has also been proven to help alleviate the discomfort experienced by patients suffering from phantom limb pain [3].

In this article we study the feasibility and practicality of using the Microsoft Kinect RGB-D camera to implement mirror box therapy in a mixed reality environment. Section 2 reviews related work. Section 3 presents our requirements and an evaluation of finger tracking SDKs. Section 4 presents our solution, which we evaluate in section 5. We conclude the paper and present an outlook on future work in section 6.

2. LITERATURE REVIEW

2.1 Mirror Box Therapy

Mirror Box therapy, which involves using a reflection of a healthy limb to simulate movement of an impaired limb, has been tested and trialled with very positive outcomes. One of the most important aspects of mirror therapy is the level of patient illusion; that is, the degree to which the patient believes that the observed mirror image is their stroke affected limb [1]. A recent study at the University of Otago in Dunedin concluded that their Augmented Reality (AR)

Mirror Box using web cams was perceptually more believable than the traditional mirror therapy technique [12].

2.2 Hand Tracking

A large variety of hand tracking algorithms has been proposed. A good survey is given in [8]. Marker-based hand tracking algorithms require the user to wear point or area markers, such as reflective spheres, LED-gloves [9], or coloured gloves [15]. Marker-based tracking applications can provide highly accurate results, but the need for auxiliary devices (markers, gloves) can be inconvenient for the user and often requires some type of calibration.

Without markers the easiest way to identify (potential) hand shapes is by using a skin colour classifier [6]. Once a (potential) hand shape has been identified based on skin colour it must be verified and its 3D position and orientation must be determined. A popular way to achieve this is by using a 3D hand model and searching for a mapping between it and the perceived hand shape subject to the model's inherent constraints (e.g., joint constraints and rigidity of bones) [14, 4, 7].

Tracking can be simplified by using depth information. Multi-camera vision and stereographic systems have been used successfully [2], but are expensive and impractical for patient home use. We chose instead the Microsoft Kinect motion sensor due to its low price and high popularity and availability. The Kinect combines a regular RGB video camera with an infrared (IR) light emitter paired with an IR light receiver.

After we started our project the Leap Motion Controller for gesture recognition was released. The technology looks promising and we look forward to investigate its performance within a small enclosed environments as used for mirror therapy. Another technology we considered is the 3Gear SDK, which uses the colour glove technology from Wang and Popović [15]. We want to avoid the need for markers and there are also doubts whether the hand and finger tracking will work inside a "Mirror Box".

3. TECHNOLOGY EVALUATION

3.1 Requirements

The patient is the primary user, with the therapist a close second. Our focus is on usability and effectiveness. We try to simulate the experience of using an optical Mirror Box as many therapist and users will have experience with such devices. Two thirds of stroke sufferers are over the age of 65, which means that ease-of-use and a simple uncluttered display are very important. Taking these issues into account and information obtained from collaborators and medical users we derived the following requirements:

Patient/Therapist Requirements

1. User friendly
2. Portable, designed with home use in mind
3. Suitable computer system requirements
4. Easy to install
5. Similar experience to traditional Mirror Box
6. Easy user interface
7. Quick Program Response
8. Believable
9. The ability to perform a reasonable set of normal rehabilitation exercises. We restrict ourselves to finger exercises involving a flat hand, e.g., bending the fingers as if playing piano
10. Ability to mirror and hide hands

Project Specific Requirements

1. Cost effective
2. Single Platform
3. Prefer 3D hand model rendering
4. Use Microsoft Kinect
5. Some form of game interaction

Best Practice Requirements

1. Future Development friendly
2. High Frame Rate
3. Developer Community Support
4. Continued driver support
5. Encourage user engagement

3.2 Set-Up

In order to simulate an optical Mirror Box we attach two $30cm^3$ boxes side by side. The wall between the boxes prevents patients from touching their own hands, which would compromise the illusion of the impaired hand moving. The Kinect is mounted about $90cm$ above ground level on a piece of wood attached to the back of both boxes. An LED light strip is fixed to the inner rim of the top of the boxes. The boxes are open at the front and the top. Figure 1 illustrates our set-up. While the Kinect offers a "near range" mode we found that even with suitable lenses finger tracking in the near range mode is worse than with the "default mode".

3.3 Evaluation of Finger Tracking SDKs

We investigated Kinect-based hand and finger tracking software and identified five SDKs, which seemed robust enough for our application and provided a free license. Four of the five SDKs make use of the OpenNI API and the OpenCV libraries: The PrimeSense SDK, OpenNI, SimpleNI and Frantracer. The final SDK tested is part of Microsoft's Kinect for Windows package.

We tested the SDKs using the set-up illustrated in figure 1. Tracking performance using the boxes was consistently worse than for an open room set-up where the hand is far way from any background objects.

The Frantracer SDK is implemented in $C\#$ using Microsoft Visual Studio and performed well for simple, slow gestures but failed to track fingers well when the hand was moved at faster speeds. It struggled when the hand was clenched in a fist as it continued to attempt to locate the fingertips. This resulted in any sharp points remaining in the depth image around the hand being represented as finger tips. The source code was not well commented, but well structured and easy to modify. The SDK offers many different variables to adjust tracking performance and computational time.

The PrimeSense SDK offers very robust hand and finger tracking, but is computationally demanding. We found that tracking worked well on computers with good graphics card and at least 8 GByte main memory, but the tracking performance was insufficient on normal lab computers.

The OpenNI SDK's hand and finger tracking performance was similar to Frantracer, but had a lower frame rate and stability.

The Simple OpenNI API is a wrapper for accessing OpenNI functionality through the "Processing" development environment [10]. We found that the provided finger tracking code performed well and had low resource requirements. However, we were unable to integrate the code into a C/C++ or Java GUI and we were unable to access the finger tracking raw data.

Figure 2 summarises the results of our evaluation.

4. DESIGN

4.1 Finger Tracking

Our evaluation shows that none of the reviewed SDKs is suitable in its current form, but the Frantracer framework provides a good compromise between tracking performance, resource requirements, and extendability. In order to use the framework in our application several modifications were implemented:

- We found that the depth data contains many erroneous values, possibly due to interference from the boxe's side walls and the shiny surface of the laminated wood. We eliminate small scale noise by performing morphological filtering and then applying a weighted moving average filter using the last 5 frames, where the most recent frame has the highest weighting. Figure 3 shows the improved performance of Frantracer when using the processed Kinect raw data.

- Using our set-up we know that a depth value must represent either the hand or the floor of the box. We assume that the hand does not touch the sidewalls of the box. By applying a binary classification to the depth image we can create a mask identifying the hand region in the RGB image. The palm center is defined by the centre point of the largest circle fitting into that region. This provides a more stable palm center estimate than when using the Frantracer framework.

- The finger tips are identified by tracing the hand contours and finding the points with the largest distance to the palm centre and convex curvature. The finger tip is the center point of the convex region. Note that if the finger is bend this method still finds the most outward joint or knuckles of the bend fingers.

- We identify finger joints by using the fact that the length ratio between finger sections is approximately 1 : 1.6 [5]. If the finger length is smaller than the palm diameter, then we assume that all finger joints are bend simultaneously. This relationship is motivated by the observation that the distal-interphalangeal (DIP)

Figure 1: The Kinect Mirror Box.

Criterion	Fran Tracer	Prime-Sense	OpenNI	Microsoft	SimpleNI
Ability to Interface	5	1	2	4	4
Stability	5	4	5	5	5
Tracking Performance	3	4	5	4	3
Frame Rate	5	2	4	4	5
Code Readability	4	3	4	4	2
Algorithms	1	2	3	3	3
User Experience	4	3	4	3	4
Hardware Setup	5	1	3	4	4
System Requirements	5	2	3	2	5
Precision	2	5	3	4	3
Resolution	2	4	4	4	3
Complexity	3	2	5	3	4
Range	4	1	3	3	4
Cross-platform Performance	5	5	5	1	2
Cross-platform Compatibility	5	2	4	4	4
Release Date	2	2	4	4	4
FINAL SCORES	530	365	510	504	512

Figure 2: Evaluation of finger tracking SDKs.

joint angle is two-third of the proximal-interphalangeal (PIP) joint angle [13].

The above algorithm yields the 5 finger tips, 14 finger joints, and the palm centre. For all of these points we have 3D locations. If the finger is bend depth values are estimated based on the above bone length and joint angle constraints. For most users, especially elderly with limited joint mobility, this provides an acceptable solution.

4.2 Skeletal Animation

A hand model in Collada format and code for rendering and animating it was obtained from the site `www.blendswap.com`. The model is fully featured with textures and skeleton for skeletal animation. We use the ASSIMP library for importing Collada models. So far we use pre-animation for finger motions, although in future we want to compute joint positions and motions using the 3D point data obtained in the previous step.

A GUI was developed for setting relevant parameters (permissible hand depth range and box dimensions) and for performing mirroring operations (see figure 4).

4.3 Interactive Game

We created a simple interactive music game for finger exercises, inspired by the popular "Dance Dance Revolution" arcade video game. The bottom of the box is represented by a chessboard as illustrated in figure 5 and the hand model is inserted into it. Yellow arrows are placed on the back wall and blue arrows move up the screen. The user must press the coloured on the chessboard pattern when yellow and blue arrows overlap. The exercise trains finger flexion and extension and molds the hand in shapes similar to grasping a ball or sphere, a popular exercise for stroke rehabilitation.

5. RESULTS

Our evaluation of finger tracking SDKs suggests that Frantracer is most suitable for our application. Tracking one hand resulted in a time delay of about 20 ms, which we regard acceptable. After implementing our improvements the frame rate on a mid-range computer without graphics card drops to about 10-15 frames per second, which is barely tolerable.

One of the biggest problem with all hand tracking algorithms is noise and the lack of stability, e.g., of palm and finger tip detection. We were able to improve stability by using morphological and low pass filtering, and exploiting domain knowledge (box dimensions).

The hand model is currently animated using pre-baked animations for each finger, which are placed and triggered

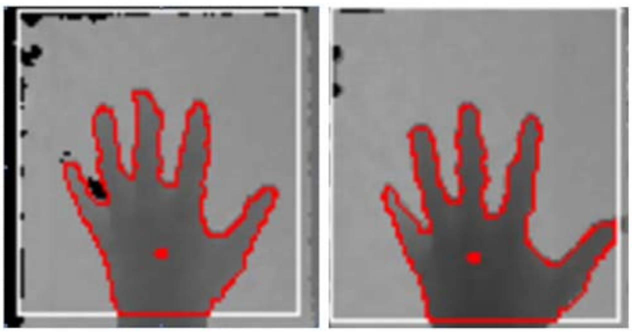

Figure 3: The hand tracking results using Frantracer before (left) and after (right) processing the Kinect raw data. The unidentified black region between fingers is successfully removed.

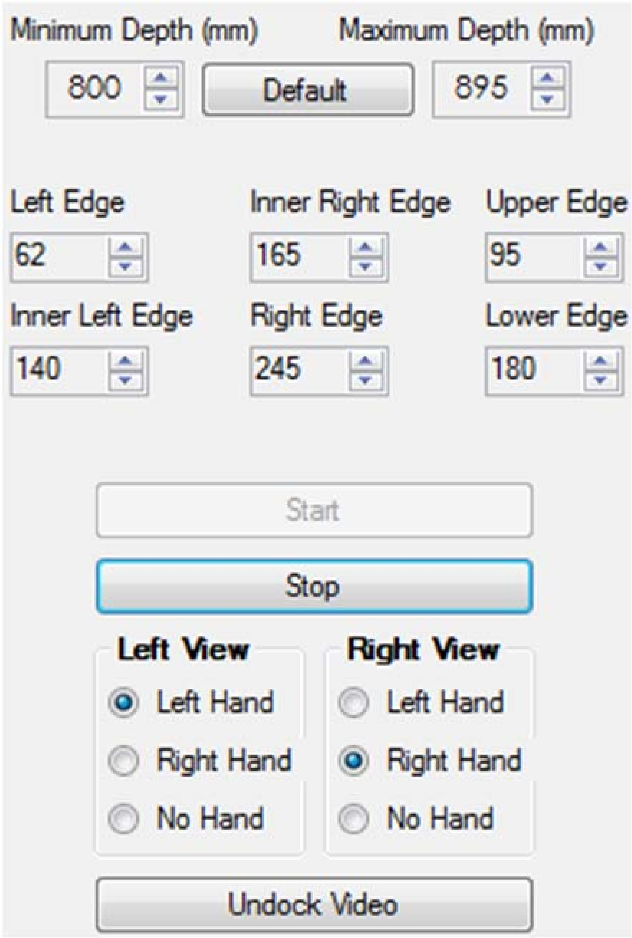

Figure 4: The control panel for the Kinect Mirror Box.

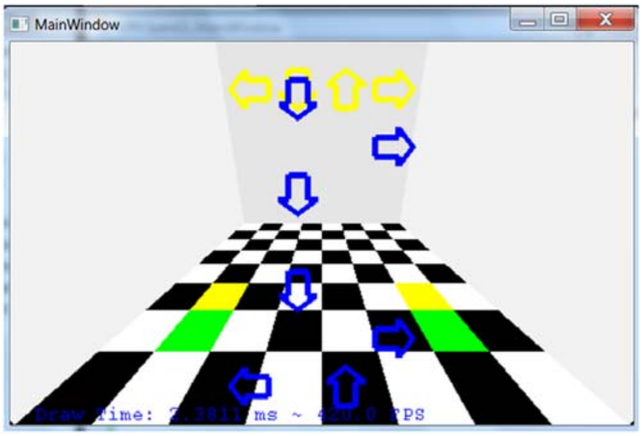

Figure 5: Kinect Mirror Box rhythm game (with the rendered hand removed).

using the finger tips and palm center. The quality of the animation is unsatisfactory, since it does not provide a good sense of immersion.

We presented a simple rhythm game to make finger exercises more interesting. Informal testing during a public exhibition indicated that users liked the idea and found the application intuitive and easy to use. However, current tracking performance (quality and range of finger motions detected), and in particular realism of the hand model (immersion) are not satisfactory yet.

6. CONCLUSION AND FUTURE WORK

We have presented a prototype of a Kinect Mirror Box for stroke rehabilitation. Our research indicates that consumer-level motion tracking sensors and mixed reality environments can make mirror box therapy more accessible and interesting. A further advantage is that performance data can be collected and (e.g., practice duration) and used by medical professionals.

Unfortunately the hand tracking performance obtained with the Kinect was not satisfactory and different motion sensors need to be evaluated in the future. Furthermore our hand animation and rendering is too simplistic and unsuitable to trick users' brain into believing that their impaired hand was moved. However, the algorithms for realistic skin rendering and skeletal animation are well known, and hence we believe that this problem can be easily overcome.

In future work we will test different consumer-level motion tracking technologies, improve the rendering and animation of the hand model, and perform a more extensive user study.

7. REFERENCES

[1] E. L. Altschuler, S. B. Wisdom, L. Stone, C. Foster, D. Galasko, D. M. E. Llewellyn, and V. S. Ramachandran. Rehabilitation of hemiparesis after stroke with a mirror. *The Lancet*, 353(9196):2035–2036, June 1999.

[2] A. A. Argyros and M. I. A. Lourakis. Binocular hand tracking and reconstruction based on 2d shape matching. In *Proc. International Conference on Pattern Recognition (ICPR '06)*, pages 207–210, 2006.

[3] B. L. Chan, R. Witt, A. P. Charrow, A. Magee, R. Howard, P. F. Pasquina, K. M. Heilman, and J. W. Tsao. Mirror Therapy for Phantom Limb Pain. *The New England Journal of Medicine*, 357(21):2206–2207, Nov. 2007.

[4] Q. Chen, N. Georganas, and E. Petriu. Real-time vision-based hand gesture recognition using haar-like features. In *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC 2007)*, pages 1–6, May 2007.

[5] R. A. Dunlap. *The golden ratio and Fibonacci numbers.* World Scientific Publishing Company, 1997.

[6] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recogn.*, 40(3):1106–1122, 2007.

[7] R. Liu, B. Wünsche, C. Lutteroth, and P. Delmas. A framework for webcam-based hand rehabilitation exercises. In *Proceedings of VISAPP 2011*, pages 626–631, 2011.

[8] F. Mahmoudi and M. Parviz. Visual hand tracking algorithms. *Geometric Modeling and Imaging–New Trends*, 0:228–232, 2006.

[9] J. Park and Y.-L. Yoon. LED-glove based interactions in multi-modal displays for teleconferencing. *International Conference on Artificial Reality and Telexistence*, pages 395–399, 2006.

[10] C. Reas and B. Fry. *Processing: A Programming Handbook for Visual Designers and Artists.* MIT Press, 2007.

[11] H. Regenbrecht, S. Hoermann, G. Mcgregor, B. Dixon, E. Franz, C. Ott, L. Hale, T. Schubert, and J. Hoermann. Augmented reality: Visual manipulations for motor rehabilitation. *Computers & Graphics*, 36(7):819–834, Nov. 2012.

[12] H. T. Regenbrecht, E. A. Franz, G. McGregor, B. G. Dixon, and S. Hoermann. Beyond the looking glass: Fooling the brain with the augmented mirror box. *Presence: Teleoperators and Virtual Environments*, 20(6):559–576, Dec. 2011.

[13] H. Rijpkema and M. Girard. Computer animation of knowledge-based human grasping. In *Proceedings of SIGGRAPH '91*, pages 339–348. ACM, 1991.

[14] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1372–1384, 2006.

[15] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. In *Proceedings of SIGGRAPH '09*, pages 1–8. ACM, 2009.