2016

Computer Graphics and Visualization based Analysis and Record System for Hand Surgery and Therapy Practice

Venkatamanikanta Subrahmanyakartheek Gokavarapu
Wright State University

COMPUTER GRAPHICS AND VISUALIZATION BASED ANALYSIS AND RECORD SYSTEM FOR HAND SURGERY AND THERAPY PRACTICE

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Science

By

VENKATAMANIKANTA SUBRAHMANYAKARTHEEK GOKAVARAPU
B.Tech., Gandhi Institute of Technology and Management, India, 2013

2016
Wright State University

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

MAY 25, 2016

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Venkatamanikanta Subrahmanyakartheek Gokavarapu ENTITLED Computer Graphics and Visualization based Analysis and Record System for Hand Surgery and Therapy Practice BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

Yong Pei, Ph.D.
Thesis Director

Mateen M. Rizki, Ph.D.
Chair, Department of Computer Science

Committee on
Final Examination

Yong Pei, Ph.D.

Mateen M. Rizki, Ph.D.
(Co-Thesis Director)

Krishnaprasad Thirunarayan, Ph.D.

Robert E.W. Fyffe, Ph.D.
Vice President for Research and Dean of the Graduate School

ABSTRACT

Gokavarapu, Venkatamanikanta Subrahmanyakartheek. M.S., Department of Computer Science and Engineering, Wright State University, 2016. Computer Graphics and Visualization based Analysis and Record system for Hand Surgery and Therapy Practice.

In this thesis, we have designed and developed a computer graphics and visualization based analysis and record system for hand surgery and therapy practice. In particular, we have designed and developed three novel technologies: (i) model-based data compression for hand motion records (ii) model-based surface area estimation of a human hand and (iii) an emulated study of hand wound area estimation.

First, we have presented a new data compression technique to better address the needs of electronic health record systems, such as file storage and privacy. In our proposed approach, we will extract the patient's hand motion information and store the motion data in a binary format, and then we use a loss less data compression to further reduce the file size. To illustrate this idea, we have built a prototype, which demonstrates our entire work flow. Our experiment results have shown the effective compression performance as well as added benefit of 3-D review, enhanced privacy and review capability at different playback speeds and view angles.

Next, we developed a new approach, to estimate the surface area of a patient's hand accurately and quickly with a low-cost imaging sensor and a graphics based hand model. Through the image capturing device, we capture infrared images of

the patient's hand. Once we get the input images, we run them through the image analysis engine to extract the required information in order to obtain a customized graphics hand object. Once customized we use the graphics hand object to calculate the surface area of the patient's hand. To illustrate this idea, we have built a prototype system, which demonstrate the entire work flow. Our experiment results have shown considerable reproducibility and consistency.

Finally, we came up with a novel system, capable of identifying simulated wounds on a human hand and estimating their area. Through the imaging device, wound analysis engines, we are able to identify simulated wounds on a patient's hand and measure their area using optical equations. To illustrate this idea, we have built a prototype system, which demonstrates our entire work flow. Our experiments show considerable reproducibility, accuracy and consistency.

In summary, we have developed prototypes for each of the approaches to demonstrate its capabilities. Experiments and analysis are carried out to study their performance and complexity. We believe these new approaches can significantly improve the current practice in hand surgery analysis and therapy practice.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Overview of Electronic Health Record Systems

Health care is one of the information rich sectors. Electronic Health Record (EHR) has been widely adopted by the health care sector to digitize health care record infrastructure. With these Electronic Health Records, patient's medical information can be accessed whenever and wherever it is needed.

An electronic health record (EHR) is also called an Electronic Medical Record (EMR). Electronic health records store a patient's health information in a digitized format. Normally, patient's information that can be stored in an electronic health record includes general and detailed medical information including, but not limited to, name, age, height, address, medical history, medication and allergies, immunization status, laboratory test results, vital signs, radiology images, etc[5].

This general and medical information of a particular patient stored in an electronic health record can be shared across different hospitals or health care organizations via Internet instantly. EHRs can store data more accurately and efficiently than traditional paper based documents. It is much more legible and easier to find a patient's record in an Electronic health record than in a paper based storage room. Electronic health records avoid the risk of creating duplicate entries for the same patient and decreases risk of lost paperwork. Some of the main advantages of EHRs include[5] :

1. Recording all the prescriptions used for a particular patient and his progress accordingly.
2. Notifying the patients about dosage, remind them of scheduled checkups, etc.
3. Information regarding a patient can be shared much faster between two health care organizations if necessary.
4. Decreasing the risk of lost paperwork and avoid creating duplicate entries for the same patient.
5. Sending and receiving bills, reports and results.

However, one of the major challenges of EHR is its ever-growing need for high data storage and processing. In this thesis, we try to illustrate such needs through a particular medical segment i.e. Hand surgery and therapy and propose a novel solution to address some of these needs. Some of the other major concerns of EHRs include patient privacy, lack of coordination between health care organizations, etc.

Much detailed information regarding an EHR can be found at `https://en.wikipedia.org/wiki/Electronic_health_record`.

1.2 Hand Motion Analysis in Medical Practices

"Working with a knowledgeable hand therapist can make the difference between success and failure in complex hand surgical cases. The therapist extends the continuum of our care, as well as functioning as coach and trainer for our patients." — Marybeth Ezaki, MD, Past President, American Society for Surgery of the Hand.

A hand/wrist injury may occur from any of the following reasons: trauma, disease, congenital or acquired deformity. After treating theses injuries/deformities through a surgery, a number of therapeutic intervention sessions are needed to help return a patient's hand to its highest level of function. These therapeutic intervention sessions are called 'Hand Therapy'. Hand therapy is the art and science of evaluating and treating injuries and conditions of the upper extremity:shoulder, arm, elbow, forearm, wrist and hand[6].

An occupational or physical therapist, through advanced continuing education, clinical experience and integration of knowledge in anatomy,becomes proficient in treatment of pathological upper extremity conditions and conducts these hand therapy sessions. A physical or hand therapist employs a variety of techniques and tools, including activity and exercise programs based on his observation of patient's progress[6].

One of the objectives of this research is to better help the hand therapist with more complete patient information and a communication platform among the hand surgeon, therapist and patient for more accurate and effective interaction through the treatment process. We plan to support that by providing capabilities to record all the previous sessions of the hand therapy for a particular patient and store them in electronic health record, which can be reviewed by the physician and therapist as needed. This recorded information can help a therapist to make well-informed treatment decisions more quickly, safely, accurately and also assure such information is not lost in transition i.e. when there is a change of therapist. 3D videos require tremendous amount of storage space. To overcome these storage and privacy issues we need a new information representation and data compression technique. As a result, we will employ a model-based approach to overcome this barrier.

1.3 Data Compression

In practice, hand motion data are captured as a video. A video is made up of a series of still image frames. These sequences of frames consist of redundant information. By eliminating this redundant information, we can reduce the video file size. There are a number of video compression algorithms that are put in place over the years to reduce the redundancy of data in a video file, which can be used in our system. The Table1.1 lists all the different international video compression standards that are present until date:

Year	Standard	Publisher	Popular implementations
1984	H.120	ITU-T	
1988	H.261	ITU-T	Videoconferencing, video telephony
1993	MPEG-1 Part 2	ISO,IEC	Video-CD
1995	H.262/MPEG-2 Part 2	ISO,IEC, ITU-T	DVD Video, Blu-ray, Digital Video Broadcasting, SVCD
1996	H.263	ITU-T	Videoconferencing, video telephony, video on mobile phones (3GP)
1999	MPEG-4 Part 2	ISO,IEC	Video on Internet (DivX, Xvid ...)
2003	H.264/MPEG-4 AVC	Sony,Panasonic, Samsung, ISO, IEC, ITU-T	Blu-ray, HD DVD, Digital Video Broadcasting, iPod Video, Apple TV, videoconferencing
2009	VC-2 (Dirac)	SMPTE	Video on Internet, HDTV broadcast, UHDTV
2013	H.265	ISO,IEC, ITU-T	

Table 1.1: History of Video Compression Standards[4]

1.3.1 MPEG-4 Video Compression

MPEG-4 is one of the most widely used video compression techniques. MPEG-4 stands for Motion Picture Experts Group Layer-4 Video. MPEG-4 video compression technique was first officially published in 1999 and later many enhancements have been made. MPEG-4 technique employs different techniques to achieve its high compression performance.

Normally, videos are sequences of image frames. There exists a lot of redundant information, e.g. the spatial and temporal redundancies. The temporal redundancy refers to similarity or correlation between the series of frames in a video. The spatial redundancy refers to the similarity or correlation between the adjacent pixels with in an image frame itself. As a result, MPEG-4 video compression techniques only have to encode the difference between the successive frames within a video. In this way, MPEG-4 technique can remove significantly the temporal redundancy in the video, which in turn decreases the video file size. To reduce the spatial redundancy in the video MPEG-4 video compression technique uses DCT-based transform coding[7].

Apart from handling the spatial and temporal redundancies in the video, MPEG-4 compression technique uses subsampled 'YUV' instead of RGB to decrease the file size. In YUV, Y refers to luminance components and U, V refers to chroma components. Specifically, the Y component carries intensity/brightness information and the U,V components carry color information. More bits are used to encode Y component than to encode U,V components because human eye is less sensitive to color than to brightness. However, MPEG-4 is a lossy compression method, which may result in loss of important motion data information[7].

Unlike, the above-mentioned conventional video compression techniques, we propose a model-based data compression technique, which gives improved compression performance with enhanced privacy.

1.4 Model Based Data Compression and Visualization

In our model-based data compression and visualization technique, we extract the hand motion information from the hand motion video and store the motion information in a text file in binary format. Later, we use our prototype called the 'Visualizer', to visualize the stored hand motion data using a hand model in the form of a reconstructed animated video. The proposed compression method can be applied for both image and video compression without loss of important motion data information. Some of the major advantages of our model-based data compression and visualization approach include:

1. Motion data is extracted and is stored which can be big data processed.

2. Extracted motion data can be reviewed through our Visualizer in the form of an animated video.

3. Motion data can be reviewed from any desired angle/view with different playback speeds.

4. Patients information is private and secure e.g. No finger print can be collected from an animated video.

We have developed a prototype system, which employs the model-based compression technique to store the hand motions and play them back in the form of an animated video. We will discuss the technical details and the workflow of our prototype, which employs model-based compression technique, in the second chapter of this thesis.

1.5 Model Based Hand Surface Area Estimation of a Human Hand

In the previous sections of this chapter, our focus was on hand movement analysis i.e. to record, extract and visualize the human hand movement with considerably less storage space and added 3-D capabilities. In this section, we will explore the use of a computer graphics model-based approach to estimate the surface area of a human hand. We propose a new approach to estimate the surface area of a human hand using our model-based approach. Some of the applications of estimating the human hand surface area are listed below:

1. It can be used by the hygienist for estimating the amount of sanitizer required for a particular hand in 'Automatic Hand Sanitizers'.
2. It can be used in exposure assessment in occupational toxicology.
3. It can be used for the development of manual equipment/protective gloves in ergonomics.

Until date, there is no proper tool that is capable of estimating the surface area of a human hand accurately in real time[8]. By using our novel hand model-based approach, we developed a system that has the potential to estimate the surface area of a human hand accurately and quickly.

We built a prototype system, which can instantly calculate and display the surface area of a user's hand. We have used the same hand model and input device (i.e. Leap Motion device) which is used in the model-based data compression approach in our prototype system to calculate the surface area of human hand.

We discuss the workflow of model-based hand surface area approximation and combinations of techniques/ procedures used to estimate the surface area of a human hand in the third chapter of this thesis.

1.6 An Emulated Study of Wound Area Estimation of a Human Hand

As an extension to our model-based hand surface area estimation, we further explored its potential to identify a wound on the human hand and then estimate its area.

This estimated wound area can be used in health care domain and has several applications. For example, it is particularly useful to track a patient's progress and his response to the treatment after a burn or wound on his/her hand. It lets the doctor and therapist know how well the patient's wound is reacting to the current treatment. Moreover, this kind of tool gives a patient much more confidence and motivation by providing an effective platform to monitor his progress visually every week. Some of the other applications of this wound area include:

1. It can be used to estimate the area of burned skin in burn treatment and therapy.
2. It can be used to estimate the total wound area out of the total hand's surface area.

As for now, we do not have access to a patient with a hand wound. Therefore, we used a Marker to emulate a wound on the hand in an emulated way to illustrate the idea. This is reason why this new section is called as 'An Emulated Study of Wound Area Estimation'.

We have built a prototype system, which can calculate the wound area of the simulated wound (i.e. black marks with the marker) on the hand. In future, through a clinical study, our system can be applied to real wounds and be validated properly.

We discuss more about the work flow and combination of image processing techniques we employed to identify and estimate the area of the simulated wound on the human hand in the fourth chapter of this thesis.

2 MODEL BASED DATA COMPRESSION AND VISUALIZATION

As mentioned in the previous chapter, storing all previous therapy sessions for every patient may require a lot of storage space. Thus limit, one of the main advantages of Electronic Health Records, exchange of patient information across different health care organizations, because it may take a lot of time to transfer a patient's health record from one health organization to another. Therefore, an effective data compression technique may save a lot of time and storage space in such scenarios. In this thesis, we propose a model-based compression technique.

2.1 Workflow of Model Based Data Compression

Figure 2.1: Workflow for Model Based Data Compression.

For our model-based data compression technique, we follow the workflow shown in the Figure 2.1. First, the input device (i.e. Leap Motion sensor), in the form of images, captures the hand movements, and then the hand motion analysis engine extracts the required information such as the hand, finger, bone's proration and orientation data and store them into a file. Later, we can review the extracted hand motion through our Visualizer, which plays the stored hand motion in the form of an animated video. Each part of this workflow is explained in much more detail in the coming sections.

2.2 Motion Capture Device

In this model-based data compression and visualization, we use a Leap Motion controller as a hand motion capture device. The Leap Motion controller is a low cost, low weight, portable, small USB peripheral device, which is designed to be placed conveniently on a desk, facing upward.

Leap Motion controller consists of two infrared cameras and three LEDs. Using these IR cameras and the LEDs, the device observes a roughly hemispherical area, to a distance of up to 1 meter. The LEDs generate IR light and the cameras generate almost 100 frames per second of reflected data, which is then sent through a USB cable to the host computer, where it is analyzed by the Leap Motion controller software using 'complex math' in a way that has not been disclosed by the company, in some way synthesizing 3D position data by comparing the 2D frames generated by the two cameras[9]. Here is an image of a Leap Motion controller shown in Figure 2.2 .

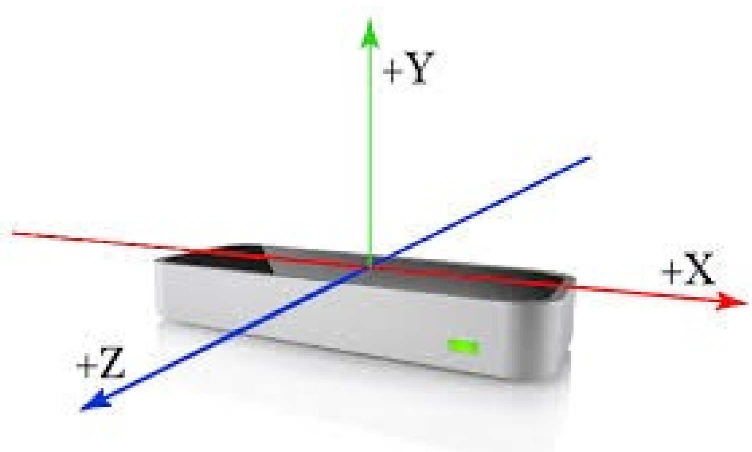

Figure 2.2: Leap Motion Controller [1]

When we place our hand on the Leap Motion sensor, we get 640x240 infrared image frame. The Leap device is capable of producing around 100 frames per second. Using these images required information such as finger count, finger lengths, finger orientation, etc., can be extracted using image processing and complex math. Much detailed information regarding Leap Motion controller can be found at `//https://en.wikipedia.org/wiki/Leap_Motion`.

2.3 Overview of Hand Motion Analysis Engine by Leap

A human hand consists of five fingers. Each finger consists of four bones: Metacarpals, Proximal phalanges, Intermediate Phalanges, Distal Phalanges. In our model-based data compression approach, we store the proration data and the orientation data of every bone in the human hand instead of storing the images to achieve data compression. We use the same proration and orientation data to reconstruct the video using the Visualizer. The used hand structure is shown in Figure 2.3.

The Leap Motion API provides the proration data (i.e. x, y, z coordinates) and the orientation data of all bones along with additional information for every frame in a structure called 'Frame'. A Frame structure consists of a 'HandList' Object. A Handlist contains 'Hand' objects in a frame. Say, we have two hands placed on the Leap device, and then the Hand list consists of two Hand objects for that particular frame. Each Hand object contains an 'Arm' object and a 'FingerList' object. The Arm object consists of information regarding the forearm. The FingerList consists of 'Finger' and some additional information regarding fingers (like finger count, leftmost one, rightmost one, etc.). Each finger is made up of bones. Therefore, the Finger object contains 'Bone' Objects. Each Bone object contains information like bone length, direction, bone width, etc.

Much detailed information regarding a Leap Frame structure can be found at

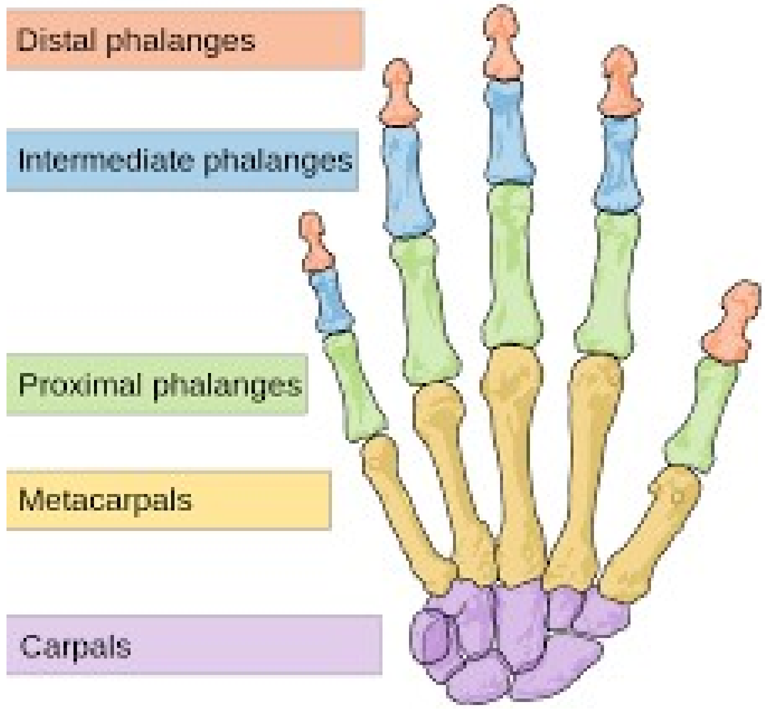

Figure 2.3: Simple Hand Structure [2]

`https://developer.leapmotion.com/documentation/csharp/api/Leap.Frame.html`. An image of the Leap Frame Structure is shown in Figure 2.4.

Apparently, each Hand object consists of a FingerList, which has five Finger objects (i.e. index, middle, ring, thumb and Little fingers) in it. Each Finger has four Bone objects (i.e. one for each of the Distal, Intermediate, Proximal and Metacarpal bones). For every 1 sec, around 100 Leap Frames are created. One Leap Frame object is created for every frame. All these Leap Frame objects provided by the Leap containing information about hand, finger, bone, etc, can be serialized and can be stored in a binary file. This saved byte information is used by the Visualizer to reconstruct the hand motion in the form of an animated video.

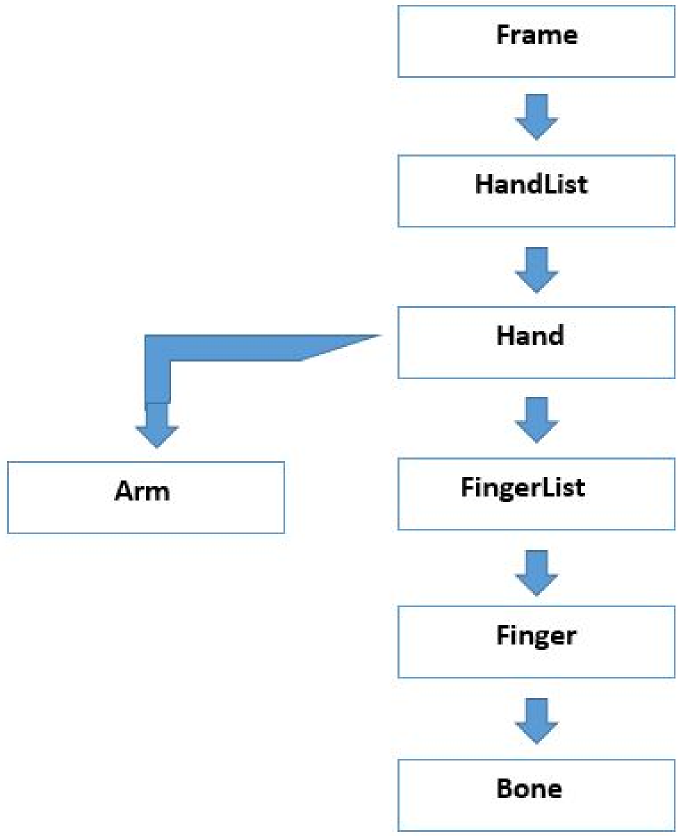

Figure 2.4: Leap Frame Structure.

2.4 Prototype Development

A Visualizer is the key part of this model-based data compression technique. Visualizer is an application, which we created using Unity, a cross platform visual game engine. Visualizer replays the hand motion using a hand model. A graphics hand model is made up of collections of polygons using 3D modelling tools like Maya. We made use of the graphics hand models from the 'Leap Motion Unity Core Assets' package. The Visualizer reads the motion data stored in the binary file and applies the proration and the orientation data to every bone, finger and forearm in the hand model to create an animated video using the functions provided by the Leap Motion API. Our prototype system provides the following capabilities for the medical professionals.

1. Record human hand's gestures and movements into a binary file.

2. Review the hand's gestures and movements stored in the file through the Visualizer.

Some of the key advantages of our Visualizer include:

1. Extract and store only motion data.

2. Record the gestures of the human hand and save it in a desired location.

3. Review the extracted hand motion in the form of an animated video.

4. View the animated video from any desired angle/view with different playback speeds.

5. Zoom in/out feature for the animated video.

6. Pause/rewind/forward features while playing the animated video.

7. Enhanced privacy and security for the patient e.g. no finger prints can be extracted from the data or animated video.

8. Accommodating big data processing for treatment profiling.

The main capabilities and the advantages of the Visualizer are explained in a much more detailed fashion in the coming sections.

2.4.1 Hand Motion Capture, Analysis and Storage

Using this Visualizer, we can record the hand movements and gestures using the 'Record' button present in the Visualizer GUI. To record the hand movements and gestures, user places either his left hand or right hand on the Leap Motion sensor, then he could see the hand model in the GUI replicating his hand movements in real time. We can change the skin color of the hand model and set the zoom angle with respect to the user's preferences. Once he presses the 'Record' button, the recording begins. He can stop the recording by pressing the 'Stop' button in the GUI, and then a popup is shown asking whether to save or discard this recording. Based on the user's selection, the video, i.e., the extracted hand motion sequences are either recorded in a binary file or discarded. A sample image of the record function in the Visualizer is shown in Figure 2.5.

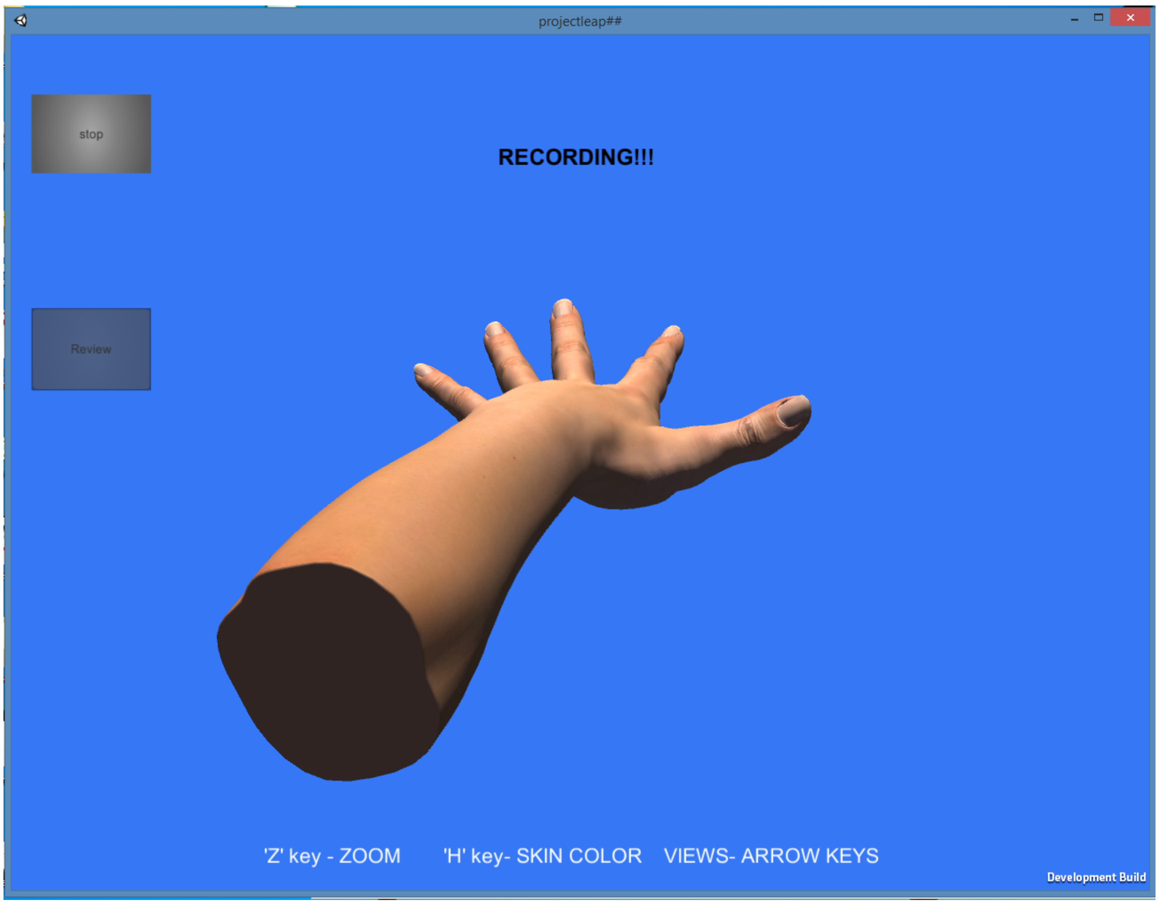

Figure 2.5: Record Capability in Visualizer.

2.4.2 Hand Motion Visualizer

Using the Visualizer, we can playback/review the recorded hand movements and gestures. To play it back the user just needs to press the 'Review' button, and select the file he wants to review. The Visualizer reads the recorded hand motion data information from the binary file, and plays the hand motion in the form of an animated video using the hand model. The review/playback capability of the Visualizer comes with a different set of features. Some of the most important ones are:

1. 3-D review.
2. Zoom in/out.
3. Pause, forward, rewind options for the review capability.
4. Reviewing the stored hand motions of the patient's hand can be carried out at different playback speeds.
5. A level of privacy/security for the patient's information.

A sample image of the Review function in the Visualizer is shown in Figure 2.6.

Unlike the traditional video playback, we have a 3-D review feature. While reviewing, the reviewer or the user can view the animated video in his desired angle. For example, if he wants to view the hand motion from the top angle, he will be able to do it irrespective of how the video is originally recorded. Say, if he wants to review the hand motion from the left, he just needs to click the 'left view' button present on the GUI to review it from the left angle. The Visualizer also comes with a zoom in/out option. A reviewer will be able to zoom in/out to particular location in the hand irrespective of how the video is originally recorded. The Visualizer supports pause, forward and rewind functionalities while playing the 3-D animated video. In addition, Visualizer is capable of playing the animated video at different playback speeds. The reviewer can access these functionalities by clicking on the pause, forward and rewind

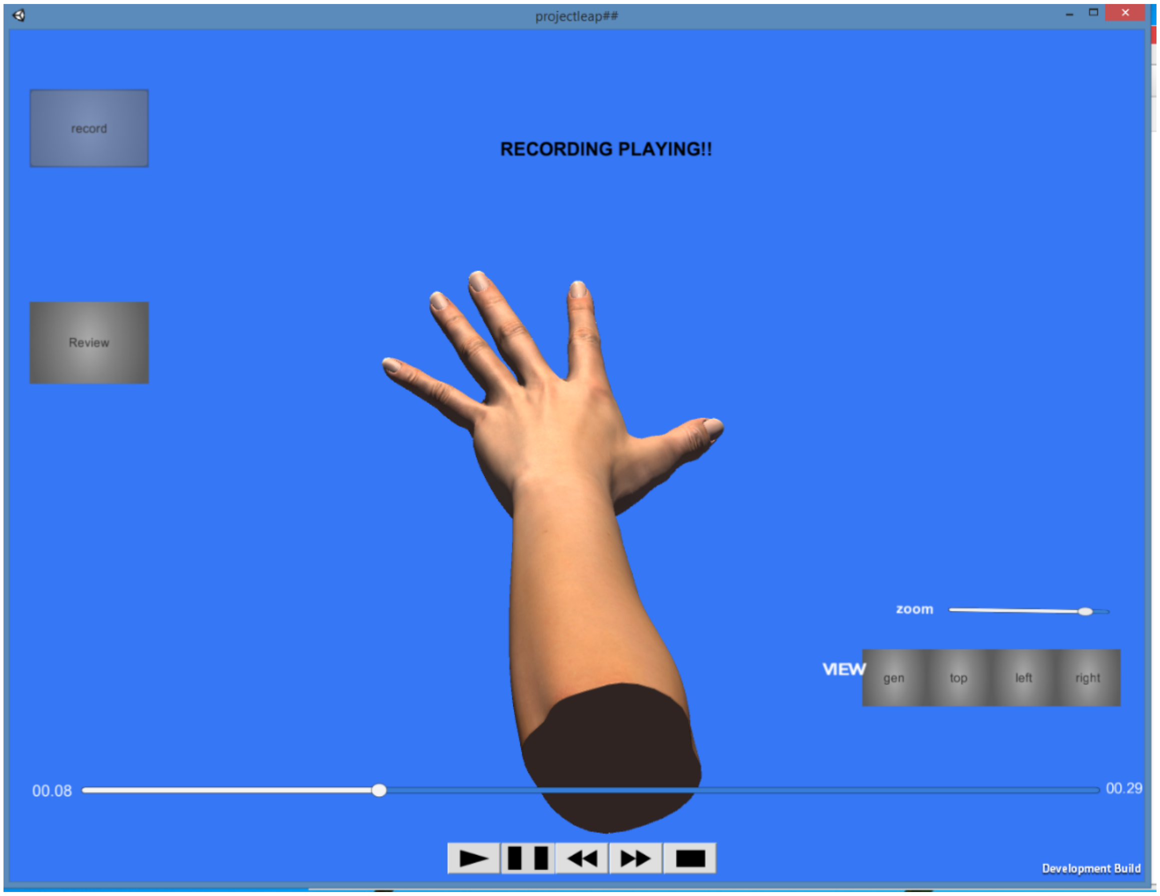

Figure 2.6: Review Capability in Visualizer.

buttons displayed on the Visualizer's GUI. Finally, yet importantly, the visualizer adds a level of security/privacy for the user's data. For improved security, encryption algorithms can also be used. The hand motions that are recorded will be stored in a binary file, which is unreadable by a human eye. The only way we can reconstruct or use the data stored is through loading it in our Visualizer. As a result, the Visualizer significantly improves privacy/security to the patient's data i.e. stored in the EHR.

The upcoming section describes about the file storage of model-based compression technique and the comparison between MPEG-4 compression technique and our model-based data compression technique.

2.5 Comparative Performance Evaluation of the Proposed System

As mentioned in the previous sections, the model-based compression technique stores hand motion data in a binary format. The binary file consists of the proration and orientation data of hand, finger and bone for every frame recorded. The Visualizer offers 'Save' option through which user will be able to save the file in the desired location on his personal computer. During Playback/Review, the user has to provide the file path (i.e. where the recorded binary file is stored previously) for reviewing the hand motion.

2.5.1 Performance Comparison at Image Level

For every frame, Leap Motion sensor provides us with two 640×240 Infrared images (i.e., one from the left camera and the second one from the right camera). Each image is of size 151 Kilo bytes. An example of the raw image from Leap Motion sensor, which is of size 151KB when in uncompressed format is shown in Figure 2.7.

Figure 2.7: Raw Image from Leap Motion Controller

For comparison, we used JPEG image encoder to compress the raw image from the Leap Motion sensor to JPEG format. Using JPEG encoder, for best quality compression, we could compress the file to 50KB. JPEG can compress the raw image to as small as 2.74 KB; however at a cost of significant distortion. A compressed JPEG image with high quality and less distortion, which consumes around 50KB of storage space, is shown in Figure 2.8.

Figure 2.8: JPEG Compressed Image with High Quality and Less Distortion Option.

A compressed JPEG image with less quality and with a significant distortion, which consumes around 2.74KB of storage space, is shown in Figure 2.9.

Figure 2.9: JPEG Compressed Image with Low Quality and High Distortion Option.

Our approach takes around 3.49KB for every frame. We are able to further compress it to 2.66KB by compressing our data file using a ZIP tool without loss of any important motion data information because we extract the motion data before any lossy image compression. From the above scenario, we could see our model-based data compression offers better image compression than JPEG when considering motion information.

2.5.2 Performance Comparison at Video Level

We recorded 35-seconds of grey scale video with a frame rate of 105 frames per second using one of the Leap Motion's infrared cameras. The file size for this uncompressed grey scale video file is around 539MB.

We used a MPEG-4 part 10 (i.e. H.264, one of the latest video encoders) to compress the raw video file collected from Leap cameras. Generally, medical grade equipment needs high quality. Therefore, with H.264 encoder using best quality option, we could compress our video file to 5.12MB.

For comparison, we processed the same frame information using our prototype. Our prototype system recorded the 35-second video and created a byte's file of size 6.50 MB. To further compress it we sent our data file through a ZIP tool. The final compressed file, which can be played using our Visualizer, is around 3.71 MB.

From the comparative study, we could clearly see that our model-based approach could offer better video compression along with some very important advantages like the 3-D review, data privacy, security to the patient's information, etc.

3 MODEL BASED SURFACE AREA ESTIMATION OF A HUMAN HAND

In this chapter, we propose a new system, which is capable of estimating the surface area of a human hand using our model-based approach. Once estimated, the surface area of a human hand can be used for various purposes. For example, hand surface area can be used in estimating the amount of sanitizer required for a particular hand in Automatic hand sanitizers, used to calculate exposure in occupational toxicology, etc.

3.1 Workflow of Model Based Hand Surface Area Estimation

The workflow of the model-based approach for hand surface area estimation is illustrated is the Figure 3.1.

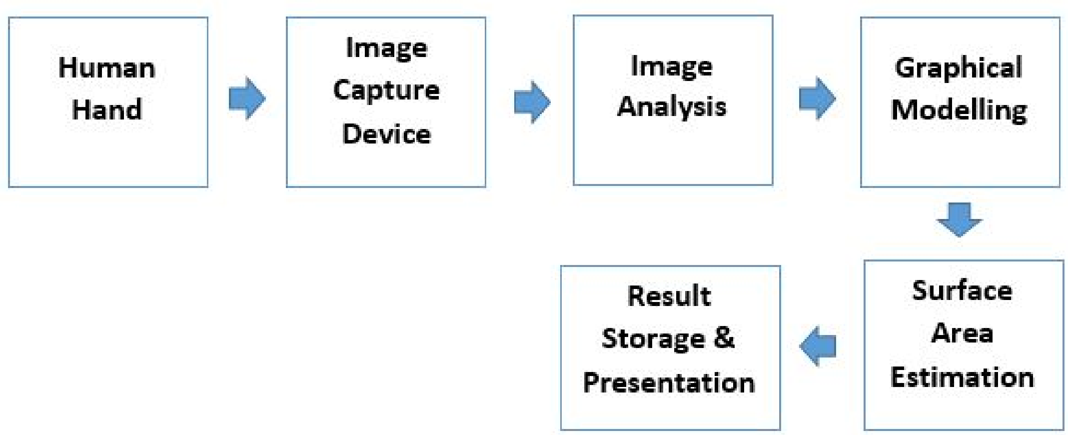

Figure 3.1: Workflow of Model Based Hand Surface Area Estimation.

In our model-based hand surface area approximation, we use a hand graphics model which is created by using 3d modelling tools (Maya in our work) to estimate the surface area of a human hand by matching the captured hand image to the model. Here is a brief summary of how the workflow goes. We use the Leap Motion device as the input device in this model-based hand surface area estimation approach. Although a Leap device is designed for the purpose of hand motion capturing, we are trying to explore a new usage by taking advantage of its infrared imaging capabilities from which a human hand can be accurately identified and segmented out from the image.

The Leap device provides 640×240 grey scale images at a rate of over 100 frames per second. However, for our Image Analysis, we will first identify images from which the human hand can be extracted with high confidence. Then, the selected hand image is segmented and extracted before we map it to our computer graphics model, resulting in a customized hand object for this particular hand.

Then, by using the customized graphics hand model, we will be able to calculate the surface area of the hand by aggregating the areas of all the polygons present in the graphical hand object. Each part of this workflow is explained in much more detail in the coming sections.

3.2 Image Capture Device

We use a Leap Motion sensor as an image-capturing device in our model-based approach to calculate the surface area of human hand. For the technical details of the Leap Motion device, please refer to Section 2.2.

Generally, a Leap Motion sensor is used as a motion capture device, as mentioned, and is used in the model-based data compression to capture hand motion. However, we want to further explore the capabilities of the Leap device and want to use it as an imaging device for our model-based hand surface area estimation system.

A Leap Motion device is capable of generating two 640×240 infrared images for every frame. A 640×240 raw hand image collected by the Leap Motion sensor is shown in Figure 3.2.

Figure 3.2: Raw Hand Image from Leap Motion sensor.

3.3 Image Analysis Engine

The image analysis engine is one of the most important components of the model based hand surface area estimation approach in order to construct customized hand object on a particular hand. In this part of the workflow, we collect the corrected leap images from the imaging device (i.e. Leap Motion controller) and process them to get the required hand geometry information. We achieve this extraction of information using image-processing tools provided by OpenCV.

The raw image from the Leap Motion controller comes with distortion due to lens curvature and other imperfections. The Figure 3.3 contains an example image to illustrate the distortion present in the raw Leap images.

This lens distortion can be corrected by using the calibration map provided by

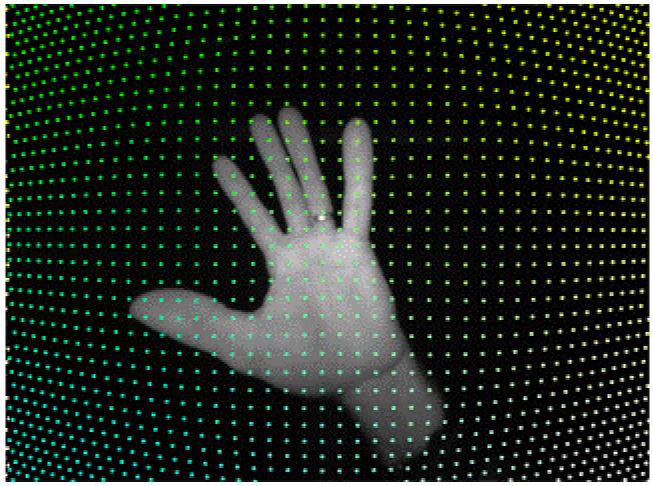

Figure 3.3: Image with Distortion in Raw Image [3]

the Leap through a correction process.

More information regarding lens distortion and the calibration map can be found at `https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Images.html`. The Figure 3.4 shows a distortion free hand image obtained by using the calibration map provided by Leap.

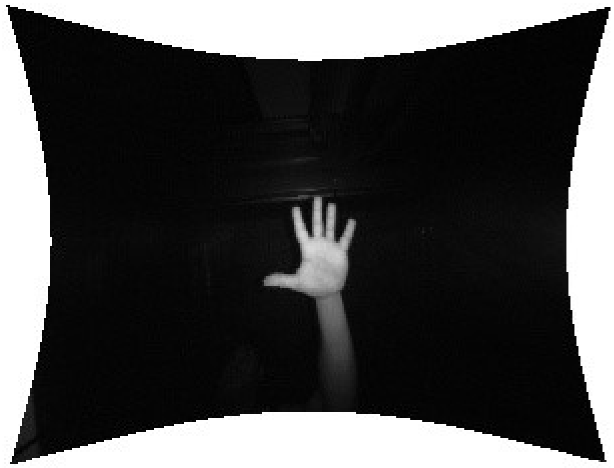

Figure 3.4: Distortion-Corrected Hand Image.

3.3.1 Cropping Hand Image

After we correct the raw image from the Leap, our hand image contains some white edges and spaces around it as shown in the Figure 3.4. To remove all these white edges, and to bring our image to a square shape, we crop the corrected image. The image displayed in the Figure 3.5 shows the hand image after the cropping procedure.

Figure 3.5: Cropped Hand Image.

3.3.2 Image Enhancement

In order to accurately segment the hand from the background in the infrared image, we further apply an automated contrast adjustment in order to enhance the edge between the hand and the background. By doing so, the objects closer to the camera will look much brighter than the ones farther away from camera as clearly shown in Figure 3.6.

Figure 3.6: Sample Enhanced Hand Image.

3.3.3 Background Suppression

Following the image enhancement, we will extract the hand object from the background. In order to reduce the impact of background on identifying the hand contour, we will first apply a threshold based filtering process to the image as shown in Figure 3.7. The threshold based filter will suppress the background significantly in the infrared image.

Figure 3.7: Sample Hand Image after Background suppression.

3.3.4 Extraction of Human Hand

After suppressing the background from the input image, we then find the contours around the human hand using the 'findContours' function given by the 'OpenCV' libraries. We draw this hand contour on to a new image using the 'drawContours' function provided by 'OpenCV' as illustrated in Figure 3.8.

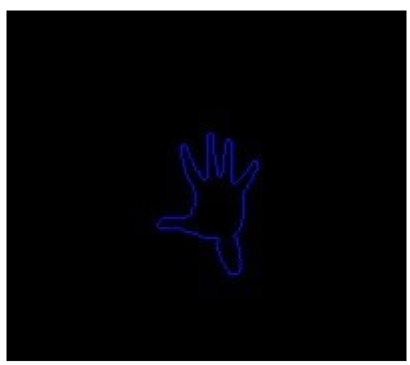

Figure 3.8: Sample Image displaying Hand Contour.

Then, we draw a convex hull to find the fingertips and depth points using the 'convexHull' function supported by the 'OpenCV' libraries. The image shown in Figure 3.9 illustrates the convex hull drawn around the hand contour.

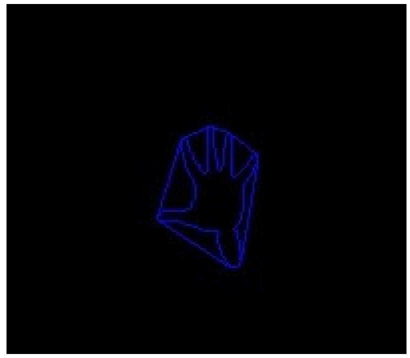

Figure 3.9: Sample Image displaying Hand Contour and Convex Hull.

3.3.5 Identification of Human Hand

We could clearly see both the contour around the human hand, and the convex hull drawn around the hand contour, from the image shown in Figure 3.9. Both the contour and the convex hull are drawn to identify the fingertips and the depth points. ‘OpenCV’ supports a function called ‘convexityDefects’ which gives us the end points (i.e. where the contour and the convex hull coincide) and the depth points. In our case, the end points refer to the fingertips and the depth points refer to the gaps between the fingers. The same is demonstrated in the image shown in the Figure 3.10, in which yellow circles refer to depth points (i.e. gap points between the fingers) and the red circles plotted refer to ends (i.e. fingertips).

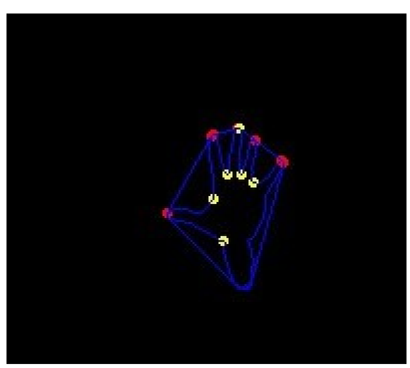

Figure 3.10: Sample Hand Image with Finger Tips and Depth Points.

Using these end and depth points, we can calculate the confidence of the input image. If the input image has enough confidence value and matches the predefined pattern, we extract the information from the input hand image that is needed to create the hand object using the computer graphics model for a particular patient's hand. An input image without a sufficient confidence value is discarded and the next image in the image sequence is considered. This process continues until we find an input hand image with sufficient confidence.

3.4 Graphical Modelling of Hand

Once the input image with high confidence value is selected, we use the end points and the depth points to customize the graphics hand model for a particular patient's hand. Figure 3.11 shows an image of the graphics hand model in a solid view.

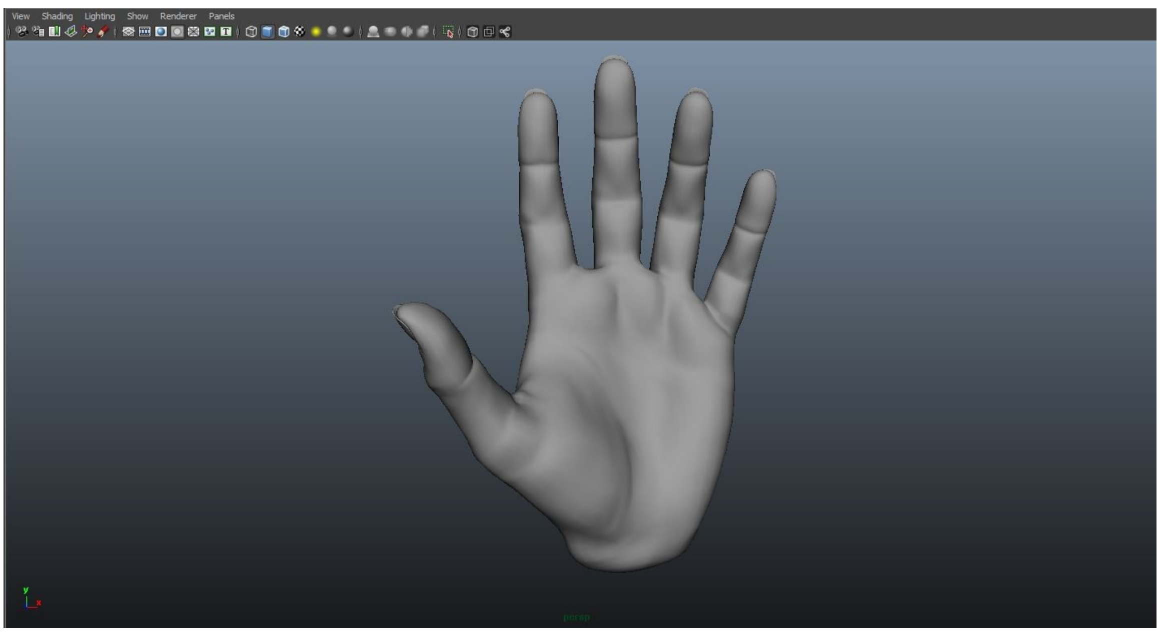

Figure 3.11: Graphics Hand Model in Solid View.

All these graphics hand models are made up of thousands of polygons using 3D modelling tools, like Maya. We have made use of the hand models provided by the "Leap Motion Unity Core Assets" package. The image shown in Figure 3.12 illustrates the idea that the hand models are made up of polygons.

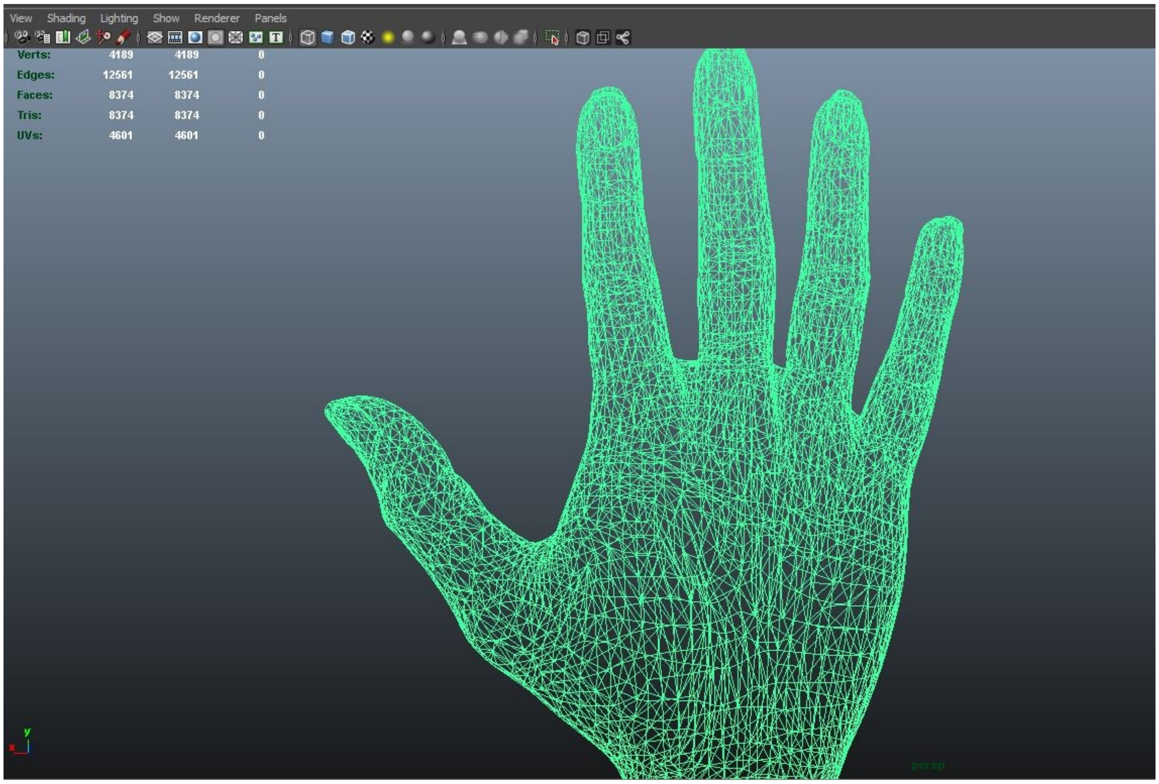

Figure 3.12: Graphics Hand Model in Wireframe View.

3.5 Surface Area Estimation of Human Hand based on a Computer Graphics Hand Model

We calculate the surface area of the patient's hand using a hand object, that is customized based on the image processed of the patient's hand. Specifically, we match the hand model using the information extracted from the input hand image; we then calculate the surface area of the user's hand by calculating the aggregate area of polygons in the hand object. The area of each polygon is calculated using the equation given in 3.1.

$$Area\ of\ the\ triangle = \frac{(cross\ product\ of\ two\ sides\ of\ the\ triangle)}{2}[10] \qquad (3.1)$$

Our hand model consists of a large number of polygons. In this thesis, we studied the effect of a set of four graphics models, which represent increasing levels of granularity by using increasing number of polygons.

The larger the number of polygons, the higher the accuracy when we estimate the hand surface area at the cost of higher computation. We have obtained experiment results using the prototype that we developed, as shown in Figure 3.13.

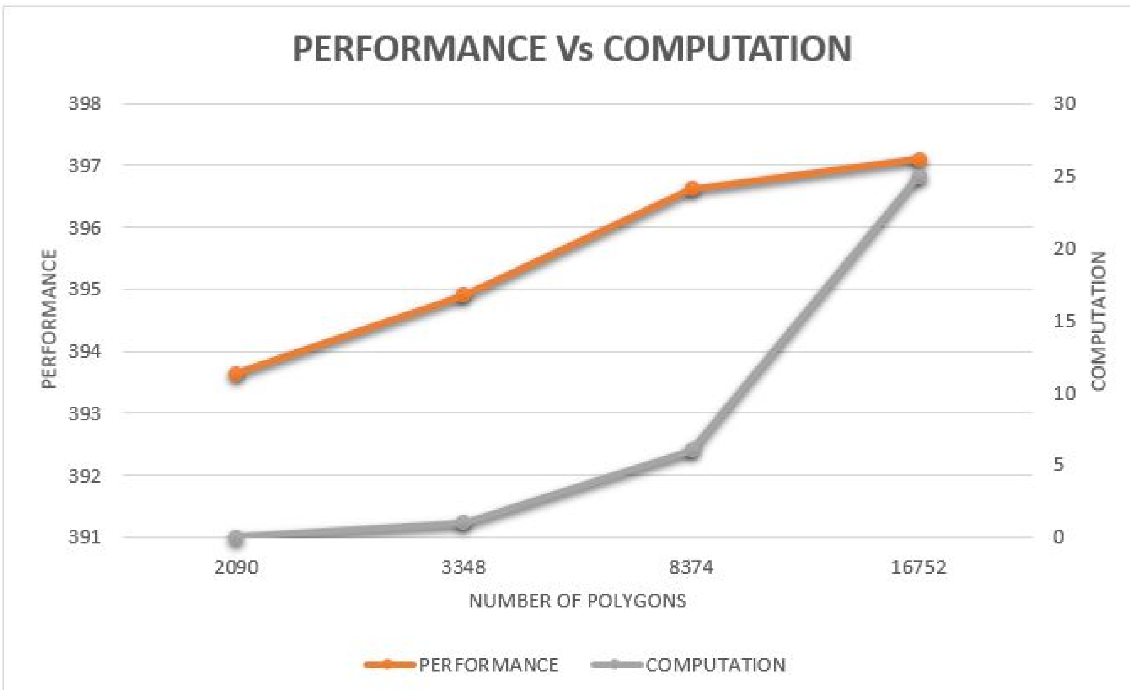

Figure 3.13: Performance vs Computation.

From the graph shown in Figure 3.13, We can clearly see as the number of polygons in the hand model increases, performance increases linearly and the complexity increases exponentially. This exponential growth of the computation is due to the time it takes to identify and collect all the vertices of the polygons to calculate the area of the hand object.

3.6 Development of Prototype System

We have built a prototype system using Unity to estimate and display the surface area of the human hand. This prototype system supports some basic functionalities, like save and reset. The image shown in Figure 3.14 shows the presentation part of our prototype system.

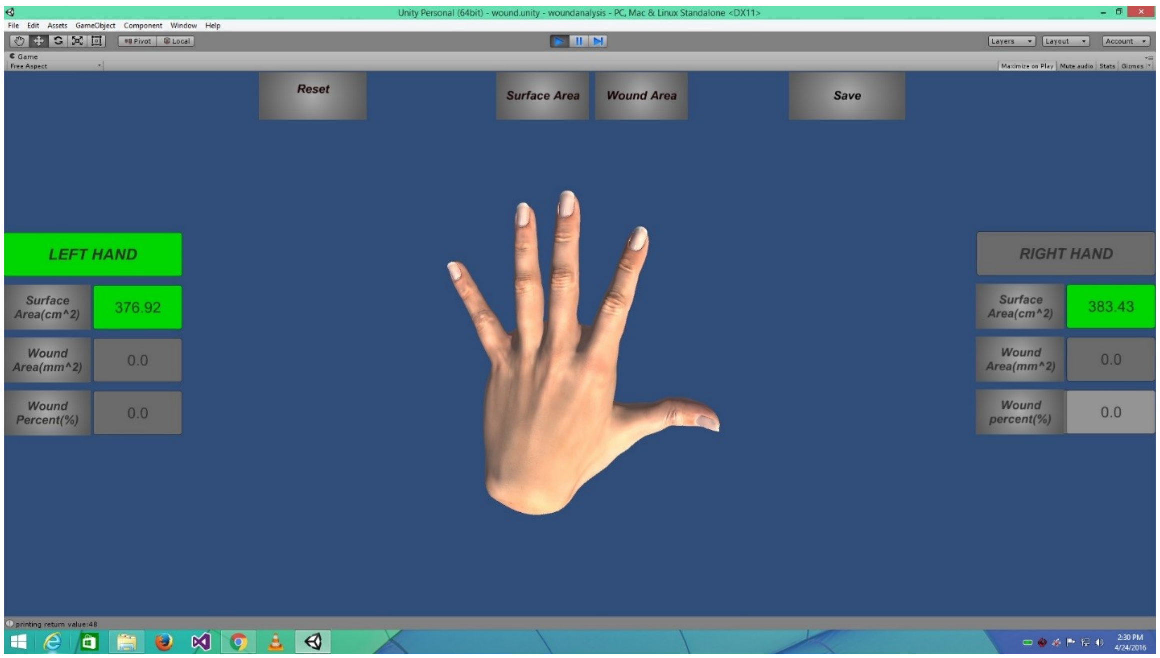

Figure 3.14: Prototype System to Calculate Surface Area.

For a user to use our system, he first needs to connect the Leap Motion device to the computer. Then when he puts his left/right hand on the Leap Motion device, he can see a animated hand on the GUI replicating all his hand movements. When he wants to measure the surface area, he needs to click the 'Surface Area' button. This calculates the surface area of the user's hand and shows the surface area value in the corresponding 'Surface Area' box. Our system also supports a 'save' function. The user can store the surface area values in desired location in a text format by clicking on the 'Save' button displayed on the GUI. The 'Reset' button on GUI clears all the old surface area values.

3.7 Experiment Results for Model Based Surface Area Estimation System

To test the consistency of our prototype system, which is capable of measuring the surface area of a user's hand, we have taken three measurements of the same hand using our prototype system. The results presented in the Table 3.1 show the consistency and reproducibility of the surface area measurements for a user's hand.

Take	Left Hand Surface Area (sq.cm)	Right Hand Surface Area (sq.cm)
1	368.18	376.89
2	383.78	377.38
3	371.52	378.21

Table 3.1: Results for Model Based Surface Area Estimation System

It is evident from Table 3.1 that the results obtained are consistent.

4 AN EMULATED STUDY OF WOUND AREA ESTIMATION OF A HUMAN HAND

In this chapter, we propose a new system, which is capable of estimating the wound area on a human hand. Once estimated, the wound area can be used in various fields. For example, the estimated wound area can be used to track the patient's progress and his response to the given treatment after a burn or wound on his hand. As for now, we do not have access to a patient with a hand wound. Therefore, we use a 'Marker' as to draw a simulated wound on the hand in an emulated way to illustrate the ideas. In the future, through the clinical study, our system can be applied to those real wounds and validated properly.

4.1 Workflow for Estimation of Simulated Wound Area on a Human Hand

The image shown in Figure 4.1 illustrates the workflow that we have developed to calculate the wound area on a human hand. Similar to the surface area estimation, we use the Leap Motion sensor as an imaging device. Leap Motion sensor consists of two infrared cameras, which are capable of generating infrared images. All these infrared images generated are sent to a wound analysis engine. The wound analysis engine collects all the raw images from the input device, process them using the image processing tools provided by OpenCV and extracts the required wound information and sends it for the wound area estimation. This wound area estimation block estimates the wound area based on the information provided by the wound analysis

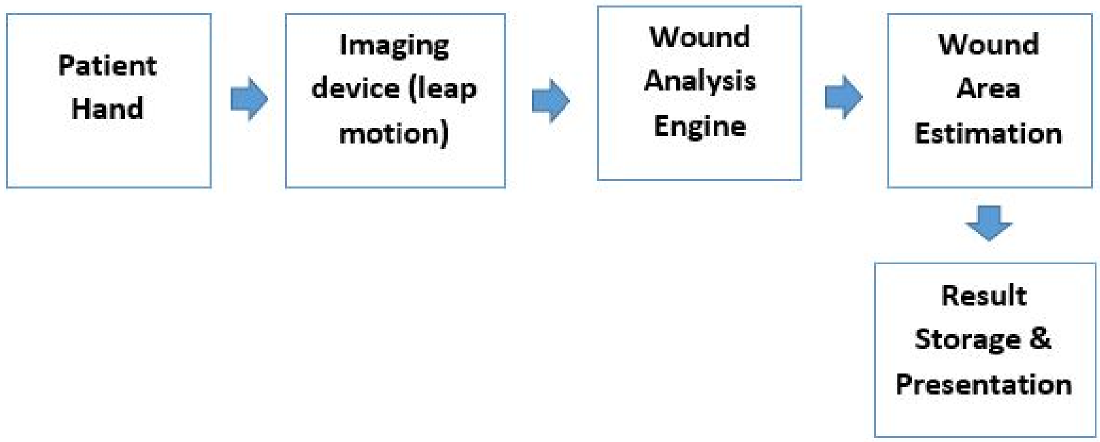

Figure 4.1: Workflow for Estimation of Simulated Wound Area on Human Hand.

engine. Once the wound area is estimated, it is displayed on the GUI using our prototype system. The coming sections explain each part of the workflow in a very detailed manner.

4.2 Imaging Device

We use a Leap Motion sensor, as an image-capturing device in our approach to capture the wound area on a human hand. For the technical details of the Leap Motion device, please refer to section 2.2.

Generally, a Leap Motion sensor is used as a motion capture device as mentioned and used in the model-based data compression. However, we want to further explore its capabilities as an imaging device for accurately identifying the wound and to estimate its area accurately. Our preliminary experiments have shown its capabilities to identify real wounds on hands.

To understand Leap's infrared imaging capabilities, we use it to take a 640×240 infrared image as shown in Figure 4.2. It becomes clearer that the image can capture the simulated wound as shown in Figure 4.2 as the dark part inside the hand contour.

Figure 4.2: Raw Image of Human Hand with a Simulated Wound.

The raw images collected by the Leap infrared cameras come with obvious fish eye effect (i.e. contains a distortion, which need to be corrected). As a result, these raw images are processed by the wound analysis engine, to correct the distortion and then to extract the wound information which is necessary for the estimation of wound area.

4.3 Image Processing based Wound Analysis Engine

The raw image from the Leap Motion controller comes with distortion due to lens curvature and other imperfections. This lens distortion can be corrected by using the calibration map provided by the Leap. Figure 4.3 shows a distortion-free hand image obtained by using the calibration. The distorted ratio of hand palm has been corrected as evident in Figure 4.3.

4.3.1 Cropping the Input Image

As we can clearly see from the distortion corrected image shown in Figure 4.3, it has some white edges around the image, which are not necessary for our purpose. Therefore, we remove those additional white areas by cropping the image using the

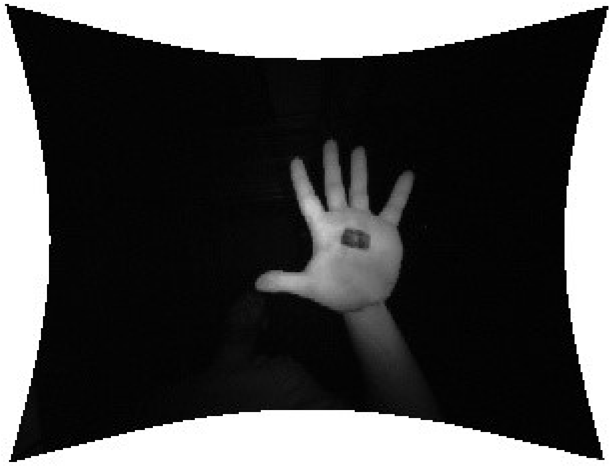

Figure 4.3: Corrected Image of Human Hand with a Simulated Wound.

OpenCV tool. The image displayed in the Figure 4.4 shows the hand image after cropping procedure.

4.3.2 Image Enhancement

To effectively identify the edge of the hand, we will sharpen the cropped image to better differentiate the human hand from the background in the infrared image. In contrast to the image enhancement in the previous chapter, we avoid the loss of wound information by using a sharpen approach instead. We achieve this sharpening of the image using Gaussian filtering. The sharpened image is shown in Figure 4.5.

Figure 4.4: Cropped Image of Human Hand with a Simulated Wound.

Figure 4.5: Sharpened Image of Human Hand with a Simulated Wound.

4.3.3 Background Suppression

In order to reduce the interference of background objects in correctly identifying the wound area, we take a 2-step approach. At first, we will identify the hand object. This is achieved by correctly locating the hand contour from the image. To achieve high accuracy, we will use a threshold based filtering process to suppress the background objects.

In the second step, we will only consider objects within the hand contour of the identified hand. As a result, any background object with similar intensity signature will not be miss-identified as a wound area.

From Figure 4.6, we could clearly see that the background is suppressed from the infrared image after the background suppression process.

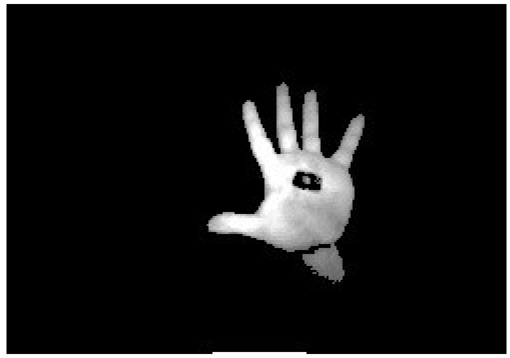

Figure 4.6: Human Hand with a Simulated Wound after Background Suppression.

4.3.4 Histogram Based Image Enhancement

We apply histogram to the above-enhanced image to make the edges and the area of the wound much clearer. The histogram analysis is applied to the entire hand area. Then the resulting optimum filter is applied to the hand area to obtain the new enhanced image. As clearly shown in Figure 4.7 the wound area is significantly distinguished out of the hand.

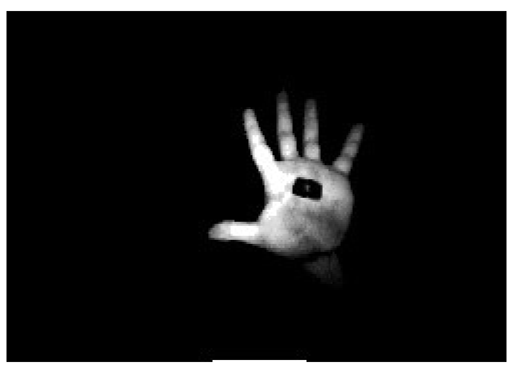

Figure 4.7: Human Hand with a Simulated Wound after Histogram based Image Enhancement.

4.3.5 Wound Area Identification and Extraction

Following the histogram based image enhancement, we find these contours around the human hand and the contours around the wound. We draw this hand and wound contours on to a new image using the 'drawContours' function provided by OpenCV as illustrated in Figure 4.8.

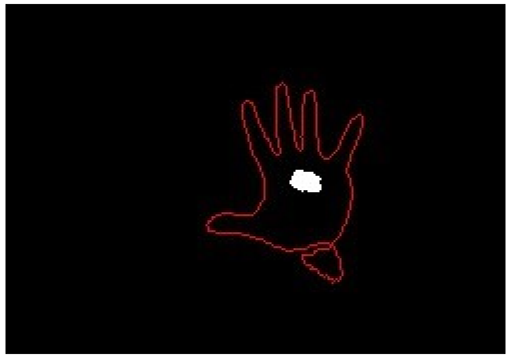

Figure 4.8: Sample Hand Image with Hand and Wound Contour.

There could be multiple wounds on the same hand. Our system is able to identify and segment out multiple wounds as shown in Figure 4.9.

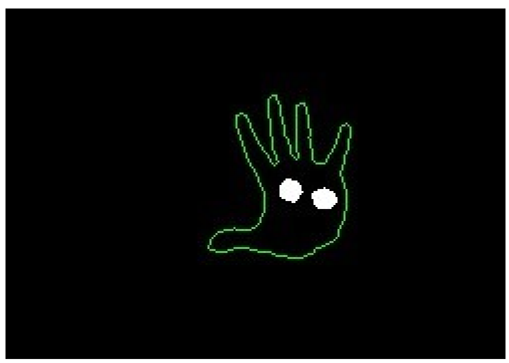

Figure 4.9: Sample Hand Image with Multiple Wounds.

4.4 Wound Area Estimation

In this wound area estimation, we find the area of the simulated wound on the human hand using optical equations based on the wound contours found as shown in Figure 4.8. To estimate the wound area we will map each pixel within the wound contour to its actual size based on the corresponding optical correlation. Our proposed approach is also capable of handling multiple wounds in the human hand. In case of multiple wounds, we calculate the aggregate of all the wound areas and display it.

4.5 Developement of a Prototype System

As mentioned earlier, our prototype can also calculate and display wound area, in percentage of hand surface area. For this to work, the user has to calculate his hand's surface area first by clicking the 'Surface Area' button displayed on the GUI. Figure 4.10 displays a sample screen shot of our prototype.

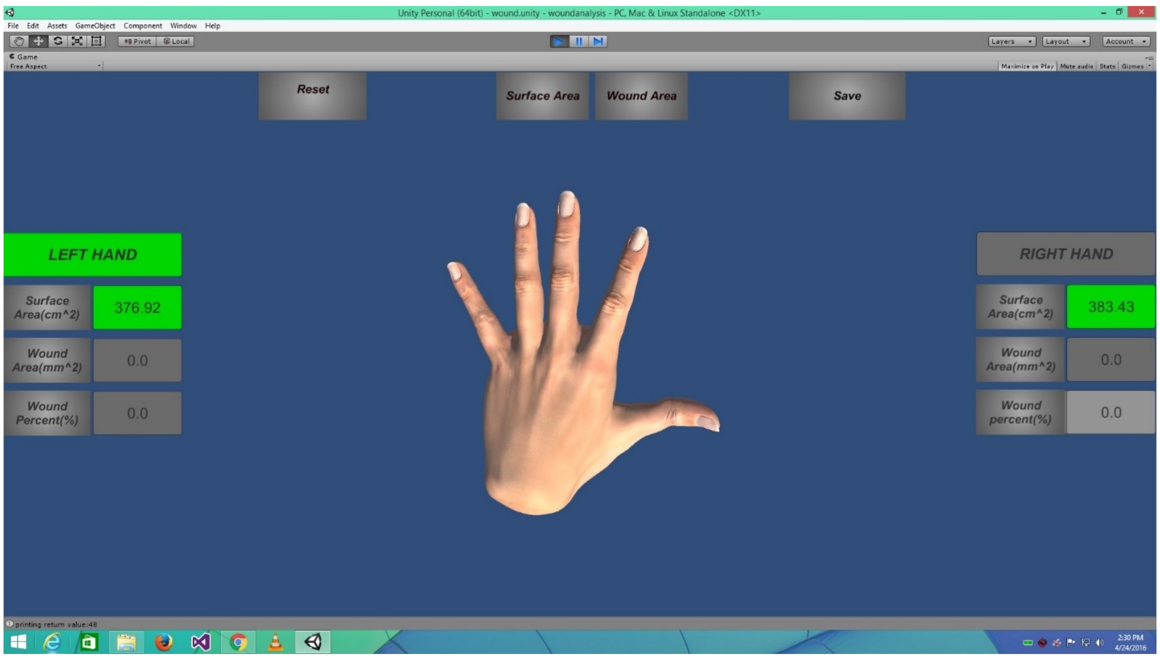

Figure 4.10: Prototype displaying Estimated Surface Area.

Later, once the user with a simulated wound in his hand places his hand and clicks on the 'WoundArea' button displayed on the GUI, the prototype system collects the input images from the Leap Motion controller, estimates the wound area and displays the estimated wound Area on the GUI. The built prototype also displays wound area as percentage of the whole hand's surface area. Our prototype has a 'Save' option in which the user can save the wound and surface area on to a text file in his desired location. Figure 4.11 shows the estimated area of a simulated wound on user's left hand.

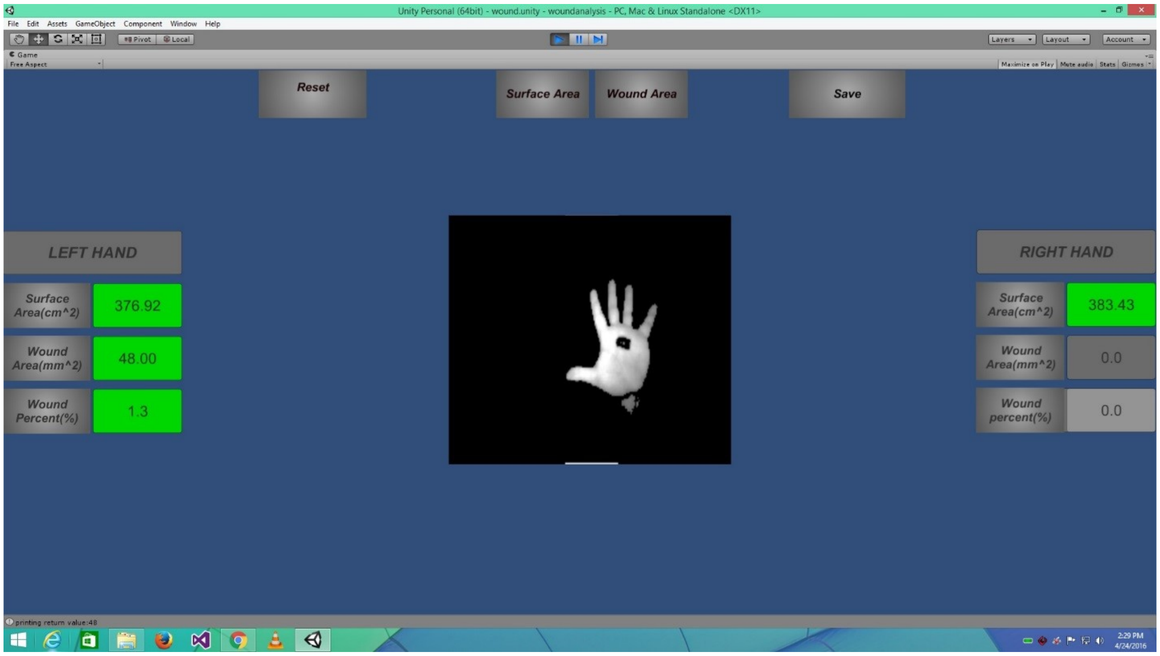

Figure 4.11: Prototype for Estimating Wound Area.

4.6 Experiment Performance Evaluations

In this section, we study the reproducibility and accuracy of measurements when using our prototype system to calculate the area of the simulated wound on a user's hand.

4.6.1 Measurements for One Simulated Wound on a Human Hand

At first, we will look at one simulated wound on a hand as shown in Figure 4.12. To test the accuracy of our prototype system, we glued a 2.5×2.5 cm black colored piece of paper on the user's hand. The results obtained by using our system are presented in Table 4.1.

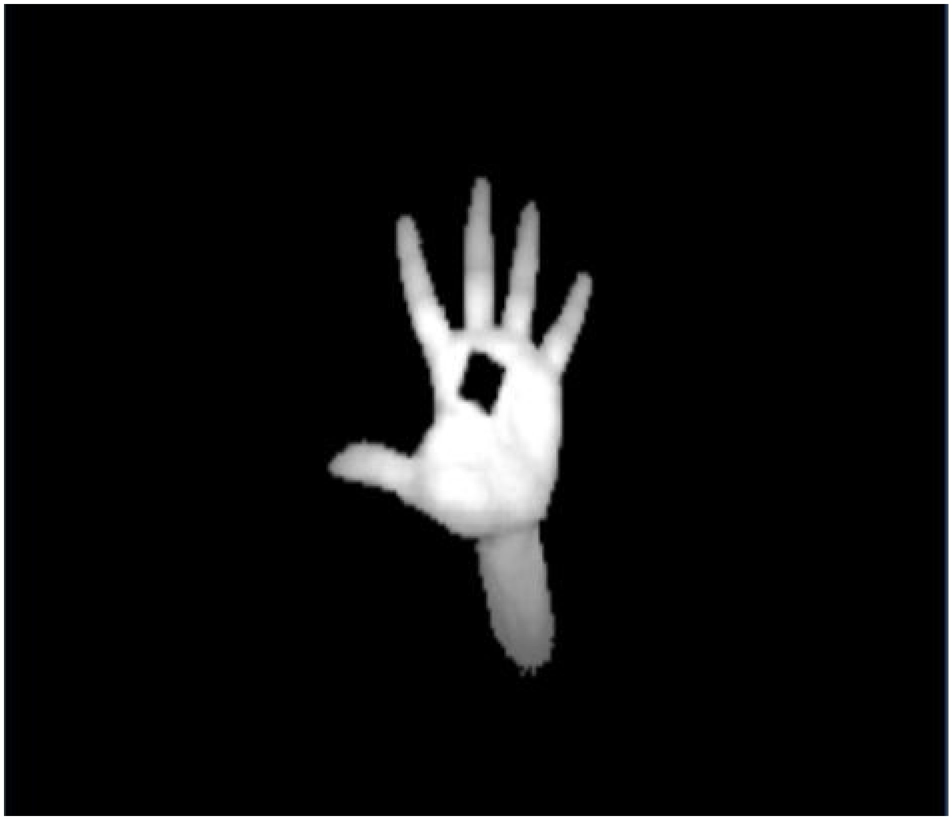

Figure 4.12: Simulated wound on User's Hand.

Take	Number of Wounds	Estimated Wound Area (sq.cm)
1	1	6.47
2	1	5.82
3	1	6.6
4	1	6.58

Table 4.1: Results for Wound Area Estimation of a Human Hand Prototype

Mean = 6.3675

Standard deviation = 0.3694

Standard deviation in percentage = 5.8%

4.6.2 Measurements for Multiple Simulated Wounds on a Human Hand

In this section, we will look at multiple simulated wounds on a hand as shown in Figure 4.13. To test the accuracy of our prototype system, we glued two 2.5×2.5 cm black colored pieces of paper on the user's hand. The results obtained by using our system are presented in Table 4.2.

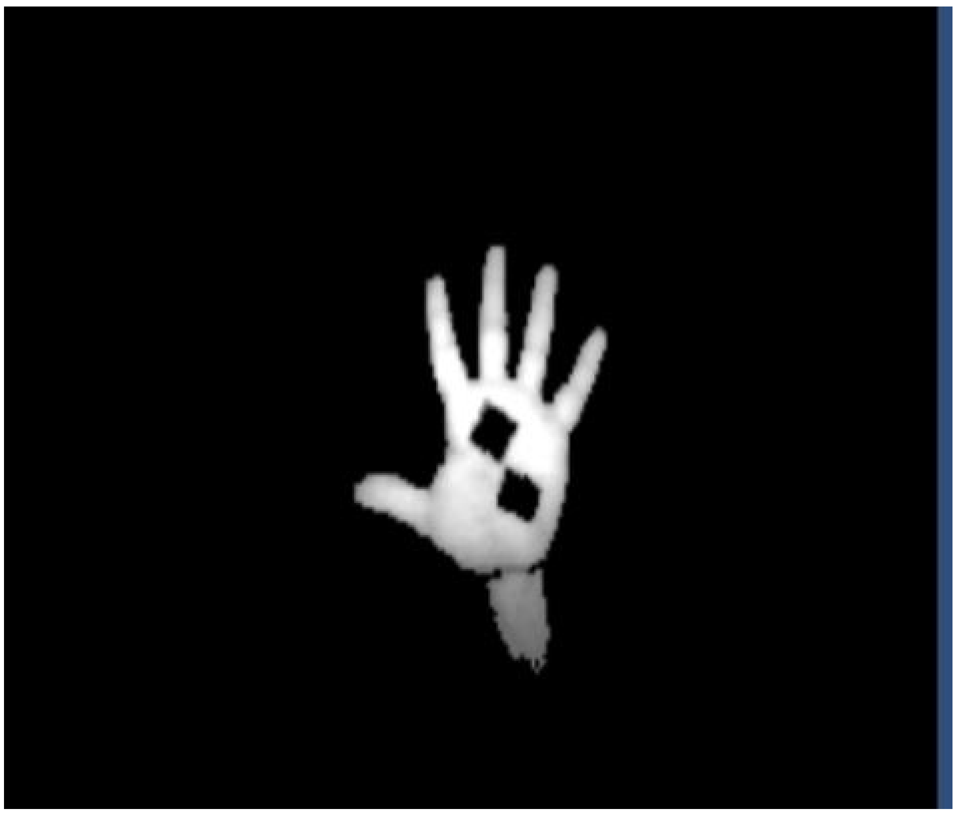

Figure 4.13: Multiple Simulated Wounds on User's Hand.

Take	Number of Wounds	Estimated Wound Area (sq.cm)
1	2	12.3
2	2	11.5
3	2	12.0
4	2	11.68

Table 4.2: Results for Wound Area Estimation of a Human Hand Prototype.

$$\text{Mean} = 11.87$$

$$\text{Standard deviation} = 0.3534$$

$$\text{Standard deviation in percentage} = 3\%$$

It is evident from Table 4.1 and Table 4.2 that the results obtained are consistent and accurate.

5 CONCLUSION AND FUTURE WORK

In this thesis, we have designed and developed a computer graphics and visualization based analysis and record system for hand surgery and therapy practice. In particular, we have developed the following novel approaches:

5.1 Model Based Data Compression

Some of the important and most critical challenges of the Electronic Health Record system include data storage and data privacy. To address these needs we came up with a new data compression technique which better helps electronic health record systems in solving challenges like data storage and data privacy.

In our model-based video compression, we extract and store the motion data of the patient's hand in a binary file. The hand surgeon or a hand therapist using our Visualizer can review the stored hand motion. The Visualizer replays the stored hand motion in the form of an animated video using a graphics hand model. To illustrate this idea, we have built a prototype, which demonstrates our entire work flow. Our experiment results have shown the effective compression performance as well as added benefit of 3-D review, enhanced privacy and review capability at different playback speeds.

5.2 Model Based Surface Area Estimation of a Human Hand

Surface area of a human hand can be used for exposure assessment in occupational toxicology, can be used in the development of manual equipment like the protective gloves in ergonomics, etc. We built a novel system, which is capable of estimating the surface area of a human hand accurately and quickly. In our system, we have collected the images of the user's hand; image processed them to get the required information and then customized a graphics hand object to match the user's hand based on the extracted information from the user's hand image. Once customized, we can calculate the area of the customized hand object to get the surface area of the user's hand. To illustrate this idea, we have built a prototype, which demonstrates the entire work flow. Our experiment results have shown considerable reproducibility and consistency.

5.3 An Emulated Study of Wound Area Estimation of a Human Hand

We propose a new approach to measure the area of the wounds on a human hand. As for now, we do not have access to a patient with a hand wound. Therefore, we used a Marker as a simulated wound on the user's hand in an emulated way to illustrate our ideas. To measure the area of the simulated wound, we get the infrared images of the user's hand using an imaging device, and process them using our wound analysis engine to identify and measure the area of the simulated wounds. To illustrate this idea, we have built a prototype, which demonstrate our entire work flow. Our experiment results have shown considerable reproducibility, consistency and accuracy.

In summary, we have developed prototypes for each of the approaches to illustrate our approach, and studied their performance and complexity. We believe these new approaches will significantly enhance the current practice in hand surgery analysis and therapy practice. In the future, we are planning to study and properly validate our prototype in a clinical study.

REFERENCES

[1] Leap Motion, "Api overview," the Leap Motion right-handed coordinate system. [Online]. Available: https://developer.leapmotion.com/documentation/java/devguide/Leap_Overview.html

[2] ——, "Api overview." [Online]. Available: https://developer.leapmotion.com/documentation/java/devguide/Leap_Overview.html

[3] ——, "Camera images," an image from one of the cameras. A grid highlighting the significant, complex distortion is superimposed on the image. [Online]. Available: https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Images.html

[4] Wikipedia, the free encyclopedia, "Data compression," May 2016, history of video compression standards. [Online]. Available: https://en.wikipedia.org/wiki/Data_compression#Video

[5] ——, "Electronic health record," April 2015. [Online]. Available: https://en.wikipedia.org/wiki/Electronic_health_record

[6] American Society of Hand Therapists, "Hand therapy," 2015. [Online]. Available: http://www.assh.org/handcare/About-Hand-Surgery/What-is-a-Hand-Therapist

[7] Mohammed Ghanbari, *Standard Codecs: Image compression to advanced video coding*, 3rd ed., ser. 54. London: Institution of Electrical Engineers, 2003.

[8] Joo-Young Lee, Jeong-Wha Choi, Ho Kim, “Determination of hand surface area by sex and body shape using alginate,” *Journal of PHYSIOLOGICAL ANTHROPOLOGY*, vol. 26, no. 4, pp. 475–483, 8 2007.

[9] Wikipedia, the free encyclopedia, “Leap motion,” 2016, technology. [Online]. Available: https://en.wikipedia.org/wiki/Leap_Motion

[10] Unity Technologies, “Unity documentation,” 2016, computing a Normal/Perpendicular vector. [Online]. Available: http://docs.unity3d.com/Manual/ComputingNormalPerpendicularVector.html

[11] Simen Andresen, “Cybernetics engineering and stuff,” 2013, hand Tracking and Recognition With OpenCV. [Online]. Available: http://simena86.github.io/blog/2013/08/12/hand-tracking-and-recognition-with-opencv/