

August 2020

Mobile Wound Assessment and 3D Modeling from a Single Image

Victor Williamson
University of Wisconsin-Milwaukee

Follow this and additional works at: <https://dc.uwm.edu/etd>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Williamson, Victor, "Mobile Wound Assessment and 3D Modeling from a Single Image" (2020). *Theses and Dissertations*. 2624.
<https://dc.uwm.edu/etd/2624>

This Dissertation is brought to you for free and open access by UWM Digital Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UWM Digital Commons. For more information, please contact open-access@uwm.edu.

MOBILE WOUND ASSESSMENT AND 3D MODELING FROM A SINGLE IMAGE

by

Victor Williamson

A Dissertation Submitted in
Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

in Engineering

at

University of Wisconsin, Milwaukee

August 2020

ABSTRACT

MOBILE WOUND ASSESSMENT AND 3D MODELING FROM A SINGLE IMAGE

by

Victor Williamson

The University of Wisconsin-Milwaukee, 2020
Under the Supervision of Professor Zeyun Yu

The prevalence of camera-enabled mobile phones have made mobile wound assessment a viable treatment option for millions of previously difficult to reach patients. We have designed a complete mobile wound assessment platform to ameliorate the many challenges related to chronic wound care. Chronic wounds and infections are the most severe, costly and fatal types of wounds, placing them at the center of mobile wound assessment. Wound physicians assess thousands of single-view wound images from all over the world, and it may be difficult to determine the location of the wound on the body, for example, if the wound is taken at close range. In our solution, end-users capture an image of the wound by taking a picture with their mobile camera. The wound image is segmented and classified using modern convolution neural networks, and is stored securely in the cloud for remote tracking. We use an interactive semi-automated approach to allow users to specify the location of the wound on the body. To accomplish this we have created, to the best of our knowledge, the first 3D human surface anatomy labeling system, based off the current NYU and Anatomy

Mapper labeling systems. To interactively view wounds in 3D, we have presented an efficient projective texture mapping algorithm for texturing wounds onto a 3D human anatomy model. In so doing, we have demonstrated an approach to 3D wound reconstruction that works even for a single wound image.

© Copyright by Victor Williamson, 2020
All Rights Reserved

This dissertation is dedicated to my parents Angela and Nolen Williamson.

TABLE OF CONTENTS

TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
1 Introduction	1
1.1 Problem Statement	1
1.2 Wound Assessment	3
2 Background & Related Work	9
2.1 3D Reconstruction From a Single Image	9
2.2 3D Wound Reconstruction	23
2.3 3D Wound Visualization on Mobile Phones	24
2.4 Mobile Wound Assessment	26
3 Machine Learning Wound Assessment	32
3.1 Unsupervised Wound Image Segmentation	32
3.2 Data-Driven Wound Image Segmentation	36
3.3 Data-Driven Wound Image Classification	41
4 3D Wound Modeling	46
4.1 3D Human Anatomy	46
4.2 3D Wound Texture Mapping	69
4.3 Intensity-Based Wound Depth Map	87
4.4 Limitations	88
5 Wound Assessment Mobile App	90
5.1 End-to-End Workflow	90
5.2 Neural Networks for Wound Assessment	91
5.3 Semi-Automatic Unsupervised Wound Segmentation	95
5.4 Mobile Application	100
6 Conclusion & Future work	105

BIBLIOGRAPHY	107
CURRICULUM VITAE	132

LIST OF FIGURES

2.1	3D-R2N2 vertebrae reconstruction	17
4.1	NYU Anatomy Labeling System	51
4.2	3D NYU Labeling System	53
4.3	3D Human Anatomy Model Front & Back	54
4.4	3D Human Anatomy Model Sides	54
4.5	3D Human Anatomy Model Top & Bottom	55
4.6	Anatomy Mapper Full Body	56
4.7	Anatomy Mapper Face	57
4.8	Anatomy Mapper Feet and Ears	58
4.9	Anatomy Mapper Hands	59
4.10	3D Human Anatomy Models	60
4.11	Final 3D Human Anatomy Model	61
4.12	3D Region Highlighting of Upper Chest	65
4.13	3-Point Image Correspondence	76
4.14	Wound 3D Intensity Displacement	88
5.1	Camera page	92
5.2	Wound segmentation	94
5.3	Wound segmentation w/ hole filling	95
5.4	Burn Wounds	96
5.5	Wound classification	97
5.6	Model I/O 3D Render	101
5.7	Web-based ThreeJS 3D Render	102
5.8	Home page	103
5.9	Wound inspection page	104

ACKNOWLEDGMENTS

First, I thank my advisor, Prof. Zeyun Yu, for his sincere guidance from start to finish; Dr. Jeffrey Niezgoda, of the AZH Wound Care and Hyperbaric Oxygen Center, for providing research direction; Dr. Angie Izard, for her application and fellowship recommendations; Amod Lagu, for mentoring me and for writing recommendations; Prof. Steve Ward, for advising me and for writing recommendations; Prof. Ethan Munson, for accepting me into UWM's PhD program and for writing recommendations; Prof. John Boyland, for homemade pizza and TA advising; Prof. Chao Zhu, for rounding out my math minor with probabilistic modeling; Prof. Jun Zhang, for providing a strong machine learning foundation; Dr. Gary Williams, for understanding my special need as a black male in higher education; Betty Warras, for answering my program related questions and for expeditiously processing departmental forms; Nakeesha Brown for handling my Program of Study form; Karen Collins and Desere Liddell, for having open arms in the Office of Graduate Diversity and Inclusion; All former and current AOP directors, May Kao Xiong, Thoy Bouakongxaya and Ramona C Tenorio, for working with my often unorthodox program schedule; Ramona Sledge and Henri-Etta Pryor, for providing a safe and comfortable place in the Black Cultural Center; Dr. Alice Brown-Collins, for her recommendations and mentoring since Undergraduate; Bishop Dr. Earl Parchia, for urging me to come home and calling me out for success; Suffragan Bishop Monica Parchia-Price, for continuing her father's legacy with Mt. Zion Nation; Prof. Susan McRoy, for successfully pleading on my behalf for an AOP nomination; Coach

Joe Sims, who affirmed my pursuit of higher education; Dr. Jacqueline Cook, for providing youth STEM programming; and last but not least, the UWM Graduate School and the AOP Fellowship, for financially supporting me for four of my six years as a PhD student.

1. Introduction

1.1 Problem Statement

Wound care can require extensive investments of time and money, leading to high emotional distress and patient suffering. The overhead of wound care costs leads to many wounds becoming ignored, misdiagnosed or untreated. Annual wound care costs in the U.S. have surpassed \$25 billion [1]. There are upwards of 6 million new U.S. wound patients every year, with that number expected to increase due to increased rates of diabetes, obesity and chronic diseases. This is compounded by aging baby boomers, rising health care costs and the uncertain future of Medicare and Social Security. Pressure ulcers and other chronic wounds are the most costly wounds to treat, owing to their prolonged and protracted treatment plans.

All over the world wound sufferers lack access to adequate health care facilities, with many hospitals being understaffed and lacking in basic supplies and equipment. This is particularly true in rural areas and traditional agrarian or hunter-gatherer societies. In densely packed cities across the developing world, water-borne diseases are exacerbated by

open sanitation systems, tropical flooding and pollution.

The silver lining in the challenges facing wound care is the large potential for mobile technology to streamline and enhance existing medical protocols, i.e. remote wound assessment. Particularly, camera-enabled smart phones enable patients and medical personnel to take pictures of wounds for storage and automated diagnosis. In so doing, medical costs can be reduced and remote access actualized.

However, wound images may not always be easy for automated software algorithms to analyse. Firstly, there are unknown camera parameters, such as the camera angle, zoom, focal length and optical distortion. Secondly, lighting in an uncontrolled environment may introduce glare and unwanted shadows. Image artifacts pose difficulties for automated segmentation and classification algorithms, and hinder 3D reconstruction.

Lastly, while it is ideal that multiple pictures be taken of the wound, in practice this is not always possible, making 3D reconstruction, which is an ill-posed problem, very difficult. However, when available, 3D data makes wound care more accurate by providing improved measurement capability and interactive visualization. Modern mobile technologies for 3D reconstruction and rendering can be integrated into mobile apps, with many mobile-based algorithms performing well enough for realtime applications.

1.2 Wound Assessment

Wound assessment is a broad term that refers to the various medical procedures and protocols pertaining to the review, diagnosis and treatment of wounds. In a typical clinical setting, this involves an initial evaluation that involves visual inspection, a physical exam and a question and answer. After first aid is administered, patients are diagnosed and treatment prescribed. Based on the wound type and severity, all of these steps can vary tremendously between different hospital units and medical practices. In this section we provide a brief overview of wound assessment to guide our discussion and to better understand how mobile technologies can be used to streamline wound care.

Wound Types

There are hundreds of wound types, but for our purposes a rough categorization classifies wounds into different types of skin lesions, skin injuries and ulcers. Skin lesions are wounds related to genetics, cancers, viruses, autoimmune disorders, allergies, etc. Skin injuries include lacerations, punctures, incisions, burns, etc. Ulcers are diabetic sores, chronic wounds, venous, arterial, pressure wounds, etc. While the technologies presented herein are applicable to all wounds, we place particular emphasis on chronic ulcers, owing largely to our medical expertise being derived from doctors who work primarily with venous and arterial wounds.

Skin Lesions

Skin lesions are generally viral and genetics related protrusions, scales and discolorations of the skin. In the context of mobile wound assessment, melanoma lesions are the most commonly assessed because of their visual simplicity enabling them to be easily consumable by automated software tools [2]. Other types of lesions including blisters, rashes, patches, cysts, nodules, scabs, boils and hives offer limitless expressions requiring multifaceted and holistic treatment programs.

Skin Injuries

On the morbid end of the scale, skin injuries may be traumatic, but they may also be surgical. Much fewer mobile wound assessment platforms assess traumatic injuries than those that assess lesions and ulcers. This is because ulcers and lesions, especially when applied to a narrow domain, exhibit similar morphology in terms of shape, color and progression. Automated algorithms are generally well behaved in these scenarios. One example of a morbid wound is burns [3, 4]. Burns have varying severity and can come from heat, chemicals, electricity, friction, radiation, etc. One example of a surgical wound is that left behind after heart surgery to implant a left ventricular assist device (LVAD) [5]. Doctors continuously monitor scars left behind at surgical sites in order to prevent infection and to ensure proper healing.

Ulcers

Ulcers are chronic wounds that can ooze and grow in size, and are symptomatic of some kind of underlying condition such as poor blood circulation, diabetes, bacterial infections, etc. Ulcers are particularly prevalent with the elderly, due to declining function and mobility, leading to scenarios ripe for developing pressure ulcers, arterial ulcers, venous ulcers, and pyoderma ulcers and gangrenous ulcers [2]. In this work, image examples are primarily drawn from skin ulcers.

Wound Analysis

The wound analysis phase of wound assessment is the manual inspection of the wound itself, either physically or by examining a picture of the wound. The goal of mobile wound assessment is to reduce the need for in person visits by sharing pictures of wounds for remote analysis. This reduces costs and commute times for the patient, and frees up physicians to prioritize the most urgent cases. Wound analysis involves measuring wound size, identifying tissues, taking culture swabs, noting orientation and specifying a location.

Tissue Type

For ulcers, which are chronic wounds, a common tissue analysis model is the so-called GSN. GSN is an acronym for granulation (G), slough (S), and necrotic (N) tissue [2]. Granulation tissue is the soft new tissue reddish in color created by the body that indicates wound healing. Slough tissue is a sticky, creamy, dull colored yellowish gray foam of dead tissue that is a

byproduct of inflammation during wound healing. Slough is a food source for bacteria and hence must be removed through a process known as debridement in order to optimize wound healing. Necrotic tissue is a hardened black colored layer of dead tissue that, like slough, must be removed via debridement.

Wound Size Measurement

The size of the wound indicates its severity and therefore serves as a reliable measurement of healing progression. This is true for lesions, skin injuries and ulcers. Measuring wound size is often an integral part of mobile wound assessment platforms. Within a 3D wound visualization platform like ours, 3D information can enhance the accuracy of automated wound size measurement. As an example, Mirzaalian Dastjerdi, et al. (2019) [6] obtain accurate wound surface area measurements by using Structure from Motion (SfM) to generate a 3D surface mesh from which the size of the wound can be measured. In circumstances where automatically making highly accurate wound measurements is too difficult, wound image area measurements can be compared over time to achieve effective wound healing tracking.

An example of how wound measurement is used for skin lesions comes from malignant melanoma. When a malignant lesion is suspected, it is removed through an excisional biopsy for diagnosis. If melanoma is diagnosed, a prognosis is initiated based on lesion thickness and composition, such as prescribing adjuvant treatment to suppress secondary tumor formation. In order to conform to the recommendations of the World Health Organization (WHO),

excisions must include at least 1 cm margins for lesions up to 1 mm of thickness, and 2 cm margins beyond that [7].

In the case of skin burn injuries, in situations of life threatening morbidity, the total body surface area burned (TBSAB) plays a crucial role from the onset, requiring accurate area and depth measurements for effective treatment [3]. Lindert & Tafazzoli-Lari (2018) [8] measure burn wound depth in children using optical coherence tomography (OCT). OCT is more ideal than CT for obtaining high resolution tissue images. In addition to size, researchers were able to identify wound surface irregularity, which includes loss of normal dermal papillary pattern, disappearing skin lines and changes to the skin's microvascular architecture. Singla, Srivastava & Mehta (2018) [8] incorporate size measurements, from OCT A-scan and B-scan images, into a machine learning classification algorithm. Using a generalized linear classifier, burn wounds are classified into one of two categories, less severe and more severe.

For diabetic foot ulcers, wound size and depth are critical factors to be considered in all of the major foot ulcer classification systems [9]. The Wagner-Meggitt classification system relies almost entirely on wound depth. The University of Texas system grades wound severity by wound depth and then does further gradation by the level of infection or ischemia. The SAD and PEDIS classification systems include both depth and size as two of the features in their five ulcer features classification system.

Wound measurements can be taken manually or digitally. Manual methods include acetate tracing, ruler or scalpel. Manual approaches run the risk of contamination, wound irritation and wound dressing disturbance. Digital measurements include adding references

within 2D images or using photogrammetry for full 3D reconstruction. As an example, Mirzaalian Dastjerdi, et al. (2019) [6] measure wound surface area by taking pictures of wounds alongside strips of taped checkered patterns on the patient. The checkered pattern has a known size and is later extrapolated onto the image of the wound. A third flavor of wound measurement techniques are semi-automated. Sánchez-Jiménez, et al. (2019) [10] built an application where physicians manually enter the distance between points on the wound from which then the entire area of the wound can be determined.

2. Background & Related Work

The foundation of our work is to apply methods of 3D reconstruction from a single image to 3D wound image visualization on mobile devices as part of a complete mobile wound assessment workflow. Firstly, we review a slew of techniques by which single photos of objects can be turned into 3D models. Then we review the growing body of literature specifically related to the creation of interactive 3D wound models. Finally, we review mobile-based 3D wound visualization and highlight issues specific to mobile devices and embedded systems.

2.1 3D Reconstruction From a Single Image

The ill-posed problem of 3D reconstruction from a single image has important implications in resource constrained settings where access to multi-view photos of a scene is not available. Particularly, single-view 3D reconstruction tends to be important for creating databases of 3D models for use in machine learning. In the area of wound assessment, it is often the case that nurses, physicians or even patients have only taken a single wound picture, or that a single picture is all that's available from another clinic.

Using 3D Camera Self-Calibration

Many single-view 3D reconstruction algorithms begin by estimating camera parameters which include focal length, optical distortion, principle point and view direction. The most common method is to identify lines that converge at vanishing points in the image, making this technique very popular for images of cityscapes and man made objects. Wilczkowiak, Boyer & Sturm (2001) [11] use a semi-automated approach by having users manually identify parallelepipeds in the image, from which camera parameters are estimated and 3D urban scenes reconstructed. Liu, Liu & Wang (2016) [12] use camera distortion, vanishing points, ellipses and coplanarity for self-calibration to reconstruct urban scenes. They identify straight lines and an ellipse in the image to determine the radial distortion, focal length and principle point. The parameters are iteratively refined as more accurate principle points are found. In the most general case, a pinhole camera model for perspective projection of 3D scenes is defined by the following matrix

$$\mathbf{P} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \quad (2.1)$$

And the project from 3D to 2D is defined as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.2)$$

We are using 3D homogeneous coordinates and therefore it is sufficient to calculate the entries of P by setting the following cross product to zero

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \times \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0 \quad (2.3)$$

The rank of the matrix in 2.3 is two, so we can remove the third row. With $\vec{\mathbf{V}} = (X, Y, Z, 1)$, and setting the rows of P as $\vec{\mathbf{p}}_1$, $\vec{\mathbf{p}}_2$ and $\vec{\mathbf{p}}_3$, this cross product can be rewritten as

$$\begin{bmatrix} y\vec{\mathbf{p}}_3^T \vec{\mathbf{V}} - \vec{\mathbf{p}}_2^T \vec{\mathbf{V}} \\ \vec{\mathbf{p}}_1^T \vec{\mathbf{V}} - x\vec{\mathbf{p}}_3^T \vec{\mathbf{V}} \\ x\vec{\mathbf{p}}_2^T \vec{\mathbf{V}} - y\vec{\mathbf{p}}_1^T \vec{\mathbf{V}} \end{bmatrix} = \vec{\mathbf{0}}_3 \quad (2.4)$$

Note that $\mathbf{p}_i = \mathbf{0} \forall i$ is a solution. We can augment the system with constraints. For example, if we have a linear system $\mathbf{Ax} = \mathbf{b}$, then we can augment with $\mathbf{c}^T \mathbf{x} = x_0$ as follows

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{c}^T \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c}^T x_0 \end{bmatrix} \quad (2.5)$$

When fully expanded to include n pairs of points and constraining $\sum_i^p i = 1$, we can set

\mathbf{M}

$$\begin{bmatrix} \mathbf{0}_4^{\vec{T}} & -x_1 & -y_1 & -z_1 & -1 & X_1 y_1 & Y_1 y_1 & Z_1 y_1 & y_1 \\ x_1 & y_1 & z_1 & 1 & \mathbf{0}_4^{\vec{T}} & -X_1 x_1 & -Y_1 x_1 & -Z_1 x_1 & -x_1 \\ & & & & \vdots & & & & \\ \mathbf{0}_4^{\vec{T}} & -x_n & -y_n & -z_n & -1 & X_n y_n & Y_n y_n & Z_n y_n & y_n \\ x_n & y_n & z_n & 1 & \mathbf{0}_4^{\vec{T}} & -X_n x_n & -Y_n x_n & -Z_n x_n & -x_n \\ & & & & \mathbf{1}_{12}^{\vec{T}} & & & & \end{bmatrix} \quad (2.6)$$

Setting $\mathbf{P}_{12} = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}]$ we can calculate P with

least squares using normal equations as follows

$$\mathbf{M} \begin{bmatrix} \mathbf{P}_{12} \\ 1 \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{0}}_{\mathbf{n}} \\ 1 \end{bmatrix} \quad (2.7)$$

We use normal equations to solve least squares as follows

$$\mathbf{M}^T \mathbf{M} \begin{bmatrix} \mathbf{P}_{12} \\ 1 \end{bmatrix} = \mathbf{M}^T \begin{bmatrix} \vec{\mathbf{0}}_{\mathbf{n}} \\ 1 \end{bmatrix} \quad (2.8)$$

Numerical solutions can be applied to solve for the specific rotation and translation components of the camera projection matrix as follows

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{array} \right] \quad (2.9)$$

3D Deformable Models

3D deformable modeling has been a popular approach to generate 3D models from 2D views. They are in essence 3D models with explicit key points of interest, or landmarks, identified so as to more easily be able to categorize the general shape of a similar 3D model. Typically, a group of 3D models are matched against a Gaussian model, with a mean position and variance, to identify the variations of each landmark. Where each approach differs is in how they estimate the conditional probability distributions for either the likelihood formula, the posterior conditional distribution, or both.

Kar, Shubham, Carreira & Malik, 2015 [13] create a likelihood model based on the non-rigid structure from motion (NRSFM) algorithm, which is maximized using the *expectation-maximization* (EM) algorithm. Taking a silhouette of an object as input, the Chamfer distance is minimized between the silhouette and the orthographic or perspective projection of the stored 3D models. The Chamfer distance measures the closeness between sets of points. For example, for sets S_1 and S_2 , The Chamfer distance, d_CD is calculated as

$$d_{CD} = \frac{1}{n} \sum_{p \in S_1} \min_{q \in S_2} |p - q| \quad (2.10)$$

Romdhani & Vetter, 2005 [14] estimate the 3D shape, texture, pose and lighting from a single human face image by the stochastic Newton optimization algorithm. Each vertex position is assumed to have a Gaussian distribution from the mean. The step formula for stochastic Newton optimization on function $f(x)$ is as follows

$$x_{k+1} = x_k - \left(\frac{1}{n} \sum_{i=1}^n f''(x_k) \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^n f'(x_k) \right) \quad (2.11)$$

Boosting

Boosting is a machine learning strategy that improves classification accuracy by stringing together a sequence of weak learners, such as simple decision trees. Ada Boost is the first and probably the most popular of these boosting algorithms. Creusot, Pears & Austin, 2013 [15] use Ada Boost to learn the landmarks of 3D facial models. Like deformable modeling, landmarks are used to reduce the overall complexity of the system. The Boosting trains on two classes, neighboring and non-neighboring vertices, and generates a landmark score for each vertex. Benhabiles, Lavoue, Vandeborre & Daoudi, 2011 [16] use Ada Boost to segment 3D models by training on a large database of manually segmented 3D meshes. The inputs are pools of geometric features identified by software. The final segmentation results are averaged from the ground-truth repository of 3D models. Kalogerakis, Hertzmann & Singh, 2010 [17] combine so-called Joint Boosting with conditional random fields (CRF). The trained CRF objective function is used to segment the models and joint boosting is used to label geometric features.

Random Forest

Like boosting, random forests use more than one underlying learning function. However, instead of being weak learners strung together, random forests run each learner individually and outputs their collective result. Shotton, et al., 2013 [18] use an ensemble of randomized decision trees to estimate the positions of 3D joints for human pose recognition. Rock, et al. 2015 [19] use random forests as a hashing function to partition a training set of 3D models into similar groups based on image features. The 3D meshes are rendered from multiple viewpoints to match against a number of potential projections. To construct a 3D model, thin plate splines (TPS) are deformed according to a constrained, symmetry-preserving energy minimizing function. Minimal energy surfaces have long been used to generate 3D models. Given a parametric surface $\mathbf{s}(u, v)$, forces \mathbf{f} , and weights w_i , Gibson & Mirtich (1997) [20] propose one such minimal energy surface

$$E = \int w_1 ||\mathbf{s}_{\mathbf{u}}||^2 + 2w_2 \mathbf{s}_{\mathbf{u}} \cdot \mathbf{s}_{\mathbf{u}} + w_3 ||\mathbf{s}_{\mathbf{v}}||^2 + w_4 ||\mathbf{s}_{\mathbf{uu}}||^2 + 2w_5 ||\mathbf{s}_{\mathbf{uv}}||^2 + w_6 ||\mathbf{s}_{\mathbf{vv}}||^2 - 2\mathbf{f} \cdot \mathbf{s} \quad du \, dv \quad (2.12)$$

Convolutional Neural Networks

Researchers are becoming increasingly audacious in their attempts to generate 3D models from 2D images using convolution neural networks (CNN). Truth be told, it is very difficult to generate a 3D model from a neural network, which are best suited for simple classification

and regression exercises. Nevertheless, Jackson, Bulat, Argyriou & Tzimiropoulos, 2017 [21] have made incredible strides using CNN's to generate 3D reconstructions of human faces. Using only a single image of a human face, it generates a textured 3D volume of the face. Their network is trained on over 60,000 face image-3D model pairs. Delanoy, Bousseau, Aubry, Isola & Efros, 2017 [22] have created a CNN that maps hand drawn line drawings to 3D voxels. As more line drawings are drawn from different angles, a more accurate 3D model is produced.

CNN's are also used in sketch-based systems. Sketch-based interfaces automatically generate 3D models as users draw. Han, Gao & Yu, 2017 [23] have built a sketch-based interface using CNN for interactive line drawing of 3D caricatured human face meshes. This approach outputs vertices directly which is more difficult than outputting 3D voxels. In a similar vein, sketch-based interfaces can be used to select from or match against an existing set of 3D models. The system of Wang, Kang & Li, 2015 [24] apply the so-called siamese CNN to match line drawings to the nearest looking 3D model in the database. For pose recognition, Wohlhart & Lepetit, 2015 [25] use a CNN to learn 3D pose descriptors from 2D images. The final 3D pose is determined by minimizing the euclidean distances between the pose descriptors of nearest neighbors. The minimization function is a measure of pose similarity between the image and the pre-trained models.

Using CNN's to reconstruct man-made objects have shown much promise. Choy, et al., 2016 [26] were able to train on hundreds of thousands of models from ShapeNet, PASCAL 3D, Online Products and MVS CAD Models. The so-called 3D Recurrent Reconstruction

Neural Network (3D-R2N2) takes in one or more images of a single object such as a car, couch, computer, etc. and produces a voxel volume that matches the object. A multi-view sequence of images of an object works best, but a single image also works. We even tested 3D-R2N2 on vertebrae with descent results as seen in figure 2.1. Dosovitskiy, Springenberg, Tatarchenko & Brox, 2017, [27] reconstruct 3D models into voxel volumes from a pre-trained CNN of thousands of chairs, cars and tables 3D models. Both approaches manually generate 2D renderings of 3D models in order to train their neural networks.

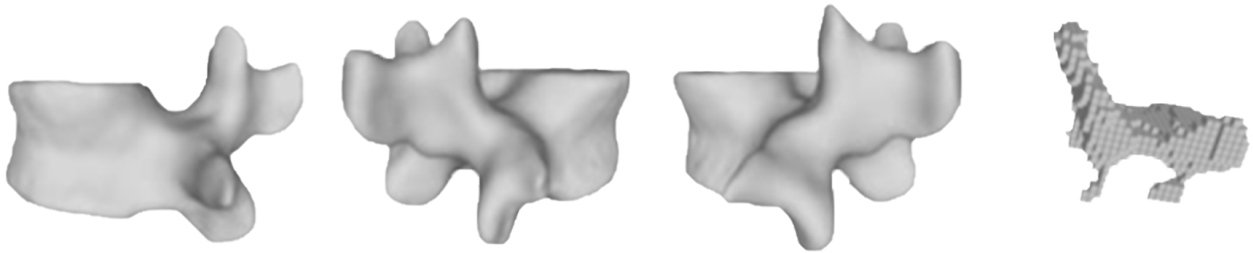


Figure 2.1: 3D-R2N2 vertebrae reconstruction

Among other applications, Han, Li, Huang, Kalogerakis & Yu, 2017 [28] use a CNN with long short-term memorized context fusion (LSTM-CF) to infer global structure about a malformed 3D model. This information is used to fill in gaping holes on the model surface with patches. Guo, Zou & Chen, 2015 [29] use a CNN to segment and label the various parts of a 3D model. The CNN is trained on a large pool of classical geometric features. The output is a label vector for each triangle that indicates its probability of belonging to each feature. Machine learning can also be used to generate 2D images from 3D models. Nalbach, Arabadzhiyska, Mehta, Seidel & Ritschel 2017 [30] do just this by training 3D models on their corresponding shading buffers during rendering. The CNN, six layers deep, is then

used to estimate the intensity for each image pixel for an input model.

Probabilistic Modeling

Probabilistic modeling systems leverage probability distributions and can be incorporated into 3D reconstruction algorithms. Rezende, et al., 2016 [31] parameterize a so-called probabilistic neural network with conditional distributions to generate 3D volumes and 3D models. The neural network is further powered by a short-term memory network (LSTM) and trains on the very large dataset of 3D models available from ShapeNet. Delage, Lee & Ng, 2006 [32] use a so-called bayesian neural network to generate 3D scenes from 2D images. Bayesian neural networks include additional conditional probabilities between neurons. These conditions constrain the model and helps the network to resolve ambiguities during 3D reconstruction.

Markov models are suitable for 3D reconstruction because they can be used to deform 3D models between state transitions. Mesh-based 3D deformations easily satisfy the Markov property because deformations to the mesh only depend on the current topological state of the vertices and faces that make up the mesh. In general, for a stochastic process $X = X_n : n = 0, 1, \dots$, and a given current state $X_n = i$, the probability that the next state $P(X_{n+1}) = j$ depends only on the current state i is

$$P_{i,j} = P(X_{n+1} = j | X_n = i) = P(X_{n+1} = j : X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) \quad (2.13)$$

Ashutosh, Sun & Ng, 2009 [33] generate 3D scenes by applying a Markov Random Field (MRF) to small homogeneous patches of the input image. It is a supervised learning ap-

proach that assimilates depth cues and ties relationships between different parts of the image. Benameur, Mignotte, Labelle & De Guise, 2005 [34] use a first order Markov processes to differentiate patients with normal spines and those with scoliosis based on 3D scan data. This is accomplished by building state transitions that deform the 3D mesh incrementally until the correct classification is reached. The process handles both aberrations to the spine shape and deformities to individual vertebrae. As is typically done, the principal component analysis (PCA) is applied to reduce the complexity of the 3D model. For a large set of data in matrix form, \mathbf{M} , of size $m \times n$, we can lower its dimension by multiplying by a column vector, w of size n to obtain a new set of data, \mathbf{m} , as follows

$$\mathbf{m} = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^T \mathbf{M}^T \mathbf{M} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \quad (2.14)$$

Humbert, Guise, Aubert, Godbout & Skalli, 2009 [35] use inferencing to reconstruct 3D models of vertebrae from orthogonal x-ray images. Inferences are made based on the geometric relationships between different vertebrae. This is a hierarchical approach and breaks down 3D vertebrae construction into two phases, firstly, by looking at the spine as a whole and then by identifying individual vertebrae. A similar statistical-based approach is taken by Cresson, et al., 2009 [36] and Zheng, Ballester, Styner & Nolte, 2006 [37] except that images are not x-rays, but rather calibrated fluoroscopic images. Aguiar, Theobalt, Stoll & Seidel, 2007 [38] apply Gaussian distribution-constrained 3D deformations that best match images taken from video of human performances. Balan & Black, 2008 [39] apply statistical priors to constrain 3D deformations when matching images of silhouettes.

Sketch-Based 3D Modeling

Sketch-based interfaces and modeling (SBIM) have gained traction over the years as a way to combine traditional drawing media with 3D modeling. We know that 3D modeling is often the bottle neck in terms of the amount of time and resources required. As of yet, SBIM does not compare to the advanced level of quality producible from industry-level modeling programs, but it is a great research domain for prototyping and proof of concepts, and can be adequate in domains with less rigorous modeling requirements.

Rivers, Durand & Igarashi (2010) [40] introduce a way to model complex man-made 3D models by combining an unbounded number of models drawn in 2D from up to three orthogonal views. Overlapping objects are combined in 2D using Constructive Solid Geometry (CSG) operations. This is achieved by first discretizing all lines drawn as polylines. For each resultant segment, an orthogonal plane that contains the segment and extends into the viewing direction is used to generate an individual surface facet on the plane. The shape of these 2D surface facets are determined from 2D CSG operations of all intersections from the silhouette cylinders on the plane. The collection of surfaces define a 3D shape. These surfaces are further subdivided into quadrilaterals. The mesh is smoothed by approximating their least-variation-of-curvature that minimizes variation of the laplacian magnitude of the two directions of principle curvature using the iterative approach of Nealen, Igarashi, Sorkine & Alexa (2007) [41]. Mesh shrinking during smoothing is precluded by fixing selected vertices to the rim of the silhouette visual hull, which is accomplished by re-tessellating after

each smoothing operation. In comparison to Google SketchUp, ten participants modeled 3 objects alternating between their approach and Google SketchUp, and ten other participants ranked the visual appeal of each model, finding that their approach prevailed with statistically significant confidence.

Li, Lee, Zhang & Jiang (2016) [42] snap drawn curves to the contours of supplied images that are first extracted as curve points using non-maximal suppress on secondary derivative from Cheng, 2009, and then segmented into disparate curves along sharp corners. Final sketched curves are selected by aligning the portions of the sketched line that correspond to image contours via energy-minimizing spline active contour models. The final sketched curves are then smoothed and closed using spline fitting to a third degree B-spline as specified in Liu, Ma, Zhang & Liveforce (2011) [43]. Finally a 3D model is created interactively using the EasyToy 3D modeling software.

Buchanan, Mukundan & Doggett, 2013 construct 3D character models from 2D cartoon characters that are facing forward without self-occluding limbs or body parts in an image with a lightly textured background. The character silhouettes are extracted from the image using the *potrace vectorization* algorithm of Selinger (2003) [44]. 2D skeletons are then extracted from these outlines by using density-conserving mesh smoothing followed by contracting points into a 1D skeleton along their normal directions and using feature size to control skeleton granularity as described by Willcocks & Li (2012) [45]. The skeleton is tested for symmetry using the pose-neutral symmetry axis, segment, junction classification scheme of Pantuwong & Sugimoto (2012) [46], followed by estimating the character’s offset angle

from center facing by calculating center line projections of joints. Arcs are generated from each outline vertex toward a bone centered point weighted according to distance and normal deviation of the nearest and normal oriented bone points. A 2D cross section profile is created by lofting a series of arcs lengthwise along the skeleton extending from the outline to the bone according to the curvature of the outline. The resultant non-overlapping cross-sections are then extruded into 3D. After extrusion, holes in the surface along the skeleton are filled in using ear clipping polygon fill of Eberly (1998) [47]. Additional joints are added along forks and inflection points in the skeleton and projected into 3D using the aforementioned offset angle from center facing. Each vertex is skinned using the weighted contribution of distance and dotted normal of the four nearest bones. The generated model is texture mapped directly from the character colors on the input image.

Prasad, Zisserman & Fitzgibbon (2006) [48] construct smooth 3D surfaces from the outline contours of 2D objects used as constraints to minimize the thin-plate energy smoothness objective function over the central finite difference approximations for the first and second derivatives of the parametric surface representation. The parametric surface is discretized into a grid of $m \times n$ matrices for each of the three axes of space. Images are segmented using the class-based object recognition *OBJCUT* algorithm.

More recently, machine learning is being used for sketch-based modeling. When sketching hair to produce 3D modeled hairstyles, Shen, Zhang, Fu, Zhou and Zheng (2019) [49] use the laplacian operator in their loss function to constrain voxel-based 3D volume produced from the S2ONet and O2VNet deep neural networks, and \tilde{y}_i ground truth, as follows

$$loss = \sum_i (\Delta(y_i) - \Delta(\tilde{y}_i))^2 \quad (2.15)$$

$$\Delta(y_i) = \sum_{j \in N_i} \frac{1}{N_i} (v_j - v_i) \quad (2.16)$$

Smirnov, Bessmeltsev and Solomon (2019) [50] use deep learning to create a 3D model made of parametric surface Coons patches from a single sketch of man-made shapes. To deal with minimal 3D training data, the data is augmented with random isometries obtained via rotation, reflection and permutations of control points.

There are a number of sketch-based 3D modeling programs that use extrusion operations to generate 3D models from hand drawn sketches. Rossa, Camozzato and Hocevar (2016) [51] generates a maze of walls from its hand drawn counterpart by extruding the walls vertically.

2.2 3D Wound Reconstruction

Wilczkowiak, Boyer and Sturm (2001) [11] reproduce 3D models of wounds using photogrammetry on two wound images. Zenteno, et al. (2018) [52] find that generating 3D wounds from structure from motion (SfM) is just as reliable for measuring the volume of cutaneous leishmaniasis ulcers as traditional 3D scanning approaches. Villa, Flies and Jacobsen (2018) [53] combine internal and external data from CT scanning and camera-based photogrammetry to create a 3D model of a wound and the surrounding anatomical regions. When photogrammetry is used, such as in Orun, Goodyer and Smith (2018) [54], we can use nor-

mal equations to track 3D surface points Δx_p with camera parameters Δx_s , where t_p and t_s are right-hand sides of the normal equations, as follows

$$\begin{bmatrix} N_p & N_{ps}^T \\ N_{ps} & N_s \end{bmatrix} \begin{bmatrix} \Delta x_p \\ \Delta x_s \end{bmatrix} = \begin{bmatrix} t_p \\ t_s \end{bmatrix} \quad (2.17)$$

Wahabzada, et al. (2017) [55] manually built 3D models of wounds from collage matrix cell cultures, and then used Hyperspectral Imaging (HSI) for unsupervised evaluation of wound healing over time. K-means clustering was employed by first projecting the images to 1D and splitting along the point c_i , where s_i are the projection points, that minimizes the following

$$c_i = \frac{1}{i} \sum_{j=1}^i (s_j - u_1) + \frac{1}{n-i} \sum_{j=i+1}^n (s_j - u_2) \quad (2.18)$$

2.3 3D Wound Visualization on Mobile Phones

Cheah, et al. (2018) [4] use the 3D Burn Resuscitation (3D Burn) mobile app to aid in the measurement of surface burn area, known as total body surface area burned (TBSAB). To do so, physicians digitally paint in 3D directly onto a 3D human model. The 3D Burn mobile app shows improvements in burn area measurement accuracy over the Rule of Palm method, Rule of Nines and the Lund and Browder chart. However, the process is slower because end-users have to digitally paint in 3D on a small device, which is tedious, and the 3D human anatomy models are generic, and do not reflect patient anatomy precisely.

A number of wound treatment applications have arisen that allow both physicians and patients to take pictures of their wounds with a smart phone. Kumar, et al. (2019) [56] compare multiple ways of generating 3D models from 3D point clouds obtained using Google ARCore and from unspecified Structure from Motion (SfM) libraries, and found that Poisson surface mesh construction is the most optimal. Ballester, et al. (2017) [57] uses mobile phones to capture two views of a human being, frontal and sagittal, and performs 3D reconstruction using a database of 3D human body models parameterized in shape and posture using Principal Components Analysis (PCA). The accuracy of the reconstructions can then be measured using mean absolute differences (MAD) between n source models, m_s , and reconstructed models, m_r , pairs

$$\frac{1}{n} \sum_i |m_s^i - m_r^i| \quad (2.19)$$

For testing, Ballester, et al. (2017) [57] use real 3D figurines, which can be augmented by tacking multiple pictures, say r , of each figurine and measuring differences between all $\binom{r}{2}$ pairs

$$\frac{1}{n} \sum_i \frac{1}{\binom{r}{2}} \sum_{s=1}^{r-1} \sum_{t=s+1}^r |m_s^i - m_t^i| \quad (2.20)$$

2.4 Mobile Wound Assessment

Background

Mobile wound assessment has grown by leaps and bounds with the proliferation of smart-phones. Their primary objective is to reduce costs by minimizing trips to the clinic and streamlining diagnosis and treatment. Many studies validate the effectiveness of mobile wound assessment [58, 59, 60, 61, 62, 63, 64, 65]. Kulikov, Sandhu & Van Leuven (2019) [66] performed a study showing that mobile wound care generally led to more timely wound assessment, diagnosis and treatment. Lo, Chang, Sen, Chuang & Chen (2015) [67] showed that one wound app shortened the average working time per patient by over twenty minutes.

Remote wound assessment is optimal when applied to specific scenarios such as those that target a specific type of wound, or a particular setting or circumstance. Salome, Alves, Alves, Dutra & Dutra (2019) [68] recommend wound cleaning procedures for different wound types. Patients use the MyFootCare app to take pictures of their wound in between wound dressings to allow clinicians to remotely track their healing progress [69]. Tolins, et al. (2019) [70] built an application to follow up with patients who were admitted to the ER because of a wound. Patients with LVAD heart implants have to be telemonitored to diagnose and prevent infection at driveline exit sites [5]. Other mobile apps target assisted living communities [71].

The most difficult aspect of mobile wound assessment is taking quality wound photos. For self-care, taking a clear photo of one's own body is non trivial. Gunter, et al. (2016) [72]

evaluated the mHealth mobile app and found that the most difficult task for postoperative patients was to take pictures of their surgical wounds. Taking pictures of wounds is easier when the mobile app's targeted end users are nurses. One such app is Nurstrial®, which gives nurses a handy way to monitor wound healing evolution [73]. In such cases, the wound picture is centered under ambient light leaving nominal margins between the wound and view finder window. Without ideal shooting conditions and clear visibility, automatic image processing tools become more complex and error prone.

Measuring Wound Size

A key metric to diagnosing wound types and tracking wound healing is to make accurate wound size measurement [74, 75]. It can be as simple as calculating the ratio of wound pixels to non wound pixels [76]. When measurements are calculated automatically on a smartphone, it saves additional trips to the clinic and minimizes invasive in-person wound measurement [77, 78]. Au, Beland, Anderson, Sasseville & Wang (2019) [79] found automated measurement done by the Swift App to be 57% faster than using the ruler method and paper charting. One of the most popular mobile wound assessment apps, Wound Vision, specializes in automated wound size measurement and mobile wound healing tracking [80]. Capturing 3D information about the wound leads to more accurate segmentation results. For example, The iDr. app uses structure from motion (SfM) to measure wound area and volume [81, 82], and Liu, Fan, Guo, Mo, Chang & Xu (2019) [83] use photogrammetry to improve measurement accuracy.

Local Peripherals

More complex systems attach peripheral devices such as sensors or leverage local area networking on the phone to aid in wound assessment. Jun, et al. (2019) [84] make three-dimensional wound measurements with the 3DWMS system via an attached laser-assisted sensor camera. Pei, et al. (2019) [85] use the phone's bluetooth for in-situ and real-time monitoring of wound bio-impedance. Lu, Yee, Meng, Harmon, Hinduja & Yi (2018) [86] use an attached thermal imaging sensor to provide augmented wound modeling for improved diagnosis. Kaile, Leiva, Mahadevan, Miguel, Vishwanathan & Godavarty (2019) [87] use a near-infrared (NIR) imaging device to measure tissue oxygenation to determine wound healing status.

Storage and Privacy

Medical records in the form of electronic medical records (EMR) and patient care management systems (PCMS) are protected under the Health Insurance Portability and Accountability Act (HIPAA) enacted in 1996. Mobile wound assessment patients reported a few concerns regarding privacy and security, but overwhelmingly supported mobile wound assessment because of its low cost and convenience. Sikka, Carlin, Pines, Pirri, Strauss & Rahimi (2012, May 2) [88] found during their trials of remote laceration ED determination that most patients were comfortable taking pictures of their wounds and sending them to physicians electronically. Santamaria & Kapp (2013) [89] argue for all electronic patient

records to be stored centrally as a big data repository, which would save costs by coalescing disparate software systems, but would also create a single point of failure for cyber attacks and software failures.

Research corroborates that sharing data remotely doesn't compromise physician performance. Chanussot-Deprez & Contreras-Ruiz (2008) [90] found that sending chronic wound images remotely allowed experts to provide rapid evaluation, diagnosis and treatment. Braun, et al. (2005) [91] use Cohen κ statistics to measure high levels of agreement between remote and face-to-face wound evaluations. Tsai, Pong, Liang, Lin & Hsieh (2004) [92] in their study showed 80%, 76%, 66% and 74% agreement among physicians when remotely diagnosing gangreen, necrosis, erythema and cellulitis or infection cases respectively.

Wound images captured on smartphones are securely transferred over the Internet and stored remotely for retrieval by an expert at a later date. Wallis, Fleming, Hasselberg, Laflamme & Lundin (2016) [93] built an app that allows practitioners to send patient wound images from a clinic to an off site burns expert. Dalya & Shedge (2016) [94] upload wound images to a database where they are segmented, preliminarily diagnosed and expert reviewed. CliniCam stores wound images as annotated Portable Document Format (PDF) documents in the EHR [95]. As demonstrated by Ye, J., et al. (2016) [96], wound data may also incorporate streaming video.

Wound Knowledge-Based Apps

Knowledge based wound apps are mobile apps whose primary function is to assist with and automate standard medical procedures and protocols. They do not perform advanced image processing and instead rely on surveys, questionnaires, mobile messaging and other methods for data. The smart mobile system created by Wang, Zhang, Huang, Tian, Hu, Cheng & Peng (2018) [97] helps nurses manage their daily tasks and outlays procedures for handling different types of wounds. Salomé & Ferreira (2018) [98] explain risks factors associated with pressure injuries, highlight progression stages, provide preliminary evaluation and recommend treatment. Gunter, et al. (2018) [99] built an app to combat surgical site infection (SSI) among post-discharge surgical patients who share pictures of their wounds and answer a daily survey detailing their recovery. The system of King (2014) [100] provides wound monitoring and wound dressing advice to nurses and general practitioners.

Wound Imaging Based Apps

By far the most popular mobile wound assessment apps employ some form of capturing and processing pictures of wounds taken with the camera phone. Going beyond simply taking a picture of the wound, these apps process the wound image to help doctors and nurses analyse and diagnose the wound. They may also provide knowledge-based surveys and questionnaires to supplement image data [101]. Choosing a platform that can run on a mobile device is of utmost importance. For example, Deserno, Kamath & Sirazitdinova (2018)

[102] do wound image pixel classification in Java via random foresting with the intention of porting it to Android's Java implementation. Even advanced image compression techniques are being applied to wound images such as Chakraborty (2019) [2] who uses the lossy set partitioning in hierarchical tree (SPIHT) compression technique to improve on transmission and segmentation of wounds.

Patterns in wound images can be exploited by computer algorithms to garner insight into wound type, size and composition. Typically this involves a segmentation step to delineate wound borders followed by a classification step to provide a diagnosis. OpenCV for computer vision applications is one open source library that can be used to process wound images, and is used for image segmentation by Pires & Garcia [103] and Wu, Guler, Cheng & Kim (2014) [104]. Schnalzer & Alcalde (2019) [105] use the iOS neural network API for skin lesion boundary detection. Goyal, Reeves, Rajbhandari & Yap (2019) [106] use MobileNet and the Inception-V2 CNN on Android to localize diabetic foot ulcers. On Windows phones, Huang, Jhan, Da, Lin & Liu (2018) [107] use the Python programming language for white balancing, anti-glare, Contrast-limited adaptive histogram equalization (CLAHE) and segmentation of wound images.

3. Machine Learning Wound Assessment

Machine learning has become the most popular approach for performing image processing on mobile devices. The basis for this shift has been the increased availability of large repositories of wound image databases. Lüneburg, et al. (2019) [5] trained U-net on 185 images manually annotated by physicians from the Hannover Medical School, and trained 745 images for infection classification. Jiao, Su, Xie and Ye (2019) [3] manually labeled 1150 burn wound images from Wuhan Hospital No. 3 using the Common Objects in Context (COCO) format. A large dataset of melanoma wound images are available through the annual International Symposium on Biomedical Imaging (ISBI) challenge.

3.1 Unsupervised Wound Image Segmentation

There a number of effective unsupervised wound segmentation algorithms that are suitable for segmenting in resource constrained environments. On the simpler end are are purely color-based segmentation algorithms that carve out colors known to be associated with wounds [108, 109, 110, 111, 112, 113, 114, 115]. Increasingly accurate unsupervised wound seg-

mentation strategies come at the cost of increased computational complexity. Many wound segmentation solutions are multi dimensional and combine multiple different strategies together in an effort to offset the shortcomings of when each is used individually.

Mean-Shift

Wound segmentation via mean-shift [116, 117, 118, 119] is a convenient technique for wound segmentation. Mean-shift works by categorizing each pixel in the image as belonging to a specific cluster. At first, a distribution is assumed on the pixels, and a window shape and size is chosen. In its simplest form, the Gaussian distribution is assumed and a circle window is utilized. At each iteration, the mean is calculated for all pixels within the window, and the entire window is moved to the new mean via what is called the mean-shift vector. This process continues until convergence, and is repeated per pixel to determine which cluster each pixel belongs to.

DBSCAN

DBSCAN is a density based clustering technique useful for wound segmentation because it can segment non spherical regions such as might arise when categorizing necrotic, slough and epithelial wound tissue [120]. Disparate wound tissues are interleaved with each other and have organic shapes. DBSCAN takes two parameters to specify pixel count and window width. It then categorizes each pixel as a core, border or outlier which is used to assign or not assign the pixel to a cluster. In addition to being able to create arbitrarily shaped

clusters, DBSCAN is robust to outliers and doesn't require a predetermined cluster count.

K-Means

k-means clustering is another common algorithm to segment wounds from images in an unsupervised manner [121, 122, 123, 124, 125, 126]. Its attractiveness lies in its conceptual simplicity. Given a value for k , which is the number of clusters, k arbitrary pixels are chosen and assigned to different clusters. The remaining pixels are assigned to a cluster based on their proximity to the cluster representative. The mean of each cluster is calculated and the process repeats using this new mean. At convergence, the variance of each cluster is examined to determine if the entire process should restart with a new set of k points.

Fuzzy C-Means

An alternative to hard segmentation schemes that assign pixels to one cluster or another, soft segmentation algorithms include a weighted value of association that a pixel has for each cluster [127, 128, 129, 130, 131, 132, 133, 134]. In the context of wound segmentation, thresholding can be applied to determine which pixels are part of the wound and which are not. The benefit of fuzzy approaches is that thresholds can be applied post segmentation, meaning that it can be modified until suitable results are achieved without resegmenting. Fuzzy c-means performs soft segmentation by iteratively updating the weights for each pixel. The algorithm proceeds similarly to k-means. Pixels are initialized with random weights and k randomly positioned clusters. At each step, each cluster center c_j is recalculated from N

pixels as follows

$$c_j = \frac{\sum_{i=1}^N w_{ij}^m x_i}{\sum_{i=1}^N w_{ij}^m} \quad (3.1)$$

Gaussian Mixture

It is also possible to apply probabilistic systems modeling to unsupervised wound image segmentation [135, 126, 136, 137]. Specifically, we assume that the image can be mapped to a given number of Gaussian distributions each centered on some image pixel. For pixel vector \mathbf{x} of size N and Gaussian parameters vector θ , we want to maximize $P(\mathbf{x}|\theta)$. This is called parametric density estimation for which can use maximum a posteriori (MAP) Bayesian estimation or max likelihood estimation (MLE). MAP is more a general approach that would be applied if we didn't know the probability distributions. In our case we assume Gaussian distributions, for which MLE is sufficient, i.e. $L(\theta|\mathbf{x})$. To optimize the distribution parameters θ_{MLE} we have

$$\theta_{MLE} = \operatorname{argmax}_{\theta} p(\mathbf{x}|\theta) \quad (3.2)$$

We use Gaussian mixture, and substitute the probability distribution with a summation over k weighted bivariate normal distributions as

$$\theta_{MLE} = \operatorname{argmax}_{\theta} \sum_{i=1}^k w_i \mathcal{N}_2(\mu_i, \Sigma_i) \quad (3.3)$$

It is difficult to find a closed form solution to this equation, so the expectation–maximization (EM) algorithm is used to do so iteratively.

Other Approaches

Among other approaches, segmentation techniques that exploit hierarchy in the wound image have been evaluated. These include the Color Structure Code (CSC) algorithm [118, 138, 118, 119], and graph-based approaches such as spectral clustering [139, 140] and graph cuts [141, 142]. Edge detection algorithms have shown promise segmenting wounds such as energy minimizing deformable splines [143, 144], Canny edge detection [145, 146, 147, 148] and toroidal decomposition [149]. To apply feature decomposition and feature extraction strategies, the wavelength and watershed transform techniques have been used [150].

3.2 Data-Driven Wound Image Segmentation

Performance Metrics for Wound Image Segmentation

If A and B are the areas of the ground truth and learned segmentation respectively, then the *dice coefficient* is defined as follows:

$$dice\ coefficient = 2 * \frac{A \cap B}{A + B} \quad (3.4)$$

The Jaccard index is similarly defined as follows:

$$Jaccard\ index = \frac{A \cap B}{A \cup B} \quad (3.5)$$

Neural Network Basics

The neural networks used for segmentation and classification of wounds employ basic ideas of neural networks. A neural network takes inputs into neurons which are forwarded through a network of neurons to compute some desired output. Each neuron has an associated weight that multiplies its input. Since we are dealing with images, the first task involves convolving the image with a kernel to identify high level features and to reduce the input size into the neural network. The activation function is the function that the neuron applies on its input, the most popular of which is the sigmoid

$$S(x) = \frac{1}{1 + e^{-x}} \quad (3.6)$$

To train the neural network, a loss function is applied to the output and then minimized using gradient descent. For e.g., for network output vector $\hat{\mathbf{y}}$ and actual output \mathbf{y} , a common function to minimize is the mean squared error (MSE)

$$\mathbf{E}[(\mathbf{y} - \hat{\mathbf{y}})] = \frac{1}{N} \sum_i (y - \hat{y})^2 \quad (3.7)$$

Another common loss function is the so-called cross-entropy function. Entropy measures the minimum average number of bits required to encode the output. If we have a discrete distribution $p(x)$, then the entropy is

$$H(x) = \sum_x p(x)(-\log(p(x))) \quad (3.8)$$

That is, $-\log(p(x))$ is the number of bits and $p(x)$ is the probability of obtaining x . The lower the average number bits, then the lower our uncertainty about the output. In the case that a neural networks classifies output classes as probabilities, then we can use entropy in what is known as cross-entropy.

$$H(y, \hat{y}) = \sum_i y_i(-\log(\hat{y}_i)) \quad (3.9)$$

Cross-entropy is the average number of bits required to encode the output $\hat{\mathbf{y}}$, given that the actual output is \mathbf{y} , and $H(y, \hat{y}) \geq H(y)$

Whichever loss function we choose, call it L , we want to minimize it by taking its derivative and performing gradient descent on each weight, w_i , in the neural network

$$w'_i = w_i + \alpha \frac{\partial L}{\partial w_i} \quad (3.10)$$

Fortunately, $\frac{\partial L}{\partial w_i}$ can be calculated recursively using each neuron reachable from w_i . Each reachable neuron, n_j , contributes $w_j \frac{dA}{dx}$ to the partial derivative, where $A(x)$ is any neuron activation function. At the end of the last neuron, we tack on the derivative of the loss function in terms of its output y' , $\frac{dL}{dy'}$. The final result is a recursive tree of calculations which is easiest expressed in pseudocode

```
function traverseNeurons(n_k)
```

```

if leaf_neuron(n_k)

    return dL/dy'

else

    return w_k * dA/dx  $\sum_j$  traverseNeurons(n_j)

```

where n_j are all the neurons that the output of neuron n_i connects to. The recursion works for all except the input neuron. The input neuron, n_i , contributes $i \frac{dA}{dx}$, i.e. replace w_i with the neuron's input value.

$$\frac{\partial L}{\partial w_i} = i \frac{dA}{dx} (\sum_j \text{traverseNeurons}(n_j)) \quad (3.11)$$

In the case of the sigmoid activation function $S(x)$, the derivative is defined elegantly in terms of its output y as $y(1 - y)$. This gives us

$$\frac{\partial L}{\partial w_i} = i y_i (1 - y_i) (\sum_j \text{traverseNeurons}(n_j)) \quad (3.12)$$

Neural Network Frameworks for Wound Segmentation

Lüneburg, et al. (2019) [5] use U-net for driveline tube segmentation and VGG-16 for infection classification, pre-trained on ImageNet and augmented with wound images subjected to various affine transformations. Jiao, et al. achieved a dice coefficient of 95% for segmentation and 67% success rate for 3-level infection classification. [3] use the Mask Regions with

Convolutional Neural Network (Mask R-CNN) and ResNet for burn wound segmentation, and achieve 85% dice score segmenting 150 test burn images.

Baghersalimi, et al. (2019) [151] incorporate the computational efficiency and speed of DenseNet’s dense blocks and add skip links that pass information to a decoder for higher wound segmentation accuracy. The so called DermoNet achieved 82.5% and 89.4% for the Jaccard and Dice coefficients respectively on the ISBI 2016 skin lesion challenge data set. Tschandl, Sinz & Kittler (2019) [152] segment skin lesions with LinkNet152, which is build on top of ResNet152 and conditional random fields (CRF) to improve the segmentation results.

Zahia, Sierra-Sosa, Garcia-Zapirain & Elmaghraby (2018) [153] use CNN to segment pressure ulcers and track wound healing by classification into necrotic, granulation and slough. Images 5x5 pixels is size are passed into the CNN for each healing stage to training the network on each healing stage color, which are dark grey (necrotic), light gray (granulation) and white (slough) after being converted to grayscale.

Elmogy, Garcia-Zapirain, Burns, Elmaghraby & Ei-Baz (2018) [154] use 3D CNN to segment pressure ulcers into one of necrotic, granulation and slough regions. This is achieved by passing five different versions of the same wound image into the neural network. The RGB, HSV, gaussian blurred and first-order edge detection filter on the current image, plus a first-order filtered version of the previously captured image of the wound

Vesal, Ravikumar & Maier (2018) [155] extend U-Net for skin lesion segmentation. Their so called SkinNet incorporates both local and global knowledge of the wound by putting a

dilation convolution layer between U-Net’s encoder and decoder layers.

Sarker, et al. (2018) [156] combine a dilated residual network (DRN) with a pyramid pooling network (PPN) for improved skin lesion segmentation. Alom, et al. (2018) [157] achieve improved performance for segmenting cancerous skin lesions by extending U-net with a recurrent residual convolutional neural network.

Qi, Le, Li and Zhou (2017) [158] use a fully connected neural network (FCNN) to segment skin lesions using the 2750 melanoma images from the International Symposium on Biomedical Imaging 2017 challenge. Chakraborty (2019) [2] uses the chrominance color channel D_r to measure wound contrast with the image background. For N wound pixels $I_w(x, y)$ and M background pixels $I_b(x, y)$, we can calculate contrast as follows

$$contrast = |\frac{1}{N} \sum I_w(x, y) - \frac{1}{M} \sum I_b(x, y)| \quad (3.13)$$

3.3 Data-Driven Wound Image Classification

Performance Metrics for Wound Image Classification

The Confusion matrix is the basis for determining the accuracy of a classification scheme. In its simplest form for binary classification, the Confusion matrix has two columns and two rows with one cell for each of true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

		Predicted	
		Yes	No
Actual	Yes	TP	FN
	No	FP	TN

Table 3.1: Confusion Matrix

The receiver operating characteristic curve (ROC) is a graph that relates the false positive rate (FPR) to the true positive rate (TPR) for varying threshold values.

$$FPR = \frac{FP}{FP + TN} \quad (3.14)$$

$$TPR = \frac{TP}{TP + FN} \quad (3.15)$$

In general, optimal thresholds are those toward the upper left of the ROC curve. To get an idea of how different learning algorithms compare for a single classification task, we can compare the area under the curve (AUC) of their respective ROC curves. Higher AUC's indicate a better performing classifier. An AUC of 1 indicates 100% accuracy, while 0 means it is always wrong.

The F_1 score is based off of precision and recall which are defined as follows

$$Precision = \frac{TP}{TP + FP} \quad (3.16)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.17)$$

The F_1 score is then obtained via

$$F_1 = 2 \cdot \frac{\textit{precision} * \textit{recall}}{\textit{precision} + \textit{recall}} \quad (3.18)$$

Neural Networks for Image Classification

As we have seen in the previous section, many applications perform both segmentation and classification of wounds. Segmentation is generally the preliminary step for obtaining a good image to classify. There are many CNN-based binary wound classification schemes. Binary wound classification can be effective even in settings where there is a dearth of training images available. In other cases, researches have found ways to augment the dataset of images by performing affine image transformations.

Among data-driven binary wound classifiers is DFUNet from Goyal, Reeves, Davison, Rajbhandari, Spragg & Yap (2018) [159], which is used to classify diabetic foot ulcers as normal or abnormal. They achieve 92.5% accuracy and 0.961 AUC. Yadav, Sharma, Singh & Goyal (2019) [160] classify burn wounds into one of two classes, those needing graft and those not needing graft. A skin graft is when skin is taken from elsewhere on the body to cover the damaged region. Nilsson & Velic (2018) [161] classify wounds as venous or non-venous. They augment 300 manually annotated images and incorporate previously trained data on the VGG-19 network. Abubakar & Ugail (2019) [162] classify wound images into a burn or ulcer. They extract image features with the VGG-face, ResNet101, and ResNet152 networks and then use an SVM classifier to achieve 99.9% accuracy. The DFU_QUTNet

NN of Alzubaidi, Fadhel, Olewi, Al-Shamma & Zhang (2019) [163] classifies a wound as a diabetic foot ulcer or normal skin. They report an F_1 score of 94.5%. Goyal, Reeves, Rajbhandari, Ahmad, Wang & Yap (2020) predict infection or ischaemia in diabetic foot ulcers. They reported 90% accuracy for ischaemia and 73% accuracy for infection detection.

González-Díaz (2019) [164] built a CNN with novel blocks to categorize eight dermatological features of skin lesions, and to identify rings and symmetrical properties to aid in skin lesion diagnosis. Zahia, Sierra-Sosa, Garcia-Zapirain and Elmaghraby (2018) [153] classify pixels into granulation, slough or necrotic by first removing flash lights. This is accomplished by converting the image into grayscale and classifying each pixel into one of two categories, flash or no flash. To obtain a flash threshold, a split is created around a histogram of the gray scale values that maximises t with selectively chosen weights $w_0(t)$ and $w_1(t)$, and global average μ as such

$$w_0(t) \left(\sum_{i=0}^{t-1} \frac{ip_i}{w_0(t)} - \mu \right)^2 + w_1(t) \left(\sum_{i=t}^{N-1} \frac{ip_i}{w_1(t)} - \mu \right)^2 \quad (3.19)$$

Elmogy, Garcia-Zapirain, Burns, Elmaghraby and Ei-Baz (2018) [154] combine pressure ulcers into healing stages over time, and create 3D lattices of wound images for classification into granulation, slough and necrotic. In a preprocessing stage, $7 \times 7 \times 3$ blocks of the lattice are convolved with a smoothing 3D Gaussian Kernel

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}|\mathbf{r}-\mathbf{r}_c|^2} \quad (3.20)$$

Recall, that for 2D images, we sample the Gaussian smoothing kernel function across a 3×3 image mask

$$\frac{1}{16} \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \quad (3.21)$$

Since we don't have a sampled version of the $7 \times 7 \times 3$ 3D Gaussian kernel offhand, we first rewrite the 3D Gaussian into its 3 variables explicitly.

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x^2+y^2+z^2)} \quad (3.22)$$

And then integrate over each 3D voxel (x_i, y_i, z_i) to obtain the corresponding sampled value at that voxel, keeping in mind to normalize the lattice to fit within three standard deviations of the kernel center, since beyond that the function is practically zero.

$$\int_{x_i}^{x_{i+1}} \int_{y_i}^{y_{i+1}} \int_{z_i}^{z_{i+1}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x^2+y^2+z^2)} dx dy dz \quad (3.23)$$

The definite integral for each voxel then becomes the sampled value that we use to discretize the Gaussian kernel and apply it to the 3D image lattice. Burdick, Marques, Weinthal & Furht (2018) [165] obtain better classification results by including information about normal skin surrounding skin lesions.

4. 3D Wound Modeling

4.1 3D Human Anatomy

3D Human Anatomy Labeling Systems

Human surface anatomy labeling plays an important role in helping physicians, students and medical personnel discuss and exchange medical data that involves human surface anatomy. Conditions that affect the skin such as chronic rashes, lesions, cancers, abrasions, burns, etc. are treated differently based on their diagnosis, region and severity. As of today, The Anatomy Mapper Web application and the NYU Human Surface Anatomy Labeling systems are two popular labeling systems used by U.S. hospitals. We have extended these surface anatomy labeling systems to 3D by segmenting anatomical regions on a human 3D model. 3D visualization of human surface anatomy brings an ease-of-use and precision that has been heretofore difficult to achieve when using just images alone. We have demonstrated its usefulness by creating an interactive mobile Web platform for 3D human surface anatomy visualization and labeling.

Advances in 3D modeling is bringing about an increase in the number of clinics, universities and researchers using 3D models to visualize human anatomy. Human 3D models of both internal and external anatomy are being created from large databases of CT and MR scans. These models are segmented into regions, and then labeled appropriately. The resultant 3D models are used in interactive 3D visualization applications that run on desktop, mobile and/or Web platforms.

We have created a 3D human surface anatomy labeling system based on a digitally sculpted 3D human model. We have based our labeling system on the NYU and Anatomy Mapper 2D labeling systems currently in use in many clinics. Both systems assist with recording, diagnosing and treating various types of skin conditions. Our 3D labeling system provides clinicians with all of the advantages of the original 2D labeling system plus the ability to visualize and interact entirely in 3D. This includes the ability to highlight regions, see label names and fully pan, rotate and scale the human model.

We have built an interactive mobile Web app to showcase the labeling system. The system relies on WebGL for 3D rendering and uses texture mapping to digitally paint anatomical surface regions. Modern HTML5 canvas tools enable the texture map image to be processed in real time for interactive use such as region highlighting and land marking. Texture mapping also makes it easy to modify anatomical regions when necessary via digital painting, therefore precluding the need to modify the 3D human model itself.

Organs

VOXEL-MAN is an ambitious effort to label 3d models of internal organs [166]. It includes interactive labeling and supports cross-sectional, regional and systematic anatomy. It's quality is buttressed by high quality CT cadaver data from the VH dataset. Sandor & Leahy (1997) [167] label a 3D model of the heart reconstructed from MR images. The Navy's Anatomy Study Guide App has interactive 3D modeling and labeling of the human brain [168]. The Anatomy Learning mobile Web app has fully segmented and interactive labeling of the human digestive system. [169]. The BioDigital Human - 3D Anatomy mobile app simulates surgical incisions on internal organs with regional segmentation and color coding [170]. In some cases, the anatomy is segmented and not labeled. For example, Chen & Bagci (2011) [171] automatically segment 3D models of the abdomen and foot via object recognition and graph cutting.

Skeletal

Witmer, Ridgely, Dufeu, & Semones (2008) [172] allow for 3D visualization and labeling of the human skull, hand, foot, legs and pelvis. These are available as 3D PDF's and as animated videos. CGTrader has a full 3D human skeleton model with labeled bones [173]. The Anatomy Learning mobile Web platform has a fully segmented and interactive labeling of the human skull [169]. The Skeletal Anatomy Android app labels 140 bones on an interactive 3D human skeleton model [174]. The Navy's Anatomy Study Guide App has an interactive 3D model and labeling of the human skeleton [168]. BoneLab is an OpenGL-based

3D skeletal visualization desktop application with color coded regions and labels accessible from a file navigation-like panel. Both English and Latin names are supported [175].

Cardiovascular

Dicken, Krass, Peitgen, Kuhnigk, Hahn & Hindennach (2003) [176] segment 3d lung models into regions called lobes and label them, and the internal blood vessels are segmented. After segmentation, one could use the automated lung bronchial tree labeling algorithm presented by Dicken, Krass, Peitgen, Kuhnigk, Hahn & Hindennach (2003) [176] or by Chalopin, Finet & Magnin (2001) [177]. The Navy's Anatomy Study Guide App provides interactive 3D modeling and labeling of the heart and vascular system [168]. Anatomy Learning has a fully segmented and interactive labeling system of the heart and shows blood vessels. [169].

Muscular

The Muscle System Pro is an interactive mobile app for 3D visualization of the muscular system [178]. Muscular regions are labeled and searched by name using auto completion. The Navy's Anatomy Study Guide App has interactive 3D modeling and labeling of the muscular system [168]. Muscular Teacher is an OpenGL-based 3D muscular visualization desktop application with color coded regions and labels accessible from a separate panel [175].

Full Body

The Complete Anatomy Platform 2020 [179] includes 7100 anatomical regions, twelve body systems and seven layers of tissues. It also includes physical muscle simulation, nerve and blood tracing and landmark mapping. The 3D Bones and Organs mobile app has 145 3D fully segmented muscles [180]. The muscles can be peeled back to reveal the human skeletal model with labels. Internal organs become visible when bones are peeled off interactively in 3D. The Essential Anatomy 3 mobile Web platform shows highly detailed and labeled views of the muscular system [181]. It shows veins and arteries, and allows layer peeling to view the skeletal system and internal organs, along with a highly detailed 3D visualization of the nervous and digestive systems, and connective tissue.

Human Surface Anatomy Labeling Systems

Palmer, Sachdeva, Lambert & Dermatologic (2010) [182] created a 2D Anatomy Mapper labeling system accessible over the Web. Prior to that, Alfred Kopf, MD of the NYU Melanoma Clinical Cooperative Group created the NYU Labeling System in the 1970's. To the best of our knowledge, no 3D human anatomy labeling system is available. We have digitally painted two 3D human models. The first is painted with the 2D regions from the Anatomy Mapper labeling system and the second is painted with 2d regions from the NYU labeling system.

NYU Labeling System

Alfred Kopf, MD of the NYU Melanoma Clinical Cooperative Group created the NYU Labeling System in the 1970's, as seen in figure 4.1. The labeling system allows clinicians to track lesion changes over time in a standardized way. It includes a numbering system for each anatomical region, all of which is hand drawn. There is also a Web interface that allows users to hover over a specific region to obtain its medical terminology. NYU labeling remains in use at the Memorial Sloan Kettering Cancer Center, Cleveland Clinic and Mayo Clinic.

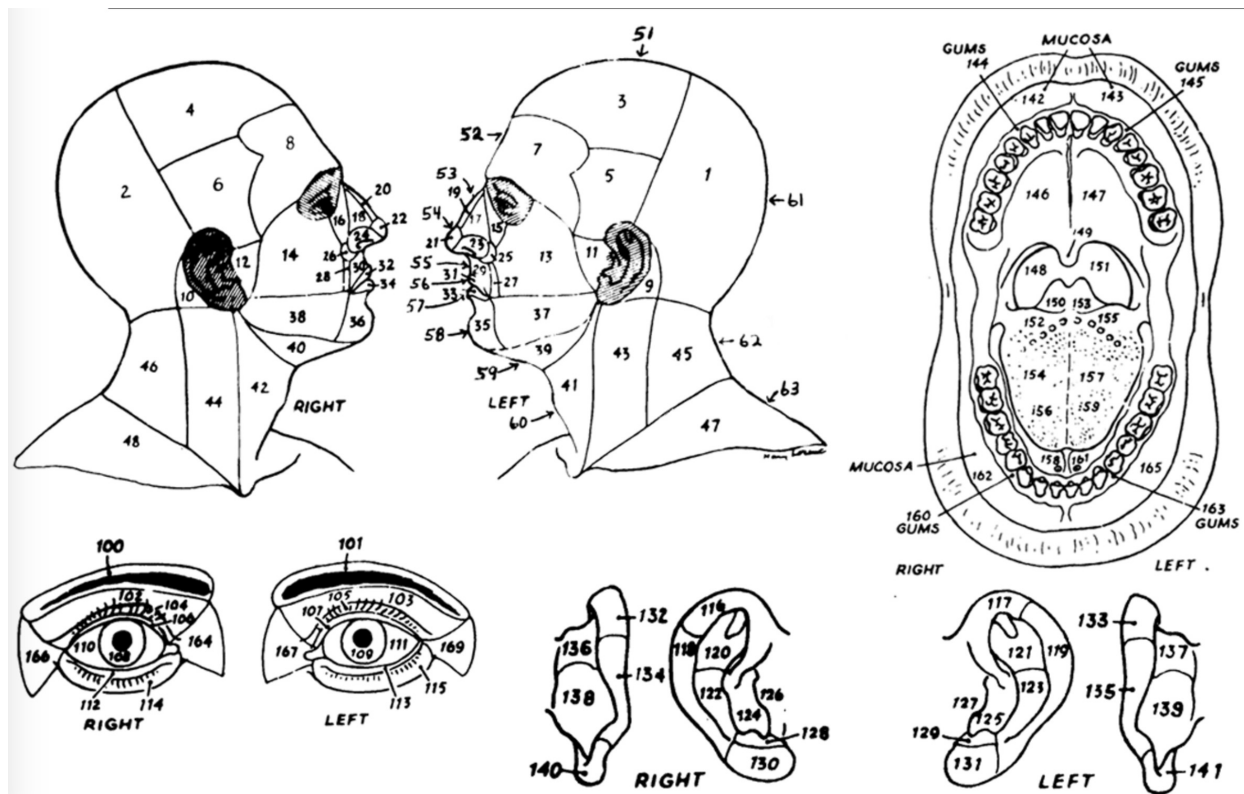


Figure 4.1: NYU Anatomy Labeling System

We have digitally painted a 3D human model with 2D regions obtained from the NYU labeling system, as shown shown in figure 4.2. We have included the complete NYU anatomical regions for the body, and modified regions for the head, hands and feet. We label our models according to initial discussions with Dr. Jeffrey Niezgoda of the AZH Wound Care and Hyperbaric Oxygen Center. Some differences between our model and that of NYU is that we only color code the fingers as a whole, rather than both the front and back of the fingers, similarly for the toes. For the head, we simplify the segmentation of the eyes, nose, chin, ears, cheeks, half spheres of the skull and forehead.

Anatomy Mapper Labeling System

Palmer, Sachdeva, Lambert & Dermatologic (2010) [182] created a 2D Anatomy Mapper labeling system accessible over the Web. The goal of the project is to provide consistent, detailed anatomy labeling to reduce the ambiguity that arises from the myriad of procedures, surgeries, biopsies and wound types for patients. For example, a single patient may have multiple wounds in the same area, along with scars from previous surgeries and biopsies that must be properly distinguished. The Web interface shows anatomical labels on hover-over which can be easily copied, for example, into an Electronic Medical Record. We have digitally painted a 3D human model with the 2D regions obtained from the Anatomy Mapper labeling system. The current Anatomy Mapper-based 3D model is shown in figures 4.3, 4.4 and 4.5.

The online Web-based Anatomy Mapper application by default doesn't show the different

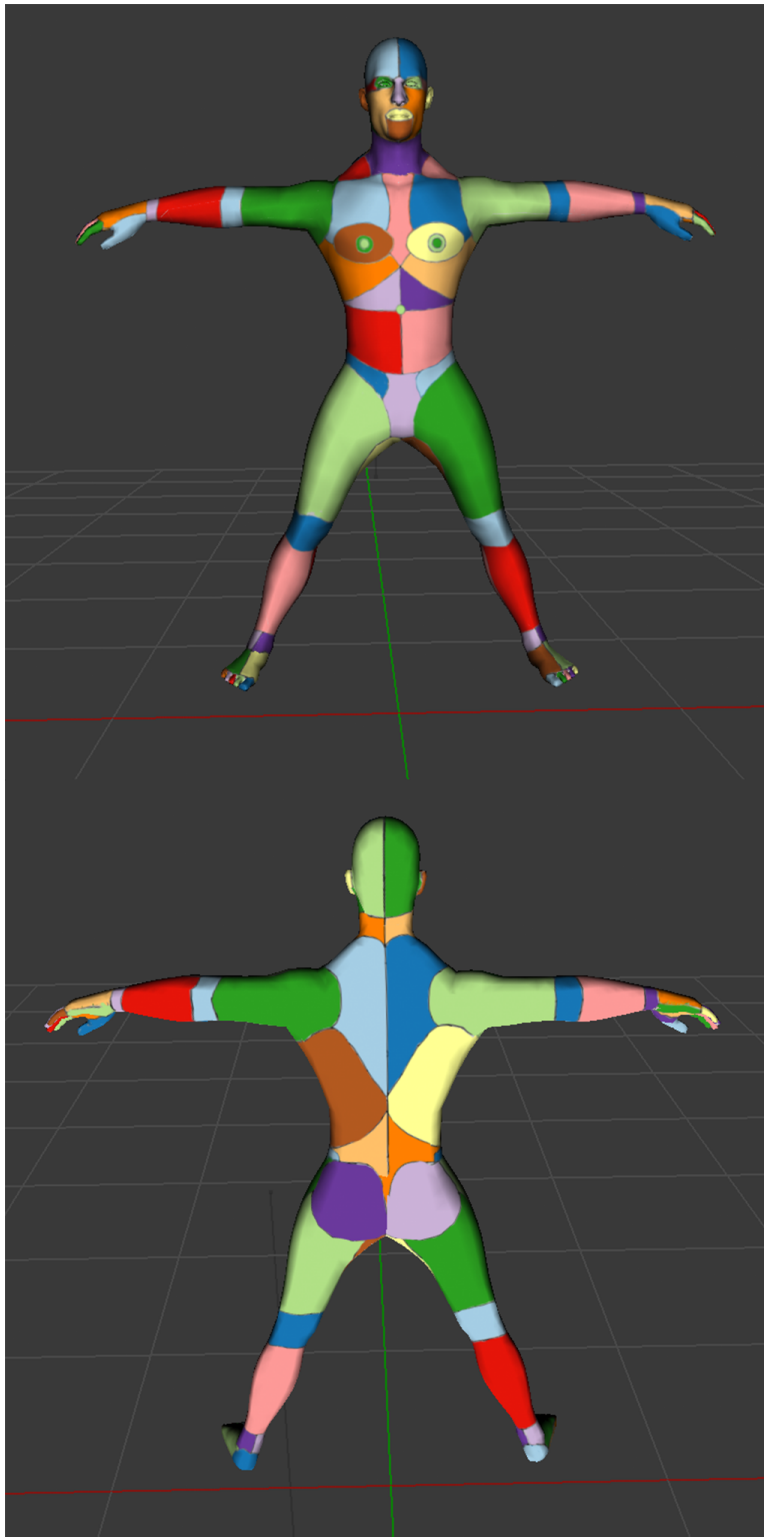


Figure 4.2: 3D NYU Labeling System

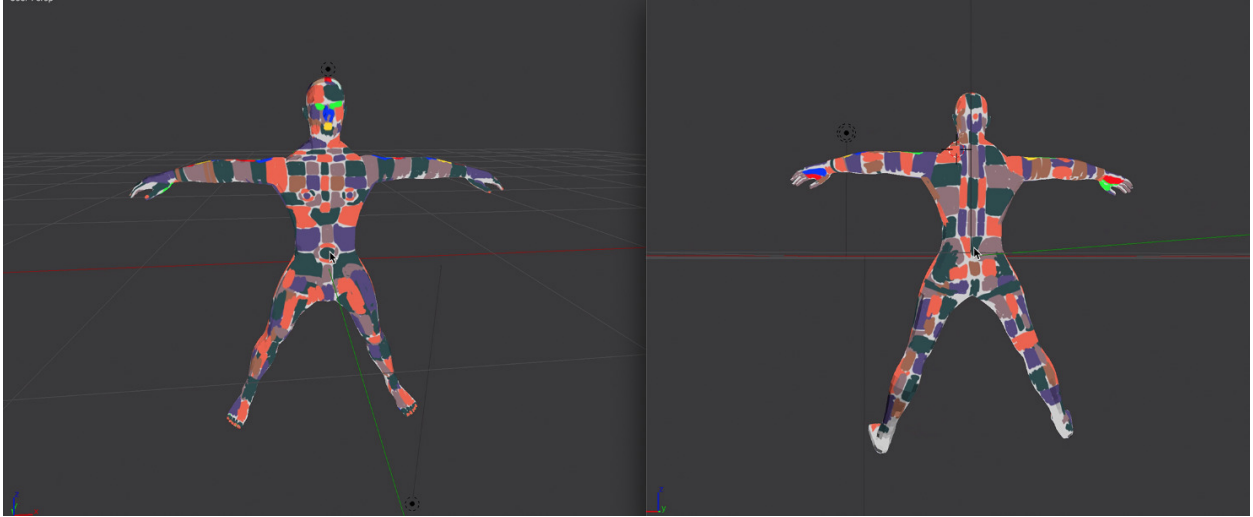


Figure 4.3: 3D Human Anatomy Model Front & Back

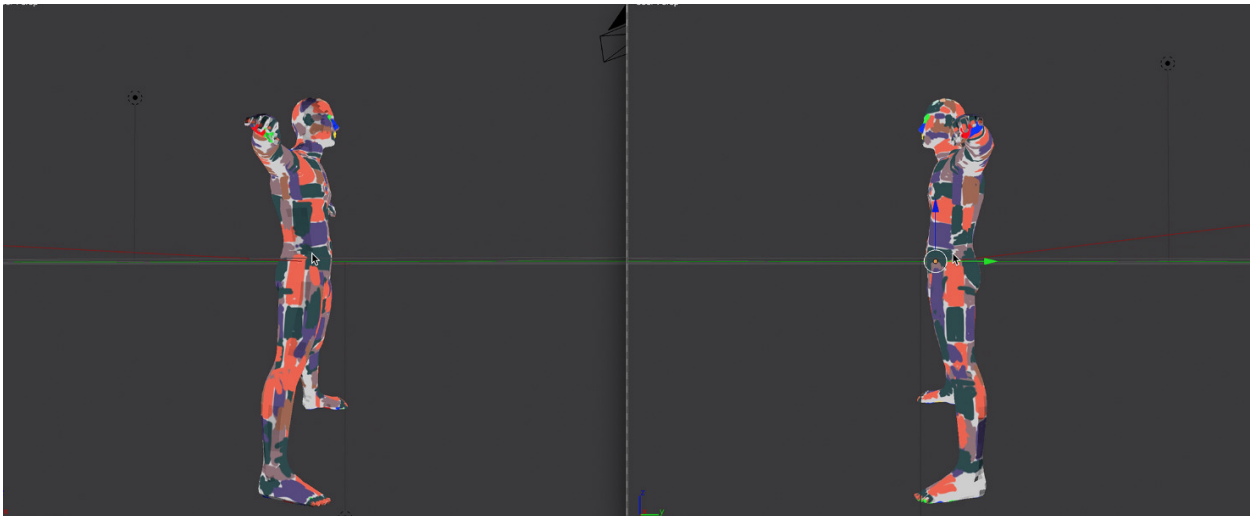


Figure 4.4: 3D Human Anatomy Model Sides

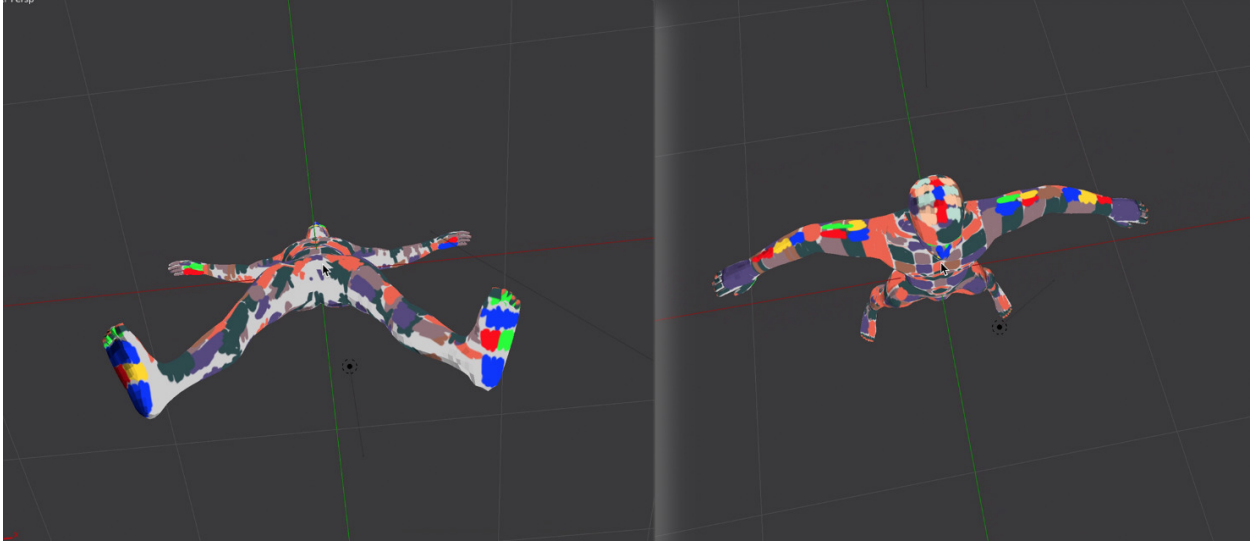


Figure 4.5: 3D Human Anatomy Model Top & Bottom

surface regions because the application first requires users to mouse over a region of the 2D image. This makes it challenging to get an overall view of the labels and tedious to transfer to a 3D model. To fix this, we loaded the Anatomy Mapper app into a browser and used the Web Inspector to use the page's area maps as templates to create filled polygons with SVG for each anatomical region as show in figures 4.6, 4.7, 4.8 and 4.9.

3D Human Anatomy Labeling

We used Photoshop CC 3D tools to digitally paint our 3D model. We used the 3D menu option "Generate UVs" to generate a texture map and were able to paint directly onto the 3D model such that the texture map was updated automatically. Ideally, we need only use 4 colors to paint the anatomical regions by the 4-color theorem, but in practice we required more than $2\times$ that.

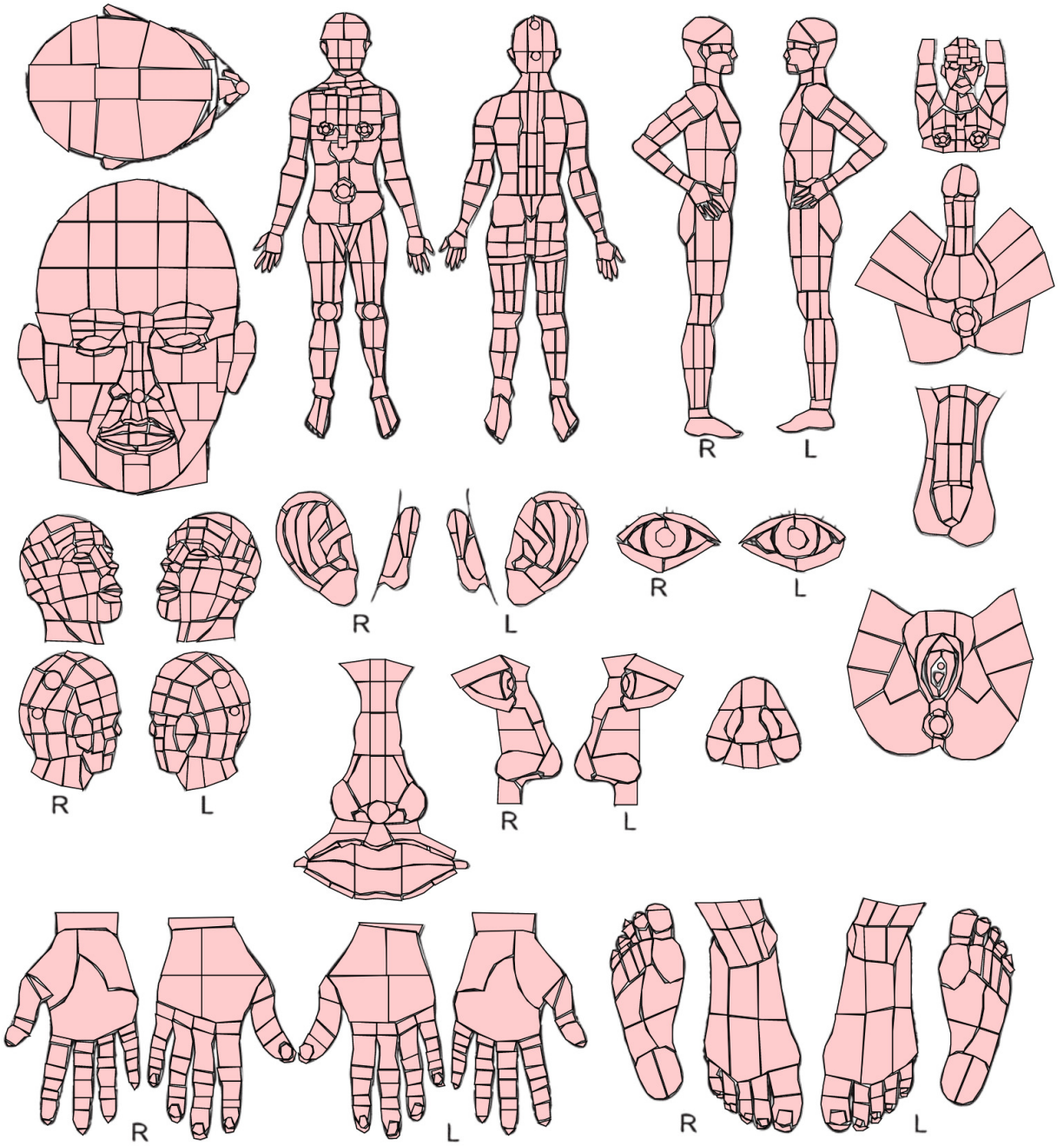


Figure 4.6: Anatomy Mapper Full Body

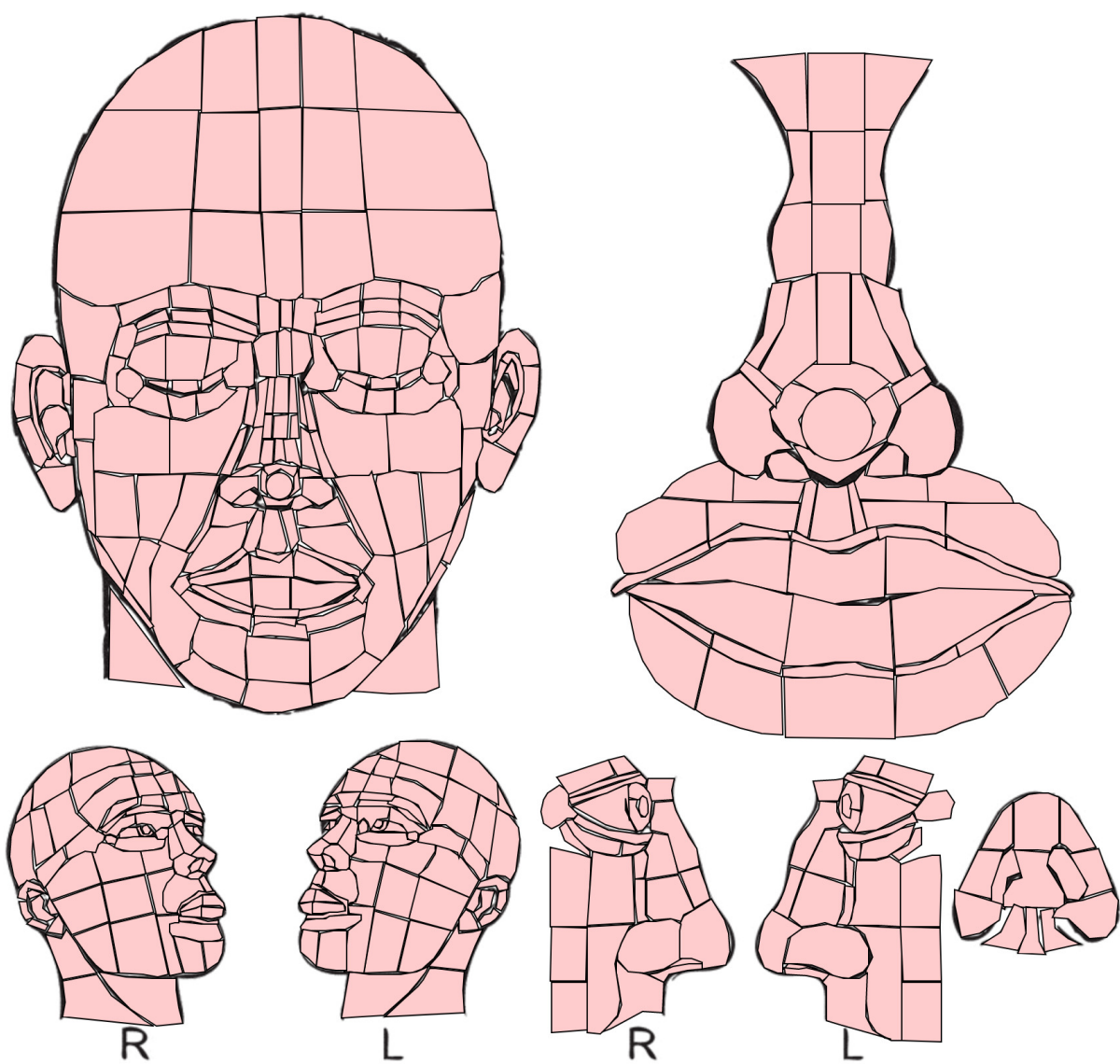


Figure 4.7: Anatomy Mapper Face

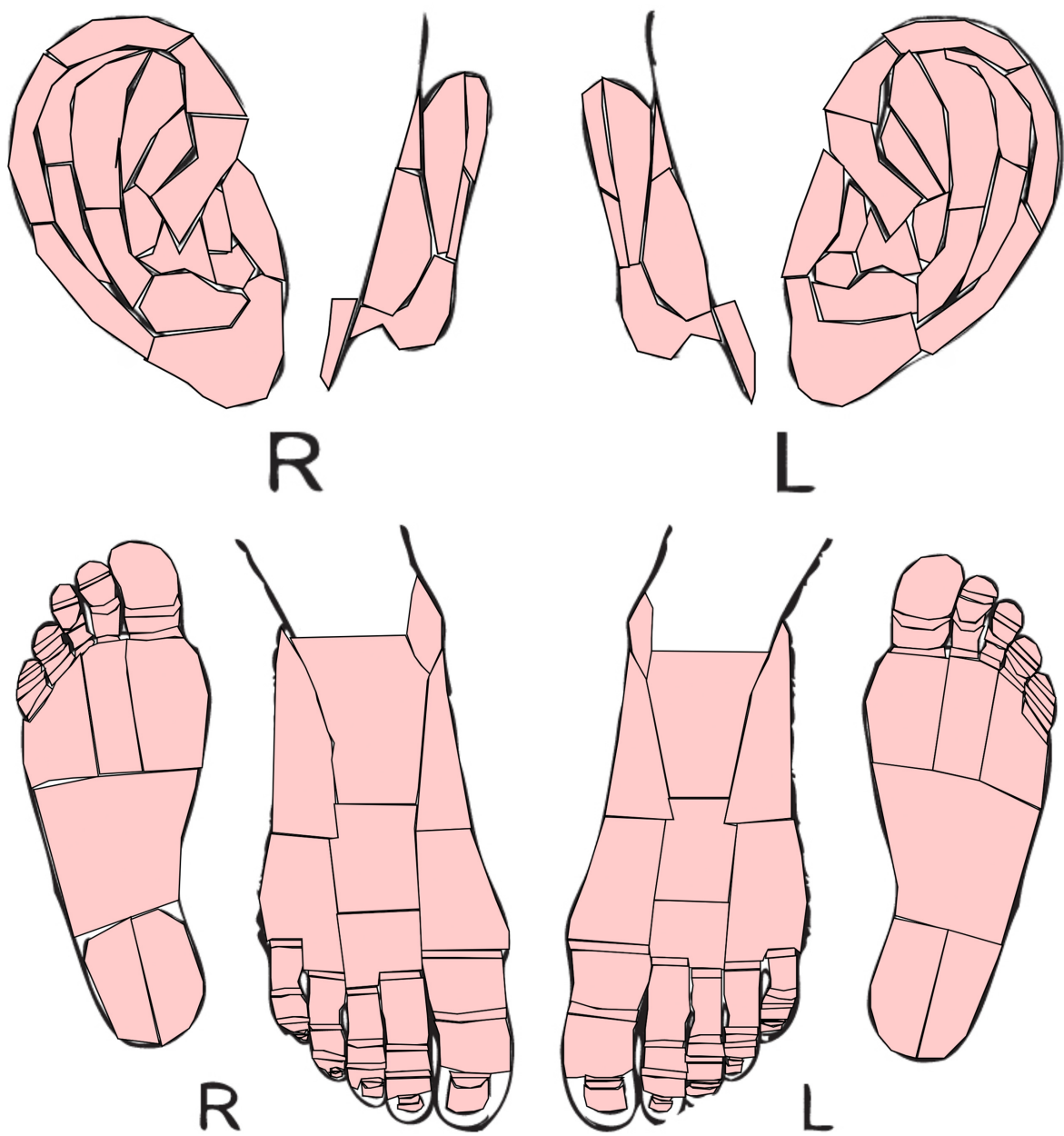


Figure 4.8: Anatomy Mapper Feet and Ears

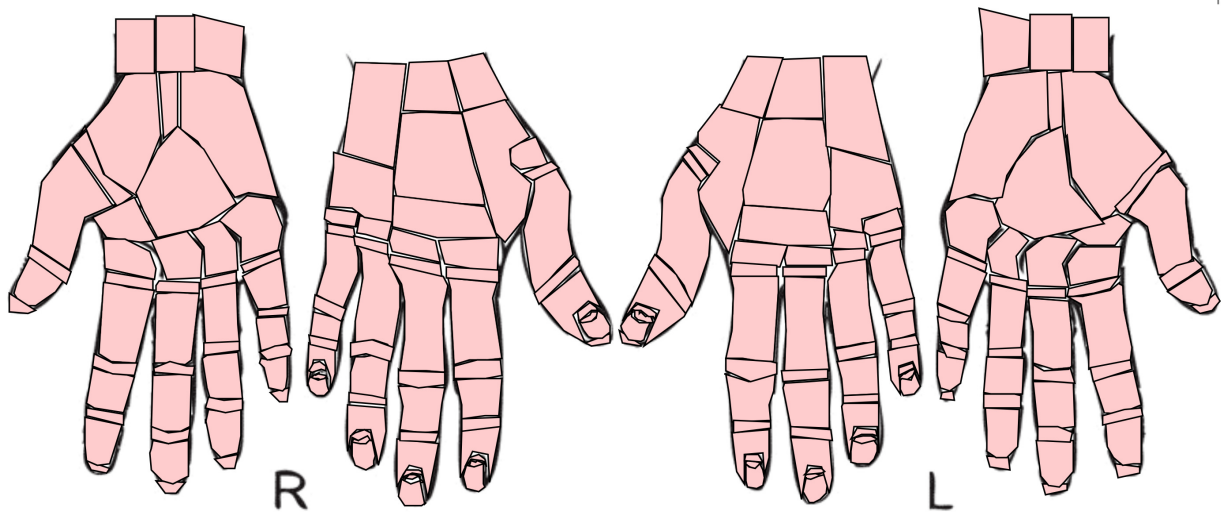


Figure 4.9: Anatomy Mapper Hands

The initial 3D human model was obtained for free from TurboSquid 3D stock models Website. We then imported the 3D model into Blender for post processing. In the default model, the human model is standing upright with arms down by the side. This is problematic, since the end goal is for users to be able to pick different regions, including the hands, arms and inner thighs. So we applied a human meta rig in Blender to deform the arms and legs by raising the arms into a wingspan and spreading the legs. The difference can be seen in figure 4.10.

After applying the meta rig in Blender and before digitally painting the NYU-based 3D model labeling system, we further cleaned up the mesh topology. Specifically, we found excessive bulging and creasing around the shoulders where the rig deformed the mesh into a wingspan. This made for a slightly inordinate male superhero style surface anatomy and generated unwanted topology below the skin surface. Also, spreading the legs caused the

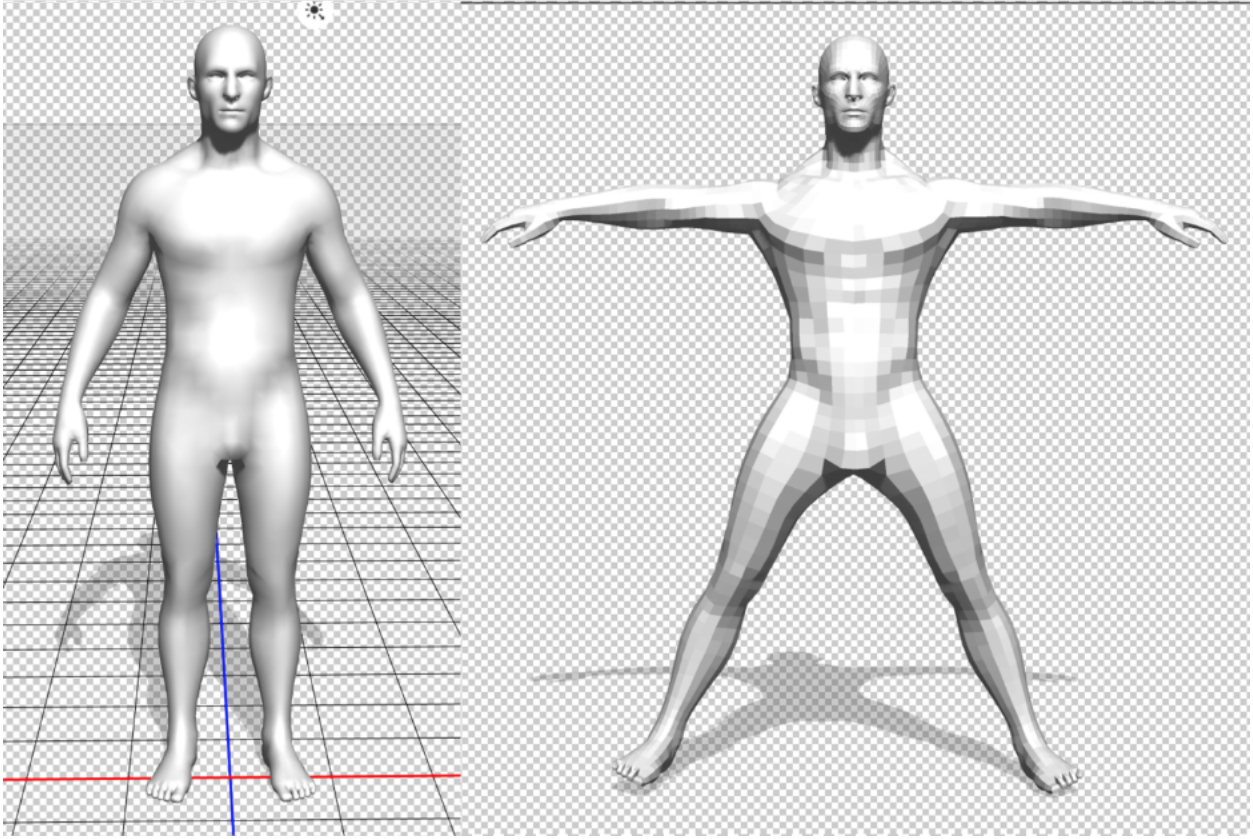


Figure 4.10: 3D Human Anatomy Models

crotch to bulge out. In both cases, we manually changed the topology of the mesh using Blender’s modeling tools for extrusion, deletion, moving, etc. We also discovered a mesh lattice for the eye balls and inner mouth, which we removed since they were not visible on the anatomy surface. The final 3D human anatomy mesh topology is shown in figure 4.11.

3D Surface Labeling

To the best of our knowledge, no 3D human surface anatomy labeling system is available. The Human Body Educational VR 3D mobile app includes a complete 3D skin model of a



Figure 4.11: Final 3D Human Anatomy Model

male with high level labels, but no detailed anatomical segmentation [183]. We note that skin surface labeling is distinct from skin anatomy labeling itself. For example, TurboSquid has a 3D model of the tissue layers beneath the skin such as the epidermis, glands, hair follicles, vessels, membranes, etc. [184]. Scientist Cindy has a rotating 3D model of the integumentary system showing the three major skin layers, the epidermis, dermis and hypodermis. Furthermore, human surface labeling may also include non-skin tissues such as finger nails, eye and inner mouth.

Previous work uses automated and semi-automatic techniques to segment 3D anatomy into regions [185, 186, 187]. Kalogerakis, Hertzmann & Singh (2010) use conditional random field (CRF) learning and feature extraction to automatically segmented and label the fingers of the hand and limbs of humanoid bodies [188]. Skin surface anatomy is more complex in that there are less obvious region boundaries. For example, Anatomy mapper has 1,437 separate anatomical regions on the human surface, including detailed segmentation and labeling of eyes, ears, fingers, toes, lips, nose and crotch. NYU has 186 anatomical regions which includes detailed labeling of the inside of the mouth. What we have done is manually segment each anatomical region in 3D using digital painting in order to obtain fine level of detail.

Mapping 2D anatomical regions to 3D regions presents its own set of challenges. This is particularly apparent when migrating the Anatomy Mapper to 3D. A close inspection of Anatomy Mapper’s regions shows four views of the human body from the front, back, left and right. Ideally, 3D regions should connect seamlessly over 360 degrees of the 3D model.

However, this is not straightforward to do using the templates where the body proportions are slightly different in each view. Therefore, it's much easier to convert the NYU labeling system to 3D since it only shows two views from the front and back and often has intuitive connecting boundaries, particularly around the legs and arms. In cases where the front regions don't align cleanly with those in the back, we can follow the natural curvature of the 3D human model so that regions look interconnected over 360 degrees of rotation.

Region Highlighting

Ali, Hartmann & Strothotte (2005) [189] discuss a variety of possible label layout options in 3D. These include color-coding, internal labeling and external labeling, where internal labels appear within the region and external labels are connected by line segments to their respective regions. For example, Götzelmann, Ali, Hartmann & Strothotte (2005) [190] label 40 regions of a 3D heart model using a combination of color coding, and internal and external labeling. We take the approach of highlighting regions on click on desktop, and on touch on mobile devices. This is the same strategy that Anatomy Mapper employs. However, Anatomy Mapper does not provide any color differentiation for regions, nor does it include mobile touch capability. I believe that combining region coloring with highlighting makes learning, visualizing and interacting with 3D surface anatomy more intuitive.

Figure 4.12 shows an example of our region highlighting. The region is surrounded with a white glowing border and the label name is displayed in the upper left. In 2D, for example with the online Anatomy Mapper, HTML polygon-shaped area maps are used to demarcate

surface regions. In 3D, the situation is a bit more complicated. The 3D model is defined by triangle faces and the regions are textured mapped onto the model, making it difficult to build polygon shaped regions in 3D space. However, the texture map image itself is 2D. We have used this fact to manually create paths from the texture map along the region boundaries for each region. Note, we do not rely on enclosed shapes because the texture map may spread polygon faces for the same region across different parts of the image. Once we have generated paths for each region, we can simply use standard HTML5 canvas path drawing tools to modify the texture directly and reload it onto the mesh.

In order to highlight the correct region, the system must be able to distinguish the region touched or clicked. As already noted, we do not keep track of an enclosed shape for each region, which is difficult to achieve. However, we have two bits of information to go on. We have the surface color and the 3D coordinates on the model surface of the click. We use the online tool, Color Brewer, to create a 12-class qualitative map coloring scheme for our region coloring. To identify a region, and its associated ID and label name, we look at the color and a 3D bounding box for each region to verify which label is clicked. This works as long as the scheme's colors are adequately distributed across the model's surface.

Human Model Procurement

We used Photoshop CC 3D tools to digitally paint our 3D model. We used the 3D menu option "Generate UVs" to generate a texture map and we were able to paint directly onto our 3D model and see the updated texture map in real time. Ideally, we would need to use



Figure 4.12: 3D Region Highlighting of Upper Chest

only 4 colors to paint the anatomical regions by the 4-color theorem, but we instead use a color scheme created specifically for maps using the online Color Brewer tool.

The default human model that we obtained from TurboSquid is a low resolution polygon model, meaning that individual polygons are noticeable on the rendered mesh. Texture mapping is applied to ameliorate this short coming, but we still felt the need to bump up the resolution to achieve a smoother look. We increased the polygon resolution minimally enough so as not to cause a noticeable slow down in rendering speed or geometric processing. Major techniques for mesh smoothing are loop subdivision, butterfly subdivision, Doo-Sabin and Catmull-Clark.

Loop subdivision is a mask-based approximation scheme for triangular meshes with C^2 continuity. Each edge is split via eighths, where v_1 and v_2 are the ends of the edge, and v_3 and v_4 are vertices opposite the edge, as follows¹

$$v = \frac{3}{8}v_1 + \frac{3}{8}v_2 + \frac{1}{8}v_3 + \frac{1}{8}v_4 \quad (4.1)$$

Each original vertex, v_0 , is then moved to a new location by averaging over the six surrounding edge mid points, $v_1, v_2, v_3, v_4, v_5, v_6$, using the rule of sixteenths as follows

$$v = \frac{10}{16}v_0 + \sum_i \frac{1}{16}v_i \quad (4.2)$$

Like loop subdivision, butterfly subdivision is for triangular meshes, except that the existing vertices are interpolated rather than approximated. Butterfly subdivision got its

¹we have ignored boundary conditions

name because edges are split by averaging over existing vertices in a neighborhood that resembles a butterfly shape. Specifically, if v_1 and v_2 are the ends of the edge, and v_3 and v_4 are vertices opposite the edge, and b_1, b_2, b_3, b_4 are the butterfly points, then, ignoring boundary cases, new vertices are split along the edges according to the following

$$v = \frac{1}{2}v_1 + \frac{1}{2}v_2 + \frac{1}{8}v_3 + \frac{1}{8}v_4 + \sum_i \frac{-1}{16}b_i \quad (4.3)$$

Doo-Sabin subdivision is a C^1 approximation scheme that is the surfaces counterpart to Chaiken's curve subdivision algorithm. Chaiken adds two new points on each curve edge, and then cuts out the original points when forming the new curve, a process known as corner cutting. The two points, q_1 and q_2 are locally computed from the edge's end points p_1 and p_2 as

$$\begin{aligned} q_1 &= \frac{3}{4}v_1 + \frac{1}{4}v_2 \\ q_2 &= \frac{1}{4}v_1 + \frac{3}{4}v_2 \end{aligned} \quad (4.4)$$

The end result of a Doo-Sabin execution is a smoothed mesh of arbitrary topology that approaches at its limit a bi-quadratic uniform B-spline. In general, the final shape produced is unpredictable and depends on the original mesh's topology. Like Chaiken's curve subdivision, Doo-Sabin cuts out the vertices of the original mesh via corner cutting by adding more faces while keeping the number of the vertices largely unchanged. To achieve this, firstly, the vertices of each face, f_i , are averaged to produce vertex F_i , second, each edge, e_j , is split at

its midpoint, E_j , by averaging the end points, and finally each new vertex, v_k , is computed from the original vertex, V_k as follows

$$v_k = \frac{V_k + E_j + E_{j-1} + F_i}{4} \quad (4.5)$$

Catmull-Clark is Pixar's famous subdivision scheme, and is used within Blender's subdivision modifier. Like Doo-Sabin, Catmull-Clark is an approximation scheme, but produces a quadrilateral mesh of C^2 continuity that is non shape-preserving as it approaches a bi-cubic uniform B-spline at its limit. It accomplishes this by increasing both the number of vertices and faces at each subdivision step. One vertex is added to the center of each face like so

$$f = \frac{1}{n} \sum_{i=1}^n v_i \quad (4.6)$$

One vertex is added between edge end points, v_1, v_2 , and adjacent face points, f_1, f_2 , as so

$$e = \frac{v_1 + v_2 + f_1 + f_2}{4} \quad (4.7)$$

And finally, each original vertex, P , is moved by performing weighted averaging between itself, the average of edge midpoints, M , and the average of face points, F , that are in its neighborhood as follows

$$v = \frac{F + 2M + (n - 3)P}{n} \quad (4.8)$$

4.2 3D Wound Texture Mapping

There are a myriad of single view wound images available on the Web and in clinics around the world. In many cases, physicians are viewing images of wounds with little to no context. One critical piece of information is the physical anatomical location of the wound. Wound images may be projectively texture mapped onto a 3D human anatomy model to obtain a 3D visualization of the wound.

Texture Mapping Methodology

Texture mapping is an established method for adding details to low polygon meshes, and it is also useful for demarcating anatomical structures [191, 192, 193]. We leverage this convenience in order to add anatomical regions by digitally painting a texture map. This is in contrast to many of the other solutions listed in this paper that manually divide the mesh into sub-components and then assign colors to each component. Using texture mapping for region coloring allows us to achieve high precision labeling for very small regions of the skin surface. For example, we are able to label minute details of the eye via painting rather than by tediously sculpting mesh topology.

We are also able to easily change the anatomical regions to add new, to remove or to modify existing regions. This is important for incorporating updated feedback from clinical surface anatomy experts. The caveat to the added flexibility of texture mapping is that it makes it more difficult to highlight regions interactively. To do so requires low level image

processing via pixel manipulation in order to highlight specific regions.

In order to interactively update the texture map image, we must process the texture image in real time. In olden times, this would require an image processing library. However, HTML5 canvas provides simple tools to enable image editing directly within the browser. We have used two methods with which to edit texture images interactively. Both methods start with loading the texture image into an HTML canvas element. To modify pixels directly, we save the canvas as image data, modify the data's pixels and then put the image data back into the canvas. If we do not need to modify pixels individually, then a second method is to draw with the drawing tools provided by the canvas. Specifically, we have used arcs and strokes to draw dots and region highlights onto our texture. Lastly, we have to give the updated texture map image to ThreeJS. This is accomplished by using the canvas' *toDataURL* function, which returns a data URL from the canvas that can be passed around and assigned within HTML.

Wound Image Alignment

In order to use projective texture mapping to project a wound image onto our 3D human anatomy model, we must first determine a correspondence between points on the image and points on the mesh. One way to do this is to allow users to pick three or more pairs of points between the image and mesh, and then solve the resulting overdetermined linear system using least squares. A user selects pairs of points, \vec{v} on the image and \vec{V}' on the mesh. Using ray tracing, points on the view plane, \vec{v}' , can be calculated from points

$\vec{\mathbf{V}}'$ that are in 3D. With these pairs of points, $\vec{\mathbf{v}}$ and $\vec{\mathbf{v}}'$, a 3×3 transformation matrix M is calculated to transform the image to the correct position on the view plane so that it can projectively texture mapped onto the 3D mesh.

$$\vec{\mathbf{v}}' = \mathbf{M}\vec{\mathbf{v}} \quad (4.9)$$

The matrix M transforms between 2D homogeneous coordinates and is defined as follows

$$\mathbf{M} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

Transformation between homogeneous coordinates requires only that \mathbf{v} and \mathbf{v}' are proportional, such that we can compute the cross product as follows

$$\vec{\mathbf{v}}' \times \mathbf{M}\vec{\mathbf{v}} = 0 \quad (4.11)$$

Set the rows of M as $\vec{\mathbf{m}}_1$, $\vec{\mathbf{m}}_2$ and $\vec{\mathbf{m}}_3$, with $\vec{\mathbf{v}} = (x_1, y_1, 1)$ and $\vec{\mathbf{v}}' = (x'_1, y'_1, 1)$, then the cross product in equation (4.11) becomes

$$\begin{bmatrix} y'\vec{\mathbf{m}}_3^T\vec{\mathbf{v}} - \vec{\mathbf{m}}_2^T\vec{\mathbf{v}} \\ \vec{\mathbf{m}}_1^T\vec{\mathbf{v}} - x'\vec{\mathbf{m}}_3^T\vec{\mathbf{v}} \\ x'\vec{\mathbf{m}}_2^T\vec{\mathbf{v}} - y'\vec{\mathbf{m}}_1^T\vec{\mathbf{v}} \end{bmatrix} = \vec{\mathbf{0}}_3 \quad (4.12)$$

Equation (4.12) can be expanded into matrix form as follows

$$\begin{bmatrix} \vec{\mathbf{0}}_3^T & -\vec{\mathbf{v}}^T & y'\vec{\mathbf{v}}^T \\ \vec{\mathbf{v}}^T & \vec{\mathbf{0}}_3^T & -x'\vec{\mathbf{v}}^T \\ -y'\vec{\mathbf{v}}^T & x'\vec{\mathbf{v}}^T & \vec{\mathbf{0}}_3^T \end{bmatrix} \begin{bmatrix} \vec{\mathbf{m}}_1 \\ \vec{\mathbf{m}}_2 \\ \vec{\mathbf{m}}_3 \end{bmatrix} = \vec{\mathbf{0}}_3 \quad (4.13)$$

The third row of the matrix in equation (4.13) is a linear combination of the first two rows by multiplying the first row by $-x$ and the second row by $-y$. The expanded form of equation (4.13) is then

$$\begin{bmatrix}
0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \\
x & y & z & 0 & 0 & 0 & -x'x & x'y & -x'
\end{bmatrix}
\begin{bmatrix}
m_{00} \\
m_{01} \\
m_{02} \\
m_{10} \\
m_{11} \\
m_{12} \\
0 \\
0 \\
1
\end{bmatrix}
= \vec{\mathbf{0}}_9 \quad (4.14)$$

To simplify equation (4.14) further, note that $\vec{\mathbf{m}}_{\mathbf{3}} = [0, 0, 1]$, so that we remove the columns that are multiplied by 0 and move the column multiplied by 1 to the other side of the equation as follows

$$\begin{bmatrix} 0 & 0 & 0 & -x & -y & -1 \\ x & y & z & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{10} \\ m_{11} \\ m_{12} \end{bmatrix} = \begin{bmatrix} -y' \\ x' \end{bmatrix} \quad (4.15)$$

There are six unknowns which means that we need at minimum six equations which are obtained from three pairs of points. Here then is the full system with n pairs of points

$$\begin{bmatrix} 0 & 0 & 0 & -x_1 & -y_1 & -1 \\ x_1 & y_1 & z_1 & 0 & 0 & 0 \\ & & & \vdots & & \\ 0 & 0 & 0 & -x_n & -y_n & -1 \\ x_n & y_n & z_n & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{10} \\ m_{11} \\ m_{12} \end{bmatrix} = \begin{bmatrix} -y'_1 \\ x'_1 \\ \vdots \\ -y'_n \\ x'_n \end{bmatrix} \quad (4.16)$$

Let the left matrix be defined as M , and we now solve the system using least squares with normal equations

$$\mathbf{M}^T \mathbf{M} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{10} \\ m_{11} \\ m_{12} \end{bmatrix} = \mathbf{M}^T \begin{bmatrix} -y'_1 \\ x'_1 \\ \vdots \\ -y'_n \\ x'_n \end{bmatrix} \quad (4.17)$$

We have implemented these equations in JavaScript using custom matrix libraries to solve linear systems using LU decomposition. Figure 4.13 shows two example correspondences. The wound image on the upper left is a diabetic arterial wound provided by Dr. Jeffrey Niezgoda from the AZH Wound Care and Hyperbaric Oxygen Center. On the right is our 3D human anatomy model. The wound image has been transformed on top of the 3D model according to three points selected interactively by the user.



Figure 4.13: 3-Point Image Correspondence

Texture Mapping

Once the segmented wound images has been properly aligned, it must be texture mapped onto the human model. To do this, we leverage barycentric coordinates. For a triangle with three vertices \mathbf{a} , \mathbf{b} and \mathbf{c} , the barycentric coordinate for point \mathbf{P} is

$$\mathbf{P}(\alpha, \beta, \gamma) = \alpha \mathbf{a} + \beta \mathbf{b} + \gamma \mathbf{c} \quad (4.18)$$

The area of the triangle is

$$\mathbf{A} = \frac{|(\mathbf{c} - \mathbf{a}) \times (\mathbf{b} - \mathbf{a})|}{2} \quad (4.19)$$

The barycentric coordinates are proportional to the area of the triangle produced from intersection of bisectors opposite the vertex as follows

$$\alpha = \frac{|(\mathbf{b} - \mathbf{P}) \times (\mathbf{c} - \mathbf{P})|}{2A} \quad (4.20)$$

$$\beta = \frac{|(\mathbf{a} - \mathbf{P}) \times (\mathbf{c} - \mathbf{P})|}{2A} \quad (4.21)$$

$$\gamma = 1 - \alpha - \beta \quad (4.22)$$

In order to determine which triangles need to be copied to the texture map, we iterate over all vertices in the model and calculate their projection to the camera view plane. Say that the position of the camera in 3D space is \mathbf{C} and the view plane is positioned at z_{vp} along the z-axis, then a projection point \mathbf{P}_p for 3D point \mathbf{P} is

$$\mathbf{P}_p = \mathbf{P} \frac{c_z - z_{vp}}{c_z - z} + \mathbf{C} \frac{z_{vp} - z}{c_z - z} \quad (4.23)$$

For each projection point \mathbf{P}_p that has a wound texture, we paste its wound texture triangle into the corresponding triangle in the texture map using *barycentric coordinates*. The trick is to iterate over the pixels with higher resolution, either from the view plane or from the texture map. If the texture map has higher resolution, then we can pull the corresponding pixel value from the image plane directly. If the image plane has higher resolution then we can take the average of all the pixels in the image plane that map to the

same pixel in the texture map and keep track of a counter c . We then can iteratively update the texture map pixel as

$$\mathbf{RGB}_{new} = \frac{\mathbf{RGB}_{old} + \mathbf{RGB}_{pixel}}{++c} \quad (4.24)$$

Projective Texture Mapping

Projective texture mapping provides a quick and dirty way to use images as 3D textures by projecting them onto 3D scenes and objects as if by a slide projector. It is important to have an offline, software-based, general purpose algorithm that's efficient enough to run on the CPU, portable and pluggable into a variety of applications. We present an offline algorithm that rasterizes a texture map directly for saving and retrieval to support applications that use UV mapped texture maps. Processing time is reduced with pixel bucketing. Pixel buckets is a data structure that reduces ray intersection calculations. A scan line algorithm is applied per polygon for each polygon in the texture map by projecting them onto the camera's view projection window. The end result is a portable offline CPU bound algorithm for projective texture mapping.

Projective texture mapping is found in graphics packages such as OpenGL , WebGL, DirectX and Metal. In contrast to 2D/3D digital painting applications where high levels of skill are employed, image projections can be applied quickly for rapid prototyping and are suitable for automated geometric processing.

In traditional graphics packages, projective texture mapping is done as part of the graph-

ics pipeline and renders displays in realtime. However, an application that creates texture maps directly without necessarily displaying it can be done offline without needing to leverage a GPU-backed rendering pipeline. This is useful, for example, for automated procedures that rasterize texture maps and save them for later use.

Offline projective mapping can be broken down into two phases. The first phase is to generate a mapping from the camera's projection window pixels to a list of polygon faces, a.k.a. pixel bucketing. In the second phase, the texture map is rasterized and saved to disk. We assume that the parameterization and UV mapping of the 3D object have already been done. In the pixel bucketing phase, a data structure is created that stores for each projection window pixel a list of faces that are intersected when a ray is cast from the camera through that pixel. In the projection phase, the faces are iterated over. A scan line algorithm traverses each face's pixels in the texture map and a check is made against the other faces in the relevant pixel bucket. If the texture map pixel is visible, it is painted with colors from the texture.

View Dependent Texture Mapping

Many applications dynamically texture 3D scenes using projective texture mapping. These texture maps can be morphed using image morphing, 2D and 3D transformations, landmark marking of critical points, 2D to 3D correspondences, etc. so that different views produce plausible looking scene textures [194, 195, 196, 197, 198]. Nakashima, Uno, Kawai, Sato & Yokoya (2015) improve upon view dependent projective texture mapping with depth maps

that help reduce artifacts [199].

Gard, Hilsmann & Eisert (2019) demonstrate how projective texture mapping is used in object tracking systems by iteratively updating object edges and projecting distorted textures [200]. Milosavljević, Rančić, Dimitrijević, Predić & Mihajlović (2016) apply projective texture mapping to enable video surveillance from 3D geographic information systems (GIS) data [201]. Lin, et al. (2019) use view dependent projective texture mapping to provide a more stable background during remote AR telementoring [202]. Criminisi & Zisserman (2013) use projective texture mapping to distort planar textures as part of a process to estimate plane orientation in images taken with uncalibrated cameras [203].

3D Digital Painting

In 3D digital painting applications, brush tips are represented as textured images that are projected onto a 3D model as the user paints in realtime [204, 205, 206]. Fu & Chen’s (2008) haptic painting system uses projective texture mapping to paint 3D models in realtime [207]. Cheah, et al. (2018) built a database of 3D wound data by digitally painting wounds on human models [4].

Many applications allow for interactive texture mapping [208, 209, 210, 211]. Gingold, I., Davidson, Han & Zorin (2008) use projective texture mapping for texture placement onto 3D models in realtime [212]. Cherin, Cordier & Melkemi (2013) show how projectively texture mapping can be done intuitively by end users [213]. Chun & Höllerer (2013) employ projective texture mapping to enable realtime interaction with 3D models on mobile phones

[214].

Augmented Reality

Augmented reality is a popular area for the use of projective texture mapping. In this context, we enhance the real world with 3D objects via 3D reconstruction, spatial positioning and 3D visualization [215, 216]. Gauglitz, Nuernberger, Turk & Höllerer (2014) use 3d augmented reality video to allow users to interact remotely [217]. Hirsch, Wetzstein, & Raskar (2014) use projective texture mapping to create a glass-free 3D theatre experience [218]. Hauswiesner, Straka & Reitmayr (2011) projectively texture map clothing onto automated 3D models of people trying on wardrobe using 3D augmented VR [219, 220].

Remote controlled agents use 3D vision for navigation and decision making. Cho & Cho (2019) demonstrate how projective texture mapping of 2D segmented scenes can be used to reconstruct 3D scene objects in realtime during remote controlled unmanned robot expeditions [221]. Buss, et al. (2010) built a 3D projective texture mapped scene whereby two remote operators can collaborate simultaneously [222]. Fondazione, Kessler, Ortin & Remondino (2012) use projective texture mapping for mobile street navigation [223].

3D Reconstruction

Projective texture mapping is commonly used to texture map scenes, landscapes, models, point clouds and other objects from photographs [224, 225, 226, 227, 228, 229, 230, 231]. In photogrammetry, 3D models are projectively texture mapped from a series of photographs

using ray triangulation [232]. It can also be accomplished from video streams [233]. Grammatikopoulos, et al. (2012) demonstrate how projective texture mapping with ortho and perspective views can be used to improve close-range photogrammetry [234]. A less precise reconstruction approach that approximates 3D shapes from segmented images, known as visual hulls, also apply projective texture mapping [235, 236]. Tsai & Lin (2007) projectively texture map mosaics of close-range photographs to texture 3D buildings in cyber cities [237]. Tang, et al. (2018) projectively map textures onto 3d meshes after realtime compression of streaming 4D video [238]. Zhang, Li, Cheng, Cai, Chao & Rui (2015) projectively texture map 3D models generated from stereo views [76].

Offline Projective Texture Mapping

Background

In modern graphical engine pipelines a 3D scene is projected onto a 2D image using a specific camera calibration, volume clipping and normalization. Conversely, projective texture mapping projects images back out into a 3D scene as if by a slide projector [239]. The underlying mesh surface parameterization is used to update or create an underlying texture map, such as a UV texture map for a triangulated mesh.

The canonical application of projective texture mapping is for lighting and shadowing effects in games and animation [240, 241, 242, 243]. Yu, Guo, Lin, Lumsdaine & Yu (2013) use projective texture maps for image-based light fields [244]. Tetzlaff & Meyer (2017) [245] demonstrate how projective texture mapping can be used to alter scene lighting.

Offline Workflow

We use the term offline in the context of projective texture mapping to refer to any procedure that generates texture maps from image projections for the purpose of saving to a file, as opposed to rendering to a display. This is what happens, for example, in Photoshop, when an image layer is merged with a 3D layer. Emphasis is placed on the projective texture mapping algorithm and image processing routines. There is no strict need for GPGPU, or general purpose GPU shaders, but we emphasize efficiency and performance within the context of CPU processing.

Pixel Bucketing

Pixel bucketing is a preprocessing stage to reduce the number of face intersection computations needed in the projective texture mapping phase. In the worst case, bucketing requires as many intersection calculations as standard ray casting. For F faces and a projection window resolution of $m \times n$, the number of intersection computations may be as high as $F * m * n$, barring any acceleration structure optimizations. In practice we expect much better performance because each projection pixel maps to only small list of faces. For example, if the mesh is a sphere, there are no more than two face intersections per pixel.

Pixel bucketing uses projection window coordinates to determine which pixel buckets a polygon face belongs in. As each face is projected onto the projection window, a scan line over the polygon determines all pixels covered by the polygon, and the polygon is added to each pixel's bucket. In the worst case we do not save more than standard ray casting. If

each polygon covers the entire projection window, then for F faces and projection window resolution $m \times n$, pixel bucketing takes time $\mathcal{O}(F * m * n)$. However, in practice we expect much better because each polygon covers a very small portion of the projection window.

Comparison to Depth Mapping

Depth mapping in graphics packages such as OpenGL is a data structure that stores for each projection pixel, the depth of the nearest face intersection. It runs in time linear to the resolution of the projection window, i.e. $\mathcal{O}(m * n)$. It can be done as a preprocessing step, but has traditionally been done in tandem with the rasterization algorithm, such that new depth values are updated dynamically. Consequently, unlike pixel bucketing, each pixel in the map stores a value, not a list.

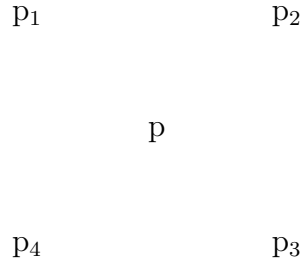
Online projective texture mapping algorithms maintain a data structure similar to a depth map, known as a shadow map. Shadow maps use depth values so that for each pixel from the perspective of a light source, it is determined which faces should be textured and which should be shadowed. Yu, Y. (1999) [246] leverages screen space coordinates in order to efficiently compute a shadow map as part of a projective texture mapping workflow.

Rasterization

To rasterize the texture map, each polygon face in the texture map is rasterized via a scan line routine. For each pixel encountered during the scan line routine, its world coordinates are calculated so that a ray can be projected to it from the camera. This is done by using

the UV mapping, that maps vertices to texture space. Firstly, barycentric coordinates are calculated for the pixel in texture space, from which we can exact the intersection point of the triangle in world space.

To color the texture map with the corresponding color from the projection image, we get the intersection of the camera ray with the projection window. This intersection will lie somewhere within a square of four projection image pixels, from which a bilinear interpolation is performed to obtain the final color for the corresponding texture map pixel. If a point $p = (x, y)$ resides inside of a square defined by four points starting at the upper left and proceeding clockwise as p_1, p_2, p_3, p_4 ,



and if the bounds of this square are $x_{min}, x_{max}, y_{min}, y_{max}$, then we obtain the bilinearly interpolated point, P , as follows

$$\begin{aligned}
 b_1 &= (x_{max} - x)(y_{max} - y) \\
 b_2 &= (x - x_{min})(y_{max} - y) \\
 b_3 &= (x - x_{min})(y - y_{min}) \\
 b_4 &= (x_{max} - x)(y - y_{min})
 \end{aligned} \tag{4.25}$$

$$P = \frac{b_1 * p_1 + b_2 * p_2 + b_3 * p_3 + b_4 * p_4}{(x_{max} - x_{min})(y_{max} - y_{min})} \quad (4.26)$$

Magnification and Minification

Magnification and minification are terms used to describe depth placement of textures in 3D space. In traditional texture mapping, a single texture map pixel may cover numerous screen pixels, and vice versa. In the former case, it is called magnification and happens when a texture is applied closer to the camera. When a polygon is farther from the camera, a single screen pixel covers multiple texture map pixels, referred to as minification. In either case, the algorithm can filter the image in order to choose an appropriate color to guard against artifacts such as aliasing, moire patterns and jaggies.

We use bilinear interpolation to choose a texture map color from the projection texture. In cases of extreme minification or magnification, however, this approach suffers from artifacts. There are solutions to this problem which are beyond the scope of this work. A popular solution is mipmapping. Mipmapping stores the same texture at different resolutions. Higher resolutions are applied closer to the camera, and lower resolutions are applied at distances farther from the camera. This improves over using bilinear interpolation alone, but suffers from hard boundaries between resolution transitions and blurring from foreshortening caused by perspective viewing. Hard boundaries can be ameliorated with trilinear filtering and perspective view blurring can be solved with anisotropic filtering.

4.3 Intensity-Based Wound Depth Map

Recovering 3D reconstruction from shading is an ill-posed problem, meaning that there is no single solution, and hence there is no single way to reconstruct a 3D model from 2D when using shading alone. Inevitably, there will be errors in the 3D reconstruction, and these will be exacerbated by the limitations of the phone camera. However, shading may help visualization in general. For e.g., wounds tend to have sharp grooves and crevices that generate intense specular reflections. There is a basic illumination model for intensity I , i.e RGB color, and vertex v as follows

$$I(v) = K_a I_a + K_d I_d (\mathbf{N} \cdot \mathbf{L}) + K_s I_s (\mathbf{V} \cdot \mathbf{R})^2, \mathbf{N} \cdot \mathbf{L}, \mathbf{V} \cdot \mathbf{R} > 0 \quad (4.27)$$

Where I_a is the ambient color and associated constant K_a , I_d is the diffuse color and associated constant K_d , \mathbf{N} is the vector from the surface, \mathbf{V} faces the viewer, \mathbf{L} faces the light source and \mathbf{R} is the reflected light which in simplest case can be set relative to the normal and incoming light source as $2(\mathbf{N} \cdot \mathbf{L})\mathbf{N} - \mathbf{L}$. We minimize the brightness error between the original image intensities for each pixel, I_o , and the reconstructed pixel intensities I_r

$$\int \int (I_o - I_r)^2 dx dy \quad (4.28)$$

We have used intensity-based depth mapping in Blender to alter the 3D surface of a wound as seen in figure 4.14. This was done by manually segmenting an arterial wound in

Photoshop and then projectively texture mapping it onto our 3D human model by merging the segmented wound image layer onto the human model 3D layer.

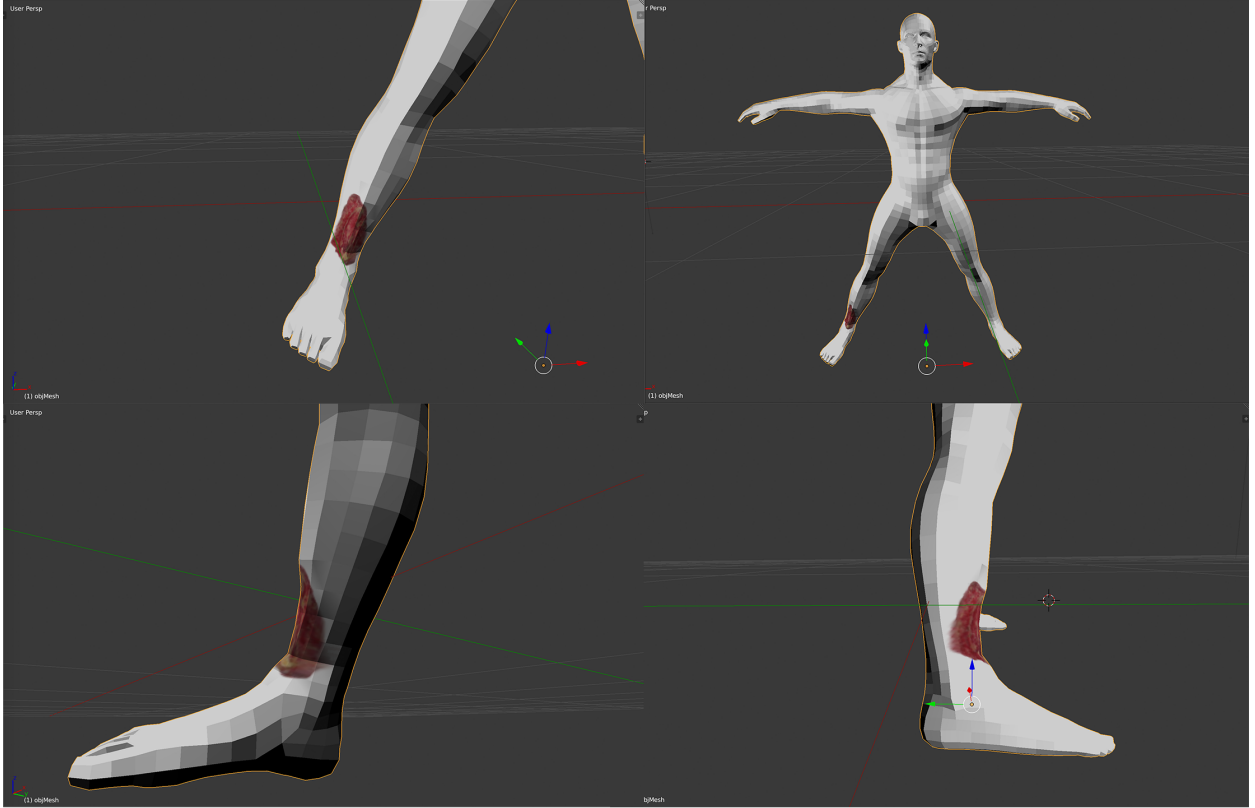


Figure 4.14: Wound 3D Intensity Displacement

4.4 Limitations

As we have noted previously, 3D reconstruction from a single image is an ill-posed problem owing to the multiplicity of possible solutions. Another limitation of our approach to 3D visualization is that we have used a standard 3D human model. In reality, human beings

have quite a significant amount of anatomical variations in gait, size, posture, shape, etc. These and other person-to-person peculiarities are ignored by our use of a common 3D model. Preferably, which is out of scope for our work, we would like to employ a automated solution that incorporates each patient's body type. Cheah, Kangkorn, Tan, Loo and Chong (2018) [4] deal with this in their Android-based mobile app by using six 3D human anatomy models instead of one. Three are female, three are male, and each is split into thin, average and overweight human body types.

5. Wound Assessment Mobile App

5.1 End-to-End Workflow

The ultimate goal of mobile wound assessment is to simplify everyone's lives by making the user experience as seamless and user friendly as possible. This means automating as much as can be safely and accurately done so, and then providing easy to use interactive tools to help users carry out their daily tasks. At one extreme are mobile apps that automate the entire workflow. For example, the approach posited by Chakraborty, et al. (2019) [2] strives to be purely computational in diagnosing chronic wound tissue. Mobile software preprocesses the wound image before automatically segmenting it using *fuzzy c-means clustering*. In the classification phase, color features are extracted and identified as one of the three major healing stages of granulation, slough or necrotic, i.e. red, yellow or black respectively.

We take a sort of middle of the line approach with an emphasis on 3D visualization instead of classification. Segmentation is semi-automated and 3D visualization is interactive. The mobile app that we are creating is a cloud-based storage application that securely stores wound images taken from camera phones. The workflow starts with the user opening the

mobile app, and then selecting the camera icon to take a single picture, as shown in figure 5.1. The wound is then segmented from the background using the neural network described below. Either the original wound photo or the segmented wound can be run through a neural network classifier to classify the wound.

The user then positions the segmented wound image onto the 3D human model. This can be done in one of two ways. The user may rotate the 3D model and line up the wound image manually, or the user may pick pairs of points between the segmented wound image and the 3D model. The location is then used to projectively texture map the wound onto the 3D model.

5.2 Neural Networks for Wound Assessment

As has already been mentioned, neural networks are the future of wound segmentation and classification. There are mobile libraries for iOS and Android that can be used to build embedded neural networks. A localized mobile-residing neural network is currently beyond the scope of this work. However, because our mobile solution is cloud-based, it is perfectly reasonable to perform CNN-based segmentation and classification on a remote machine. The primary benefit of this approach is that we offload high intensity computations from the device to a high-end GPU. In our lab we use the NVIDIA RTX 2080Ti GPU.

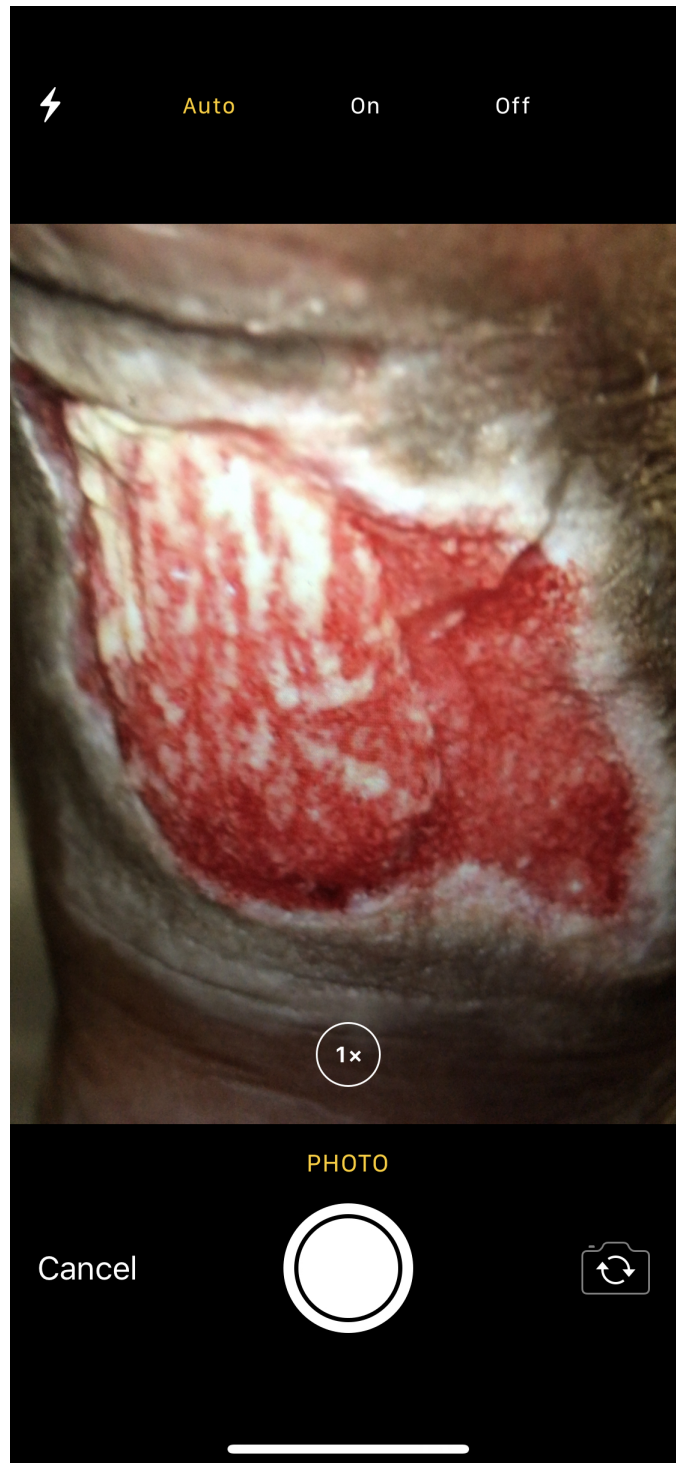


Figure 5.1: Camera page

Neural Network for Wound Segmentation

We leverage the work of Wang (2019) [247] who built a deep neural network to segment chronic ulcers from wound images. Of the 305 wound images obtained from Dr. Jeffrey Niezgoda at the AZH Wound Care and Hyperbaric Oxygen Center, 259 images were used for training and 46 images for testing. A dice coefficient of 85% is achieved. Two segmentation results are shown in figure 5.2.

Wang, et al. (2020) [248] updated to a new segmentation NN based on MobileNetV2 with transfer learning and hole filling. Additional training data is made available from the Medetec Wound Database. These enhancements have improved the segmentation dice score to 86.78%. Specifically, pre-trained data is transferred from the Pascal VOC segmentation dataset and is used to initialize the NN weights. Holes can be caused by fibrinous tissue or noise. The final segmentation, then, fills in these holes by finding connected components within the segmentation boundary.

Neural Network for Wound Classification

We leverage the work of Rostami (2019) [249] to classify burn wound images as either deep thermal and deep thickness or superficial. A total of 94 images were used to train AlexNet using transfer learning, where 20 were used for training and 74 for testing. In addition, training data was augmented 16x using image affine transformations. A result of 89.2% accuracy was achieved. Examples of each type of burn wound appears in figure 5.4

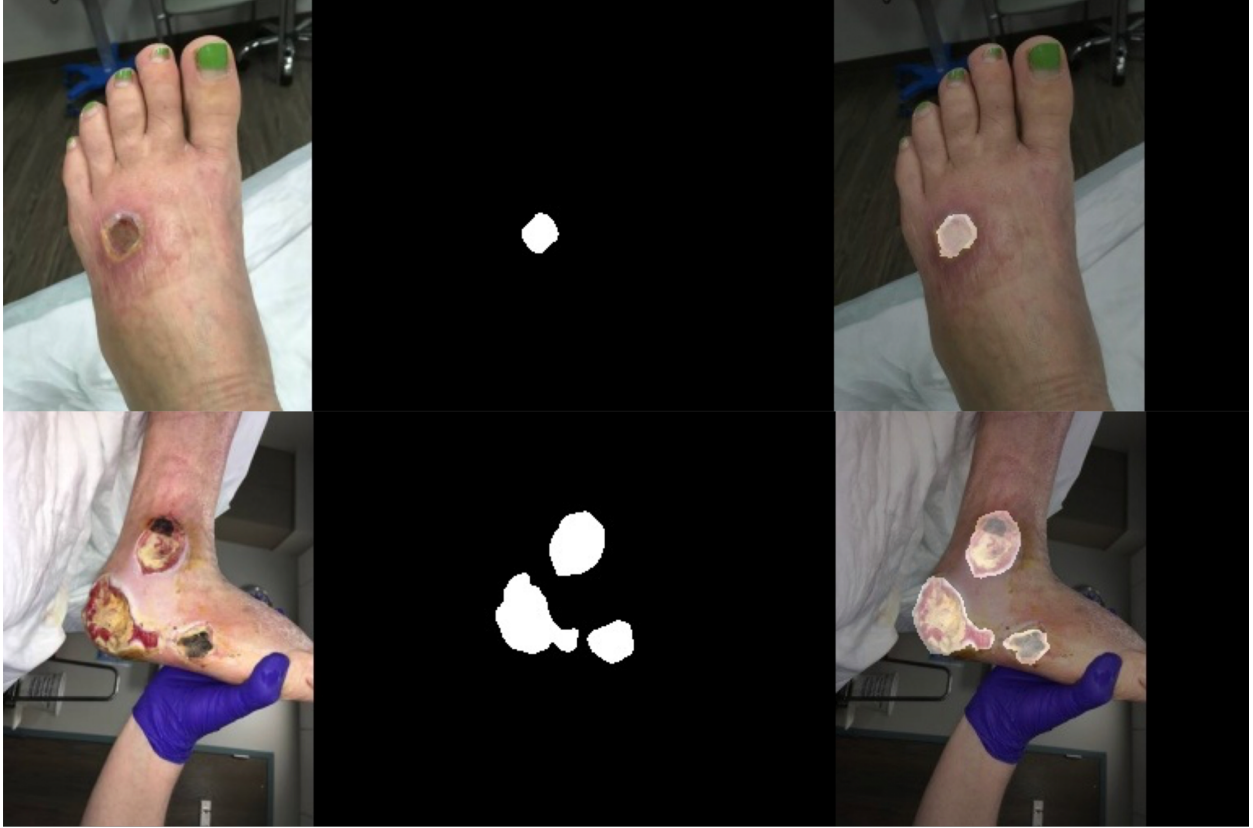


Figure 5.2: Wound segmentation

(a) First column, original image (b) Second column, manual ground truth (c) Third column, the prediction

Rostami, et al. (2020) have updated their research with a more sophisticated classification network that classifies wounds into one of venous, diabetic, pressure or surgical wound, as shown in figure 5.5. This is particularly useful for our work which emphasizes chronic wounds. Furthermore, classification is done via patches which could mitigate the need for segmentation in some contexts.

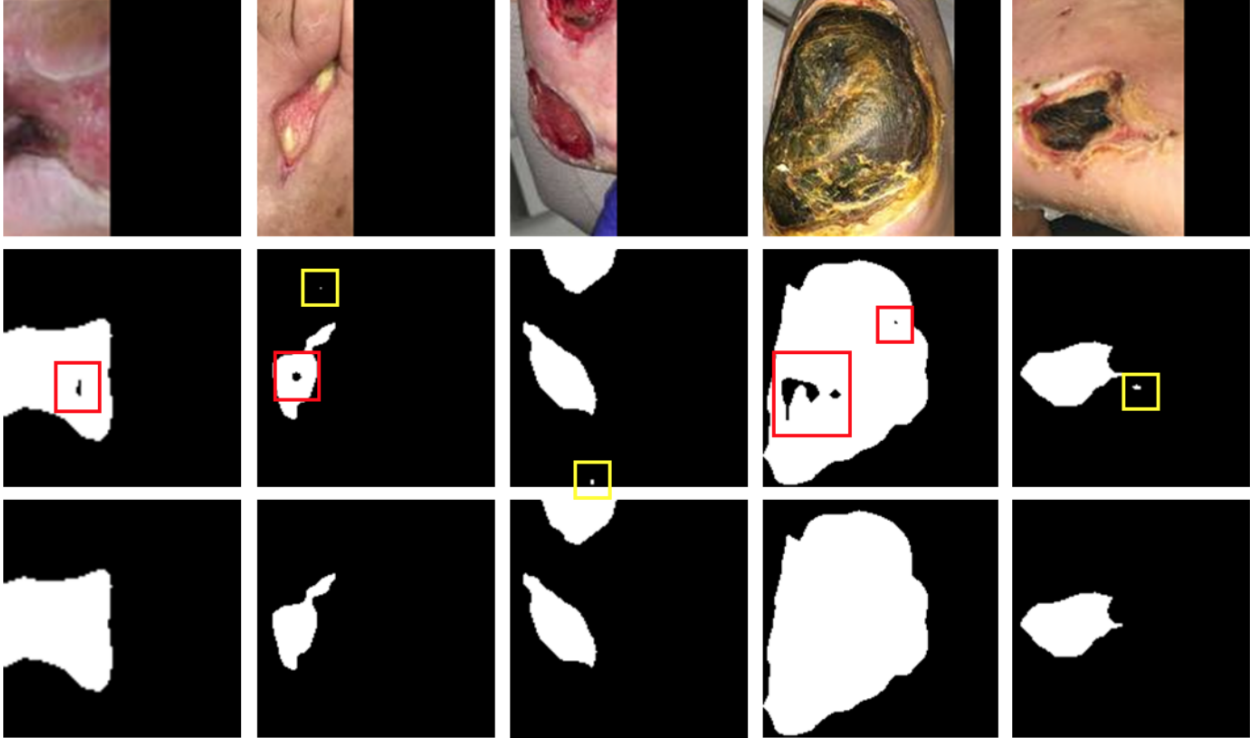


Figure 5.3: Wound segmentation w/ hole filling

(a) First row, wound images (b) Second row, segmentation results (c) Third row, with hole filling

5.3 Semi-Automatic Unsupervised Wound

Segmentation

We have created a semi automated wound image segmentation tool for mobile platforms suitable for resource constrained environments. Mobile wound assessment has matured dramatically over the last decade, with mobile wound image capture becoming a go-to for clinicians, nurses, doctors and medical professionals alike who want to streamline wound monitoring and treatment. Modern mobile wound assessment segments wound images and

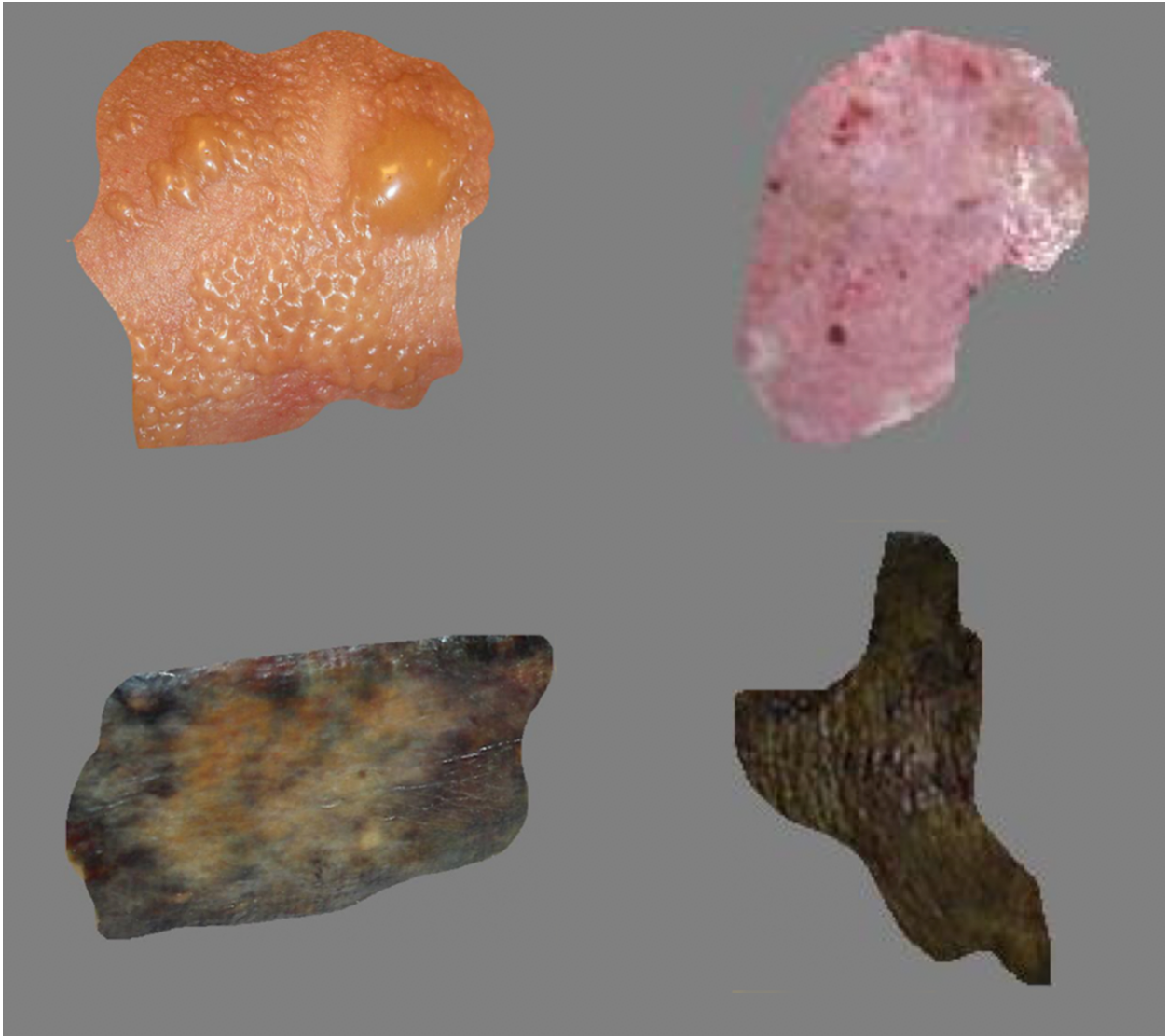


Figure 5.4: Burn Wounds

(a) The first row are superficial burn wounds (b) The second row are deep burn wounds



Figure 5.5: Wound classification

(a) First row, diabetic (b) Second row, venous (c) Third row, pressure (d) Fourth row, surgical

classifies them for automated diagnosis. Convolutional neural networks have become the de facto standard to achieve the most accurate segmentation and classification results. However, CNN's require a large repository of labeled ground truth wound image data which is prohibitively expensive in resource constrained environments. To ameliorate this shortcoming, we have combined industry standard quick selection tools to manually adjust the segmented wound region after local unsupervised wound image segmentation.

Unsupervised Wound Segmentation

Novel advances in telemedicine have made wound image segmentation viable on mobile devices. Wound segmentation is a standard procedure on mobile wound assessment platforms that perform tasks such as tissue classification, wound measurement and wound healing monitoring. While machine learning has become the most popular and most accurate tool for wound segmentation, it is limited by the quality and size of the training data set. Furthermore, gathering manual ground truth wound segmentation data requires expert knowledge from wound doctors and hence remains time consuming and error prone. When automatic segmentation fails, many target applications still require interactive manual modification of the segmented region.

Ambiguity often arises in attempting to determine exactly where the boundary of the wound lies. Experts make subjective judgements when delineating wound borders leading to segmentation variations that potentially adversely affect wound dressing preparation [250]. Our approach demonstrates how traditional unsupervised wound segmentation can be com-

combined with an easy to use selection tool that modifies the segmented region to one's liking. In so doing, high quality wound segmentation is achieved that is customized to the individual user and application.

Semi-Automatic Segmentation

Hettiarachchi, Mahindaratne, Mendis, Nanayakkara & Nanayakkara (2013) [251] allow wound regions segmented via active contour models to be modified manually before using automated camera calibration to measure wound size. A manual component to wound segmentation is useful in contexts where precise segmentation is ambiguous. For example, in applications that have varying requirements for visualizing differing wound tissues and the surrounding skin. To meet these intricacies, we propose using marching ants and a resizable quick selection tool similar to what is available in Photoshop.

Before the end user interactively modifies the wound region, we use an automated unsupervised algorithm to provide a preliminary segmentation to work with. In so doing, we obtain the best of both worlds. Unsupervised wound segmentation quickly hones in on the target wound region, and quick selection allows for easy modification and customization. Once the wound is segmented from the image, the segmented wound image is stored for later retrieval such as would be required by the majority of the wound assessment apps discussed in this paper.

5.4 Mobile Application

3D Visualization

The 3D visualization of the wound is the piece that ties all of the work we have done together. The good news is that we can use existing 3D rendering pipelines to view and interact with our 3D human model. In figure 5.6, we have used the native iOS Model I/O API to render the mesh. In figure 5.7, we have used the ThreeJS WebGL framework to render the model over the Web. The iOS Model I/O API is smoother because it is GPU-accelerated natively by the iOS Metal framework. It is also fully equipped with physics-based arc ball rotation to achieve a very organic feel. ThreeJS is portable to both iOS and Android, making it the favorable solution for prototyping the concepts of our mobile wound assessment platform.

Wound Browsing & Inspection

We have built a prototype for the wound application. The home page of our application is the wound gallery, as seen in figure 5.8. The wound gallery is a scrollable list showing all of the wound images taken for the current patient. The navigation bar of the application allows the end user to take a picture, to add notes and to login. When the end user clicks on a wound image, the wound inspection page comes up. This page allows the end user to perform two-finger touch enabled panning, scaling and rotation of the wound in order to better inspect it, as seen in figure 5.9.

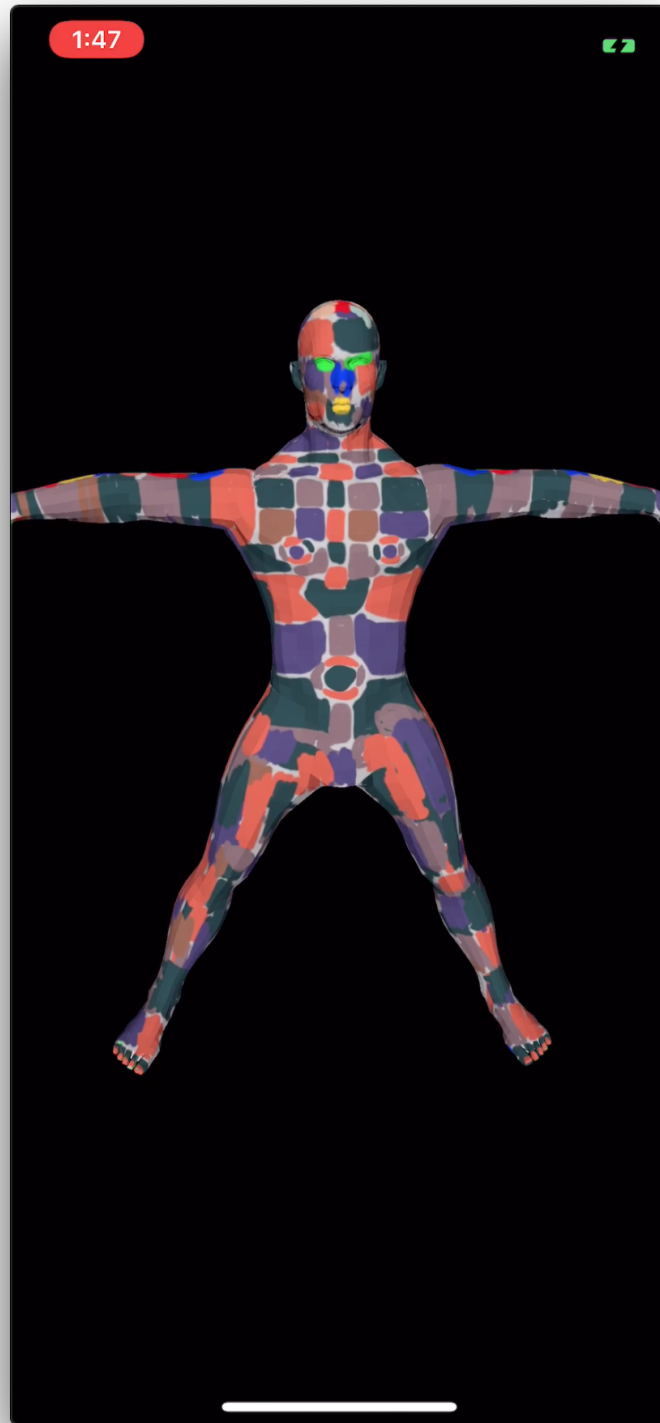


Figure 5.6: Model I/O 3D Render
101

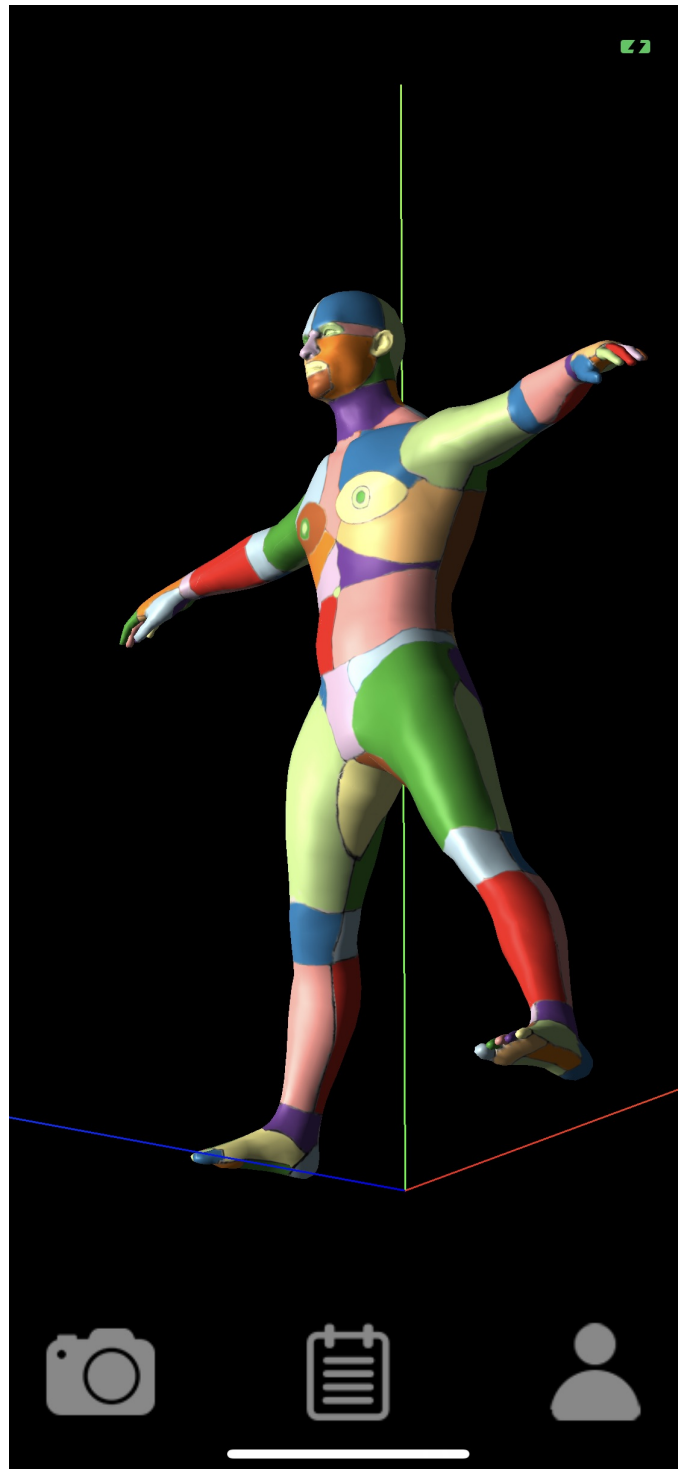


Figure 5.7: Web-based ThreeJS 3D Render



Figure 5.8: Home page

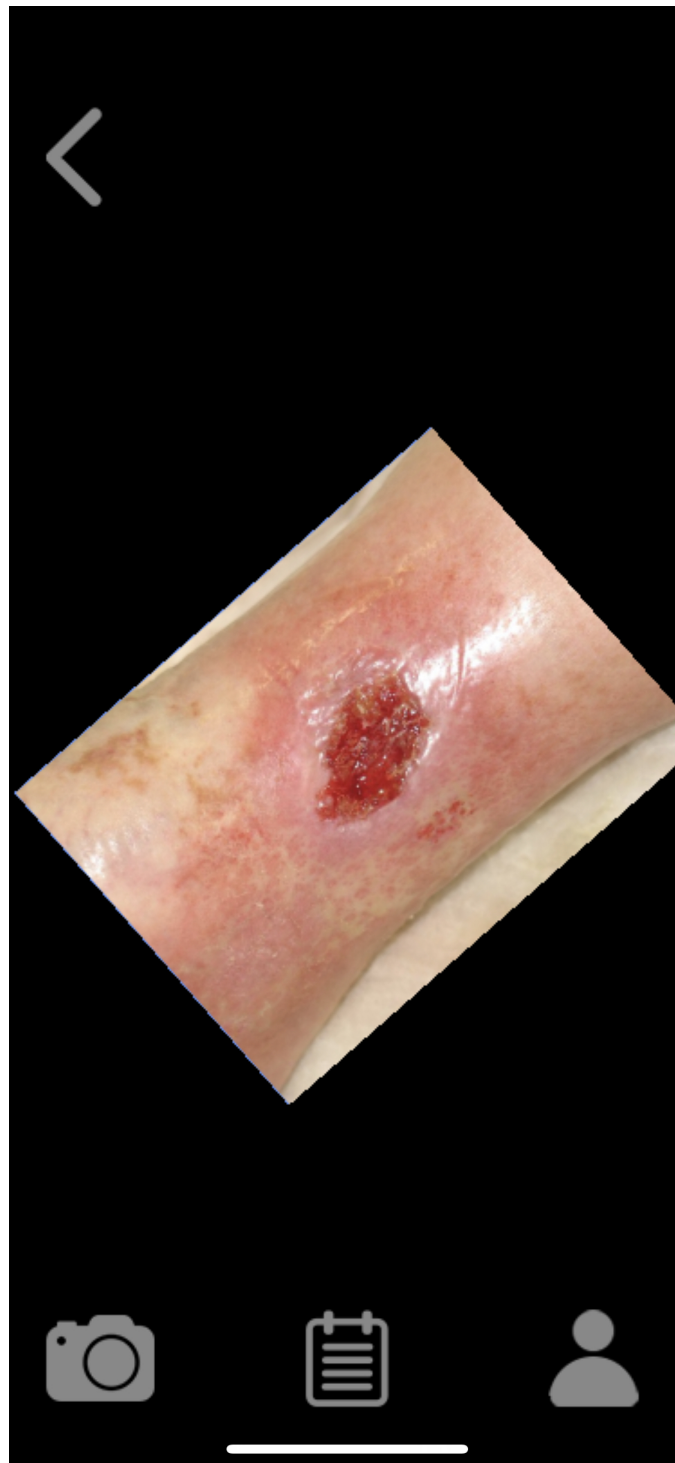


Figure 5.9: Wound inspection page

6. Conclusion & Future work

We have designed a complete mobile wound assessment platform for the remote care and management of wounds. We capitalize on the worldwide proliferation of camera-enabled smartphones to enable wound assessment even in under-resourced and remote environments. We have integrated machine learning algorithms that use convolution neural networks (CNN's) to automatically segment the wound from the image and to classify it into a number of different types based on its tissue composition. We follow this up with semi-automated interactive 3D visualization of the wound. The wound is positioned on a digital 3D human model when the end-user creates pairs of point correspondences between the wound image and the human model, which gives an over-determined linear system that's solvable via least squares. An efficient projective texture mapping algorithm is then used to project the wound as a texture onto the 3D model. The humanoid model is fully segmented and labeled in 3D with the human surface anatomy labels that come from the NYU and Anatomy Mapper labeling systems.

In the future we'd like to incorporate a knowledge-based decision tree with data collected from interviews with domain-level expert physicians. The mobile system will ask the user

a series of questions as determined by the underlying decision tree and then provide a preliminary diagnosis. Patient answers would be stored on the server and made available to physicians for later review. Ideally, physicians would be notified via an automated signaling system which automatically prioritizes wound cases based on their level of urgency.

Over the long term we'd like to build a large database of chronic wound images for use in machine learning tasks such as image segmentation, classification and 3D reconstruction. This would allow the system to be smarter and more judicious in its decision making, further automating workflows and reducing burden on medical personnel. For example, instead of selecting corresponding points between the wound image and the 3D model, an AI-based algorithm would automatically determine the location of the wound based on the wound texture and surrounding body parts.

Finally, since there aren't many databases of 3D wound models, neither openly accessible nor private, we'd like to incorporate results from the iPhone dual camera to obtain 3D depth map data from wound pictures. Depth maps can be used to build a 3D wound model database which would provide data for machine learning algorithms. The goal is to provide more accurate 3D modeling of wounds that would be made possible by more rigorous validation. The other benefit of using the iPhone's dual camera is its potential to incorporate 3D wound data directly within the mobile wound assessment workflow.

BIBLIOGRAPHY

- [1] George Han and Roger Ceilley. *Chronic Wound Healing: A Review of Current Management and Treatments*. Mar. 2017. DOI: 10.1007/s12325-017-0478-y.
- [2] Chinmay Chakraborty. “Computational approach for chronic wound tissue characterization”. In: *Informatics in Medicine Unlocked* (2019). ISSN: 23529148. DOI: 10.1016/j.imu.2019.100162.
- [3] Chong Jiao et al. “Burn image segmentation based on Mask Regions with Convolutional Neural Network deep learning framework: more accurate and more convenient”. In: *Burns & Trauma* 7.1 (Dec. 2019), p. 6. ISSN: 2321-3876. DOI: 10.1186/s41038-018-0137-9. URL: <https://burnstrauma.biomedcentral.com/articles/10.1186/s41038-018-0137-9>.
- [4] A. K. W. Cheah et al. “The validation study on a three-dimensional burn estimation smart-phone application: accurate, free and fast?” In: *Burns & Trauma* 6.1 (Dec. 2018). DOI: 10.1186/s41038-018-0109-0.
- [5] Noël Lüneburg et al. “Photographic LVAD Driveline Wound Infection Recognition Using Deep Learning.” In: *Studies in health technology and informatics* 260 (2019), pp. 192–199. ISSN: 1879-8365. URL: <http://www.ncbi.nlm.nih.gov/pubmed/31118337>.
- [6] Houman Mirzaalian Dastjerdi et al. “Measuring surface area of skin lesions with 2D and 3D algorithms”. In: *International Journal of Biomedical Imaging* 2019 (2019). ISSN: 16874196. DOI: 10.1155/2019/4035148.
- [7] Beth G Goldstein and Adam O Goldstein. “Diagnosis and management of malignant melanoma”. In: *American family physician* 63.7 (2001), p. 1359.
- [8] Judith Lindert and Kianusch Tafazzoli-Lari. “Optical coherence tomography provides an optical biopsy of burn wounds in children—a pilot study”. In: *Journal of Biomedical Optics* 23.10 (Oct. 2018), p. 1. ISSN: 1083-3668. DOI: 10.1117/1.JBO.23.10.106005. URL: <https://www.spiedigitallibrary.org/journals/journal-of-biomedical-optics/volume-23/issue-10/106005/Optical-coherence-tomography-provides-an-optical-biopsy-of-burn-wounds/10.1117/1.JBO.23.10.106005.full>.

- [9] John Doupis and Aristidis Veves. “Classification, diagnosis, and treatment of diabetic foot ulcers.” In: *Wounds: a compendium of clinical research and practice* 20.5 (2008), pp. 117–126.
- [10] David Sánchez-Jiménez et al. “Development of Computer Vision Applications to Automate the Measurement of the Dimensions of Skin Wounds”. In: *Proceedings* 19.1 (July 2019), p. 18. ISSN: 2504-3900. DOI: 10.3390/proceedings2019019018. URL: <https://www.mdpi.com/2504-3900/19/1/18>.
- [11] Marta Wilczkowiak, Edmond Boyer, and Peter Sturm. “Camera calibration and 3D reconstruction from single images using parallelepipeds”. In: *Proceedings of the IEEE International Conference on Computer Vision* (2001). DOI: 10.1109/ICCV.2001.937510.
- [12] Dan Liu, Xuejun Liu, and Meizhen Wang. “Camera Self-Calibration with Lens Distortion from a Single Image”. In: (2016). DOI: 10.14358/PERS.82.5.325.
- [13] Abhishek Kar et al. “Category-specific object reconstruction from a single image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1966–1974.
- [14] Sami Romdhani and Thomas Vetter. “Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior”. In: *Computer Vision and Pattern Recognition* 2 (2005), pp. 986–993.
- [15] Clement Creusot, Nick Pears, and Jim Austin. “A machine-learning approach to key-point detection and landmarking on 3D meshes”. In: *International journal of computer vision* 102.1-3 (Mar. 2013).
- [16] Halim Benhabiles et al. “Learning Boundary Edges for 3D-Mesh Segmentation”. In: *Computer Graphics Forum* 30.8 (Dec. 2011).
- [17] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. “Learning 3D mesh segmentation and labeling”. In: *ACM Transactions on Graphics (TOG)* 29.4 (July 2010).
- [18] Jamie Shotton et al. “Real-time human pose recognition in parts from single depth images”. In: *Communications of the ACM* 56.1 (Jan. 2013).
- [19] Jason Rock et al. “Completing 3D object shape from one depth image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2484–2493.
- [20] Sarah F F Gibson and Brian Mirtich. *MERL L A MITSUBISHI ELECTRIC RESEARCH LABORATORY A Survey of Deformable Modeling in Computer Graphics*. Tech. rep. 1997, pp. 97–116. URL: <http://www.merl.com>.
- [21] Aaron S Jackson et al. *Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression*. Mar. 2017.

- [22] Johanna Delanoy et al. *What You Sketch Is What You Get: 3D Sketching using Multi-View Deep Volumetric Prediction*. July 2017.
- [23] Xiaoguang Han, Chang Gao, and Yizhou Yu. *DeepSketch2Face: A Deep Learning Based Sketching System for 3D Face and Caricature Modeling*. June 2017.
- [24] Fang Wang, Le Kang, and Yi Li. “Sketch-based 3d shape retrieval using convolutional neural networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1875–1883.
- [25] Paul Wohlhart and Vincent Lepetit. “Learning descriptors for object recognition and 3d pose estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3109–3118.
- [26] Christopher B Choy et al. “3d-r2n2: A unified approach for single and multi-view 3d object reconstruction”. In: *European Conference on Computer Vision*. Springer International Publishing, Oct. 2016, pp. 628–644.
- [27] Alexey Dosovitskiy et al. “Learning to generate chairs, tables and cars with convolutional networks”. In: *Computer Graphics Forum* 39.4 (Apr. 2017), pp. 692–705.
- [28] Xiaoguang Han et al. *High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference*. 2017.
- [29] Kan Guo, Dongqing Zou, and Xiaowu Chen. “3d mesh labeling via deep convolutional neural networks”. In: *ACM Transactions on Graphics (TOG)* 35.1 (Dec. 2015).
- [30] Oliver Nalbach et al. “Deep Shading: Convolutional Neural Networks for Screen Space Shading”. In: *Computer Graphics Forum* 36.4 (July 2017), pp. 65–78.
- [31] Danilo Jimenez Rezende et al. “Unsupervised learning of 3d structure from images”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 4996–5004.
- [32] Erick Delage, Honglak Lee, and Andrew Y Ng. “A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image”. In: *Computer Vision and Pattern Recognition* 29.4 (2006), pp. 2418–2428.
- [33] Ashutosh Saxena, Min Sun, and Andrew Y Ng. “Make3d: Learning 3d scene structure from a single still image”. In: *IEEE transactions on pattern analysis and machine intelligence* 31.5 (May 2009), pp. 824–840.
- [34] Said Benameur et al. “A hierarchical statistical modeling approach for the unsupervised 3-D biplanar reconstruction of the scoliotic spine”. In: *IEEE Transactions on Biomedical Engineering* 52.12 (2005), pp. 2041–2057.
- [35] Ludovic Humbert et al. “3D reconstruction of the spine from biplanar X-rays using parametric models based on transversal and longitudinal inferences”. In: *Medical engineering & physics* 31.6 (July 2009), pp. 681–687.

- [36] Thierry Cresson et al. “Coupling 2D/3D registration method and statistical model to perform 3D reconstruction from partial x-rays images data”. In: *Engineering in Medicine and Biology Society*. Annual International Conference of the IEEE, 2009, pp. 1008–1011.
- [37] Guoyan Zheng et al. “Reconstruction of patient-specific 3D bone surface from 2D calibrated fluoroscopic images and point distribution model”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2006, pp. 25–32.
- [38] Edilson De Aguiar et al. “Marker-less deformable mesh tracking for human shape and motion capture”. In: *Computer Vision and Pattern Recognition*. 2007, pp. 1–8.
- [39] Alexandru O Bălan and Michael J Black. “The naked truth: Estimating body shape under clothing”. In: *European Conference on Computer Vision*. 2009, pp. 15–29.
- [40] Alec Rivers, Fredo Durand, and Takeo Igarashi. “3D modeling with silhouettes”. In: *ACM SIGGRAPH 2010 Papers, SIGGRAPH 2010*. 2010. ISBN: 9781450302104. DOI: 10.1145/1778765.1778846. URL: <https://dl.acm.org/citation.cfm?id=1778846>.
- [41] Andrew Nealen et al. “FiberMesh: Designing freeform surfaces with 3D curves”. In: *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*. 2007. DOI: 10.1145/1275808.1276429. URL: <https://dl.acm.org/citation.cfm?id=1276429>.
- [42] Chunxiao Li et al. “Sketch-based 3D modeling by aligning outlines of an image”. In: *Journal of Computational Design and Engineering* 3.3 (2016), pp. 286–294. ISSN: 22885048. DOI: 10.1016/j.jcde.2016.04.003. URL: <https://www.sciencedirect.com/science/article/pii/S228843001530049X>.
- [43] Yong-Jin Liu et al. *EasyToy: Plush Toy Design Using Editable Sketching Curves*. Tech. rep. 2011. DOI: 10.1109/MCG.2009.147.
- [44] Peter Selinger. *Potrace: a polygon-based tracing algorithm*. Tech. rep. 2003. URL: <http://autotrace.sourceforge.net/>.
- [45] Chris G. Willcocks and Frederick W.B. Li. “Feature-varying skeletonization: Intuitive control over the target feature size and output skeleton topology”. In: *Visual Computer* 28.6-8 (June 2012), pp. 775–785. ISSN: 01782789. DOI: 10.1007/s00371-012-0688-x.
- [46] Natapon Pantuwong and Masanori Sugimoto. “A novel template-based automatic rigging algorithm for articulated- character animation”. In: *Computer Animation and Virtual Worlds* 23.2 (Mar. 2012), pp. 125–141. ISSN: 15464261. DOI: 10.1002/cav.1429.
- [47] D Eberly. “Triangulation by ear clipping. Geometric Tools, LLC”. In: (1998).

- [48] Mukta Prasad, Andrew Zisserman, and Andrew Fitzgibbon. “Single view reconstruction of curved surfaces”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 1345–1352. ISBN: 0769525970. DOI: 10.1109/CVPR.2006.281. URL: <http://www.robots.ox.ac.uk/%E2%88%BC%7B%5C%%7D7Bmukta,vgg%7B%5C%%7D7Dhttp://www.research.microsoft.com/%E2%88%BCawf>.
- [49] Yuefan Shen et al. “DeepSketchHair: Deep Sketch-based 3D Hair Modeling”. In: (Aug. 2019). arXiv: 1908.07198. URL: <http://arxiv.org/abs/1908.07198>.
- [50] Dmitriy Smirnov, Mikhail Bessmeltsev, and Justin Solomon. “Deep Sketch-Based Modeling of Man-Made Shapes”. In: (June 2019). arXiv: 1906.12337. URL: <http://arxiv.org/abs/1906.12337>.
- [51] Pedro Rossa, Daniel Camozzato, and Rafael Hocesvar. “3D Model Generation from Freehand Drawings”. In: (2016), pp. 226–229. URL: www.unity3d.com.
- [52] Omar Zenteno et al. “Volumetric monitoring of cutaneous leishmaniasis ulcers: can camera be as accurate as laser scanner?” In: *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization* (2018). ISSN: 21681171. DOI: 10.1080/21681163.2018.1546623.
- [53] Chiara Villa, Mitchell J. Flies, and Christina Jacobsen. “Forensic 3D documentation of bodies: Simple and fast procedure for combining CT scanning with external photogrammetry data”. In: *Journal of Forensic Radiology and Imaging* 12 (Mar. 2018), e2–e7. ISSN: 22124780. DOI: 10.1016/j.jofri.2017.11.003. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2212478017300916>.
- [54] Ahmet B. Orun, Eric Goodyer, and Geoff Smith. “3D non-invasive inspection of the skin lesions by closerange and low-cost photogrammetric techniques”. In: *Image Analysis and Stereology* (2018). ISSN: 18545165. DOI: 10.5566/ias.1730.
- [55] Mirwaes Wahabzada et al. “Monitoring wound healing in a 3D wound model by hyperspectral imaging and efficient clustering”. In: *PLoS ONE* (2017). ISSN: 19326203. DOI: 10.1371/journal.pone.0186425.
- [56] Syamantak Kumar et al. “A comparison of open source libraries ready for 3D reconstruction of wounds”. In: 2019, p. 9. DOI: 10.1117/12.2513411.
- [57] Alfredo Ballester et al. “Data-driven three-dimensional reconstruction of human bodies using a mobile phone app API-ecosystem for cross-sectorial exchange of 3D personal data (BODYPASS) View project Data-driven three-dimensional reconstruction of human bodies using a mobile phone ”. In: *J. Digital Human* 1.4 (2017), pp. 361–388. DOI: 10.1504/IJDH.2016.10005376. URL: <https://doi.org/10.1504/IJDH.2016.084581>.

- [58] Kenneth A. McLean et al. “Can a smartphone-delivered tool facilitate the assessment of surgical site infection and result in earlier treatment? Tracking wound infection with smartphone technology (TWIST): Protocol for a randomised controlled trial in emergency surgery patients”. In: *BMJ Open* 9.10 (Oct. 2019), e029620. ISSN: 20446055. DOI: 10.1136/bmjopen-2019-029620.
- [59] Jonathan S. Abelson et al. “Mobile health apps and recovery after surgery: What are patients willing to do?” In: *American Journal of Surgery* 214.4 (Oct. 2017), pp. 616–622. ISSN: 18791883. DOI: 10.1016/j.amjsurg.2017.06.009.
- [60] John De Heide et al. “Get the picture: A pilot feasibility study of telemedical wound assessment using a mobile phone in cardiology patients”. In: *Journal of Cardiovascular Nursing* 32.2 (2017), E9–E15. ISSN: 15505049. DOI: 10.1097/JCN.0000000000000377. URL: <http://www.ncbi.nlm.nih.gov/pubmed/27811584>.
- [61] P. C.B. Khong, M. S.W. Yeo, and C. C. Goh. “Evaluating an iPad app in measuring wound dimension: A pilot study”. In: *Journal of Wound Care* 26.12 (Dec. 2017), pp. 752–760. ISSN: 09690700. DOI: 10.12968/jowc.2017.26.12.752.
- [62] Young Joon Jun et al. “A Mobile Application for Wound Assessment and Treatment: Findings of a User Trial”. In: *International Journal of Lower Extremity Wounds* 15.4 (Dec. 2016), pp. 344–353. ISSN: 15526941. DOI: 10.1177/1534734616678522. URL: <http://www.ncbi.nlm.nih.gov/pubmed/27881691>.
- [63] T. Kanazawa et al. “Use of smartphone attached mobile thermography assessing subclinical inflammation: A pilot study”. In: *Journal of Wound Care* 25.4 (Apr. 2016), pp. 177–182. ISSN: 09690700. DOI: 10.12968/jowc.2016.25.4.177.
- [64] Joshua Budman et al. “Design of A Smartphone Application for Automated Wound Measurements for Home Care”. In: *Iproceedings* 1.1 (Oct. 2015), e16. ISSN: 2369-6893. DOI: 10.2196/iproc.4703.
- [65] Ravi K. Chittoria. “Telemedicine for wound management”. In: *Indian Journal of Plastic Surgery* 45.02 (May 2012), pp. 412–417. ISSN: 0970-0358. DOI: 10.4103/0970-0358.101330. URL: <http://www.thieme-connect.de/DOI/DOI?10.4103/0970-0358.101330>.
- [66] Pavel S. Kulikov, Prabjot K. Sandhu, and Karen A. Van Leuven. “Can a smartphone app help manage wounds in primary care?” In: *Journal of the American Association of Nurse Practitioners* 31.2 (Feb. 2019), pp. 110–115. ISSN: 2327-6924. DOI: 10.1097/JXX.000000000000100. URL: <http://journals.lww.com/01741002-201902000-00007>.
- [67] Ching Kan Lo et al. “A Mobile Nursing App Applying to the Wound Care and Drug Administration of Patients”. In: *Proceedings - 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2015*. Institute of Electrical and Electronics Engineers Inc., Sept. 2015, pp. 399–402. ISBN: 9781479988730. DOI: 10.1109/IMIS.2015.61.

- [68] Geraldo Magela Salome et al. *A Mobile App for Wound Cleansing OPEN ACCESS*. Tech. rep. 2019, p. 2662. URL: <http://clinicsinsurgery.com/>.
- [69] Ross Brown et al. “Myfootcare: A mobile self-tracking tool to promote self-care amongst people with diabetic foot ulcers”. In: *ACM International Conference Proceeding Series*. New York, New York, USA: Association for Computing Machinery, Nov. 2017, pp. 462–466. ISBN: 9781450353793. DOI: 10.1145/3152771.3156158. URL: <http://dl.acm.org/citation.cfm?doid=3152771.3156158>.
- [70] Molly L. Tolins et al. “Wound Care Follow-Up From the Emergency Department Using a Mobile Application: A Pilot Study”. In: *Journal of Emergency Medicine* 57.5 (Nov. 2019), pp. 629–636. ISSN: 07364679. DOI: 10.1016/j.jemermed.2019.07.017.
- [71] Hossein Nejati et al. “Smartphone and mobile image processing for assisted living: Health-monitoring apps powered by advanced mobile imaging algorithms”. In: *IEEE Signal Processing Magazine* 33.4 (July 2016), pp. 30–48. ISSN: 10535888. DOI: 10.1109/MSP.2016.2549996.
- [72] Rebecca Gunter et al. “Evaluating Patient Usability of an Image-Based Mobile Health Platform for Postoperative Wound Monitoring.” In: *JMIR mHealth and uHealth* 4.3 (Sept. 2016), e113. ISSN: 2291-5222. DOI: 10.2196/mhealth.6023. URL: <http://www.ncbi.nlm.nih.gov/pubmed/27683059> <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5062001>.
- [73] François André AllAeRt. *A smartphone application for clinical research on wound healing*. Apr. 2018. DOI: 10.23736/S1593-232X.18.00412-5.
- [74] Tik Wai Kiral Poon and Marcia R. Friesen. “Algorithms for size and color detection of smartphone images of chronic wounds for healthcare applications”. In: *IEEE Access* (2015). ISSN: 21693536. DOI: 10.1109/ACCESS.2015.2487859.
- [75] Paul J.F. White, Blake W. Podaima, and Marcia R. Friesen. “Algorithms for smartphone and tablet image analysis for healthcare applications”. In: *IEEE Access* 2 (2014), pp. 831–840. ISSN: 21693536. DOI: 10.1109/ACCESS.2014.2348943.
- [76] Chi Zhang et al. *MeshStereo: A Global Stereo Model with Mesh Alignment Regularization for View Interpolation*. Tech. rep. 2015, pp. 2057–2065.
- [77] Jui Tse Hsu et al. “Chronic wound assessment and infection detection method”. In: *BMC Medical Informatics and Decision Making* 19.1 (May 2019), p. 99. ISSN: 14726947. DOI: 10.1186/s12911-019-0813-0. URL: <https://bmcmmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-0813-0>.
- [78] Rashmi Mukherjee, Suman Tewary, and Aurobinda Routray. “Diagnostic and Prognostic Utility of Non-Invasive Multimodal Imaging in Chronic Wound Monitoring: a Systematic Review”. In: *Journal of Medical Systems* 41.3 (Mar. 2017), pp. 1–17. ISSN: 1573689X. DOI: 10.1007/s10916-016-0679-y.

- [79] Yunghan Au et al. "Time-Saving Comparison of Wound Measurement Between the Ruler Method and the Swift Skin and Wound App". In: *Journal of Cutaneous Medicine and Surgery* 23.2 (Mar. 2019), pp. 226–228. ISSN: 12034754. DOI: 10.1177/1203475418800942. URL: <http://journals.sagepub.com/doi/10.1177/1203475418800942>.
- [80] Abhiraj Gupta. "Real time wound segmentation/management using image processing on handheld devices". In: *Journal of Computational Methods in Sciences and Engineering*. Vol. 17. 2. IOS Press, Jan. 2017, pp. 321–329. DOI: 10.3233/JCM-170706.
- [81] Adam Yee, John Harmon, and Steven Yi. "Quantitative Monitoring Wound Healing Status Through Three-dimensional Imaging on Mobile Platforms". In: *Journal of the American College of Clinical Wound Specialists* 8.1-3 (Jan. 2016), pp. 21–27. ISSN: 22135103. DOI: 10.1016/j.jccw.2017.11.001.
- [82] Adam Yee et al. "IDr: An Intelligent Digital Ruler App for Remote Wound Assessment". In: *Proceedings - 2016 IEEE 1st International Conference on Connected Health: Applications, Systems and Engineering Technologies, CHASE 2016*. Institute of Electrical and Electronics Engineers Inc., Aug. 2016, pp. 380–381. ISBN: 9781509009435. DOI: 10.1109/CHASE.2016.78.
- [83] Chunhui Liu et al. "Wound area measurement with 3D transformation and smartphone images". In: *BMC Bioinformatics* 20.1 (Dec. 2019), pp. 1–21. ISSN: 14712105. DOI: 10.1186/s12859-019-3308-1.
- [84] Dongkeun Jun et al. "Efficacy of the Mobile Three-Dimensional Wound Measurement System in Pressure Ulcer Assessment". In: *Journal of Wound Management and Research* 15.2 (Sept. 2019), pp. 78–84. ISSN: 2586-0402. DOI: 10.22467/jwmr.2019.00829.
- [85] Xiachuan Pei et al. "Flexible wireless skin impedance sensing system for wound healing assessment". In: *Vacuum* 168 (Oct. 2019), p. 108808. ISSN: 0042207X. DOI: 10.1016/j.vacuum.2019.108808.
- [86] Minta Lu et al. "Enhance wound healing monitoring through a thermal imaging based smartphone app". In: *Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications*. Ed. by Jianguo Zhang and Po-Hao Chen. Vol. 10579. SPIE, Mar. 2018, p. 60. ISBN: 9781510616479. DOI: 10.1117/12.2293674. URL: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10579/2293674/Enhance-wound-healing-monitoring-through-a-thermal-imaging-based-smartphone/10.1117/12.2293674.full>.
- [87] Kacie Kaile et al. "Low-cost smartphone based imaging device to detect subsurface tissue oxygenation of wounds". In: *Optics and Biophotonics in Low-Resource Settings V*. Ed. by David Levitz and Aydogan Ozcan. Vol. 10869. SPIE, Feb. 2019, p. 37. ISBN: 9781510623804. DOI: 10.1117/12.2510425. URL: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10869/2510425/>

Low-cost-smartphone-based-imaging-device-to-detect-subsurface-tissue/
10.1117/12.2510425.full.

- [88] Neal Sikka et al. “The Use of Mobile Phone Cameras in Guiding Treatment Decisions for Laceration Care”. In: *Telemedicine and e-Health* (2012). ISSN: 1530-5627. DOI: 10.1089/tmj.2011.0216.
- [89] Nick Santamaria and Suzanne Kapp. “Smart Homecare Technology and TeleHealth Dovepress Telewound care-providing remote wound assessment and treatment in the home care setting: current status and future directions”. In: *Smart Homecare Technology and TeleHealth* (2013), pp. 1–35. DOI: 10.2147/SHTT.S34353. URL: <http://dx.doi.org/10.2147/SHTT.S34353>.
- [90] Caroline Chanussot-Deprez and José Contreras-Ruiz. “Telemedicine in wound care”. In: *International Wound Journal* 5.5 (Dec. 2008), pp. 651–654. ISSN: 17424801. DOI: 10.1111/j.1742-481X.2008.00478.x. URL: <http://doi.wiley.com/10.1111/j.1742-481X.2008.00478.x>.
- [91] Ralph Peter Braun et al. “Telemedical wound care using a new generation of mobile telephones: A feasibility study”. In: *Archives of Dermatology* 141.2 (Feb. 2005), pp. 254–258. ISSN: 0003987X. DOI: 10.1001/archderm.141.2.254.
- [92] Hui Hong Tsai et al. “Teleconsultation by using the mobile camera phone for remote management of the extremity wound: A pilot study”. In: *Annals of Plastic Surgery* (2004). ISSN: 01487043. DOI: 10.1097/01.sap.0000130703.45332.3c.
- [93] Lee A. Wallis et al. “A smartphone app and cloud-based consultation system for burn injury emergency care”. In: *PLoS ONE* 11.2 (Feb. 2016). ISSN: 19326203. DOI: 10.1371/journal.pone.0147253.
- [94] Vidyashree Dalya and D K Shedge. *Design of Smartphone-based Wound Assessment System*. Tech. rep. 1. 2016, pp. 975–8887.
- [95] Adam Landman et al. “A mobile app for securely capturing and transferring clinical images to the electronic health record: description and preliminary usability study.” In: *JMIR mHealth and uHealth* 3.1 (Jan. 2015), e1. ISSN: 2291-5222. DOI: 10.2196/mhealth.3481. URL: <http://www.ncbi.nlm.nih.gov/pubmed/25565678><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4296096>.
- [96] Junna Ye et al. “A telemedicine wound care model using 4G with smart phones or smart glasses: A pilot study”. In: *Medicine (United States)* 95.31 (Aug. 2016). ISSN: 15365964. DOI: 10.1097/MD.00000000000004198.
- [97] Yutong Wang et al. “Sketch-based shape-preserving tree animations”. In: *Computer Animation and Virtual Worlds* 29.3-4 (May 2018), e1821. ISSN: 15464261. DOI: 10.1002/cav.1821. URL: <http://doi.wiley.com/10.1002/cav.1821>.

- [98] Geraldo Magela Salomé and Lydia Masako Ferreira. “Developing a Mobile App for Prevention and Treatment of Pressure Injuries”. In: *Advances in Skin & Wound Care* 31.2 (Feb. 2018), pp. 1–6. ISSN: 1527-7941. DOI: 10.1097/01.ASW.0000529693.60680.5e. URL: <http://journals.lww.com/00129334-201802000-00013>.
- [99] Rebecca L. Gunter et al. “Feasibility of an Image-Based Mobile Health Protocol for Postoperative Wound Monitoring”. In: *Journal of the American College of Surgeons* 226.3 (Mar. 2018), pp. 277–286. ISSN: 18791190. DOI: 10.1016/j.jamcollsurg.2017.12.013.
- [100] Brenda King. “Influencing dressing choice and supporting wound management using remote ‘tele-wound care’”. In: *British Journal of Community Nursing* 19.SUPPL. 6 (June 2014). ISSN: 14624753. DOI: 10.12968/bjcn.2014.19.Sup6.S24.
- [101] Norhan Shamloul, Mondana H Ghias, and Amor Khachemoune. “The Utility of Smartphone Applications and Technology in Wound Healing.” In: *The international journal of lower extremity wounds* 18.3 (Sept. 2019), pp. 228–235. ISSN: 1552-6941. DOI: 10.1177/1534734619853916. URL: <http://www.ncbi.nlm.nih.gov/pubmed/31198071>.
- [102] Thomas M. Deserno, Sanmathi Kamath, and Ekatherina Sirazitdinova. “Machine learning for mobile wound assessment”. In: *Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications*. Ed. by Jianguo Zhang and Po-Hao Chen. Vol. 10579. SPIE, Mar. 2018, p. 42. ISBN: 9781510616479. DOI: 10.1117/12.2293704. URL: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10579/2293704/Machine-learning-for-mobile-wound-assessment/10.1117/12.2293704.full>.
- [103] Ivan Miguel Pires and Nuno M Garcia. “Wound Area Assessment using Mobile Application”. In: (). DOI: 10.5220/0005236502710282.
- [104] Kyle Wu et al. *Mobile Wound Assessment using Novel Computer Vision Methods*. Tech. rep. 2014. DOI: 10.1016/j.jamcollsurg.2014.07.150.
- [105] Bianca Schnalzer and Baptiste Alcalde. “Skin lesion boundary detection with neural networks on iOS devices”. In: *ACM International Conference Proceeding Series*. New York, New York, USA: Association for Computing Machinery, May 2019, pp. 161–166. ISBN: 9781450371995. DOI: 10.1145/3340037.3340057. URL: <http://dl.acm.org/citation.cfm?doid=3340037.3340057>.
- [106] Manu Goyal et al. “Robust Methods for Real-Time Diabetic Foot Ulcer Detection and Localization on Mobile Devices”. In: *IEEE Journal of Biomedical and Health Informatics* 23.4 (July 2019), pp. 1730–1741. ISSN: 21682208. DOI: 10.1109/JBHI.2018.2868656.

- [107] Cheng Hsien Huang et al. “Automatic Size Measurement and Boundary Tracing of Wound on a Mobile Device”. In: *2018 IEEE International Conference on Consumer Electronics-Taiwan, ICCE-TW 2018*. Institute of Electrical and Electronics Engineers Inc., Aug. 2018. ISBN: 9781538663011. DOI: 10.1109/ICCE-China.2018.8448729.
- [108] Khajista Nizam et al. “Enhancement in the Identification of Slough Tissue in Chronic Wound Assessment”. In: Institute of Electrical and Electronics Engineers (IEEE), Feb. 2020, pp. 154–158. DOI: 10.1109/icsipa45851.2019.8977785.
- [109] Marco D. Cirillo et al. “Tensor Decomposition for Colour Image Segmentation of Burn Wounds”. In: *Scientific Reports* 9.1 (Dec. 2019), pp. 1–13. ISSN: 20452322. DOI: 10.1038/s41598-019-39782-2.
- [110] Samsunnisha Patel, Rachna Patel, and Dhara Desai. “Diabetic foot ulcer wound tissue detection and classification”. In: *Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2017*. Vol. 2018-Janua. Institute of Electrical and Electronics Engineers Inc., Jan. 2018, pp. 1–5. ISBN: 9781509032945. DOI: 10.1109/ICIIECS.2017.8276007.
- [111] Tallha Akram et al. “Skin lesion segmentation and recognition using multichannel saliency estimation and M-SVM on selected serially fused features”. In: *Journal of Ambient Intelligence and Humanized Computing* (Sept. 2018), pp. 1–20. ISSN: 18685145. DOI: 10.1007/s12652-018-1051-5.
- [112] S S Suganthi and S Ramakrishnan. “Analysis of Chronic Wound Images Using Factorization Based Segmentation and Machine Learning Methods”. In: (2017). DOI: 10.1145/3155077.3155092. URL: <https://doi.org/10.1145/3155077.3155092>.
- [113] Dhiraj Manohar Dhane et al. “Fuzzy spectral clustering for automated delineation of chronic wound region using digital images”. In: *Computers in Biology and Medicine* 89 (Oct. 2017), pp. 551–560. ISSN: 18790534. DOI: 10.1016/j.compbiomed.2017.04.004.
- [114] Ali Haider et al. “Computer-assisted image processing technique for tracking wound progress”. In: *IEEE International Conference on Electro Information Technology*. Vol. 2016-Augus. IEEE Computer Society, Aug. 2016, pp. 750–754. ISBN: 9781467399852. DOI: 10.1109/EIT.2016.7535333.
- [115] Zhao Zhang, William V. Stoecker, and Randy H. Moss. “Border detection on digitized skin tumor images”. In: *IEEE Transactions on Medical Imaging* 19.11 (2000), pp. 1128–1143. ISSN: 02780062. DOI: 10.1109/42.896789.
- [116] Ashish Bhelonde et al. “Flexible wound assessment system for diabetic patient using android smartphone”. In: *Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015*. Institute of Electrical and Electronics Engineers Inc., Jan. 2016, pp. 466–469. ISBN: 9781467379090. DOI: 10.1109/ICGCIoT.2015.7380509.

- [117] Chaghan Wang et al. “A unified framework for automatic wound segmentation and analysis with deep convolutional neural networks”. In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2015-Novem* (2015), pp. 2415–2418. ISSN: 1557170X. DOI: 10.1109/EMBC.2015.7318881.
- [118] Hazem Wannous, Yves Lucas, and Sylvie Treuillet. “Enhanced assessment of the wound-healing process by accurate multiview tissue classification”. In: *IEEE Transactions on Medical Imaging* 30.2 (2011), pp. 315–326. ISSN: 02780062. DOI: 10.1109/TMI.2010.2077739. URL: <http://www.nigms.nih.gov>.
- [119] Hazem Wannous. “Robust tissue classification for reproducible wound assessment in telemedicine environments”. In: *Journal of Electronic Imaging* 19.2 (2010), p. 023002. ISSN: 1017-9909. DOI: 10.1117/1.3378149. URL: <https://hal.archives-ouvertes.fr/hal-00648504>.
- [120] Wellington S. Silva et al. “A two-phase learning approach for the segmentation of dermatological wounds”. In: *Proceedings - IEEE Symposium on Computer-Based Medical Systems*. Vol. 2019-June. Institute of Electrical and Electronics Engineers Inc., June 2019, pp. 343–348. ISBN: 9781728122861. DOI: 10.1109/CBMS.2019.00076.
- [121] Emmanuel Agu et al. “The smartphone as a medical device: Assessing enablers, benefits and challenges”. In: *2013 IEEE International Workshop of Internet-of-Things Networking and Control, IoT-NC 2013*. IEEE Computer Society, 2013, pp. 48–52. ISBN: 9781479908264. DOI: 10.1109/IoT-NC.2013.6694053.
- [122] Ahmad Fadzil Mohd Hani et al. “Lecture Notes in Computer Science Detection and Classification of Granulation Tissue in Chronic Ulcers”. In: *LNCS 7066* (2011), pp. 139–150. DOI: 10.1007/978-3-642-25191-7_14. URL: <https://www.researchgate.net/publication/221365159>.
- [123] Ahmad Fadzil M. Hani et al. “Assessment of chronic ulcers using digital imaging”. In: *2011 National Postgraduate Conference - Energy and Sustainability: Exploring the Innovative Minds, NPC 2011*. 2011. ISBN: 9781457718847. DOI: 10.1109/NatPC.2011.6136334.
- [124] Hannah Inbarani H., Ahmad Taher Azar, and Jothi G. “Leukemia Image Segmentation Using a Hybrid Histogram-Based Soft Covering Rough K-Means Clustering Algorithm”. In: *Electronics* 9.1 (Jan. 2020), p. 188. ISSN: 2079-9292. DOI: 10.3390/electronics9010188. URL: <https://www.mdpi.com/2079-9292/9/1/188>.
- [125] Ihor Farmaha et al. “Wound image segmentation using clustering based algorithms”. In: *New Trends in Production Engineering* 2.1 (Nov. 2019), pp. 570–578. DOI: 10.2478/ntpe-2019-0062.

- [126] Lian Luo et al. "Multi-Feature Clustering Approach for Firearm Wound Identification on CT Images". In: *Proceedings of 2019 IEEE International Conference on Mechatronics and Automation, ICMA 2019*. Institute of Electrical and Electronics Engineers Inc., Aug. 2019, pp. 1594–1599. ISBN: 9781728116983. DOI: 10.1109/ICMA.2019.8816493.
- [127] Chinmay Chakraborty, Bharat Gupta, and Soumya K. Ghosh. "Identification of Chronic Wound Status under Tele-Wound Network through Smartphone". In: *International Journal of Rough Sets and Data Analysis* 2.2 (July 2015), pp. 58–77. ISSN: 2334-4598. DOI: 10.4018/ijrsda.2015070104.
- [128] Chinmay Chakraborty. "Computational approach for chronic wound tissue characterization". In: *Informatics in Medicine Unlocked* 17 (Jan. 2019), p. 100162. ISSN: 23529148. DOI: 10.1016/j.imu.2019.100162.
- [129] K.S. Babu, Asit Subudhi, and Sukanta Sabut. "Segmentation of Diabetic Wound by Multidimensional Clustering for Quantitative Assessment of Healing Process". In: *Current Medical Imaging Reviews* 14.1 (Mar. 2018), pp. 71–76. ISSN: 15734056. DOI: 10.2174/1573405613666161209104534.
- [130] Thaina Aparecida Azevedo Tosta et al. "Application of Evolutionary Algorithms on Unsupervised Segmentation of Lymphoma Histological Images". In: *Proceedings - IEEE Symposium on Computer-Based Medical Systems*. Vol. 2017-June. Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 89–94. ISBN: 9781538617106. DOI: 10.1109/CBMS.2017.69.
- [131] Chinmay Chakraborty, Bharat Gupta, and Soumya K Ghosh. "Chronic wound characterization using bayesian classifier under telemedicine framework". In: *Medical Imaging: Concepts, Methodologies, Tools, and Applications*. Vol. 7. 1. 2016, pp. 741–751. ISBN: 9781522505723. DOI: 10.4018/978-1-5225-0571-6.ch030. URL: <https://www.researchgate.net/publication/316034296>.
- [132] Chinmay Chakraborty et al. "Telemedicine Supported Chronic Wound Tissue Prediction Using Classification Approaches". In: *Journal of Medical Systems* 40.3 (2016), pp. 1–12. ISSN: 1573689X. DOI: 10.1007/s10916-015-0424-y. URL: <https://www.springer.com/series/11970>.
- [133] Sirisha Konakala et al. *A Novel Threshold Technique and Fuzzy C-Means Algorithm for Segmentation of Wound*. Tech. rep. 7. 2013, pp. 975–8887. URL: www.ijcaonline.org.
- [134] Chandan Chakraborty et al. "Segmentation of Chronic Wound Areas by Clustering Techniques Using Selected Color Space Chronic Wound Management View project Segmentation of Chronic Wound Areas by Clustering Techniques Using Selected Color Space". In: *Article in Journal of Medical Imaging and Health Informatics* 3 (2013), pp. 22–29. DOI: 10.1166/jmihi.2013.1124. URL: http://www.medetec.co.uk/files/medetec%7B%5C_%7Dimage%7B%5C_%7Ddatabases.html.

- [135] Chandan Chakraborty et al. “Assessment of Segmentation Techniques for Chronic Wound Surface Area Detection”. In: (2018). DOI: 10.1007/978-981-10-8237-5_68. URL: https://doi.org/10.1007/978-981-10-8237-5_68.
- [136] Begonya Garcia-Zapirain et al. “Automated framework for accurate segmentation of pressure ulcer images”. In: *Computers in Biology and Medicine* 90 (Nov. 2017), pp. 137–145. ISSN: 18790534. DOI: 10.1016/j.compbiomed.2017.09.015.
- [137] Gabriel F.C. Campos, Sylvio Barbon, and Rafael G. Mantovani. “A meta-learning approach for recommendation of image segmentation algorithms”. In: *Proceedings - 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2016*. Institute of Electrical and Electronics Engineers Inc., Jan. 2017, pp. 370–377. ISBN: 9781509035687. DOI: 10.1109/SIBGRAPI.2016.058.
- [138] Hazem Wannous, Yves Lucas, and Sylvie Treuillet. *Combined Machine Learning with Multi-view Modeling for Robust Wound Tissue Assessment Combined Machine Learning with Multi-view Modeling for Robust COMBINED MACHINE LEARNING WITH MULTI-VIEW MODELING FOR ROBUST WOUND TISSUE ASSESSMENT*. Tech. rep. May 2010, pp. 92–104. URL: <https://hal.archives-ouvertes.fr/hal-00648511>.
- [139] Dhiraj Manohar Dhane et al. “Spectral Clustering for Unsupervised Segmentation of Lower Extremity Wound Beds Using Optical Images”. In: *Journal of Medical Systems* 40.9 (Sept. 2016), pp. 1–10. ISSN: 1573689X. DOI: 10.1007/s10916-016-0554-x.
- [140] Lukasz A. Paluchowski et al. “Can spectral-spatial image segmentation be used to discriminate experimental burn wounds?” In: *Journal of Biomedical Optics* 21.10 (May 2016), p. 101413. ISSN: 1083-3668. DOI: 10.1117/1.jbo.21.10.101413.
- [141] K Sundeep Kumar, C F Jacob, and B Eswara Reddy. “UNSUPERVISED WOUND IMAGE SEGMENTATION”. In: *ICTACT JOURNAL ON IMAGE AND VIDEO PROCESSING* (2014), p. 3. DOI: 10.21917/ijivp.2014.0107.
- [142] Urban Pavlovčič et al. “Wound perimeter, area, and volume measurement based on laser 3D and color acquisition”. In: *BioMedical Engineering Online* 14.1 (Apr. 2015), pp. 1–15. ISSN: 1475925X. DOI: 10.1186/s12938-015-0031-7.
- [143] Ananya Barui et al. “Precise Segmentation and Classification of Epithelial Rete-Pegs Signature in Assessing Lower Limb Wound Healing Progression”. In: *Journal of Medical and Biological Engineering* (2018). DOI: 10.1007/s40846-018-0442-6.
- [144] Dorra Nouri et al. “Colour and multispectral imaging for wound healing evaluation in the context of a comparative preclinical study”. In: *Imaging* 866923 (2013). DOI: 10.1117/12.2003943. URL: <https://hal.archives-ouvertes.fr/hal-00837744>.
- [145] Te-Wei Ho et al. “Evaluation of Surgical Wound Segmentation using Quantitative Analysis”. In: (2017). DOI: 10.1145/3162957.3162970. URL: <https://doi.org/10.1145/3162957.3162970>.

- [146] Hsueh-Fu Shih et al. "Surgical wound segmentation based on adaptive threshold edge detection and genetic algorithm". In: *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*. Ed. by Yulin Wang et al. Vol. 10225. SPIE, Feb. 2017, p. 1022517. DOI: 10.1117/12.2266105. URL: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2266105>.
- [147] Xingyu Li and Konstantinos N Plataniotis. "Computational scratch assay - A new frontier for image analysis: Preliminary study of multi-cellular segmentation". In: *2017 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2017 - Proceedings*. Vol. 2018-Janua. 2018, pp. 186–190. ISBN: 9781509059904. DOI: 10.1109/GlobalSIP.2017.8308629.
- [148] A Cardona et al. "Automatic image segmentation method for in vitro wound healing assay quantitative analysis". In: *IFMBE Proceedings*. Vol. 49. 2015, pp. 381–384. ISBN: 9783319131160. DOI: 10.1007/978-3-319-13117-7_98. URL: <https://www.researchgate.net/publication/273574138>.
- [149] Begoña García-Zapirain et al. "Classification of pressure ulcer tissues with 3D convolutional neural network". In: *Medical and Biological Engineering and Computing* (2018). ISSN: 17410444. DOI: 10.1007/s11517-018-1835-y.
- [150] Kai Hu et al. "A Skin Lesion Segmentation Method Based on Saliency and Adaptive Thresholding in Wavelet Domain". In: *Studies in Computational Intelligence*. Vol. 810. Springer Verlag, Nov. 2020, pp. 445–453. DOI: 10.1007/978-3-030-04946-1_43.
- [151] Saleh Baghersalimi et al. "DermoNet: densely linked convolutional neural network for efficient skin lesion segmentation". In: *EURASIP Journal on Image and Video Processing* 2019.1 (Dec. 2019), p. 71. ISSN: 1687-5281. DOI: 10.1186/s13640-019-0467-y. URL: <https://jivp-urasipjournals.springeropen.com/articles/10.1186/s13640-019-0467-y>.
- [152] Philipp Tschandl, Christoph Sinz, and Harald Kittler. "Domain-specific classification-pretrained fully convolutional network encoders for skin lesion segmentation". In: *Computers in Biology and Medicine* (2019). ISSN: 18790534. DOI: 10.1016/j.combiomed.2018.11.010.
- [153] Sofia Zahia et al. "Tissue classification and segmentation of pressure injuries using convolutional neural networks". In: *Computer Methods and Programs in Biomedicine* 159 (June 2018), pp. 51–58. ISSN: 0169-2607. DOI: 10.1016/J.CMPB.2018.02.018. URL: <https://www.sciencedirect.com/science/article/pii/S0169260717314864?via%7B%5C%7D3Dihub>.
- [154] Mohammed Elmogy et al. "Tissues Classification for Pressure Ulcer Images Based on 3D Convolutional Neural Network". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, Oct. 2018, pp. 3139–3143. ISBN: 978-1-4799-7061-2. DOI: 10.1109/ICIP.2018.8451119. URL: <https://ieeexplore.ieee.org/document/8451119/>.

- [155] Sulaiman Vesal, Nishant Ravikumar, and Andreas Maier. “SkinNet: A Deep Learning Framework for Skin Lesion Segmentation”. In: (2018). arXiv: 1806.09522.
- [156] Md. Mostafa Kamal Sarker et al. “SLSDep: Skin Lesion Segmentation Based on Dilated Residual and Pyramid Pooling Networks”. In: (May 2018). arXiv: 1805.10241. URL: <http://arxiv.org/abs/1805.10241>.
- [157] Md Zahangir Alom et al. “Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation”. In: (2018). arXiv: 1802.06955.
- [158] Jin Qi et al. “Global and Local Information Based Deep Network for Skin Lesion Segmentation”. In: (2017). arXiv: 1703.05467.
- [159] Manu Goyal et al. “DFUNet: Convolutional Neural Networks for Diabetic Foot Ulcer Classification”. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* (2018), pp. 1–12. ISSN: 2471-285X. DOI: 10.1109/tetci.2018.2866254. arXiv: 1711.10448. URL: <https://ieeexplore.ieee.org/abstract/document/8464076/>.
- [160] D. P. Yadav et al. “Feature extraction based machine learning for human burn diagnosis from burn images”. In: *IEEE Journal of Translational Engineering in Health and Medicine* 7 (2019). ISSN: 21682372. DOI: 10.1109/JTEHM.2019.2923628.
- [161] C Aguirre Nilsson and M Velic. “Classification of ulcer images using convolutional neural networks”. In: (2018). URL: <https://odr.chalmers.se/handle/20.500.12380/255746>.
- [162] Aliyu Abubakar and Hassan Ugail. “Discrimination of Human Skin Burns Using Machine Learning”. In: *Advances in Intelligent Systems and Computing*. Vol. 997. Springer Verlag, 2019, pp. 641–647. ISBN: 9783030228705. DOI: 10.1007/978-3-030-22871-2_43.
- [163] Laith Alzubaidi et al. “DFU_QUTNet: diabetic foot ulcer classification using novel deep convolutional neural network”. In: *Multimedia Tools and Applications* (2019). ISSN: 15737721. DOI: 10.1007/s11042-019-07820-w.
- [164] Iván González-Díaz. “DermaKNet: Incorporating the Knowledge of Dermatologists to Convolutional Neural Networks for Skin Lesion Diagnosis”. In: *IEEE Journal of Biomedical and Health Informatics* 23.2 (2019), pp. 547–559. ISSN: 21682194. DOI: 10.1109/JBHI.2018.2806962.
- [165] Jack Burdick et al. “Rethinking Skin Lesion Segmentation in a Convolutional Classifier”. In: *Journal of Digital Imaging* (2018). ISSN: 1618727X. DOI: 10.1007/s10278-017-0026-y.
- [166] Bernhard Preim and Patrick Saalfeld. “A survey of virtual human anatomy education systems”. In: *Computers and Graphics (Pergamon)* 71 (Apr. 2018), pp. 132–153. ISSN: 00978493. DOI: 10.1016/j.cag.2018.01.005.

- [167] Stephanie Sandor and Richard Leahy. “Surface-Based Labeling of Cortical Anatomy Using a Deformable Atlas”. In: *IEEE Transactions on Medical Imaging* 16.1 (1997), pp. 41–54. ISSN: 02780062. DOI: 10.1109/42.552054.
- [168] *US Navy Releases New 3-D Medical Study Aid App*. https://www.navy.mil/submit/display.asp?story%7B%5C_%7Ddid=81642. URL: https://www.navy.mil/submit/display.asp?story%7B%5C_%7Ddid=81642 (visited on 02/05/2020).
- [169] *Home*. <http://anatomylearning.com/en/>. URL: <http://anatomylearning.com/en/> (visited on 02/05/2020).
- [170] *BioDigital Human - 3D Anatomy - Apps on Google Play*. <https://play.google.com/store/apps/details?id=com.biodigitalhuman.humanAndroid>. URL: <https://play.google.com/store/apps/details?id=com.biodigitalhuman.humanAndroid> (visited on 02/05/2020).
- [171] Xinjian Chen and Ulas Bagci. “3D automatic anatomy segmentation based on iterative graph-cut-ASM”. In: *Medical Physics* 38.8 (July 2011), pp. 4610–4622. ISSN: 00942405. DOI: 10.1118/1.3602070. URL: <http://doi.wiley.com/10.1118/1.3602070>.
- [172] Lawrence M. Witmer et al. “Using CT to peer into the past: 3D visualization of the brain and ear regions of birds, crocodiles, and nonavian dinosaurs”. In: *Anatomical Imaging: Towards a New Morphology*. Springer Verlag, 2008, pp. 67–87. ISBN: 9784431769323. DOI: 10.1007/978-4-431-76933-0_6.
- [173] *Human Skeleton 3D pose — CGTrader*. <https://www.cgtrader.com/3d-models/character/anatomy/human-skeleton-0213061d-5512-4d73-bb78-0c41c107bf74>. URL: <https://www.cgtrader.com/3d-models/character/anatomy/human-skeleton-0213061d-5512-4d73-bb78-0c41c107bf74> (visited on 02/05/2020).
- [174] *Skeletal Anatomy 3D – (Android Apps) — AppAgg*. <https://appagg.com/android/medical/skeletal-anatomy-3d-31915732.html?hl=en>. URL: <https://appagg.com/android/medical/skeletal-anatomy-3d-31915732.html?hl=en> (visited on 02/05/2020).
- [175] *Free Interactive 3D Human Anatomy Software - Instant Fundas*. <https://www.instantfundas.com/2011/08/free-interactive-3d-human-anatomy.html>. URL: <https://www.instantfundas.com/2011/08/free-interactive-3d-human-anatomy.html> (visited on 02/05/2020).
- [176] Jan-Martin Kuhnigk et al. “Lung lobe segmentation by anatomy-guided 3D watershed transform”. In: *Medical Imaging 2003: Image Processing*. Vol. 5032. SPIE, May 2003, p. 1482. DOI: 10.1117/12.480321.
- [177] Claire Chalopin, Gérard Finet, and Isabelle E. Magnin. “Modeling the 3D coronary tree for labeling purposes”. In: *Medical Image Analysis* 5.4 (Dec. 2001), pp. 301–315. ISSN: 13618415. DOI: 10.1016/S1361-8415(01)00047-0.

- [178] *Muscle System Pro*. <https://3d4medical.com/apps/muscle-system-pro>. URL: <https://3d4medical.com/apps/muscle-system-pro> (visited on 02/05/2020).
- [179] *Complete Anatomy Platform 2020 - Apps on Google Play*. [\&hl=en{_}US](https://play.google.com/store/apps/details?id=com.a3d4medical.completeanatomy). URL: https://play.google.com/store/apps/details?id=com.a3d4medical.completeanatomy%7B%5C%7Dhl=en%7B%5C_%7DUS (visited on 02/05/2020).
- [180] *3D Bones and Organs (Anatomy) - Apps on Google Play*. <https://play.google.com/store/apps/details?id=com.biodigitalhuman.humanAndroid>. URL: <https://play.google.com/store/apps/details?id=com.hssn.anatomy3dlite> (visited on 02/05/2020).
- [181] *Essential Anatomy 3 - 3D4Medical*. <https://3d4medical.com/apps/essential-anatomy-3>. URL: <https://3d4medical.com/apps/essential-anatomy-3> (visited on 02/05/2020).
- [182] M Palmer et al. “Web-Based Human Surface Anatomy Labeling System for Dermatologic Health Care Environments”. In: ().
- [183] *Human body (male) educational VR 3D - Apps on Google Play*. https://play.google.com/store/apps/details?id=com.rendernet.humanmale{\&hl=en{_}US. URL: https://play.google.com/store/apps/details?id=com.rendernet.humanmale%7B%5C%7Dhl=en%7B%5C_%7DUS (visited on 02/05/2020).
- [184] *3D human skin structure - TurboSquid 1222133*. <https://www.turbosquid.com/3d-models/3d-human-skin-structure-1222133>. URL: <https://www.turbosquid.com/3d-models/3d-human-skin-structure-1222133> (visited on 02/06/2020).
- [185] Liberty S. Hamilton et al. “Semi-automated anatomical labeling and inter-subject warping of high-density intracranial recording electrodes in electrocorticography”. In: *Frontiers in Neuroinformatics* 11 (Oct. 2017). ISSN: 16625196. DOI: 10.3389/fninf.2017.00062.
- [186] J. L. Lancaster et al. “Automated labeling of the human brain: A preliminary report on the development and evaluation of a forward-transform method”. In: *Human Brain Mapping*. Vol. 5. 4. 1997, pp. 238–242. DOI: 10.1002/(SICI)1097-0193(1997)5:4<238::AID-HBM6>3.0.CO;2-4. URL: [https://onlinelibrary.wiley.com/doi/abs/10.1002/\(SICI\)1097-0193\(1997\)5:4%7B%5C%7D3C238::AID-HBM6%7B%5C%7D3E3.0.CO;2-4](https://onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-0193(1997)5:4%7B%5C%7D3C238::AID-HBM6%7B%5C%7D3E3.0.CO;2-4).
- [187] Kensaku Mori et al. “Automated anatomical labeling of the bronchial branch and its application to the virtual bronchoscopy system”. In: *IEEE Transactions on Medical Imaging* 19.2 (2000), pp. 103–114. ISSN: 02780062. DOI: 10.1109/42.836370. URL: <https://ieeexplore.ieee.org/abstract/document/836370/>.

- [188] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. “Learning 3D mesh segmentation and labeling”. In: *ACM SIGGRAPH 2010 Papers, SIGGRAPH 2010*. 2010. ISBN: 9781450302104. DOI: 10.1145/1778765.1778839. URL: <http://www.dgp.toronto.edu/%7B~%7Dkalo/papers/LabelMeshes/>.
- [189] Kamran Ali, Knut Hartmann, and Thomas Strothotte. “Label layout for interactive 3D illustrations”. In: (2005). ISSN: 1213–6964 (online).
- [190] Timo Götzelmann et al. “Form Follows Function: Aesthetic Interactive Labels”. In: *Proceedings of the First Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging 2005 (CAe 2005, May 18–20, 2005, Girona, Spain)*. W. Purgathofer, 2005, pp. 193–200. DOI: <http://dx.doi.org/10.2312/COMPAESTH/COMPAESTH05/193-200>. URL: <http://www.google.com/url?sa=t%7B%5C%7Dsource=web%7B%5C%7Dct=res%7B%5C%7Dcd=1%7B%5C%7Dved=0CAoQFjAA%7B%5C%7Durl=http://www.medvis-book.de/MedVisBookMaterial/Chapter20/ca05.pdf%7B%5C%7Ddei=uMNZS%7B%5C%7D%7B%5C%7Dc0Ynj-Qaehr0L%7B%5C%7Dusg=AFQjCNFhTMblhK3VFt7hi1am-lyNpEu-Eg%7B%5C%7Dsig2=%7B%5C%7DjxJ7ts1EVvAcoI17rEOEA>.
- [191] Xianwang Wang et al. “Endoscopic video texture mapping on pre-built 3-D anatomical objects without camera tracking”. In: *IEEE Transactions on Medical Imaging* 29.6 (June 2010), pp. 1213–1223. ISSN: 02780062. DOI: 10.1109/TMI.2009.2028341.
- [192] *Human Anatomy UV Coordinates Setup for Texturing · 3dtotal · Learn — Create — Share*. <https://3dtotal.com/tutorials/t/human-anatomy-uv-coordinates-setup-for-texturing-dylan-swan-character>. URL: <https://3dtotal.com/tutorials/t/human-anatomy-uv-coordinates-setup-for-texturing-dylan-swan-character> (visited on 03/02/2020).
- [193] Taehyun Rhee, Ulrich Neumann, and J P Lewis. “Human hand modeling from surface anatomy”. In: *Proceedings of the Symposium on Interactive 3D Graphics*. Vol. 2006. 2006, pp. 27–34. ISBN: 159593295X. DOI: 10.1145/1111411.1111417.
- [194] Thierry Molinier, David Fofi, and Patrick Gorria. “Novel view synthesis for projective texture mapping on real 3D objects”. In: *Machine Vision Applications in Industrial Inspection XIV*. Vol. 6070. 2006, 60700A. ISBN: 0819461105. DOI: 10.1117/12.642618. URL: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/6070/60700A/Novel-view-synthesis-for-projective-texture-mapping-on-real-3D/10.1117/12.642618.short>.
- [195] Damien Porquet, Jean Michel Dischler, and Djamchid Ghazanfarpour. “Real-time high-quality view-dependent texture mapping using per-pixel visibility”. In: *Proceedings - GRAPHITE 2005 - 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*. 2005, pp. 213–220. ISBN: 1595932267. DOI: 10.1145/1101389.1101432.

- [196] Alvin W K Soh et al. "Texture mapping of 3D human face for virtual reality environments". In: *International Journal of Information Technology* 8.2 (2002), pp. 54–65. URL: http://www2.docm.mmu.ac.uk/STAFF/E.Prakash/research/cep%7B%5C_%7Dpapers%7B%5C_%7Dedmond/Edmond%7B%5C_%7DIJIT%7B%5C_%7DTexture%7B%5C_%7DFace.pdf.
- [197] Chris Buehler et al. "Unstructured lumigraph rendering". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001*. Association for Computing Machinery, 2001, pp. 425–432. ISBN: 158113374X. DOI: 10.1145/383259.383309.
- [198] Paul Debevec, Yizhou Yu, and George Borshukov. "Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping". In: Springer, Vienna, 1998, pp. 105–116. DOI: 10.1007/978-3-7091-6453-2_10.
- [199] Yuta Nakashima et al. "AR image generation using view-dependent geometry modification and texture mapping". In: *Virtual Reality* 19.2 (June 2015), pp. 83–94. ISSN: 14349957. DOI: 10.1007/s10055-015-0259-3.
- [200] Niklas Gard, Anna Hilsmann, and Peter Eisert. "Projection Distortion-based Object Tracking in Shader Lamp Scenarios". In: *IEEE Transactions on Visualization and Computer Graphics* 25.11 (Nov. 2019), pp. 3105–3113. ISSN: 19410506. DOI: 10.1109/TVCG.2019.2932223.
- [201] Aleksandar Milosavljević et al. "Integration of GIS and video surveillance". In: *International Journal of Geographical Information Science* 30.10 (Oct. 2016), pp. 2089–2107. ISSN: 13623087. DOI: 10.1080/13658816.2016.1161197.
- [202] Chengyuan Lin et al. "Robust high-level video stabilization for effective AR telementoring". In: *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., Mar. 2019, pp. 1038–1039. ISBN: 9781728113777. DOI: 10.1109/VR.2019.8798331.
- [203] A Criminisi and A Zisserman. "Shape from Texture: Homogeneity Revisited". In: *microsoft.com*. 2013, pp. 9.1–9.10. DOI: 10.5244/c.14.9. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2000/09/Criminisi%7B%5C_%7Dbmvc2000.pdf.
- [204] Bill Baxter et al. "DAB: Interactive haptic painting with 3D virtual brushes". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001*. New York, New York, USA: Association for Computing Machinery, 2001, pp. 461–468. ISBN: 158113374X. DOI: 10.1145/383259.383313. URL: <http://portal.acm.org/citation.cfm?doid=383259.383313>.
- [205] Paul Haeberli and Mark Segal. *Texture Mapping as a Fundamental Drawing Primitive*. Tech. rep. 1993. URL: <http://www.graficaobscura.com/texmap/index.html>.
- [206] Mostafa Hassan. *Proposed workflow for UV mapping and texture painting*. Tech. rep. URL: www.bth.se.

- [207] Yongxiao Fu and Yonghua Chen. “Computer-Aided Design and Applications Haptic 3D Mesh Painting based on Dynamic Subdivision”. In: *Computer-Aided Design & Applications* 5.4 (2008), pp. 131–141. DOI: 10.3722/cadaps.2008.131–141. URL: <http://www.cadanda.com>.
- [208] David DeBry et al. “Painting and rendering textures on unparameterized models”. In: *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02*. 2002, pp. 763–768. ISBN: 1581135211. DOI: 10.1145/566570.566649.
- [209] Takeo Igarashi and Dennis Cosgrove. “Adaptive unwrapping for interactive texture painting”. In: *Proceedings of the Symposium on Interactive 3D Graphics*. 2001, pp. 209–216. DOI: 10.1145/364338.364404.
- [210] Y. Iwakiri, Y. Omori, and T. Kanko. “Practical texture mapping on free-form surfaces”. In: *Proceedings - Pacific Conference on Computer Graphics and Applications*. Vol. 2000-January. IEEE Computer Society, 2000. ISBN: 0769508685. DOI: 10.1109/PCCGA.2000.883898.
- [211] / P Brunet. *Efficient and Handy Texture Mapping on 3D Surfaces*. Tech. rep. 3. 1999.
- [212] Yotam I. Gingold et al. “A direct texture placement and editing interface”. In: *UIST 2006: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*. 2008, pp. 23–31. ISBN: 1595933131. DOI: 10.1145/1166253.1166259.
- [213] Nicolas Cherin, Frederic Cordier, and Mahmoud Melkemi. “Texture mapping of images with arbitrary contours”. In: *Journal of WSCG* 21.3 (2013), pp. 193–204. ISSN: 12136972.
- [214] Wendy H. Chun and Tobias Höllerer. “Real-time hand interaction for augmented reality on mobile phones”. In: *International Conference on Intelligent User Interfaces, Proceedings IUI*. New York, New York, USA: ACM Press, Mar. 2013, pp. 307–314. ISBN: 9781450320559. DOI: 10.1145/2449396.2449435. URL: <http://dl.acm.org/citation.cfm?doid=2449396.2449435>.
- [215] Ulrich Neumann et al. “Augmented virtual environments (AVE): dynamic fusion of imagery and 3D models”. In: *Proceedings - IEEE Virtual Reality*. Vol. 2003-Janua. 2003, pp. 61–67. ISBN: 0769518826. DOI: 10.1109/VR.2003.1191122. URL: <https://ieeexplore.ieee.org/abstract/document/1191122/>.
- [216] Denis Kalkofen et al. “Visualization Techniques for Augmented Reality”. In: *Handbook of Augmented Reality*. Springer New York, 2011, pp. 65–98. DOI: 10.1007/978-1-4614-0064-6_3.

- [217] Steffen Gauglitz et al. “World-stabilized annotations and virtual scene navigation for remote collaboration”. In: *UIST 2014 - Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. New York, New York, USA: Association for Computing Machinery, Inc, Oct. 2014, pp. 449–460. ISBN: 9781450330695. DOI: 10.1145/2642918.2647372. URL: <http://dl.acm.org/citation.cfm?doid=2642918.2647372>.
- [218] Matthew Hirsch, Gordon Wetzstein, and Ramesh Raskar. “A compressive light field projection system”. In: *ACM Transactions on Graphics*. Vol. 33. 4. Association for Computing Machinery, July 2014, pp. 1–12. DOI: 10.1145/2601097.2601144. URL: <https://dl.acm.org/doi/10.1145/2601097.2601144>.
- [219] Stefan Hauswiesner, Matthias Straka, and Gerhard Reitmayr. “Free viewpoint virtual try-on with commodity depth cameras”. In: *Proceedings of VRCAI 2011: ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications to Industry*. New York, New York, USA: ACM Press, 2011, pp. 23–30. ISBN: 9781450310604. DOI: 10.1145/2087756.2087759. URL: <http://dl.acm.org/citation.cfm?doid=2087756.2087759>.
- [220] P Wang et al. “An image-based texture mapping technique for apparel products exhibition and interior design”. In: *Displays* 24.4-5 (2003), pp. 179–186. ISSN: 01419382. DOI: 10.1016/j.displa.2004.01.004. URL: www.elsevier.com/locate/displa.
- [221] Seoungjae Cho and Kyungeun Cho. “Real-time 3D reconstruction method using massive multi-sensor data analysis and fusion”. In: *Journal of Supercomputing* 75.6 (June 2019), pp. 3229–3248. ISSN: 15730484. DOI: 10.1007/s11227-019-02747-3.
- [222] M. Buss et al. “Development of a Multi-modal Multi-user Telepresence and Teleaction System”. In: *The International Journal of Robotics Research* 29.10 (Sept. 2010), pp. 1298–1316. ISSN: 0278-3649. DOI: 10.1177/0278364909351756. URL: <http://journals.sagepub.com/doi/10.1177/0278364909351756>.
- [223] Nicolas Paparoditis et al. “Stereopolis II a multi-purpose and multi-sensor 3d mobile mapping system for street visualisation and 3d metrology”. In: *Revue Francaise de Photogrammetrie et de Teledetection* 200 (2012), pp. 69–79. ISSN: 17689791. URL: <https://www.researchgate.net/publication/236846208>.
- [224] Gene Yu et al. *Multiview Geometry for Texture Mapping 2D Images Onto 3D Range Data Computer Vision and Pattern Recognition Multiview Geometry for Texture Mapping 2D Images Onto 3D Range Data* £. Tech. rep. 2006. URL: <https://www.researchgate.net/publication/237208168>.
- [225] Carlos A. Vanegas, Daniel G. Aliaga, and Bedřich Beneš. “Building reconstruction using manhattan-world grammars”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 358–365. ISBN: 9781424469840. DOI: 10.1109/CVPR.2010.5540190.

- [226] Lionel Baboud et al. “Automatic photo-to-terrain alignment for the annotation of mountain pictures”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2011, pp. 41–48. ISBN: 9781457703942. DOI: 10.1109/CVPR.2011.5995727.
- [227] Hoang Hiep Vu et al. “High accuracy and visibility-consistent dense multiview stereo”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.5 (2012), pp. 889–901. ISSN: 01628828. DOI: 10.1109/TPAMI.2011.172.
- [228] Okan Erat et al. “Drone-Augmented human vision: Exocentric control for drones exploring hidden areas”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.4 (Apr. 2018), pp. 1437–1446. ISSN: 10772626. DOI: 10.1109/TVCG.2018.2794058.
- [229] Patrick Stotko et al. “SLAMCast: Large-Scale, Real-Time 3D Reconstruction and Streaming for Immersive Multi-Client Live Telepresence-Supplemental Material”. In: (2019). DOI: 10.1109/TVCG.2019.2899231. URL: <http://dx.doi.org/10.1109/TVCG.2019.2899231>.
- [230] Chunmei Hu, Yanmin Wang, and Wentao Yu. “Mapping digital image texture onto 3d model from lidar data”. In: *ISPRS Congress*. 2008, pp. 611–614. URL: http://www.isprs.org/congresses/beijing2008/proceedings/5%7B%5C_%7Dpdf/106.pdf.
- [231] Citation Mastin et al. “Automatic registration of LIDAR and optical images of urban scenes”. In: *ieeexplore.ieee.org*. 2010. ISBN: 9781424439911. DOI: 10.1109/CVPRW.2009.5206539. URL: <http://hdl.handle.net/1721.1/59839>.
- [232] Joseph R Spann and Karen S Kaufman. “PHOTOGRAMMETRY USING 3D GRAPHICS AND PROJECTIVE TEXTURES”. In: *Image (Rochester, N. Y.) XXXIII* (2000), pp. 2–9.
- [233] Jinhui Hu, Suyu You, and Ulrich Neumann. “Texture painting from video”. In: *13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2005, WSCG’2005 - In Co-operation with EUROGRAPHICS, Full Papers*. 2005, pp. 119–125. ISBN: 8090310079.
- [234] Lazaros Grammatikopoulos et al. *Automatic multi-view texture mapping of 3D surface projections*. Tech. rep. 2012. URL: <https://www.researchgate.net/publication/228367713>.
- [235] Xianfeng Gu, SJ Gortler, and Hugues Hoppe. “Silhouette mapping”. In: ... *Report TR-1-99*, ... (1999). URL: http://nrs.harvard.edu/urn-3:HUL.InstRepos:23017274%20http://research.microsoft.com/en-us/um/people/hoppe/silmap%7B%5C_%7Dtr.pdf.
- [236] Chulhan Lee, Junho Cho, and Kyoungsu Oh. “Visual hull with silhouette maps”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 4563 LNCS. 2007, pp. 88–96. ISBN: 9783540733348. DOI: 10.1007/978-3-540-73335-5_10.

- [237] F. Tsai and H. C. Lin. “Polygon-based texture mapping for cyber city 3D building models”. In: *International Journal of Geographical Information Science* 21.9 (Oct. 2007), pp. 965–981. ISSN: 13658816. DOI: 10.1080/13658810601034929.
- [238] Danhang Tang et al. “Real-time compression and streaming of 4D performances”. In: *SIGGRAPH Asia 2018 Technical Papers, SIGGRAPH Asia 2018*. Vol. 37. 6. Association for Computing Machinery, Inc, Dec. 2018, pp. 1–11. ISBN: 9781450360081. DOI: 10.1145/3272127.3275096. URL: <https://dl.acm.org/doi/10.1145/3272127.3275096>.
- [239] Cass Everitt. “Projective texture mapping”. In: *Citeseer* (). URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.6914%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf%20http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.104.6914>.
- [240] Igor Naigovzin. *Real-time Shadow Mapping Joined Project in Computer Graphics for Rafael Armament Development Authority*. Tech. rep. 2006. URL: https://webee.technion.ac.il/Sites/People/cgm/Computer-Graphics-Multimedia/Undergraduate-Projects/archive/2006/RAFAEL%7B%5C_%7Dfast%7B%5C_%7Dshadows/FastShadowProject/Project%20Book/Papers/Report.pdf.
- [241] Jung Ju Choi and Hwan Jik Lee. “Rendering stylized highlights using projective textures”. In: *Visual Computer* 22.9-11 (Sept. 2006), pp. 805–813. ISSN: 01782789. DOI: 10.1007/s00371-006-0070-y.
- [242] Mark Segal et al. “Fast shadows and lighting effects using texture mapping”. In: *Computer Graphics (ACM)* 26.2 (1992), pp. 249–252. ISSN: 00978930. DOI: 10.1145/142920.134071. URL: <https://dl.acm.org/doi/pdf/10.1145/133994.134071>.
- [243] Peiyi Shen and Philip Willist. “Texture for volume character animation”. In: *Proceedings - GRAPHITE 2005 - 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*. 2005, pp. 255–264. ISBN: 1595932267. DOI: 10.1145/1101389.1101438.
- [244] Zhan Yu et al. “Line Assisted Light Field Triangulation and Stereo Matching”. In: (2013), pp. 2792–2799. DOI: 10.1109/ICCV.2013.347.
- [245] Michael Tetzlaff and Gary Meyer. “Image-based relighting using environment maps”. In: *Archiving 2017 - Final Program and Proceedings*. Vol. 2017-May. Society for Imaging Science and Technology, 2017, pp. 23–27. DOI: 10.2352/issn.2168-3204.2017.1.0.23.
- [246] Yizhou Yu. “Efficient visibility processing for projective texture mapping”. In: *Computers and Graphics (Pergamon)* 23.2 (1999), pp. 245–253. ISSN: 00978493. DOI: 10.1016/S0097-8493(99)00034-5. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.9255%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf>.
- [247] Chuanbo Wang. “Identifying Wounds with AI”. Milwaukee, 2019.

- [248] Chuanbo Wang et al. “Fully Automatic Wound Segmentation with Deep Convolutional Neural Networks”. In: ().
- [249] Behrouz Rostami. “Burn Wound Classification using Deep Convolutional Neural Networks and Transfer Learning”. Milwaukee, 2019.
- [250] Rachel Khoo and Shirley Jansen. “The Evolving Field of Wound Measurement Techniques: A Literature Review.” In: *Wounds : a compendium of clinical research and practice* 28.6 (June 2016), pp. 175–81. ISSN: 1943-2704. URL: <http://www.ncbi.nlm.nih.gov/pubmed/27377609>.
- [251] N. D.J. Hettiarachchi et al. “Mobile based wound measurement”. In: *IEEE EMBS Special Topic Conference on Point-of-Care (POC) Healthcare Technologies: Synergy Towards Better Global Healthcare, PHT 2013*. 2013, pp. 298–301. ISBN: 9781467327664. DOI: 10.1109/PHT.2013.6461344.

CURRICULUM VITAE

VICTOR WILLIAMSON EDUCATION

2014–2020, University of Wisconsin at Milwaukee, Computer Science, Ph.D

- Dissertation: MOBILE WOUND ASSESSMENT AND 3D MODELING FROM A SINGLE IMAGE
- Computer graphics specialization
- 3D medical visualization
- Mobile-based 2D digital painting & animation

2008–2010, Massachusetts Institute of Technology, Computer Science, M.Eng

- Thesis: GOAL-ORIENTED WEB SEARCH
- Implement website to search Web for videos, images, and news using Planner module to dynamically optimize results
- Write website and Planner module in Python
- Use Web services YouTube Data API, Truveo XML, Google AJAX, and Yahoo BOSS
- Develop threat model
- Examine security issues of realtime third party code execution
- Research Google App Engine
- Optimize search results using video ratings
- Remote debug third party code in realtime

2000–2005, Massachusetts Institute of Technology, Computer Science, B.S

- Artificial intelligence emphasis

- Humanoid robotics

EXPERIENCE

March 2010–June 2014, Sr. Programmer Analyst, VERIZON WIRELESS, WARREN, NJ

- Architect for mobile point of sale iPad app
- Development lead for retail iPad queue management
- Build framework to render HTML screens via Apache Velocity and Trim-Path
- Implement data layer as JSON Web services
- Transform XML to JSON; Build wizard for Iconic Portal phone sales
- Design and implement Iconic Portal Shopping Cart for e-commerce
- Build In-Store pickup framework for online orders.
- Implement Android DroidX application for retail express accessory check-out
- Run usability tests with retail greeters and store reps
- Use servlet filter and cookie storage modules to authorize and authenticate users across TLS and green zone environments
- Use HP Fortify and input filtering to defend against XSS and SQL injection
- Investigate HTML 5 local storage facilities for Web views
- Implement Webkit scroll with 3D transforms

July 2008–August 2008, Software Engineering Intern, BROOKHAVEN NATIONAL LABORATORY, UPTON, NY

- Design and implement Quench Reasoning System (QRS) to analyze post-mortem data from the Relativistic Heavy Ion Collider (RHIC)

- Diagnose real magnet quenches
- Analyze power supply voltage and beam loss of quench to rank most faulty power supplies and areas of beam loss
- Debug C++ with Data Display Debugger on Red Hat Linux
- Integrate QRS with logs
- Capture and display beam and voltage data bar and line graphs
- Tweak alpha values against real data

July 2005–June 2008, Software Engineer, IBM, WESTFORD, MA

- Implement JAVA and JavaScript code for Sametime
- Build with Dojo JS Toolkit and JMS messaging services
- Investigate metrics and statistics to gather Sametime usage and personalize user experience
- Design and implement session manager to prevent race conditions and save memory in Sametime Gateway
- Design and implement configuration layer for Sametime Gateway
- Integrate config with IBM WebSphere Application Server
- Use DTO, DAO and BO Design Pattern
- Use Dbpersist framework to persist objects in memory
- Support SSL configurations between clustered IBM WebSphere servers and public IM providers Google Talk, Yahoo Messenger, and AOL
- Integrate with XMPP and SIP; Build installer for IBM Workplace Forms ISMP

PROJECTS

Jan. 2005–June 2005, Advanced Undergraduate Project, *Prof. Tomás Lozano-Pérez, MIT*

- Implement and test Bayesian spam filter and compare performance to Support Vector Machines (SVM) filters
- Use Java to parse email
- Train and test on spam messages
- Achieve 90% success rate

PUBLICATIONS

2020

- Wang, Chuanbo, D. M. Anisuzzaman, Victor Williamson, Behrouz Rostami, Jeffrey Niezgoda, Sandeep Gopalakrishnan, and Zeyun Yu. "Fully Automatic Wound Segmentation with Deep Convolutional Neural Networks."

PRESENTATIONS

August 2015, GEM Conference, *Boston, MA*

- Title: The loss of African history and culture
- Alum research
- Technical presentation

TEACHING

Sept. 2014–May 2015, Data Structures and Algorithms, *UW-Milwaukee*

- Teaching assistant
- CompSci 351
- Duration: One year

Sept. 2018–May 2019, Intermediate Computer Programming, *UW-Milwaukee*

- Teaching assistant

- CompSci 251
- Duration: One year

CONFERENCES

October 2018, Blacks in Technology, *St. Paul/Minneapolis, MN*

- The Inaugural Blacks in Technology Conference, BITCON2018, is a 3-day interactive platform for professionals, entrepreneurs, influencers, subject matter experts, students and thought leaders

August 2015, GEM Conference, *Boston, MA*,

- The National GEM Consortium 39th boarding meeting and annual conference

August 2007, NSBE Conference, *Columbus, OH*

- The revolution will not be televised

August 2005, NSBE Conference, *Boston, MA*

- Job interviewing

AWARDS

Sept. 2019–June 2020, AOP R1 Fellow, *UW-Milwaukee*

- The Advanced Opportunity R1 Fellowship are is a one year fellowship for minority or disadvantaged students who have achieved dissertator status

Sept. 2015–June 2018, AOP Fellow, *UW-Milwaukee*

- Advanced Opportunity Fellowships are designed to assist underrepresented racial/ethnic minority and disadvantaged non-minority students in graduate study to enter and complete a graduate degree at UWM.

March 2014, POS Mobile Acts of Excellence, *Verizon Wireless*

- iPad-based retail greeter
- Store queue management

June 2008–Dec. 2010, GEM Fellow, *MIT*

- The mission of The National GEM Consortium is to enhance the value of the nation’s human capital by increasing the participation of under-represented groups
- Duration: Three semesters

July 2006, IBM CEO Ovation for Sametime Acts of Excellence, *IBM*

- Manage SSL configuration, certificates and Java KeyStore
- Semi-automated deployment and installation

PATENTS

May 2017, Patent No. 2014024469, *Verizon Wireless*

- Title: Converting XML to JSON with configurable output
- A system to automatically convert XML to JSON using the streaming Java API for XML (StAX)
- A device is configured to receive information that identifies an XML data structure, an input key including information that identifies an element in the XML data structure, and an output key including information that identifies a manner in which the element is to be identified in a JSON data structure. The device is configured to parse the XML data structure to determine an element identified by the input key, and generate the JSON data structure by storing an association between the element and the output key in the JSON data structure. The device is configured to populate the JSON data structure with content from the XML data structure, where the content is identified by the input key and is placed in a location in the JSON data structure identified by the output key. The device is configured to store the JSON data structure.

December 2007, Patent No. 20090157817, *IBM*

- Title: Unsynchronized event pool to improve performance of event driven IM gateway
- An efficient multi-threaded object-sharing event pool
- The present invention discloses a solution for using an unsynchronized event pool in an IM gateway. In the solution, a set of object pools can be established, each including a set of event objects ordered relative to each other in a circular fashion. The IM gateway can detect an incoming message. A CPU processing thread for handling the incoming message can be determined. One of the unsynchronized event object pools associated with the thread can be ascertained. A current one of the ordered event objects within the pool can then be determined. The current one can be an object ordered after a last event object used from the pool. The thread can utilize the determined event object to handle a task for the incoming message, which negates a need and a time to create a new event object for the thread.