

THREE-DIMENSIONAL RECONSTRUCTION AND EVALUATION OF DECUBITUS USING A 3D SCANNER

Szabina Molnár-Zékány 

PhD student, University of Miskolc

3515 Miskolc-Egyetemváros, e-mail: sz.zekany@gmail.com

Szilvia Árvai-Homolya 

associate professor, University of Miskolc, Institute of Mathematics, Department of Analysis

3515 Miskolc-Egyetemváros, e-mail: szilvia.homolya@uni-miskolc.hu

Abstract

The aim of this study is to improve the field of patient care in medicine by utilizing information technology, as well as to help, assist nurses and doctors in their work and accelerate patients' recovery. In the case of long-term, mostly elderly patients, a condition known as chronic pressure ulcer, decubitus, or pressure sore may develop. Treating these decubitus ulcers takes time and patience. In order to find the appropriate treatment, it can be helpful to gather as much data as possible about the wound and categorize the severity type based on these collected data. This problem will be addressed and explored using the potential of artificial intelligence and three-dimensional digitisation.

Keywords: *patientcare, wound scanner, 3D scanner, 3D reconstruction*

1. Introduction

In the case of immobile patients who permanently immobilized in one position, pressure ulcers develop due to prolonged pressure concentrated on a single point. The most common areas affected are the lumbar region, but pressure ulcers can also develop on the heels.

Immobilized and wheelchair-bound patients experience reduced blood circulation in continuously pressured body parts, which can lead to skin and tissue damage in the affected areas. Ideally, pressure ulcers should be prevented by incorporating regular physical activity, frequent changes in body position, proper nutrition, and the use of various pressure-reducing mattresses (Anders et al., 2010). However, there are cases where intervention comes too late or preventive measures cannot be applied due to the patient's condition.

Furthermore, the development of pressure ulcers is facilitated by conditions such as diabetes, certain vascular diseases, or a weakened immune system (He et al., 2020). Once pressure ulcers have formed, they need to be properly treated and cared for to prevent worsening and, to the extent possible, promote healing. To facilitate healing, various measurements are taken at the wound site to assess its severity (Bansal et al., 2005).

2. Neural network

To perform the necessary measurements on the wound and obtain meaningful results, several preparatory steps are required.

After scanning, the measurement results are determined based on recognizing the wound, finding its contour, and covering it to reconstruct the original skin surface as much as possible. The solution to these tasks relies on a neural network. The neural network's operation and training require a preliminary dataset to teach the network for the specific task. Various types of neural networks exist, depending on the problem to be solved. In this case, the appropriate type of network is a Convolutional Neural Network (CNN), which is designed for image processing and computer vision tasks. This type of network processes images and learns to produce a specific output for a given input through image analysis (Li et al., 2022).

Among the convolutional neural networks, we utilize a pre-defined layer network called MobileNetV2. To analyze the wound, we need to determine precisely where the wound is located on the skin surface. After the training process, the convolutional neural network will assist us in this task. It takes an overhead image of the wound as input and provides an output mask, indicating the wound's location. The mask is a black-and-white image, where colors between black (#000000) and white (#FFFFFF) represent the wound surface. Neural networks work in a way that the output is a probability value. This is why the masks obtained from the network contain shades of gray between pure black and white.

The network determines the color of each pixel according to the likelihood of that pixel corresponding to a wound or healthy skin surface in the original image. A completely black color indicates a healthy skin surface, while a completely white color indicates a region with a pressure ulcer. The values on the grayscale between black and white indicate the probability of a pixel belonging to the wound region.

2.1. Collecting data

To train the neural network, we need data, specifically images that can be used as a training dataset for the Convolutional Neural Network (CNN).

Data collection can be done in two main ways. There are internet databases where we can find data suitable for our neural network, and we can also create our own data. The advantage of using our own data is that we can fully customize the data collection conditions, the quality of sampling, and the amount of data we gather. However, this is a more time-consuming task. If others have worked on a similar problem we are trying to solve, and there is a good quality and accessible internet dataset available, it may be worth considering its use. However, there are cases where the field is completely unexplored and no free, pre-existing, and accessible data is available. In such situations, collecting our own data becomes necessary.

For wound reconstruction, we use both pre-existing and self-generated data. The internet-sourced data helps augment the available dataset, but it might not cover the entire scope of our specific wound reconstruction problem. Since the wound scanner in this version is designed for scanning and processing pressure ulcers in the sacral region, the available internet images might primarily focus on masking tissue changes on the heels. To address this limitation and provide a comprehensive dataset for the neural network's training, we create our own images of wounds using the scanner, ensuring that the data fits our specific requirements for wound reconstruction.

2.2. Scanning

To collect the data and perform the wound reconstruction, we need the three-dimensional digital representation of the wound, which is obtained through a 3D scanning technology (Abid et al., 2019).

There are several scanning methods available, each providing solutions for different problems (Thali et al., 2003). The most well-known scanning technologies include (Pesce et al., 2015):

- Photogrammetry
- Structured light technology (Rashaan et al., 2016)
- Stereo camera
- LiDAR

In our case, the most suitable solution is the LiDAR technology. LiDAR (Light Detection and Ranging) is a remote sensing technology that uses laser beams to measure the distance to objects and creates a three-dimensional point cloud of the object (Filko et al., 2023). The LiDAR system consists of an emitter that sends out laser beams and a sensor that detects the reflected laser beams. By knowing the time of emission and the time of arrival of the reflected beams, we can calculate the time it takes for the light to bounce back from the object. This information is then used to determine the position and spatial topography of objects and create the point cloud. (Raj et al., 2020)

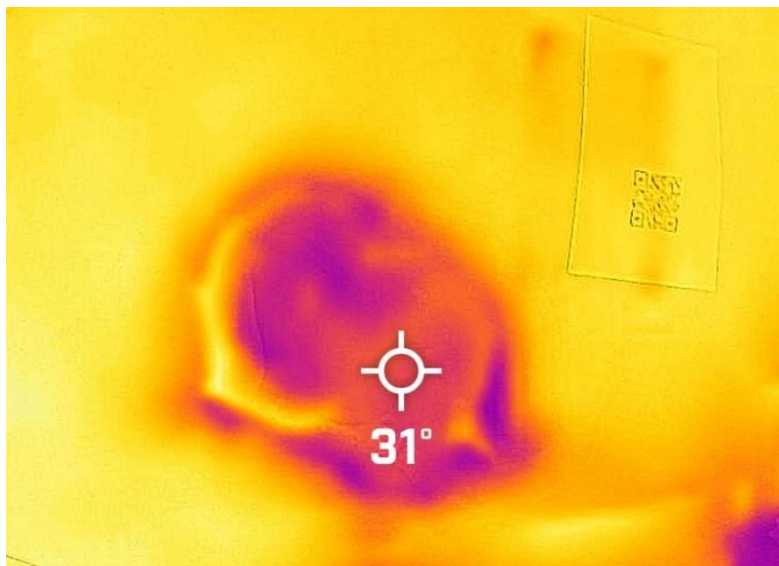


Figure 1. Thermal image of the wound

LiDAR scanners can operate in different ranges of light, from non-visible wavelengths to the visible light spectrum. The visible light spectrum ranges from 400 to 700 nanometers. The LiDAR scanner used for scanning with the iPhone 13 Pro emits light with a wavelength of 940 nanometers, which falls outside the visible light spectrum. The built-in LiDAR scanner in the phone can be operated through various applications. It is designed to work more accurately with the help of the surrounding cameras and additional sensors built into the phone, such as motion sensors.

When scanning wounds beside the patient's bed, it is essential to use a user-friendly, wireless, and portable scanner that can be easily operated by nurses to efficiently digitize the three-dimensional surface of the wound. The iPhone's cameras can also capture texture information of the three-dimensional object, providing an RGB scan instead of a single-color grayscale scan. This will be helpful during network training to identify wound contours based on colors. Furthermore, a thermal camera can be attached to the phone to create a thermal map of the wound, which can provide additional information about the wound surface and refine the results obtained so far. For instance, the temperature of fresh

wounds is typically higher due to the early stage of inflammation. While this might be visible to the naked eye in many cases, there are smaller developing skin lesions where the extent of the newly formed wound may not be readily apparent.

During the scanning process, to ensure patient data confidentiality and enable simple anonymous identification, QR codes are used and affixed to the skin surface for the duration of the scanning. The scanning procedure follows these steps:

- Nurses position the patient correctly (usually lying on their side or, in certain cases, on their stomach).
- The area around the wound is cleaned.
- QR codes are placed on the intact skin surface near the wound.
- If necessary, the hanging skin is lifted by the nurse.
- Scanning is performed using the handheld iPhone LiDAR with the attached thermal camera, along with additional lighting.

The files generated as a result of the scanning process include:

- Point cloud/mesh: In .obj file format, with .jpg texture and the corresponding texture mapping in the .mtl file.
- 2D RGB images: Images captured from multiple angles and directions.
- Thermal camera images.

By using QR codes for identification and maintaining patient data discretion, the scanning process can be conducted efficiently while protecting the privacy of the patients. The resulting files, including the point cloud and images, provide valuable information for wound reconstruction and analysis (Filko et al., 2021).

2.3. Data preparation, data cleansing and annotation

To perform predictions using the neural network, which means obtaining an output from a given input, a well-trained neural network is required. The essence of training the network lies in associating the input data with the corresponding output data. By providing known outputs based on the input parameters, we train the network. To achieve this, the images of individual pressure ulcers must have their corresponding masks, which indicate the exact output for a given input. This process, where we determine the precise output for a given input, is called annotation. In our case, we have two-dimensional RGB images as inputs, and we need to determine the corresponding output, which is a black and white two-dimensional image. It is important to note that we only use two colors, black and white, because we are not providing probabilities to the network, but rather we know precisely which pixels represent the wound and which pixels represent healthy skin.

Before creating the masks for the wounds, we need to prepare the images for annotation. Although we have taken 2D RGB images of the wounds, they might not have been captured from the appropriate angle. To address this, we use a solution where we utilize the scanned meshes and capture top-view images from them. The meshes may contain irrelevant parts or elements from the wound's surroundings (e.g., blankets, bed parts, the patient's leg, or the hand of a nurse reaching in).



Figure 2. Mesh before and after cleansing

In the first step, we need to clean the meshes by removing unnecessary areas. This is done by separating and deleting the unnecessary parts of the point cloud. After the cleaning process, we are left with a roughly rectangular-shaped mesh, which includes the wound, the surrounding healthy skin, and the QR code. The *Figure 2* illustrates the wound before and after the cleaning process.



Figure 3. Input mask for the convolutional neural network

After the cleaning process, we still need to perform mesh alignment. The essence of alignment is to fix the origin, check the position of the mesh, adjust its orientation, and perform transformations (translation, rotation, scaling) until the wound surface becomes horizontal with the xy plane. Once this is done, we capture a top-view image of the wound, as shown in the left image of the figure. Subsequently, we can create the corresponding mask, as seen in the right image of the figure. The mask image should have the same width and length as the original RGB image. (Fangzhao et al., 2018)

2.4. Training

Having a larger and more diverse dataset with training examples generally leads to better results and improved accuracy in the neural network. In your case, the dataset can be further expanded with self-captured images. The top-view images of wounds and their corresponding masks are stored in separate folders, with matching names to enable the network to associate each image with its corresponding mask. (Wang et al., 2015). The training process involves using the MobileNet V2 convolutional neural network (Wang et al., 2020). The number of epochs is set to 100, meaning the entire training dataset passes through the network's training process that many times.

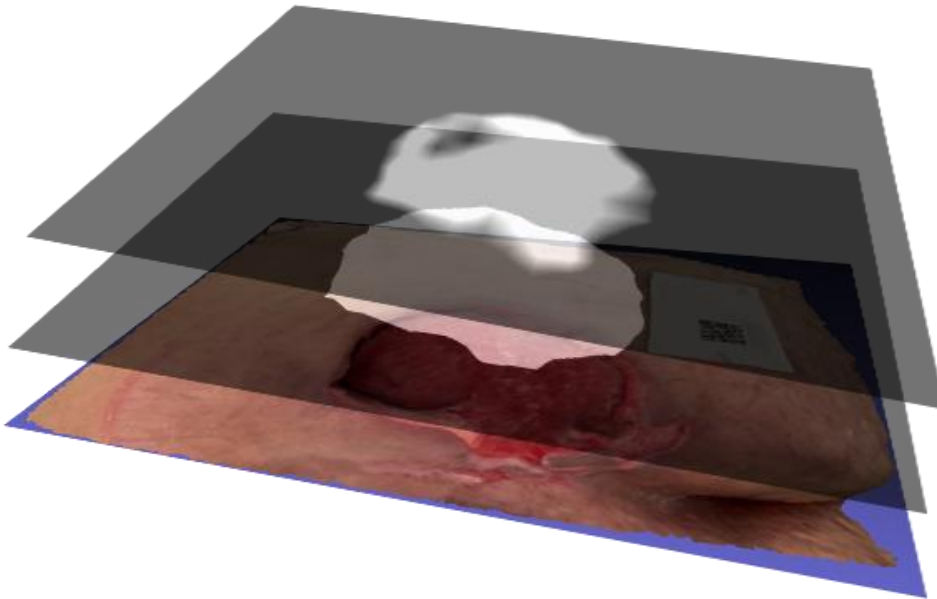


Figure 4. *The original wound area and the one found by the neural network*

The batch size determines how many subsets of the training data are sent to the network in a single epoch. In this case, the batch size is set to 10. The learning rate is set to 0.0001, which determines the extent to which the model's parameters are modified during each iteration. It adjusts the step size used to modify the weights during the learning process, effectively controlling the rate at which the network approaches the desired outcome and minimizes the error. Selecting an appropriate learning rate is important. If the learning rate is too large, it is possible to overshoot the global minimum of the loss function and not achieve the best results.

The binary cross-entropy loss function is used, which summarizes the network's error for each output. Based on this value, the network updates its weights at the beginning of the next learning cycle in steps defined by the learning rate. The Adam algorithm is used as the optimization function in the network, responsible for updating and adjusting the weights. To summarize, during the learning process, the network goes through multiple iterations (epochs) and reads the training dataset in smaller subsets (batch size).

During each iteration, it adjusts the weights using the Adam optimization function in steps determined by the learning rate to minimize the loss function's value. After the training process is complete, you obtain a neural network that provides the best possible output accuracy based on the configured parameters.

3. Contour detection

After training the network, upon an input image, the network produces a mask prediction showing the probabilities of individual pixels being part of the decubitus. Within this mask, we need to locate the predicted boundary of the wound area.

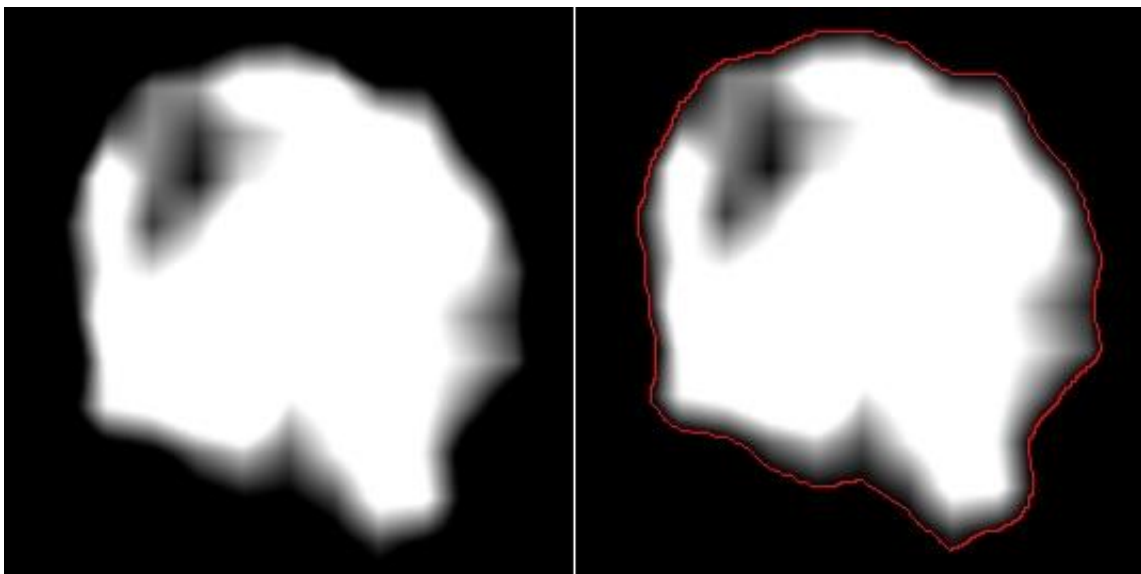


Figure 5. Finding contour pixels on the mask

To cover the wound surface, we must find the contour of the wound. The discovery of the contour along the predicted wound area in the mask occurs by iterating through the pixels of the image and identifying those pixels that differ from black. These points are not only true for the outermost pixel points of the wound surface, so we precisely examine the surrounding pixels of a given pixel. For example, if we find a non-black pixel on the left edge of the wound, that pixel will only be a contour point if there is a black pixel to its left, a pixel with a different color than black to its right, and either a black or a different-colored pixel above or below it, depending on whether it is on the top left or top right of the wound surface. In simple terms, there are two cases where a pixel point cannot be a contour point.

One case is when its neighbors are pixels with colors different from black, as this would indicate that we are inside the wound. The other case is when all the surrounding pixels are black, which would mean that we are definitely outside the boundary of the wound. For visualization purposes, the contour is drawn in red, and you can see an example in *Figure 5*.

4. Contour pixels to 3D points

Once we have the contour, we need to project it onto the three-dimensional mesh. The image containing the mask has a certain resolution, and as a result, the contour pixels have x and y coordinates. We will use these values in our calculations to project the contour onto the mesh.

The mesh is surrounded by a so-called bounding box. The bounding box is the minimal rectangular prism that encloses the mesh completely. In three-dimensional space, the bounding box has coordinates for its eight corners, and for each corner, we know its spatial xyz coordinate values. We will need the coordinates of four corners that form a rectangle parallel to the xy plane and located in the positive direction of the z-axis. We will align the contour found on this plane/rectangle. However, we need a mathematical transformation because in most cases, the 2D mask image's width and height do not match the dimensions of the top-view rectangle of the bounding box that surrounds the mesh. For each contour pixel, we need to convert its xy position relative to the bounding box, which can be done using the following formula:

$$x = \frac{\text{contour_pixels_x}}{\text{img_width}} * (\text{max_x} - \text{min_x}) + \text{min_x} \quad (1)$$

$$y = \frac{\text{contour_pixels_y}}{\text{img_height}} * (\text{min_y} - \text{max_y}) + \text{max_y} \quad (2)$$

To obtain the new x-coordinate value step-by-step, we follow these calculations:

- **contour_pixels_x/img_width:** The current x-coordinate value is divided by the width of the image.
- **(max_x-min_x):** The max_x represents the outermost value of the bounding box on the x-axis in the positive direction, while the min_x represents the outermost value in the negative direction, with the center located at the origin. To obtain the size of the bounding box, we subtract the min_x from the max_x. This gives us the dimensions of the bounding box along the x-axis.
- **(contour_pixels_x/img_width * (max_x-min_x)):** The scaled x-coordinate value obtained by calculating contour_pixels_x/img_width is multiplied by the size of the bounding box. This step rescales the x-coordinate value to fit within the dimensions of the bounding box along the x-axis. The result of this multiplication will be the new x-coordinate value relative to the bounding box.
- **(contour_pixels_x/img_width * (max_x-min_x)) + min_x:** After multiplying the scaled x-coordinate value by the size of the bounding box, you add the min_x value. This addition is necessary to shift the x-coordinate values properly so that they align with the bounding box.

Without this step, the points would be shifted towards the negative direction on the x-axis.

We perform the same calculation and translation along the y-axis to obtain the new y-coordinate values for the points.

Along with the newly calculated x and y coordinate values, a z-coordinate value is added. This allows us to transform and position the 2D pixel points in the three-dimensional space, and from this point onwards, we refer to them as 3D points. These points can be considered as a point cloud, but they are situated on a plane parallel to the xy-axis, meaning that the z-coordinate is the same for all points. The example in *Figure 6* shows the representation of the transformed 3D points of the contour pixels from the 2D image.

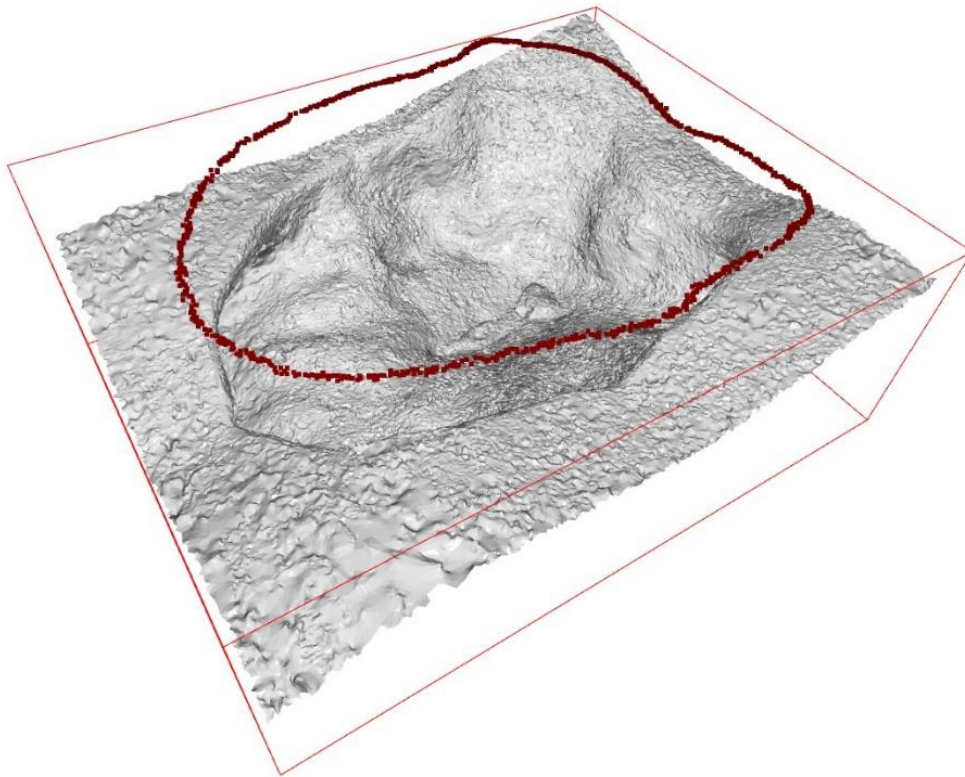


Figure 6. 2D contourpoints on a flat surface

5. Projection

Currently, the contour points are located on the xy plane of the bounding box. However, we need to map these points to the corresponding points on the 3D mesh, where the z-coordinate of the contour point represents the coordinate of the mesh surface at that point. To achieve this, we use a projection process.

The projection involves casting a ray from each point in the -z direction, essentially extending it to infinity. However, the ray will intersect with the mesh surface at some point rather than continuing infinitely. Where the ray intersects with the mesh, we examine that point. This intersection point is not just a single point; rather, it corresponds to one of the vertices that make up the mesh.

The mesh consists of vertices, edges connecting the vertices, and triangular polygons formed by these edges, which we call faces. The vertices have xyz coordinates, and we can save them in a sequence in an array. When performing the projection for each point, we determine which vertex it intersects and save the vertex coordinates. These saved points will be used for visualization, and we plot them as shown in *Figure 7*.

This process enables us to project the 2D contour points onto the 3D mesh, and each point in the contour corresponds to a specific vertex on the mesh's surface.

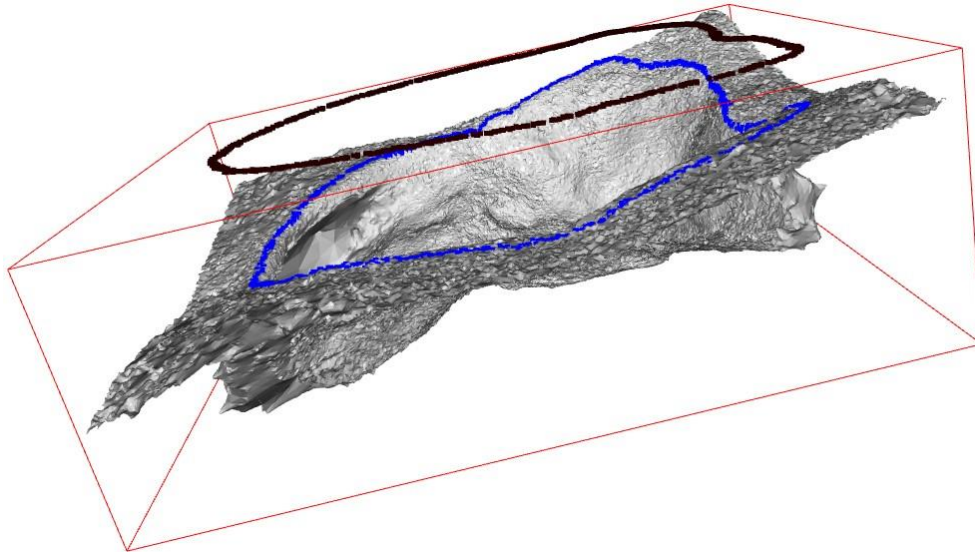


Figure 7. 2D points projected into 3D space

6. Polarcoordinate-system and reconstruction

To close the wound surface, we will utilize the points forming the contour of the 3D mesh. First, we need to find the center of the wound on the xy plane. To do this, we select the point with the largest and smallest x and y coordinates among the contour points, then, we take the average of these points.

We perform the same calculation for the z-axis, resulting in an average value in the z-direction. Once we have the center point, we connect the rest of the points to it and cover the wound by creating polygons (vertices) placed between the edges formed by connecting these points. To achieve this, we need to iterate through the points of the contour area. However, it is crucial that the points are ordered sequentially in the array in a clockwise manner. Unfortunately, the order of the points may not always be obvious, and they could be randomly placed in the array, meaning that we need to sort them somehow.

To arrange the points in order, we can switch to the polar coordinate system and sort them accordingly. In the polar coordinate system, the reference point is called the pole (the origin), and the radial axis is the ray starting from the pole in the reference direction (Tripathi et al., 2005). The distance from the pole is called the radial distance or simply the radius, and the angle is called the angular coordinate. In our case, the polar axis will be the positive x-axis. By applying a simple formula, we can calculate the distance and angle for each point in the polar coordinate system. This allows us to sort the points and arrange them in the desired clockwise order, facilitating the process of closing the wound surface by connecting the points with polygons.

$$r = \sqrt{x^2 + y^2} \quad (3)$$

$$\varphi = \arctan\left(\frac{y}{x}\right) \quad (4)$$

After sorting the values, we can begin closing the wound. We have the wound center point and the array containing the projected points along the contour of the wound on the 3D mesh. To create the vertices that will enclose the wound surface, we take three points at a time, connect them with edges, and form a triangle in the three-dimensional space. We then fill this triangle with a polygon to create each vertex.

To connect the vertices properly, we need to iterate through the array to link each new vertex to the previous one.

The essence of the iteration is to take the first element of the array, the second one, and the wound center point, connecting and covering them. In the next step, we take the second point (which was part of the previous vertex) and the third point, along with the wound center point, and again connect and cover them. This process continues until we return to the starting point. It can be observed that the algorithm's steps can be generalized during each iteration. We take the i -th and $i+1$ -th elements, along with the wound center point, connecting and covering them. We repeat this process until we return to the starting point. Furthermore, it is noticeable that each new vertex has two points that were part of the previous vertex, which means an edge is common between the new and the previous vertex.

This logic ensures a seamless closure of the mesh, making it watertight. A watertight three-dimensional object is one where there are no gaps, and every edge is connected to another vertex. This is essential to accurately measure the volume of the wound. If the three-dimensional object is not watertight, we cannot measure its volume accurately (Kriangkrai et al., 2020).

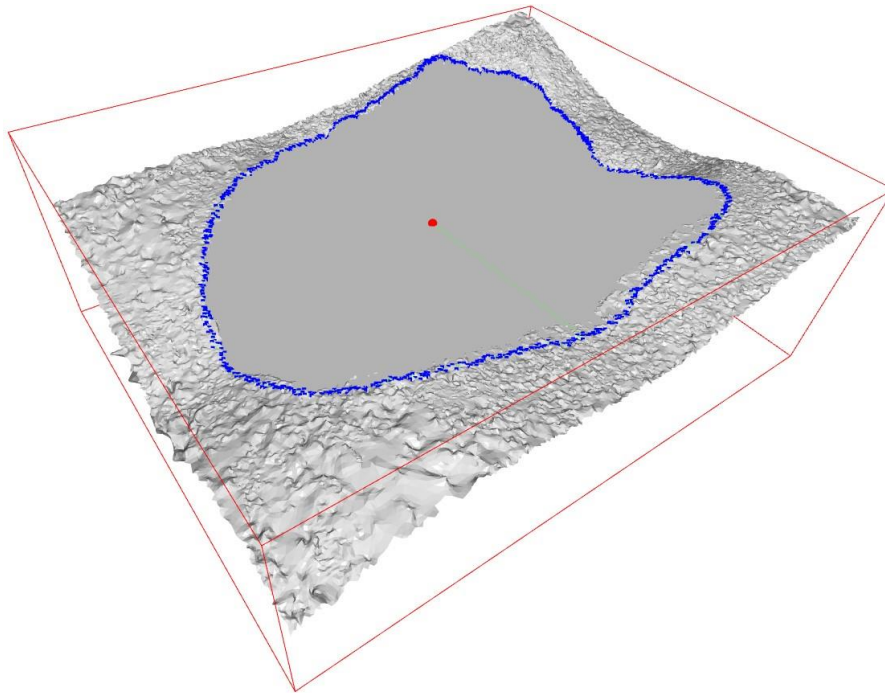


Figure 8. Reconstructed decubitus

The program was written and the neural network was implemented in Python. The three-dimensional object was visualized using the built-in functions of the Open3D library, which greatly simplifies the graphical representation. Open3D provides an easy-to-use interface for 3D visualization (Open3D, 2021). For the calculations on the three-dimensional object, the PyMeshlab library was utilized (PyMeshLab, 2022). This library contains functions for performing geometric operations on meshes. It includes a function that can calculate the volume of the object if the mesh is watertight. Although the volume calculation can be manually implemented using the vertices of the wound and the enclosing

shell, thanks to the PyMeshlab library, this step is not necessary. However, as a sanity check, it might be useful to implement the manual volume calculation algorithm for verification purposes (Llorens, 2021).

The perimeter of the wound can be easily calculated by summing the lengths of the lines connecting the contour points. In summary, the Python program uses the Open3D library for 3D visualization, PyMeshlab library for geometric calculations on the mesh, and the neural network for the wound segmentation task. The program's functionality includes volume calculation (both using PyMeshlab and manual method for validation), perimeter calculation, and visualization of the 3D object with the wound and enclosing shell.

7. Conclusion

During the development, it became evident that proper treatment of pressure ulcers is essential to make patients' long hospital stays more bearable. To achieve this, having as much information about the wound as possible is crucial. One way to collect data and perform measurements on the wound is through three-dimensional wound scanning and reconstruction. We have chosen to utilize the easily accessible and fairly accurate LiDAR technology of the iPhone 13 Pro smartphone for bedside scanning. The results of the development provide adequately accurate outcomes, which close to real measures.

Like most research and program implementations, this one can also be further improved. The computation of evaluation results with decimal precision is achievable with some post-processing. Additionally, the reconstruction of the wound surface can be assisted by the use of further artificial neural networks, as well as a heuristic solution, which involves slicing the wound parallel to the x and y axes at specified intervals. The points where these slices intersect can be used to construct a graph/network structure, and the points on this network can aid in reconstructing the uneven surface of the skin in more detail.

References

- [1] Tripathi, M. M. (2005). *Coordinate Geometry: Polar Coordinates Approach*. Alpha Science International.
- [2] Abid, M. J. (2019). *3D scanning applications in medical field: A literature-based review*, *Clinical Epidemiology and Global Health*, (Vol. 7, pp. 199–210). <https://doi.org/10.1016/j.cegh.2018.05.006>
- [3] Pesce, L., Galantucci, L.M., Percoco, G., Lavecchia, F. (2015). *A low-cost multi camera 3D scanning system for quality measurement of non-static subjects*, 3rd CIRP Global Web Conference. <https://doi.org/10.1016/j.procir.2015.04.015>
- [4] Li, Z., Yang, W., Peng, S., Liu, F. (2022). *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*. *IEEE Trans. Neural Netw. Learn Syst.*, 2022 Dec, 33 (12), 6999–7019, <https://doi.org/10.1109/TNNLS.2021.3084827>
- [5] Raj, T., Hashim, F. H., Huddin, A. B., Ibrahim, M. F., Hussain, A. (2020). Survey on LiDAR Scanning Mechanisms. *Electronics*, 9, 741, <https://doi.org/10.3390/electronics9050741>
- [6] Wang, C., Yan, X., Smith, M., Kochhar, K., Marcie, R., Warren, S. M., Wrobel, J., Lee, H. (2015). A unified framework for automatic wound segmentation and analysis with deep convolutional

- neural networks. *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2415-8, <https://doi.org/10.1109/EMBC.2015.7318881>
- [7] Wang, C., Ansiuzzaman, D. M., Williamson, V., Dhar, M. K., Niezgoda, J., Gopalakrishnan, S., Yu, Z. (2020). Fully automatic wound segmentation with deep convolutional neural networks. *Sci. Rep.*, 10, 21897, <https://doi.org/10.1038/s41598-020-78799-w>
- [8] Bansal, C., Scott, R., Stewart, D., Cockerell, C. J. (2005). Decubitus ulcers: a review of the literature. *Int. J. Dermatol.*, Oct. 44 (10), 805–10, <https://doi.org/10.1111/j.1365-4632.2005.02636.x>
- [9] Li, F., Wang, C., Liu, X., Peng, Y., Jin, S. (2018). A Composite Model of Wound Segmentation Based on Traditional Methods and Deep Neural Networks. *Computational Intelligence and Neuroscience*, pp. 1–12, <https://doi.org/10.1155/2018/4149103>
- [10] Llorens J., (2021). *Stockpile volume computation with Open3D* [Online]. <https://jose-llorens-ripolles.medium.com/stockpile-volume-with-open3d-fa9d32099b6f>
- [11] *Open3D Documentation* (2021). [Online] Open3D. <http://www.open3d.org/docs/latest/index.html>
- [12] *PyMeshlab Documentation* (2022). [Online] <https://pymeshlab.readthedocs.io/en/latest/index.html>
- [13] Filko, D., Nyarko, E. K. (2023). 2D/3D Wound Segmentation and Measurement Based on a Robot-Driven Reconstruction System. *Sensors*, 23, 3298, <https://doi.org/10.3390/s23063298>
- [14] Rashaan Z., Stekelenburg C., van der Wal M. et al. (2016). Three-dimensional imaging: a novel, valid, and reliable technique for measuring wound surface area. *Skin. Res. Technol.*, 22 (4), 443–450, <https://doi:10.1111/srt.12285>
- [15] Anders J., Heinemann A., Leffmann C., Leutenegger M., Profener F., von Renteln-Kruse W. (2010). Decubitus ulcers: pathophysiology and primary prevention. *Dtsch. Arztebl. Int.*, 107 (21), 371–382, <https://doi:10.3238/arztebl.2010.0371>
- [16] Filko, D., Marijanović, D., Nyarko, E. K. (2021). Automatic Robot-Driven 3D Reconstruction System for Chronic Wounds. *Sensors*, 21, 8308, <https://doi.org/10.3390/s21248308>
- [17] Kriangkrai T., Suriya N. (2020). An Integrated Hardware and Software Application to Support Wound Measurement Using a 3D Scanner and Image Processing Techniques. *The Open Biomedical Engineering Journal*, 14 (1), 55–73, <https://doi:10.2174/1874120702014010055>
- [18] He, X., Yang, S., Liu, C., Xu, T., Zhang, X. (2020). Integrated Wound Recognition in Bandages for Intelligent Treatment. *Adv. Healthcare Mater.*, 9, 2000941. <https://doi.org/10.1002/adhm.202000941>
- [19] Thali, M. J., Braun, M., Dirnhofer, R. (2003). Optical 3D surface digitizing in forensic medicine: 3D documentation of skin and bone injuries. *Forensic Sci. Int.*, 137 (2–3), 203–208. <https://doi:10.1016/j.forsciint.2003.07.009>