Automatic size measurement and boundary tracing of wound on a mobile device

Cheng-Hsien Huang, Sing-Da Jhan, Cheng-Hsuan Lin, Wei-Min Liu, *Member, IEEE*
Dept. of Computer Science and Information Engineering,
National Chung Cheng University, Chia-Yi County, Taiwan

***Abstract*--** **In the current clinical environment, most preliminary wound assessment methods from the caregiver are time-consuming and less accurate, and they are still performed visually or manually. Although some custom-designed software tools may exist, many of them were built on commercialized platforms, whose cost is higher than an open source one. We implemented a python-based user interface according to the current wound measurement workflow. It can be more conducive for nursing staff or the general public to use.**

I. INTRODUCTION

In the current clinical environment, most wound assessment are still performed visually or manually with time-consuming, subjective, and inefficient methods. Fig. 1 shows an example of such measurement. Although some software tools may exist or have similar capabilities, such as ImageJ, Amira-Avizo, MIPAR, …etc, most of them are commercialized or built on commercialized developing platforms, and therefore are less flexible. We implemented a python-based user interface (Fig. 2) particularly designed for measuring wound area. It provides several necessary image processing methods for wound assessment. An innovative function is that a user can provide an object with known area as a reference to calculate the pixel size. After going through three preprocessing steps, white balance [6], anti-glare [1,4], and CLAHE [7], and two segmentation algorithms, level set [3] and snake model [2], the program will automatically frame the wound and calculate the area.

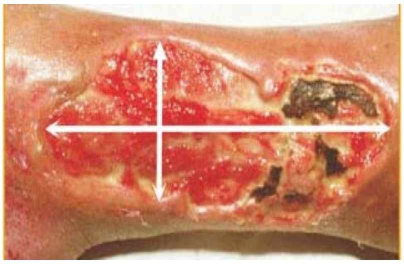
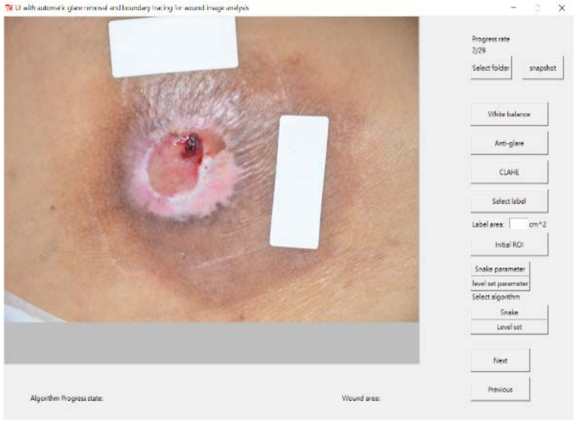


Fig.1. A rough wound size measurement [13]

II. LITERATURE SURVEY

The work from Loizou *et al.* [11] used different texture features and geometrical measures to quantify the rate of wound healing. To begin with, they normalized the intensities of the grayscale wound image to increase the contrast. Then they calculated a threshold value, and binarized the wound image with thresholding to locate the wound area. Finally its boundary was dilated as an initial contour of the snake model to further smooth its trace.



Progress rate
2/29

Label area: cm^2

Algorithm Progress state:

Wound area:

Fig.2.Our UI for wound image analysis

Duckworth *et al.* [10] designed a clinically affordable non-contact wound measurement device. People can evaluate the wound area immediately. The device takes a picture on wound and transfer image to cellphone via Bluetooth. A user draws the initial contour on his phone. Then an iterative edge detection algorithm will find the appropriate boundary for the wound, and its area can be derived.

Anti-glare is usually a necessary preprocessing procedure. Without it the boundary tracing work on a reflective wet wound could be erroneous. Miyazaki *et al.* [12] transformed image from RGB space to M space, used thresholding techniques to define the reflection area, and reconstructed it by referring its neighborhood.

Although some algorithms, software, and devices for wound image analysis were invented, very few of them allow the user to use a reference label with known size, and compare the number of pixels between wound area and the reference label to derive the wound area in metric unit. We believe

III. IMAGE PREPRCESSING AND SEGMENTATION

In our UI, users can input any image they want to process by loading it from a folder or taking a snapshot from the lens of the device they are using. After the image is taken, the next step is image preprocessing and segmentation.

A. Preprocessing

There are three optional image preprocessing methods, white balance, anti-glare, and CLAHE. Wound image after preprocessing will spend less calculation time for image segmentation and have more accurate result.

1. *White balance*

The step firstly calculates the histograms for each of RGB channels of the image, removes 5% of the brightest and darkest parts of the image. Then, it rescales the remaining 95% such that the brightest pixels correspond to 255 and the darkest ones correspond to zero. If there is a lot of noise in the background, white balance should be performed at specific area to avoid influence from irrelevant area.

2. Anti-glare

The step applies Otsu's method [1] to find the threshold automatically and binarize the picture. Next, the dilation operation is performed on the white part of the specific area to find out all the possible reflection places. These places will be processed by an inpaint operation proposed by Telea [8] for anti-glare reconstruction.

3. Contrast-limited adaptive histogram equalization (CLAHE)

The CLAHE algorithm divides the image into many blocks and performs local histogram equalization (LHE) individually. we compare the intensity of each pixel with a population threshold, and uniformly distribute those pixels above the threshold to the rest of the histogram before applying LHE. After LHE the bilinear interpolation is performed to remove artifacts near the borders.

B. Segmentation

After image preprocessing, user will have to input some initial conditions, and choose one of the algorithms, level set or snake, to start image segmentation.

1. Level set

Given a rectangle region of interest (ROI) in the wound image as an initial mask, the level set algorithm compares all the edge pixels of the mask with their two neighboring pixels (one inside the mask and the other outside the mask) respectively in each iteration. If the difference between the gradient of the edge pixel and one of its neighboring pixels is less than a threshold, the neighboring pixel will replace the original edge pixel and become a new one.

When a series of iterations do not result in significant change in the mask, it means the level set algorithm has found the boundary of the wound.

2. Snake

Given a rectangle region of interest (ROI) in the wound image as an initial mask, the Snakes model algorithm [2,9] defines an energy function of the image, look inward along the edge of the object until the trace minimizes the function.

IV. RESULT AND CONCLUSIONS

Due to the page limitation, we skipped the results of each preprocessing. Fig. 3 showed the boundary tracing results of the wound and the reference label using Snake model to demonstrate the efficacy of our UI in wound size assessment.

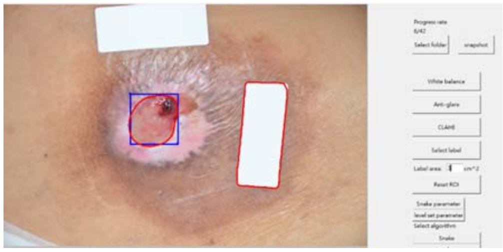


Fig. 3. Wound tracing by Snake model

Our UI and image processing algorithms are all python-based, so it is convenient to integrate both parts. A user can try several times to find the most suitable algorithm for their wound image. Unexpected results may appear in some of the blurrier images, and that is why we need white balance, anti-glare, and CLAHE to perform optional preprocessing. It is still possible that Level set and Snake model will not do well even after such preprocessing, but at least the general outline of the wound could be delineated automatically.

Currently the UI can be installed on Window-based mobile devices. We plan to re-package the UI as an APP to make it available on the Android-based devices. The problem to be solved is that the camera resolution on different mobile devices could vary a lot. Over-high resolution may bring up more noise in the image and also slow down the calculation. On the contrary, the other way may cause miscalculation due to less edge information.

ACKNOWLEDGMENT

We appreciate the support from the nursing department, Taichung Veterans General Hospital to retrieval the wound image data, and the funding support from Ministry of Science and Technology (106-2221-E-194 -036 -), Taiwan.

REFERENCES

[1] Liu Jianzhuang, et al., Automatic Thresholding of Gray-Level Pictures Using Two-Dimensional Otsu Method, China 1991 International Conference on Circuits and Systems

[2] http://scikit-image.org/docs/dev/auto-_examples/edges/plot_active_countours.html

[3] https://github.com/kevin-keraudren/chanvese

[4] https://docs.opencv.org/3.3.1/df/d3d/tutorial_py_inpainting.html

[5] http://blog.rhesoft.com/2014/07/19/python-for-android-tutorial-2-taking-a-picture/

[6] http://pillow.readthedocs.io/en/3.1.x/reference/ImageOps.html

[7] https://opencv-python-tutroals.readthedocs.io/en/latest/

[8] Alexandru Telea, An Image Inpainting Technique Based on the Fast Marching Method, Journal of Graphics Tools, Vol. 9, No. 1: 25-36

[9] Michael Kass, et al., Snakes: Active Contour Models, Interational Journal of Computer Vision, 321-331(1988)

[10] Mark Duckworth, et al., A Clinically Affordable Non-Contact Wound Measurement Device, 2008, Rehabilitation Engineering and Assistive Technology Society of North America

[11] Christos P. Loizou, et al., Evaluation of wound healing process based on texture image analysis, 2013, Journal of Biomedical Graphics and Computing, Vol. 3, No. 3

[12] Daisuke Miyazaki, et al., Polarization-based Inverse Rendering from a Single View, in Proceedings of International Conference on Computer Vision, pp.982-987, Nice, France, 2003.10

[13] http://www.bimeco.org/Software.html