

# Wound Image Analysis System for Diabetics

Lei Wang<sup>\*a</sup>, Peder C. Pedersen<sup>a</sup>, Diane Strong<sup>b</sup>, Bengisu Tulu<sup>b</sup>, Emmanuel Agu<sup>c</sup>,  
Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, 100 Institute  
Road, MA, USA 01609; <sup>b</sup>School of Business, Worcester Polytechnic Institute, 100 Institute Road,  
MA, USA 01609; <sup>c</sup>Department of Computer Science, Worcester Polytechnic Institute, 100 Institute  
Road, MA, USA 01609;

## ABSTRACT

Diabetic foot ulcers represent a significant health issue, and daily wound care is necessary for wound healing to occur. The goal of this research is to create a smart phone based wound image analysis system for people with diabetes to track the healing process of chronic ulcers and wounds. This system has been implemented on an Android smart phone in collaboration with a PC (or embedded PC). The wound image is captured by the smart phone camera and transmitted to the PC via Wi-Fi for image processing. The PC converts the JPEG image to bitmap format, then performs boundary segmentation on the wound in the image. The segmentation is done with a particular implementation of the level set algorithm, the *distance regularized level set evolution* (DRLSE) method, which eliminates the need for re-initialization of level set function. Next, an assessment of the wound healing is performed with color segmentation within the boundaries of the wound image, by applying the K-Mean color clustering algorithm based on the red-yellow-black (RYB) evaluation model. Finally, the results are re-formatted to JPEG, transmitted back to the smart phone and displayed. To accelerate the wound image segmentation, we have implemented the DRLSE method on the GPU & CPU cooperative hardware platform in data-parallel mode, which has greatly improved the computational efficiency. Processing wound images acquired from UMASS Medical Center has demonstrated that the wound image analysis system provides accurate wounds area determination and color segmentation. For all wound images of size around 640 x 480, with complicated wound boundaries, the wound analysis consumed max 3s, which is 5 times faster than the same algorithm running on the CPU alone.

**Keywords:** Diabetics, Level Set, K-Mean, Smartphone & PC, GPU & CPU

## 1. INTRODUCTION

For individuals with type 2 diabetes, foot ulcers represent a significant health issue affecting 5-6 million in the US [1-4]. Such ulcers are painful, susceptible to infection, persistent and slow to heal [5]. According to published reports [6], there were approximately 71,000 diabetes related lower limb amputations in the U.S. in 2004, constituting the primary cause of non-traumatic lower limb amputations. Even for successful healing, the cost for healing diabetic foot ulcers is estimated at \$15,000 per year per individual [7]. Overall diabetes-related health care cost was estimated at \$174 billion in 2007 and is expected to increase in the coming years [7].

Good daily care can improve wound healing, reduce the likelihood of amputations, minimize complications related to the disease and ulcers and reduce the overall health care cost. Nowadays, the standard foot ulcer treatment for patients with diabetes is a combination of daily self-care and physician examination during regular visits to a wound clinic. The wound is examined by experienced physicians, and the healing status compared to that of previous visits, based on which adjustments to the treatment strategy, if any, are made. This approach does little to encourage direct involvement by the patient in the overall wound care.

On the other hand, according to the Institute of Medicine [8], information technology (IT) enables patients to be active participants in their own care and play an important role in the wound management process. For example, the Sihouette product suite produced by ARANZ Medical [9] is capable of imaging and measuring the size of a wound. Moreover, it can integrate wound images with other essential data to provide an accurate and complete record of the assessment to track wound healing progression. Another example is the MOWA [10], which is an Android based software application for smartphones and tablets. The software differentiates types of tissues found in a pressure ulcer. The analysis can be performed on photos taken with the on-board camera or on uploaded pictures from other sources. Manual assistance,

however, is required to enable it to calculate the area of the lesion. The software can provide suggestions about treatments.

These product development efforts indicate that IT devices, such as PCs, tablets and even smartphones, might play a role in the daily wound care. The computational and data storage resources for current PCs and laptops are capable of efficient image processing including wound segmentation, area determination and color separation. While currently limited in memory and CPU power, smart phones with high-resolution cameras may eventually become ideal devices for image processing. The challenge lies in implementing the image analysis algorithms so efficiently that the analysis results are completed within a couple of seconds. Therefore, optimized implementation of highly efficient processing algorithms and establishment of a powerful hardware platform are both required.

In this paper, we present a novel wound image analysis system, utilizing the distance regularization level set evolution for wound boundary determination and the K-mean color segmentation algorithm for assessing healing status. We have implemented the DRLSE algorithm as a narrow-band restricted version based on both the GPU and CPU. After the wound boundary is determined, the wound area is further analyzed based on the RYB wound evaluation model [11] [12]. We use the fast K-mean algorithm to segment the wound area based on color and the size of each type of tissue is computed. In addition, we designed and constructed an image capture box to simplify the image capture process of wounds on the sole of the foot where diabetic foot ulcers most commonly occur. The image processing takes place as a collaboration between a smart phone and a modern low-cost PC. The data communication between these two devices is based on Wi-Fi technology.

The paper is organized as follows. In Section 2, the DRLSE algorithm is briefly reviewed. The overall wound image processing method and the system structure is introduced in Section 3. The hardware system is described in Section 4, while the experimental results are presented in Section 5. Finally, we summarize our work in Section 6.

## 2. LEVEL SET BASED SEGMENTAION

### 2.1 Theoretical Formulation

Wound area segmentation is a key element in the image analysis because it provides the foundation for the subsequent processes. The segmentation algorithm must perform well for a wide range of wound images. We chose the level set based algorithm due to the relatively fine resolution that can be achieved, the handling of slightly tilted lines and corners, and the precise and easy calculation of surface normal vectors [13-16]. Furthermore, because joining surfaces are handled implicitly, the wound contour is self-adaptive to the topological change during the evolution. This means that the implicit contour accurately follows shapes that change topology, for example when a shape splits into two, develops holes, or the reverse of these operations. This property of level set algorithm allows direct numerical computation on image grids. Because of these features, level set based algorithms have been widely used in image segmentation.

The original level set methods for capturing dynamic contours was introduced by Osher and Sethian [13]. The basic idea is to use the zero level set of a higher dimensional function to represent the target contour. This higher dimensional function is called the *level set function* (LSF)  $\phi(x, y, t)$ . In this case, finding a contour is accomplished by the evolution of the LSF from a random initial state to convergence on the target boundary. Caselles and Malladi [14] [15] first brought the level set idea to image processing and computer vision research. The most basic curve evolution equation is presented in a partial differential equation (PDE) as given in eq. (1).

$$\frac{\partial \phi}{\partial t} = -F |\nabla \phi| \quad (1)$$

where  $F$  is the speed function that controls the motion of the contour and  $\nabla$  is the gradient operator. We assume that the curve movement is always proportional to  $F$  in a direction normal to the curve itself. In image segmentation case, the speed function  $F$  is formulated as in eq. (2).

$$F = -g_I(F_A + F_G) \quad (2)$$

where  $F_A$  causes the curve to uniformly expand or contract depending on the local geometry information.  $F_G$  is a function of local curvature, which keeps the propagating curve smooth and conforms to the edge distribution of the original image. The term  $g_I$  is a stopping term defined in eq. (3).

$$g_I = \frac{1}{1 + |\nabla(G_o \otimes I(x, y))|^2} \quad (3)$$

where  $G_o \otimes I(x, y)$  filters the original image by a Gaussian smoothing filter to minimize noise.  $g_I$  is maximized when the gradient value is smallest, which means the  $F$  is completely performed on the curve when no edge is detected. The gradient value increases greatly when the current contour is located in the vicinity of the boundaries, in which case  $g_I$  is approximating to 0. This cancels the influence of  $F$  and then the evolution of the contour will be stopped [16].

## 2.2 Distance Regulation Level Set Evolution (DRLSE)

The conventional level set methods suffer from the irregular boundary determination (*i.e.*, the detected boundary may be irregular and misshapen [17], and the LSF may lose the signed distance property ( $|\nabla \phi|=1$ ), which is the basis for the successful evolution) [17] [18]. Hence, after each iteration we have to reinitialize the LSF. This re-initialization leads to a large computation burden and can reduce the accuracy of the zero level set evolution as well.

Li et al. [18] proposed a novel evolution function, which includes a distance-regularization term. This forces the LSF to maintain a desired shape and implies that the signed distance property of the LSF is preserved during the entire evolution process [18]. The evolution equation is modified from the gradient operation in eq. (1) to the expression in eq. (4):

$$\frac{\partial \phi}{\partial t} = \mu \mathfrak{R}(p, \phi) + \lambda L(g, \phi) + \alpha A(g, \phi) \quad (4)$$

where the term  $\mathfrak{R}(p, \phi)$  is the distance-regularization term mentioned above,  $L(g, \phi)$  is minimized when the zero level contour is located at the object boundary, in other words,  $L(g, \phi)$  is the stopping term. The term  $A(g, \phi)$  is used to speed up the motion of the zero level set contour in the evolution process, while  $p$  is the potential function and  $g$  is defined as in eq. (3). The controlling parameters  $\alpha, \lambda$  and  $\mu$  are set by users. Details regarding the function definitions are given in [18].

Finally, we formulate the numerical calculation of eq. (4) with a simple finite difference scheme

$$\phi^{k+1}_{i,j} = \phi^k_{i,j} + \Delta t L(\phi^k_{i,j}) \quad (5)$$

where  $k$  represents the iteration times,  $\Delta t$  is the time step,  $L(\phi^k_{i,j})$  represents the right side of eq. (4) while replacing  $\phi$  with  $\phi^k_{i,j}$ , which denotes the LSF value at iteration time  $k$  on the position  $(i,j)$  in the image grid. As is commonly known, applying the finite difference scheme to the discretization of the PDE like eq. (4) is justified. To reduce the computational demands, a narrow-band level set algorithm has been proposed to restrict the computation of the level set evolution to a narrow neighborhood area around the zero level set (usually an 8-neighbor area surrounding the zero crossing points) [19]. According to Li et al. [18], the narrow-band implementation of the model expressed in eq. (5) allows the use of a large time step, which will further speed up the evolution process.

Based on the discussion provided by Li et al. in [18], the algorithm structure can be summarized as follows:

- **Step 1: Initialize** the curve and LSF as a binary step function, which has been set according to different locations ( $c$  for the pixels within the initial curve,  $-c$  outside the curve and  $0$  for the pixels on the curve, where  $c$  is a random positive integer and is set to 2 in our case). Then, construct the initial narrowband region;
- **Step 2: Update the LSF** as  $\phi^{k+1}_{i,j} = \phi^k_{i,j} + \Delta t L(\phi^k_{i,j})$  only over the narrow band area;
- **Step 3: Update the narrow-band** based on the new zero-crossing points on the previous defined narrow band area;

- **Step 4: Assign values to new pixels in narrowband**, if  $\phi_{i,j}^k > 0$ , set  $\phi_{i,j}^k$  to  $h$ , otherwise set it to  $-h$ , where  $h$  is a constant with a default value 3 (the same value of the width of the neighborhood area);
- **Step 5: Check the termination condition**. If it is satisfied, quit and return the curve. Otherwise, return to step 2.

### 3. WOUND IMAGE ANALYSIS SYSTEM

#### 3.1 Wound Image Analysis Software System

In our wound image processing, we have two goals. The first goal is to find the wound area accurately, and the second is to analyze the wound area based on some evaluation models. In our case, the wound boundary is determined by an efficient implementation of the DRLSE algorithm. Then the wound area is segmented into different tissues based on color. In our proposed implementation, there are six software-based modules as shown in Figure 1. This section will discuss the function of the modules in Figure 1 in detail.

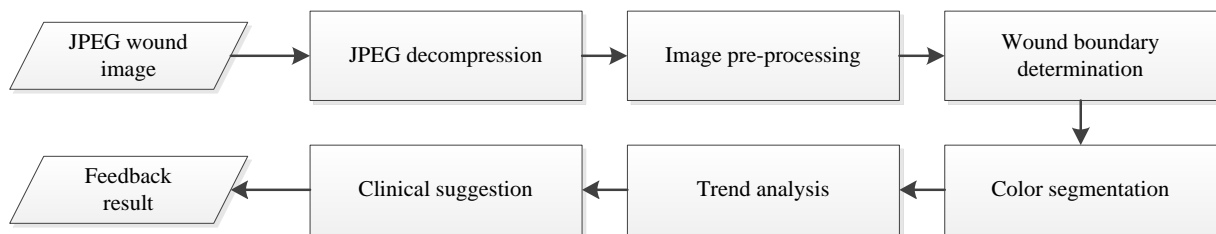


Figure 1. Wound image analysis software system

##### 3.1.1 Image pre-processing

The wound image is captured by the smart phone's camera and stored in JPEG format. Therefore, the first step is to decompress the image into bitmap format, followed by pre-processing prior to edge (wound boundary) detection: The wound image is smoothed by a Gaussian operator for noise reduction, followed by edge enhancement to provide a well-defined and continuous edge information. In our system, we use a hue-preserving enhancement algorithm [20], which increases both the intensity and saturation while retaining the hue information for all pixels. Due to the RGB-CMY color model changing step, the Gamut problem (the problem caused by enhanced color component values exceeding the allowable maximum value such as 255 for 8-bit deep image) will be prevented [20]. As the last step, an effective edge detector is performed on the enhanced image to obtain the binary image with the edge pixel set to 255 and the remaining pixels set to 0. This binary image is the input to the subsequent level set based boundary determination module. We choose the Canny edge detector [21] which will provide us a binary image with edge points marked as 1.

##### 3.1.2 Wound boundary determination

The narrow-band based DRLSE algorithm was introduced in Section 2. Building on the previous discussion, the implementation details of the candidate algorithm for wound boundary determination are discussed here. The flowchart of the wound boundary determination process is shown in Figure 2.

The DRLSE can be implemented with a simple finite difference scheme on the image grid [18]. The evolution equation is shown as eq. (5). By iteratively updating the level set function, the implicit contour of the level set function will converge towards the actual wound boundary from its initial position.

Given the complicated nature of typical wounds, the level set evolution process could be prematurely stopped by an internal boundary within the wound area if we were to place the initial curve inside the wound area and expanded it gradually. To avoid this problem, we choose to set the initial curve on the healthy skin area and make sure that it encloses the entire wound area. As the level set evolution continues, the initial curve will keep contracting and approach the actual wound boundary with little resistance, since the healthy skin color is much more uniform than the wound area. In our case, a simple skin detection method [22] based on color will be employed to locate the skin area. Afterwards, the initial curve will be set as the boundary of the skin area.

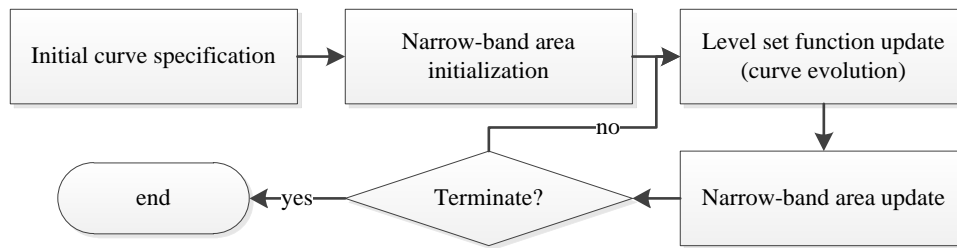


Figure 2. Flowchart of the wound boundary determination based on DRLSE algorithm

As discussed in [18], the narrow band area is defined as the union of the 3x3 neighborhoods of the zero crossing points of the level set function. We call a grid point  $(i, j)$  a zero crossing point if either  $\phi_{i-1, j}$  and  $\phi_{i+1, j}$  or  $\phi_{i, j-1}$  and  $\phi_{i, j+1}$  are of opposite signs [18]. After each update of the level set function, the narrow band area will be re-calculated accordingly.

After each iteration, we have to decide whether to stop the evolution process based on some rules. In our case, we will make this decision by checking the evolution rate  $R$  of the level set function defined as in eq. (6)

$$R = \frac{|N_k - N_{k-1}|}{N_{k-1} + 1} \quad (6)$$

where  $N_k$  represents the number of grid points (pixels) located in the implicit contour of the current level set function,  $N_{k-1}$  represent the same number in the previous iteration. Hence, the numerator represents the absolute change of the level set function. Denominating it by  $N_{k-1}$ , we can obtain the changing rate of the level set function after the current update. The reason to add 1 to  $N_{k-1}$  is to ensure that the denominator is never zero. In our case, we consider a change rate smaller than 0.001 as a sign of convergence, and we use a change rate of 0.001 as the criterion for stopping the evolution process.

The related results of the wound boundary determination are shown in Section 4.

### 3.1.3 Color segmentation

After the boundary of the wound area has been determined, color segmentation is performed based on clinical criteria. The red-yellow-black (RYB) model [11] is used, which requires an efficient color clustering algorithm to identify the color category of each pixel in the wound area. The RYB system classifies the wound as red, yellow, or black tissues, respectively, which represent the different phases of the tissue on the continuum of the wound healing process. Specifically, red tissues are viewed as the inflammatory (reaction) phase, the proliferation (regeneration) phase or the maturation (remodeling) phase. Yellow tissues typically represent infected tissue or tissue containing slough, which has not begun to heal. Finally, black tissues contain necrotic tissue.

Based on the RYB wound evaluation model, the wound analysis task consists of clustering the pixels within the wound boundary into certain color categories centered around the corresponding standard color values. Therefore, all classical clustering methods can in principle be applied to solve this task. For this research a fast clustering algorithm called K-mean has been selected [23]. In data mining,  $K$ -mean clustering is a method of cluster analysis, which aims to partition  $n$  observations into  $k$  clusters, in which each observation belongs to the cluster with the nearest mean [24]. In our case, all the pixels within the wound boundary can be viewed as observations. The three colors referred in RYB model are regarded as cluster centers. The color segmentation process is shown in Figure 3.

To perform color segmentation, several operational parameters need to be specified. At first, the color difference between the color cluster center and the target pixel is calculated by the standard color difference in the CIE Lab model [25]. Secondly, in practice we specify the initial centers according to some empirical values such as the Macbeth Color Checker table provided in [26] rather than leave them to be randomly chosen. Finally, the termination condition is fulfilled if the new cluster center assignment is the same as the previous updating result. By this operation, the

converging speed will be increased, making the color clustering process more efficient. Based on experimental results, all test images are segmented well enough with at most 5 iterations.

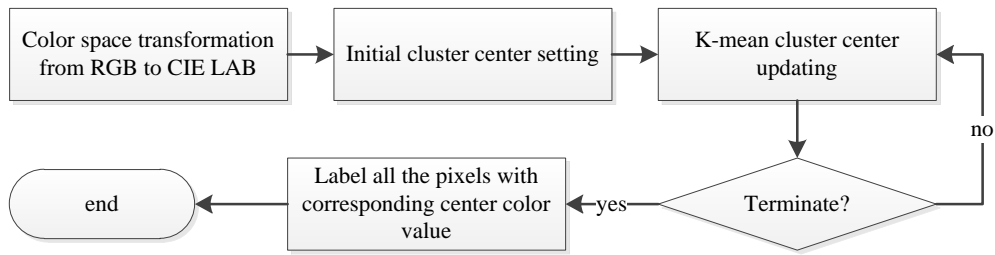


Figure 3. K-mean color segmentation flowchart

## 4. SMART PHONE & PC COLLABORATIVE IMPLEMENTATION

### 4.1 Smart phone & PC cooperative system

The experimental results show that the wound image processing time on current smart phones alone is too long for practical use. Therefore, we chose a solution where the processing is performed on an embedded PC, which provides the needed computational capability. Thus, the PC and smart phone constitute a collaborative implementation. The basic structure for this collaborative system is shown as Figure 4.

Since the wound analysis system contains two computational components, the communication scheme is important. In this system, we selected a communication scheme in the application layer based on HTTP protocol, which is the most frequently used protocol in the application layer in TCP/IP network protocol model, rather than the common method of manipulating the transport layer by socket programming based on TCP/IP protocol. The reason is that the operation in the highest layer (application layer) in the TCP/IP model will be more generalized and can be applied to different settings in lower layers of this network model. In this case, we do not attempt to determine which communication protocol the two systems in the transport and internet layer use.

The smartphone plays the role of client in the communication scheme. It accomplishes the following tasks sequentially: 1) acquire the wound image and save on the specified directory on the SD card; 2) make a request to the server (PC) and send the JPEG file of the wound image to the PC; and 3) receive the processed image file sent back from the PC and display it on the screen. The PC is viewed as the server, which: 1) listens to the requests of the client and receives the image file from the client, 2) performs the wound image analysis, and 3) sends the JPEG file of the processed image back to the smartphone (client) as a response. The server program on the PC runs continuously and listens for a request from the client process on a smartphone, which means that any smartphones getting access to the Wi-Fi are able to send requests to the server. Hence, this communication scheme is able to handle multi-client and a single server situation when several smartphones are attempting to connect to the server at the same time with some conflict-avoidance schemes, such as the TDM (time division multiplexer which allocates each client a unique slot to communicate with the server).

### 4.2 GPU & CPU collaborative implementation of the level set algorithm

In spite of the narrow-band optimization, the level set method is still too computationally demanding. Hence, we searched for strategies that permit the boundary detection of the wound image to be carried out in an acceptably short time. In our system, we implement the iterative LSF updating, which is the most computationally demanding part of the level set algorithm, on a GPU (Graphic Processing Unit), which is available on most of today's PCs and laptops. The major difference between CPU and GPU implementation is that the latter supports parallel computation to a much greater extent. Unlike the traditional, sequential way of programming on CPUs, a GPU's programs deal with complex parallel reductions to best utilize the available parallel resources [27].

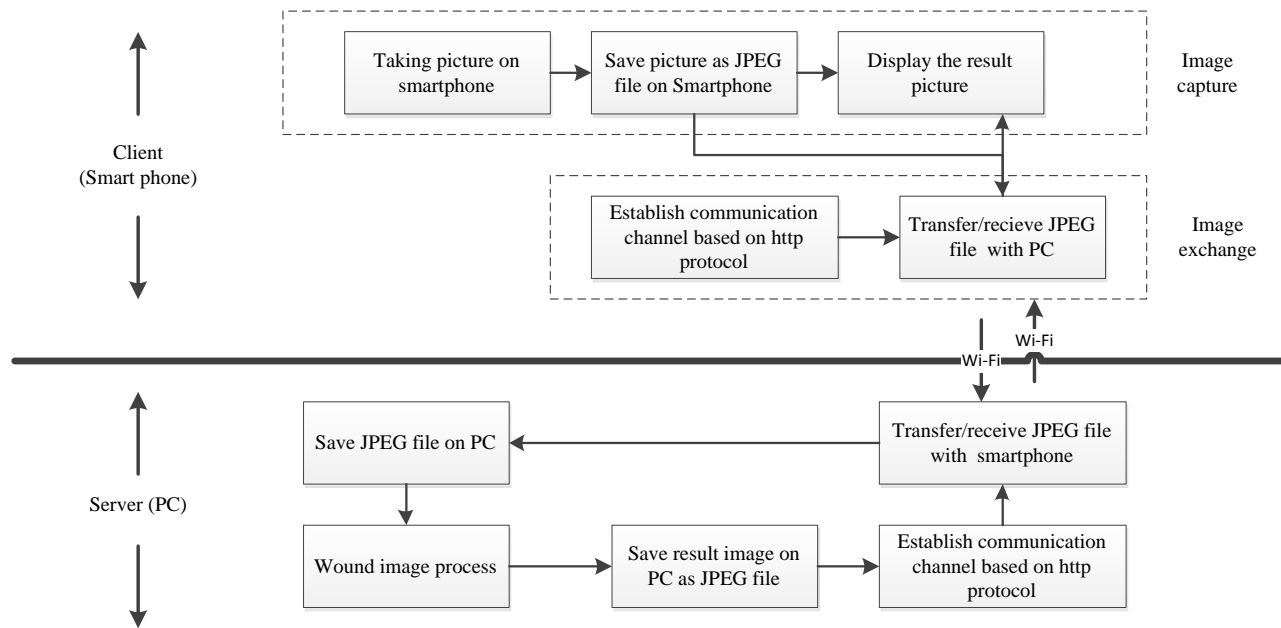


Figure 4 Smartphone & PC collaborative system

The traditional narrow band method avoids unnecessary computation by only updating field elements near the level set curve. However, according to [28], even the computations near the level set curve can be avoided in regions where the level set field has locally converged. This observation motivates the method of tracking the active computational domain according to both the temporal and spatial derivatives of the level set field. Afterwards, the level set updating only happens on the pixels belonging to the specified active computational domain [28]. The active domain is updated before every iteration. The original version of this algorithm is based on the 3D image situation [28], but we have transformed it to operate in the 2D wound image situation.

In our system, the GPU programming is implemented with Open Computing Language (OpenCL) which is a framework for writing programs that execute across heterogeneous platforms, such as CPUs and GPUs. In our case, we view each pixel in the current active computational domain as the basic processing unit (called ‘work item’ in OpenCL). Then we can maximize the usage of all the streaming cores at the same time by running a kernel (similar to the concept of function in C) on as many units as possible at the same time. The programming flow chart for modified level set algorithm based on OpenCL is shown in Figure 5. We can see from this figure that the host program is running on a CPU including the function of target image reading, edge calculation, platforms and memory initialization. On the other hand, the actual level set algorithm is encapsulated in kernels running on a GPU. All the kernels will process only one pixel at a time. In addition, the intermediate processing results of kernels will be stored in global memory on the GPU and will not be read back to the host until the end of the algorithm. In this way, the memory access latency from different platforms can be reduced as much as possible.

## 5. IMAGE CAPTURE FOOT BOX

Most foot ulcers happen on the sole of the foot. Due to the physical limitation, it is difficult for a patient with diabetes to take pictures of such wounds without assistance. To facilitate the image capture process, we have designed an image capture device with the structure shown as Figure 6(a). On top is a plate made from wood. This top plate is divided into two connected parts, where the first part has a rectangular opening of the size of an adult human foot, on which the foot is placed. Across this opening, there is a support bar that allows the foot to rest comfortably. The second part allows the smartphone to be placed in a precise location with holes for camera lens and flash. There is an angle between the two parts to allow the image capture. Below the top plate is another plate with an up-facing mirror, angled slightly so that the flash will illuminate the sole of the foot and so that image of the sole of the foot is captured by the camera.

When using this device, the patient places his/her foot on the top plate and is able to observe the wound through the smartphone camera. After ensuring a good image of the sole of the foot, the patient takes a picture of the wound by a voice command like “cheese” or “take a photo” to the smartphone. The captured image is saved on the phone automatically and transmitted to the PC for analysis.

The dimension of both plates is determined empirically from the size of adult human feet and the dimensions of android smartphones. The distance between the two plates is determined by the simple optical rules of the lens and the parameters of the camera. For the top plate, we set 12.5” x 6.5” for the part on which to place the foot and 12.5” x 5” for the part on which to place the smartphone. The size of the bottom plate is set as 12.5” x 10.5”. The distance between the foot part on the top plate and the lower plate is set to be 5”. The angle between the two parts on the top plate is set to be 57°. Accordingly, the angle between the mirror and the lower plate is set as 33°.

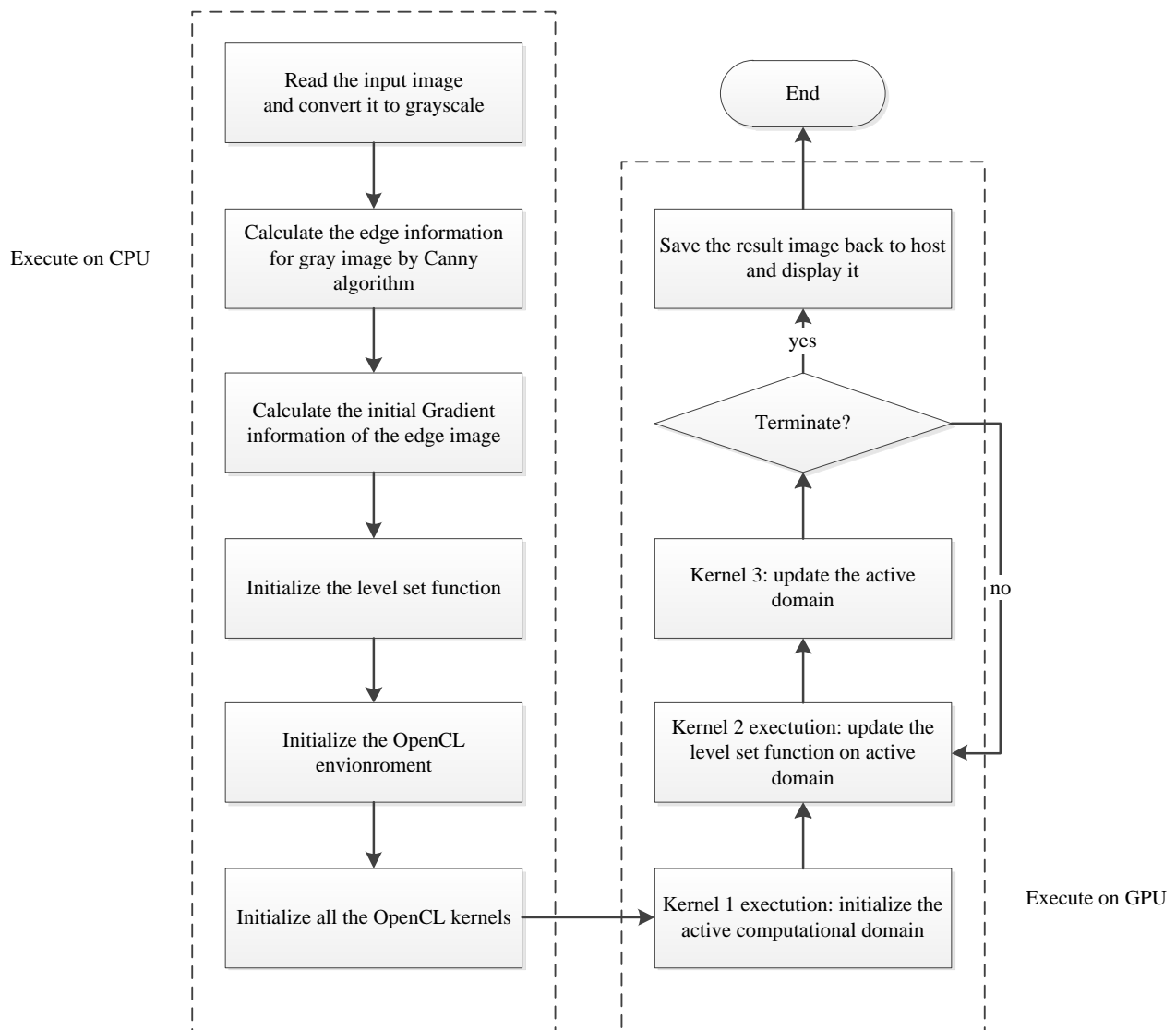


Figure 5 Programming flowchart of level set algorithm in OpenCL



A photo of the actual image capture foot box is shown in Figure 6 (a), and a sample image of a foot with a simulated wound is shown in Figure 6 (b). We can see from this figure that the entire foot is correctly imaged by the smartphone camera.



Figure 6. Design of image capture foot box. (a) photo of the actual foot box; note the smart phone; (b) sample image taken with the foot box.

## 6. EXPERIMENTAL RESULTS

### 6.1 Experimental environment

As mentioned in Section 3, the proposed wound image analysis system is based on a combined smartphone and embedded PC system. In our case, we use the Motorola Droid phone running Android OS 2.3 together with the Habey Embedded PC EPC-6567, which has a small dimension (7.87" x 8.86" x 1.77" (W x L x H)) and high computation capability (CPU dual-core 1.8GHz + ATI Radeon HD 6350 GPU). All the software is developed in JAVA with assistance of JAVACL library for GPU programming, Android 2.3 toolkit for smartphone programming and JAVA servlet package for network programming.

### 6.2 Wound image analysis result

We have tested our algorithm on sample wound images obtained from patients at UMASS Medical Center, without any attributing information; four such images are shown in Figure 7. In practice, we down sample the original wound images by a factor 0.25 in both horizontal and vertical directions. The size of down sampled images is around 640 x 480, which turns out to give sufficient resolution for wound analysis.

The wound boundary determination result is shown in Figure 8. The detected wound area is marked by blue color in all result images. We can see from these results that the DRLSE algorithm performs very well on sample image (a) and (b) since the skin color in these images are normal and uniform. The algorithm can accurately capture the wound contours. For the last two sample images, however, the segmentation results contain some small false wound area surrounding the real target. This is easy to understand, as the skin areas in these two images are not flattened and contain a lot of outlier edge information, which may unexpectedly stop the LSF evolution.

The computation time for the accelerated DRLSE algorithm are shown as in Table 1. We can see that the time efficiency is enhanced by about a factor 5 with CPU + GGPU implementation. Hence, we may assume that the enhanced DRLSE algorithm basically satisfies the time requirements for practical wound analysis.



Figure 7. Sample images from wound clinic center in UMASS

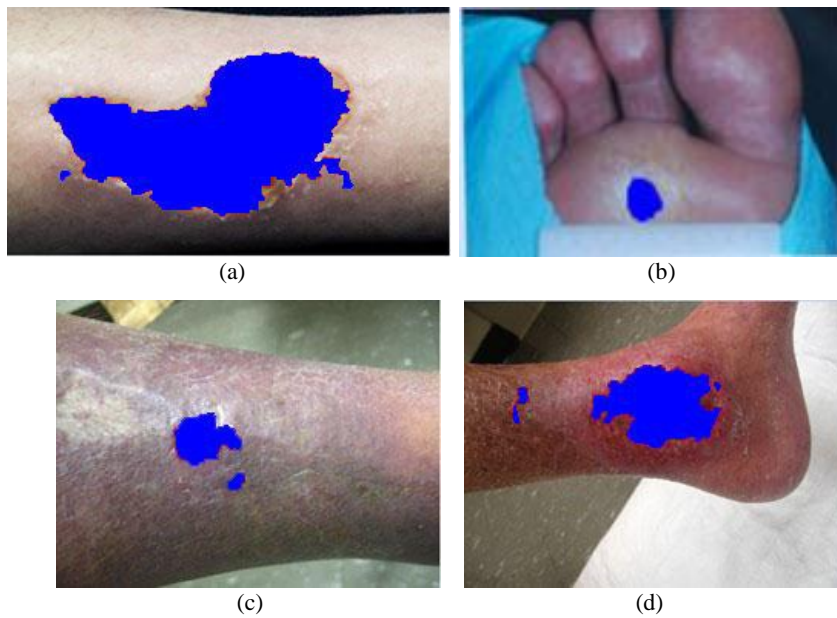


Figure 8. Wound boundary determination results.

Table 1: Efficiency Evaluation

Image No.	(a)	(b)	(c)	(d)
Computing time (s) with GPU & CPU	3.1	2.5	2.2	2.8
Computing time (s) with CPU only	15.2	11.9	13.2	12.4

### 6.3 Color segmentation results

To explicitly illustrate the color segmentation results, we chose two wound images whose colors indicate different types of wound tissues according to the RYB wound model as shown in Figure 9 (a) and (b). The color segmentation results are shown in Figure 9 (c) and (d). In the result images, each pixel in the wound area is marked by the corresponding standard color of the category to which it belongs (red, yellow or black).

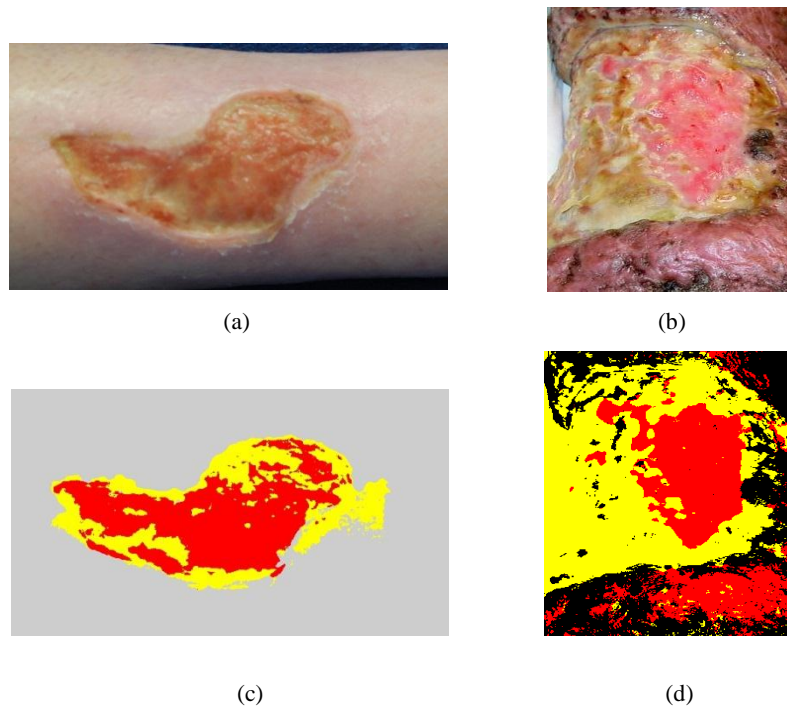


Figure 9. Color segmentation results. (a) original wound image 01; (b) original wound image 02; (c) color separated wound image 01; (d) color separated wound image 02.

Based on the results seen in the Figure 9 (c) and (d), the color segmentation looks promising. Most pixels within the wound area are classified to the color corresponding to the observed color despite several misclassified points, which always are located at the mixed area of tissues of different colors. The computation time for the fast K-mean algorithm is less than one second for the two wound images. Experienced clinicians will be consulted to further evaluate the results above.

### 6.4 Test result for the entire system

Finally, we test the entire wound image analysis system. In this trial, we send a sample wound image file from the smartphone to the PC by Wi-Fi and determine the wound boundary on the PC. Finally we transmit the result image file back to the smartphone and display it on the screen. The area determination results are shown in Figure 10, and they appear to be correct. Note that the target image used in this test consists of one original wound and another two manually added wounds. The purpose of using this image is to illustrate the good robustness of the level set algorithm against topological change.



(a) (b)  
Figure 10. Test result of the entire system (a) the original image; (b) the result image

The whole process including transmission and processing takes no more than 4 seconds, which is acceptable for a real-time system.

## 7. CONCLUSION

We have designed a wound image analysis system, based on a smart phone and PC collaborative hardware platform. In the system, the DRLSE algorithm was applied to the image to determine the boundaries for all wound areas and measure the size of each wound by pixel. To achieve further efficiency improvement, the DRLSE algorithm was parallelized and implemented on a GPU. Finally, the wound areas were classified to different color tissues based on RYB wound evaluation model. Based on evaluation of 7 sample wound images, we found that the DRLSE algorithm provides very promising segmentation results on ordinary wound images with uniform skin color. However, when the entire foot is infected and the skin color is complicated, the algorithm does not always find a well-defined boundary of the wound area. The total image analysis time, including segmentation time, does not exceed 3 seconds for all wound images tested, which is acceptable for current real-time image processing system. Finally, the K-mean clustering method works as expected on the color segmentation module.

In the future work, we are aiming for improving the efficiency of DRLSE algorithm implementation further by optimizing the memory allocation scheme on CPUs and GPUs. We plan to set an automatic parameter adjustment scheme of DRLSE algorithm in order to be adaptive to different wound image conditions. In addition, a trend analysis module will be designed and the wound image analysis result feedback format will be developed in order to give patients a clear understanding of the wound status and wound healing trend. Finally, we will involve experienced clinicians and patients with diabetes to further evaluate this system.

## ACKNOWLEDGEMENT

This research was supported by the National Science Foundation under Grant IIS-1065298. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

- [1] Buckley K.M., Adelson L.K. and Agazio J.G., "Reducing the Risks of Wound Consultation: Adding Digital Images to Verbal Reports," *Journal of Wound Ostomy & Continence Nursing*, 36(2): pp. 163-170 (2009).
- [2] Falanga , "The Chronic Wound: Impaired Healing and Solutions in the Context of Wound Bed Preparation," *Blood Cells, Molecules and Diseases*, 32(1): pp. 88-94(2004).

- [3] Hess C.T. and Kirsner R.S., "Orchestrating Wound Healing: Assessing and Preparing the wound bed," *Advances in Skin & Wound Care*, 16(5): pp. 246-257(2003).
- [4] Rees R.S. and Bashshur N., "The Effects of TeleWound Management on Use of Service and Financial Outcomes," *Telemedicine and e-Health*, 2013(6): pp. 663-674(2007).
- [5] National Institute of Health. NIH's National Diabetes Information ClearingHouse. Available from: [www.diabetes.niddk.nih.gov](http://www.diabetes.niddk.nih.gov).
- [6] Gruber D., "Consumer Engagement for Diabetes Technology: Easier Said Than Done" *Journal of Diabetes Science and Technology*, 4(3): pp. 754-758 (2010).
- [7] Glasgow R.E. and Anderson R.M., "In Diabetes Care, moving from compliance to adherence is not enough, Something Entirely Different is Needed," *Diabetes Care*, 22(12): pp. 2090-2091(1999).
- [8] Kohn, L.T., Corrigan, J.M., and Donaldson, M.S., "Crossing the quality chasm: a new health system for the 21st century health care services," Washington, DC: Committee on Quality of Health Care in America, Institute of Medicine, National Academy Press, 2001.
- [9] ARANZ Medical. Sihouette product suite. Available from <http://www.aranzmedical.com/solutions/silhouette-for-clinical-research/>.
- [10] HealthPath. MOWA: mobile wound analyzer. Available from <http://www.healthpath.it/mowa.html>.
- [11] Kransner D., "Wound Care How to Use the Red-Yellow-Black System," *the American Journal of Nursing*, 95 (5), pp. 44-47 (1995).
- [12] Mekkes J.R. and Westerhof W., "Image Processing in the Study of Wound Healing," *Journal of Clinic in Dermatology*, 13 (4), pp. 401-407 (1995).
- [13] Osher S., Sethian J., "Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulation," *Journal of Comput. Phys.*, 79(1), pp.12-49 (1988).
- [14] Caselles V., Catta F., Coll T. and Dibos F., "A geometric model for active contours," *Journal of Numberische Mathematik*, 66, pp.1-31(1993).
- [15] Malladi R., Sethian J.A. and Vemuri B.C., "Evolutionary Fronts for Topology Independent Shape Modeling and Recovery," *Proc. Of the 3<sup>rd</sup> ECCV, Stockholm, Sweden*, pp.3-13 (1994).
- [16] Caselles V., Kimmel R. and Sapiro G., "Geodesic Active Contour" *Journal of Comput. Vis.*, 22(1), pp.61-79 (1997).
- [17] Sethian J.A., "Level Set Methods and Fast Marching Methods," Cambridge, U.K.: Cambridge Univ. Press, 1999.
- [18] Li C.M., Xu C.Y., Gui C.F., "Distance Regularized Level Set Evolution and Its Application to Image Segmentation," *IEEE Trans. Image Processing*, 19, pp. 3243-3254(2010).
- [19] Adalsteinsson D. and Sethian J., "A Fast Level Set Method for Propagating Interfaces," *Journal of Comput. Phys.*, 118(2), pp. 269-277(1995).
- [20] NaiK S.K. and Murthy C.A., "Hue-Preserving Color Image Enhancement Without Gamut Problem," *IEEE Trans. Image Processing*, 12(2), pp. 1591-1598 (2003).
- [21] Canny, J., "A Computational Approach To Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6), pp. 679-698, 1986.
- [22] Vezhnevets, V., Sazonov, V., Andreeva, A., "A survey on pixel-based skin color detection techniques," *GRAPHICON03*, pp. 85-92(2003).
- [23] Tataraju, S. and Mehta, A., "Image segmentation using k-means clustering, EM and normalized cuts (Technical Report)," Irvine: University Of California, (2008) Available from [http://www.ics.uci.edu/wdramanan/teaching/ics273a\\_winter08/projects/avim\\_report.pdf](http://www.ics.uci.edu/wdramanan/teaching/ics273a_winter08/projects/avim_report.pdf).
- [24] MacQueen, J. B., "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press. pp. 281-297(1967).

[25] Hunter, R.S., "Photoelectric Color-Difference Meter," Journal of the Optical Society of America 38(7), pp. 661(1948)

[26] Pascale D., "RGB Coordinates of the Macbeth Color Checker," (2012) Available from:  
<http://www.babelcolor.com/download/RGB%20Coordinates%20of%20the%20Macbeth%20ColorChecker.pdf>.

[27] Roberts M., Packer J., Sousa M.C. and Mitchell J.R., "A Work-Efficient GPU Algorithm for level set segmentation," Journal of High Performance Graphics pp. 123-132 (2010).

[28] Lefohn A.E., Kniss J.M., Hansen C.D. and Whitaker R.T., "A Streaming Narrow-Band Algorithm: Interactive Computation and Visualization of Level Sets," IEEE Tran. visualization and computer graphics, 10, pp. 422-433 (2004).