# Mobile Framework for CT Image Reconstruction

M.S.M. Yusoff[1], R. Sulaiman[1], K. Shafinah[2], R. Fatihah[2] and J. Abdullah[3]
*[1]Institute of Visual Informatics, National University of Malaysia (UKM), 43600 Bangi, Selangor, Malaysia.*
*[2]Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, National University of Malaysia.*
*[3]Industrial Division, Malaysian Nuclear Agency, Bangi, 43000 Kajang, Selangor, Malaysia.*
*soyapi81@gmail.com*

***Abstract*—Mobile devices have conquered the world from a common daily usage as e-mail to a complex application as Global Positioning System. The mobile devices have a potential to be developed as a computed device with an application to reconstruct images from computed tomography. The mobile CT application was developed to visualize the CT datasets by plotting out a test dataset to form a sinogram image on the mobile device's screen. The image was obtained by reconstructed the CT datasets using filtered backprojection image processing algorithm. The CT datasets were filtered by using filtered datasets before the image reconstruction processes. The filtering process was a method to remove the blurring effect of the backprojection algorithm.**

***Index Terms*—Mobile; Data Flow Diagram; Datasets; Sinogram; Backprojection; CT Image Reconstruction; Filter.**

## I. INTRODUCTION

The computed tomography (CT) mobile computing is a prototype developed for creating a sinogram data from a bitmap picture and then reconstructing the sinogram data back to the image picture. This conceptual study can be extended to the real CT scan data problems. However, mobile devices claim several boundaries such as small screen viewer size, low pixel resolution, limited available memory and processing power [1]. Mobile devices have served a new method to reach information and the mobile environment is different from a desktop computer (e.g. the screen is smaller, lower computing power) [2]. The increases and penetration of mobile devices in daily lives are transformed and invigorating end users with new experiences, but suffocate with limited computing, power and storage spaces that are predicaments for a sophisticated application required by certain practice fields [3]. Fortunately, in the last few years, exceptional renovations were engineered with mobile devices, smartphones and a tablet with upgraded multi-core processors and graphics processing cores which admitting new application possibilities [4].

The mobile app design requirement is followed formal process notation provided by Gane and Sarson. The Data Flow Diagram (DFD) method is a masterful designing tool to model the system functionality by establishing the data associated with the process for the overall system. It will offer a process view of the system as well as the decomposition view, but sink into time consuming to utilize the model [5]. The DFD is a structured analysis and design method as well as a perceivable tool to represent logic models and shows the data transformation in a system, along with decomposition of the details data flows and function, but it is not a process or procedure modeling method [6].

### *A. Sinogram*

A sinogram was referred to an image obtained from the CT scan devices before the data being manipulated with image reconstruction algorithms. The sinogram was a radon transformation [7] of an image in polar coordinate space. Furthermore, sinogram is defined as the result of the radon transform progression which contains the image's information about the projection data [8]. The radon transform is a collection of projection from many different object's angles [9]. The transformation function of data f(x, y) is given as

$$h(\rho,\theta) = \iint_{-cc}^{cc} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dxdy \qquad (1)$$

where $\rho = x\cos\theta + y\sin\theta$ .

### *B. Image Reconstruction*

The image was reconstructed using the filtered backprojection method which was obtained from an open source GitHub project. The CT program code was contributed by Damien Farrell of University College Dublin, Ireland. The image reconstruction program code was written in a Java language. The equation of backprojection function can be expressed [10] as

$$g(x,y) = \int_0^{\pi} \hat{g}(x\cos\theta + y\sin\theta, \theta)\,d\theta \qquad (2)$$

## II. EXPERIMENTAL

This research incorporates the DFD system requirement to gather the information about the external entities and the process flow of the input and output variables of the mobile CT.

### *A. System Design Requirements*

Prior to the construction of a graphical DFD a concise Contact Diagram (CD) is an important piece of information to build to represent the top level process flow of a system and all its external entities and later to decompose them into graphical DFD lower levels. The Contact Diagram (CD) for the development of the mobile CT imaging application is shown in Figure 1.

The first DFD level for the development of the mobile CT imaging application is shown in Figure 2. Three derivatives DFD level 1 of the contact diagram of the image reconstruction system namely registration, sinogram and image reconstruction. Four main data store in the diagram for storing and retrieving data. Specifically, data store for the user, raw data, sinogram and image slice. An additional data store might involve that link to four main data store, but not shown in the diagram such as a user group, user access and user activity log. This first level DFD needs to be further structured into lower level DFD for a more detail perspective of the system.

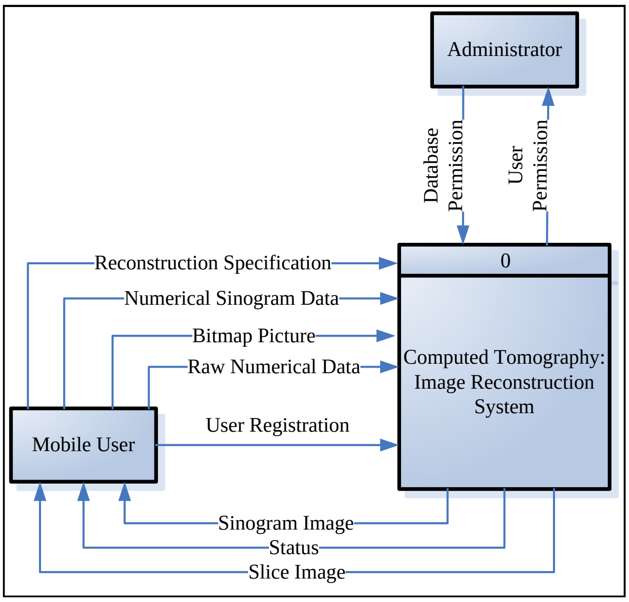


Figure 1: Contact diagram for mobile CT application

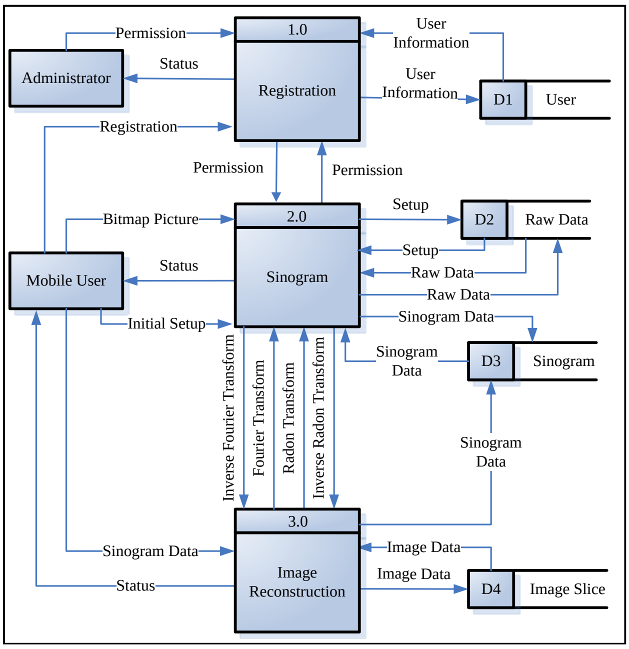


Figure 2: Data Flow Diagram Level 1 for Mobile CT

## B. System Interface Design

The interaction design was necessary to support the process to develop a mobile CT image reconstruction application for a quantitative CT data. According to Karlson (2007) the interface design guidelines concerning the interaction of objects for mobile device operation were useful to the designers and users.

## C. Mobile Design Concept

Figure 3 shows the screen interface menu of the mobile CT application for normal operation users. The image reconstruction button will bring the users to the reconstruction interface as shown in Figure 4.

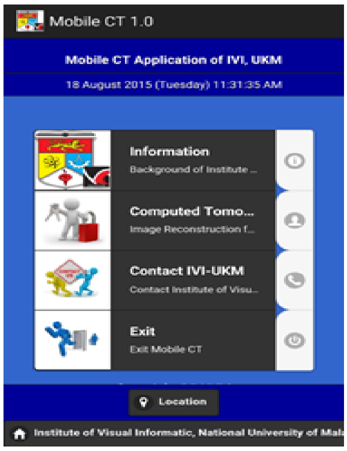


Figure 3: The menu interface for mobile CT


Figure 4: The mobile interface for CT image reconstruction

## III. RESULTS AND DISCUSSION

The datasets used to conduct the experiment for the mobile CT application was obtained from a similar source of the program source code. It was contributed by Damien Farrell of University College Dublin, Ireland at GitHub.

### A. Sinogram

Figure 5 shows the visual image of the raw CT datasets typically called an unfiltered sinogram.

### B. Sinogram Filter

Figure 6 shows the visual image of the filtered CT datasets typically called filtered sinogram using the shepplogan filter kernel. The filter was a common tool to remove the blurring effect of the backprojection process to gain the original object image. The mobile CT prototype was able to filter the datasets by using one of the filter algorithms, for example, ramp filter, shepplogan filter, hamming filter, hann filter, cosine filter and blackman filter.

The Shepplogan filter was one-dimensional kernel [11] that was introduced in backprojection image reconstruction. The filter was created by Larry Shepp and Benjamin F. Logan in 1974. The filtering technique was actually the convolution process between the raw CT datasets and the shepplogan filter datasets using one-dimensional fast Fourier transform (FFT).

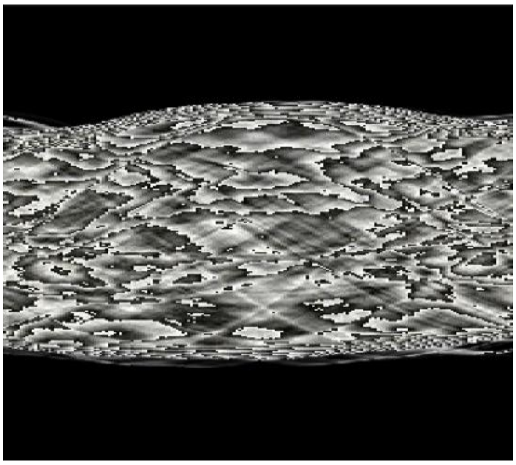

Figure 5: The sinogram of unfiltered datasets

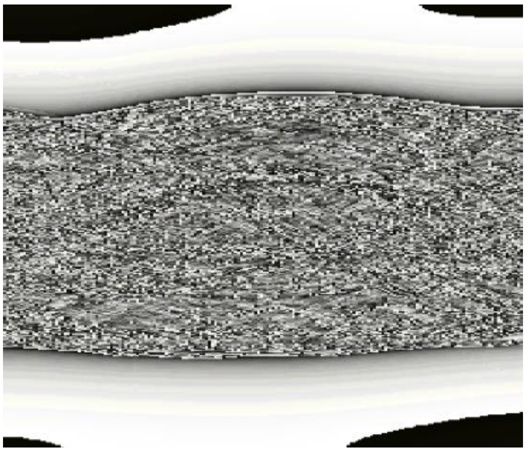

Figure 6: The sinogram of datasets using shepplogan filter

The shepplogan filter kernel datasets are generated by a loop for statement presented in the program pseudocode 1.

**Pseudocode 1** The shepplogan filter algorithm

```
for (i = 1; i <= width; i++)
{
inc = pi*i;
half_width = width/2;
ang = inc/half_width;
sn = sin(ang);
shepp_logan_filter[i] =  inc*(sn/ang);
}
```

The second experiment was implemented using the hemming filter kernel. Figure 7 shows the visual image of the filtered CT datasets using the hemming filter kernel. The hemming filter kernel datasets are generated by a loop for statement presented in the program pseudocode 2.

**Pseudocode 2** The hemming filter algorithm

```
tau = width*cutoff;
for (i = 1; i <= width; i++) {
filter[i] = 0;
if (i <= tau){
      inc = pi*i;
      cs = cos(inc/tau);
      hemming_filter[i] =  inc * (0.54 + 0.46 * cs);
   }
}
```

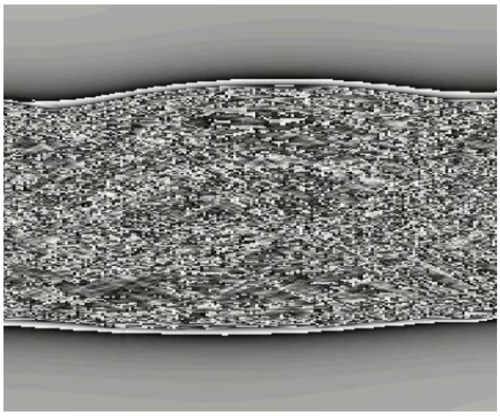

Figure 7: The sinogram of datasets using hemming filter

*C. CT Image*

Figure 8 shows the filtered backprojection image using shepplogan filter with nearest interpolation.

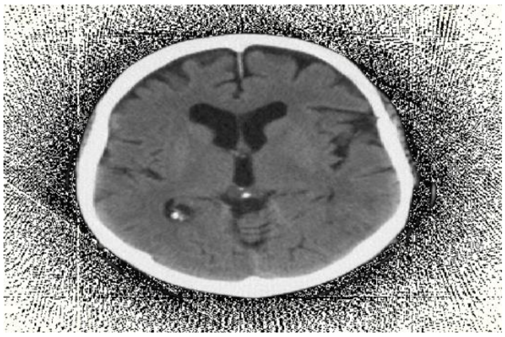

Figure 8: The filtered image using shepplogan filter

Figure 9 shows the filtered backprojection image using a hemming filter with nearest interpolation but have negative values of pixels.

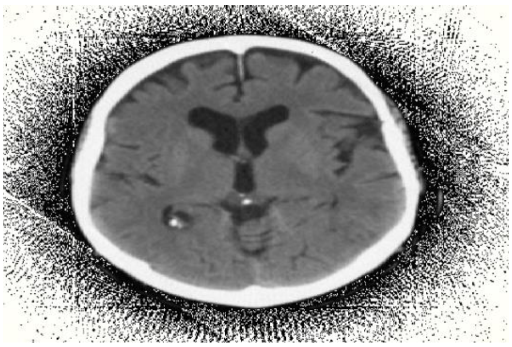

Figure 9: The filtered image using hemming filter

Figure 10 shows the final filtered backprojection gray scale image without negatives values. The minimun value for the pixel was set to zero.

The pseudocode 3 shows the backprojection algorithm that was used to reconstruct the image for the experiment.

The platform of the experiment uses operating system Windows 7 with Intel Core i5-2410M, CPU (2.3GHz), 4GB DDR3 RAM and 750 GB HDD. The CT mobile application was developed using an Android Developer Tools of the Android Open Source Project on eclipse platform, JDT, CDT, EMF, GEF and WTP but later was migrated to Android Studio v1.3.2 (August 2015).

**Pseudocode 3** The backprojection algorithm

```
for (phi=0; phi<180; phi+=1) {
      sintab[i] = sin(phi * pi / 180);
      costab[i] = cos(phi * pi / 180);
       i++;
 }
size = 512; scans = 512; views= 180;

xcenter = size / 2;
ycenter = size / 2;
scale =  size * sqrt(2) / scans;

for (x = -xcenter; x < xcenter; x++)   {
   for (y = -ycenter; y < ycenter; y++) {
        if  ( x <= xcenter  &&  y <= Ycenter  )  {
             for (phi=0; phi<180; phi+= 1) {
                   pos = -x * costab[i] + y * sintab[i];
                    if (interp == "nearest") {
                       S = round(pos / scale);
                       S += scans / 2;
                        if (S < scans && S > 0) {
                            val += projection[i][S];
                         }
                   }
                   i++;
             }
S = 0;i = 0;
x1 = x+xcenter;
y1 = y+ycenter;
bpimage[x1][y1] = (val/views);
val = 0;
        }
   }
}
```

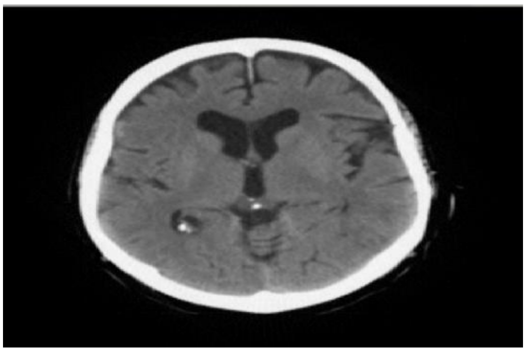

Figure 10: The final image without negative values

Besides the mobile devices, the BlueStacks application and the Android Studio emulator can be used to display the sinogram and the reconstructed images during the development stage.

## IV. Conclusion

The mobile CT application prototype was successfully developed and tested using mobile devices as well as emulated android devices. The image reconstruction technique using mobile devices was introduced as a new dimension in computed tomography. The application prototype was able to complete the computational processing to obtain the sinogram and the image. The results show that mobile CT application was a reality of today's computing technology.

In conclusion, this mobile CT prototype is able to reconstruct the image from the sinogram data by using image processing technique known as a backprojection method with the shepplogan filter. The mobile CT application was using a nearest interpolation technique to calculate the pixel image position on the screen. Further work is needed to transform the 2D pixel slice image to the 3D voxel volumetric model. The research could extend to reconstruct images from the 3D datasets resources.

## Acknowledgment

This research was supported by the National University of Malaysia research grant HEJIM-FTSM-FKAB-MTDC-101101005.

## References


[1] H., Harun, N., Jailani, M.A., Bakar, M.S., Zakaria and S., Abdullah, 2009. A generic framework for developing map-based mobile application. in Electrical Engineering and Informatics, 2009. ICEEI '09. International Conference on. 434-440.

[2] D., Martin, C., Lamsfus, and A., Alzua, 2011. Mobile Context Data Management Framework. in Multimedia and Ubiquitous Engineering (MUE). 2011 5th FTRA International Conference on. 73-78.

[3] V., Koufi ,F., Malamateniou, G., Vassilacopoulos, and A., Prentza, 2012. An Android-Enabled Mobile Framework for Ubiquitous Access to Cloud Emergency Medical Services. in Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on. 95-101.

[4] P., Tanskanen, K., Kolev, L., Meier, F., Camposeco, O., Saurer, and M., Pollefeys , 2013. Live Metric 3D Reconstruction on Mobile Phones. in Computer Vision (ICCV). 2013 IEEE International Conference on. 65-72.

[5] N.,Olayan, V., Patu, Y., Matsuno, and S.,Yamamoto, 2013. A Dependability Assurance Method Based on Data Flow Diagram (DFD). in Modelling Symposium (EMS), 2013 European. 113-118.

[6] Q., Li and Y.L., Chen, 2009. Data Flow Diagram. in Modeling and Analysis of Enterprise and Information Systems, ed: Springer Berlin Heidelberg. 85-97.

[7] W.R., Madych, 2004. Radon's Inversion Formulas. Transactions of the American Mathematical Society. 356:4475-4491.

[8] Z., Zakaria , N.H., Jaafar, N. A. M., Yazid, M.S. B., Mansor, M.H.F., Rahiman and R.A., Rahim , 2010. Sinogram concept approach in image reconstruction algorithm of a Computed Tomography System using MATLAB. in Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on. 500-505.

[9] A., Ramotar and J., Orchard. 2006. General Geometry CT Reconstruction. University of Waterloo. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.3285

[10] P. Toft , 1996. Part II -Inverse Radon Transform and PET. PhD Thesis.

[11] A., Matani and K., Terakawa, 2002. Artifact reduction filtering method for CT images. in Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint, 2:1035-1036.