

# IMAGE RECONSTRUCTION ALGORITHM FOR ELECTRICAL CAPACITANCE TOMOGRAPHY

Arko\*

## ABSTRAK

### ALGORITMA REKONSTRUKSI CITRA UNTUK TOMOGRAFI KAPASITANSI.

Rekonstruksi citra untuk tomografi kapasitansi atau *Electrical Capacitance Tomography (ECT)* kebanyakan menggunakan peta sensitivitas (*sensitivity maps*) dari sensor sebagai faktor penimbang. Komputasinya cepat, karena hanya memerlukan operasi penjumlahan hasil-kali (*multiply-and-accumulate - MAC*) yang sederhana. Akan tetapi citra yang dihasilkan biasanya mengalami pengaburan yang disebabkan oleh pengaruh medan-lunak (*soft-field effect*) dari sensor. Dalam makalah ini sebuah metode rekonstruksi citra iteratif dengan pembatasan proporsional (*proportional thresholding*) yang mampu meningkatkan kualitas citra secara nyata akan dibahas. Strategi implementasi, tingkat komputasi dan kecepatan rekonstruksi yang dapat dicapai akan dibandingkan bila digunakan komputer personal (PC) dan Prosesor Pengolah Sinyal (DSP). Implementasi pada PC memanfaatkan Watcom C++ 10.6 dan Microsoft Visual C++ 5.0. Citra hasil eksperimen dibandingkan dengan citra yang direkonstruksi dengan perangkat lunak komersial yang ada. Algoritma yang dikembangkan ternyata mampu meningkatkan kualitas citra secara nyata hanya dengan beberapa kali iterasi. Metode rekonstruksi ini dengan mudah dapat dieksploitasi untuk aplikasi on-line.

## ABSTRACT

**IMAGE RECONSTRUCTION ALGORITHM FOR ELECTRICAL CAPACITANCE TOMOGRAPHY (ECT).** Most image reconstruction algorithms for electrical capacitance tomography (ECT) use sensitivity maps as weighting factors. The computation is fast, involving a simple multiply-and-accumulate (MAC) operation, but the resulting image suffers from blurring due to the soft-field effect of the sensor. This paper presents a low cost iterative method employing proportional thresholding, which improves image quality dramatically. The strategy for implementation, computational cost, and achievable speed is examined when using a personal computer (PC) and Digital Signal Processor (DSP). For PC implementation, Watcom C++ 10.6 and Visual C++ 5.0 compilers were used. The experimental results are compared to the images reconstructed by commercially available software. The new algorithm improves the image quality significantly at a cost of a few iterations. This technique can be readily exploited for online applications.

---

\* Pusat Pengembangan Teknologi Informasi dan Komputasi - BATAN

## INTRODUCTION

Most image reconstruction algorithms for ECT use sensitivity maps as weighting factors to implement linear back projection (LBP) [1,2,3]. The computation is fast, involving a simple multiply-and-accumulate (MAC) operations. The resultant image suffers from blurring due to the soft-field effect of the sensor. To improve image quality, iterative methods of image reconstruction have been of great interest [4,5]. By the end of 1996, the author simulated iterative reconstruction using a finite element method yielding promising results [6]. The most recent implementation of an iterative algorithm was based on Landweber iteration method [7]. Unfortunately, all these iteration methods share a common drawback in that the computational run-time cost is of the order of a few seconds or longer. They are thus inappropriate for fast online applications, such as flow imaging.

This paper presents a low-cost, iterative method with *proportional thresholding* to improve the image quality. The iteration is based on the calculation of the image and capacitance by utilising only the sensitivity maps. The strategy for implementation, computational cost, and achievable speed are examined when using a personal computer (PC) and a Digital Signal Processor (DSP). For PC implementation, Watcom C++ 10.6 and Visual C++ 5.0 compilers were used. The experimental results are compared to the original image reconstructed by commercially available software. The new algorithm improves the image quality significantly with additional computation cost as low as a single iteration. This technique can be readily exploited for online application.

## ITERATIVE ALGORITHM WITH PROPORTIONAL THRESHOLDING

The iteration is based on the calculation of both image and capacitance by utilising the sensitivity maps. The threshold could be fixed to **0** to give pixel values of the image in the range of **0** - **1**. The authors prefer to use a simple proportional threshold of  $0.5 * C_{av}$ , where  $C_{av}$  is the average of the normalised measured capacitance  $C_{meas}$ . Other values have been suggested too by previous researchers [1]. Early image reconstruction using LBP relied upon a sensitivity map computed assuming a homogeneous permittivity distribution. The images exhibited large soft-field error. By using proportionally adaptive thresholding and iteration, the blurring is reduced significantly and quickly. After a single iteration the shapes of the objects become clearer.

A low-cost iterative method, with proportional thresholding, to improve the image quality quickly is shown in a flowchart of Figure 1. LBP is used to get the first estimate of permittivity distribution. After applying the proportional threshold to the

first image, the calculated capacitance **Ccalc** is computed using the thresholded image **Gth** and the same sensitivity maps **S** as opposite to LBP -- a process called linear forward solver (LFS). The capacitance difference **Cdif** between the **Cmeas** and **Ccalc** is used to reconstruct an image difference **Gdif**, which shows strong negative values at locations where the blurring in the previous image **Gth** occurs. When **Gdif** is added up to the **Gth** and then the same adaptive threshold is applied, the blurring is effectively cancelled. If more accurate image is desired, more iteration can be done, so the stop criteria could be the number of iteration. From our experience, a single iteration is enough to give significant image improvement when high frame rate is needed.

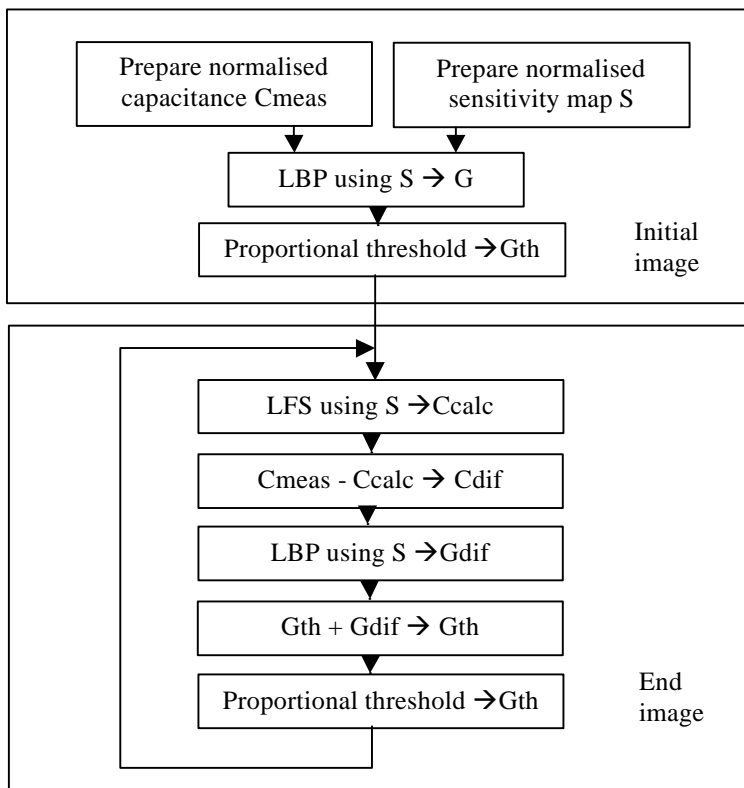


Figure 1. Flowchart of an iterative image reconstruction algorithm with proportional thresholding.

## STRATEGY FOR IMPLEMENTATION

The measured capacitance was collected from an array of  $N = 8$  electrodes mounted around a circular pipe, depicted in Figure 2, to give  $M = N*(N-1)/2 = 28$  measurements. The new algorithm is implemented as a square grid of  $32 \times 32$ , having a total number of internal small squares  $P=812$  within the circular sensing area, as shown in Figure 3.

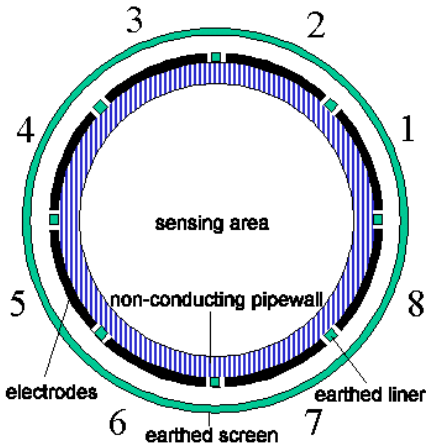


Figure 2. Cross sectional view of the circular pipe with  $N=8$  sensing electrodes

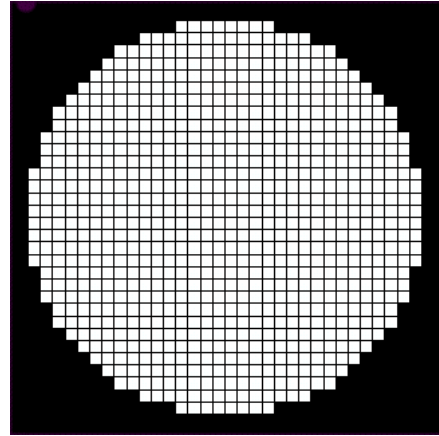


Figure 3. Square grid of  $32 \times 32$  with  $P=812$  elements within the circular area

Before going any further, let us examine the basic equation for LBP algorithm given here:

$$G(p) = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N S_{ij}(p) \cdot C_{ij}}{\sum_{i=1}^{N-1} \sum_{j=i+1}^N S_{ij}(p)} \quad (1)$$

$G(p)$  Grey level of the calculated pixel at  $p^{\text{th}}$  pixel (Normalised between 0 - 1)

$S_{ij}(p)$  Sensitivity at  $p^{\text{th}}$  pixel for electrode pair  $i$  and  $j$   
 $C_{ij}$  Normalised capacitance value for electrode pair  $i$  and  $j$   
 $N$  Number of electrodes

Since the denominator and  $S_{ij}(p)$  in equation (1) are basically two arrays of known constants, by using an algebraic expansion one can write down normalised weighting constants  $W_{ij}(p)$  in the form of :

$$W_{ij}(p) = \frac{S_{ij}(p)}{\sum_{i=1}^{N-1} \sum_{j=i+1}^N S_{ij}(p)} \quad (2)$$

It is thus more advantageous to calculate  $W_{ij}(p)$  once only prior to the image reconstruction. Even better is to save the normalised weighting constants  $W_{ij}(p)$  instead of  $S_{ij}(p)$  as a normalised sensitivity map. By doing so, the long latency division operation on virtually any processor is eliminated from the reconstruction algorithm. LBP simplifies to multiply-and-accumulate (MAC) operations between constants  $W_{ij}(p)$  and one variable  $C_{ij}$  as below :

$$G(p) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N W_{ij}(p).C_{ij} \quad (3)$$

The above calculation is more convenient and faster to implement. If desired, the algorithm can be implemented using a DSP chip that employs fast multiply-and-accumulate operations.

The calculation of **Ccalc** follows a similar way and results in the following equation

$$Ccalc_{ij} = \sum_{p=1}^P WT_{ij}(p).G(p) \quad (4)$$

where  $WT_{ij}(p)$  is the sensitivity maps normalised along the image vector, that is:

$$WT_{ij}(p) = \frac{S_{ij}(p)}{\sum_{p=1}^P S_{ij}(p)} \quad (5)$$

Again, the  $WT_{ij}(p)$  can be calculated beforehand and stored as a look-up table, as they are all constants. Equation (4) involves no slow division operation.

To summarise, the equations for calculating pixel value  $G(p)$  in equation (3) and  $Ccalc_{ij}$  in equation (4) are very similar and involve the same number of multiply-and-accumulate (MAC) operation. They are thus very simple to implement. For 812 pixels and 28 measurements, as depicted in Figure 3 and 4 earlier, each equation theoretically requires only  $28*812 = 22736$  MACs.

The following is a piece of code that implements the LBP in equation (3) to obtain an image.

```
//aMeas = 28;
//aIntPixel = 812;
for (i=0; i<aMeas; i++)
for (k=0; k<aIntPixel; k++)
    aImage[k] = aImage[k] + aCN[i] * aWSen[i*aIntPixel+k];
```

When the code is compiled, using Visual C++ 5.0 and Watcom C++ 10.6 compilers, its equivalent assembler code is much longer. The following tables summarises three cases; the last column is for DSP implementation.

# Instruction	Visual C++ 5.0	Watcom 10.6	DSP2185
Init loop i	2 + (7)	2 + (6)	12 ~ includes setup
Init loop k	2 + (7)	2 + (6)	5
Internal looping	14	16	1
Total instruction count for one LBP	2+ 28*( 7 + 2 + 812*( 7+14) ) = <b>477 710</b>	2+ 28*( 6 + 2 + 812*( 6+16) ) = <b>500 418</b>	12+(5+28)(812*1) = <b>26 808</b> clock cycles

Figure 4. Comparison of the total instruction counts for linear back projection.

The table tells us that a single LBP algorithm wastes much time performing address calculation and fetching data. The actual work of multiply-and-accumulate is far from optimum when implemented on a PC. Exact calculation is not straightforward, as each instruction in a PC needs a different number of clocks. Even if LBP is implemented on the newest PC, time is still wasted for the loading/storing data, since the chip is not optimised for numerical processing alone. As an illustration, floating point multiplication (FMUL) and addition (FADD) each need 3 clock cycles to execute, assuming the data has been pre-loaded in registers. Floating point division (FDIV) is a multi-clock, long latency operation [9]. When LBP is implemented in a

DSP2185 [10], it needs only 1 instruction to execute the internal loop, and the total instruction count is 26808. During each clock cycle, the DSP performs all data loading/storing/indexing and executes the intended operation in parallel. The reconstruction rate (LBP) for a DSP running at 33MHz would be  $33.E6 / 26808 = 1230$  frames per second theoretically. For calculating back Ccalc (i.e. LFS), the DSP requires only  $12+28*(812+5) = 22888$  clocks. Nowadays, there are plenty of more powerful DSPs on the market to choose from.

Further reduction in reconstruction time and storage requirements can be achieved by using a smaller number of pixels per frame. Experiments by the author showed that even with a grid of 16x16 (using 208 inner pixels), see Figure 5, a good qualitative image can be obtained within much less time. For flow imaging, speed is important and qualitative images by LBP using a small number of iterations are sometimes considered adequate to accommodate cross correlation for velocity profiling.

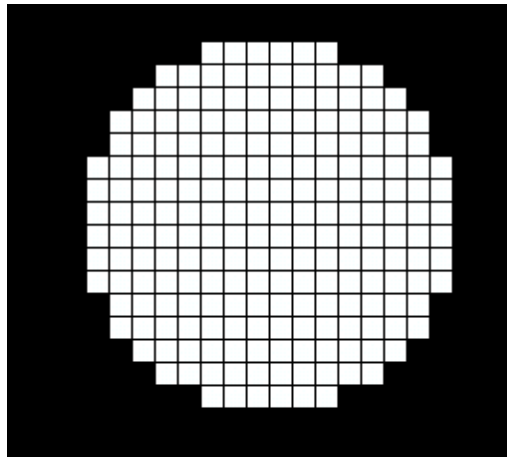


Figure 5. Square grid of 16x16 with 208 elements within the circular are to increase the speed of image reconstruction

## EXPERIMENTAL RESULTS

The experimental results are compared to the original image reconstructed by commercially available software. The new algorithm improves the image quality significantly with additional cost of computation of as low as a single iteration. Therefore, this technique is can be readily exploited for online application.

Shown in Figure 6 are the images reconstructed by using LBP of the commercial software PCECT v2.1 from Process Tomography Ltd (first column) and

the new algorithm (second, third and fourth columns). The blurring indicated both by the original image by PCECT software (first column) and the original image with proportional thresholding alone (second column) is still comparably dominant. Figure 6a and 6b show images of two non-touching circular rods of different sizes. The original images in the first column exhibit severe blurring, but after just one iteration, a large part of the blurred area is eliminated. The last column shows much better images after 5 iterations to give the closest true distribution. Figures 6a and 6b demonstrate the superb capability of the algorithm to reconstruct image properly compared to LBP.

Figure 6c and 6d are images of a single rod (large and small respectively) placed arbitrarily in the sensing area. Again, they are successfully reconstructed by the new algorithm with much better quality. Figures 6e,f,g show how the algorithm can reduce blur and enhance air-solid interface for a stratified and slug flows, respectively.

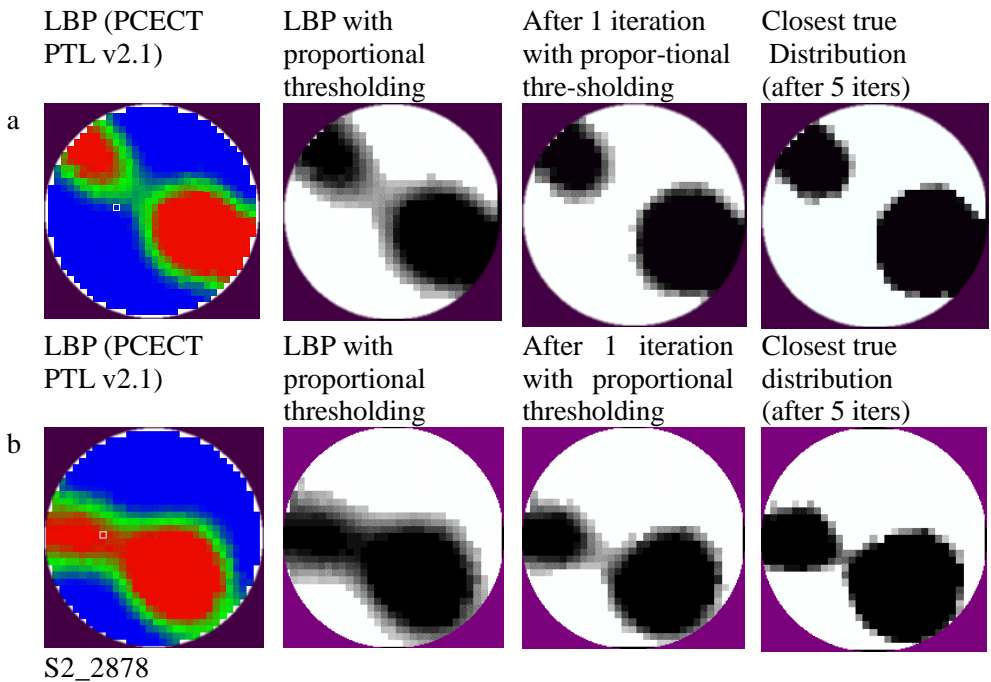


Figure 6. Experimental Results

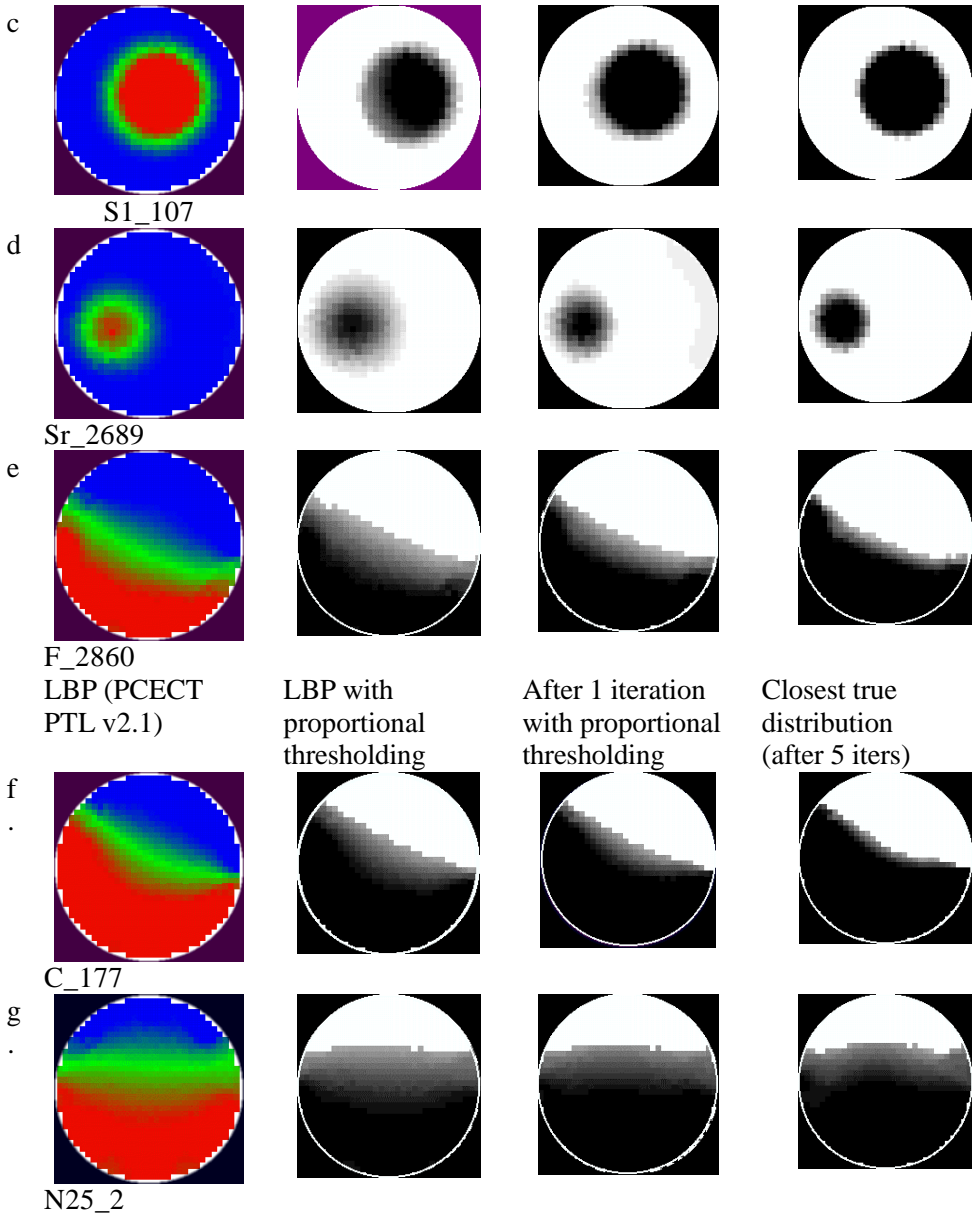


Figure 6. Experimental Results.

Now is an issue of speed. Here is a summary. The new algorithm was tested extensively on a Pentium PC @133MHz as an entry level PC, running Window 95. The C code was developed and optimised both at high level manually and at assembly level by using optimising compilers from Visual C++ 5.0 and Watcom C++ 10.6. Both compilers produced optimised code of similar quality.

For the test purpose, the number of measurement data was 28 and the number of pixels was 812. The PC performed 100 iterations per second, and during each iteration the PC had to perform LFS, LBP, calculate capacitance difference, update and threshold the resultant image. As the images in Figure 6 demonstrated earlier, even with one iteration the image was quite satisfactory. The frame rate achievable was 58 frames per second when one LBP plus one iteration was used, to give image quality shown in the third column of Figure 6. When the pixel number was reduced to 208 pixels per frame, 110 frames per second were at hand. With faster computers or DSPs the frame rate could be increased to accommodate higher frame rates.

## **DISCUSSION AND CONCLUSION**

The image quality was significantly improved with additional computing cost of as low as one iteration. Compared to the reconstruction times published by many previous iterative approaches of order a few second or longer, this new approach was much faster. The use of proportionally adaptive thresholding which depended upon the average value of the measured capacitance was satisfactory to reduce the blurring effectively. To conclude, this new approach should be used to increase the image quality of the ordinary LBP since the computational cost is relatively cheap.

## **REFERENCES**

1. O. Isaksen, *A review of reconstruction techniques for capacitance tomography*, Meas. Sci. & Tech., 7 (3), (1996) 325-337
2. C.G. Xie, S. Huang, B.S. Hoyle, R.Thorn, C.Lenn, D. Snowden, M.S. Beck, *Electrical capacitance tomography for flow imaging: system model for development of image reconstruction algorithms and design of primary sensor*, IEE Proceeding-G, 139 (1), (1992), pp.89-98
3. R.A. Williams, M.S. Beck, *Process Tomography -- Principles, Techniques, and Applications*, Butterworth Heinemann, (1995)

4. O. Isaksen, J E Nordtvedt, *Capacitance Tomography : Reconstruction based on optimisation theory*, Tomographic techniques for process design and operation, Edited by M.S Beck e.a, Computational Mechanics Publications, (1992) p. 213-24
5. Q. Chen, B.S. Hoyle, H.J. Strangeways, *Electric field interaction and enhanced reconstruction algorithm in capacitance process tomography*, Tomographic techniques for process design and operation, Edited by M.S Beck e.a, Computational Mechanics Publications, (1992) p. 205-212
6. W.Q. Yang, J.C. Gamio, M.S. Beck, *A fast iterative reconstruction algorithm for capacitance tomography*, Sensor and their applications VIII, (1997) p. 47-52
7. W.Q. Yang, D.M. Spink, T.A. York, H.McCann, *An image reconstruction algorithm based on Landweber iteration method for capacitance tomography*, Submitted to IEEE Trans. Image Processing (1999)
8. W.Q. Yang, S. Liu, *Electrical capacitance tomography with square sensor*, World 1st Conference on Industrial Process Tomography, Manchester (1999)
9. Intel Architecture Software Developer's Manual Vol 2:Instruction Set Reference
10. Analog Devices, *ADSP-2185 DSP Datasheet Rev.0*, Analog Devices Inc. (1997)