

# 國立交通大學

資訊科學與工程研究所

## 博士論文

以 X 光顯微鏡進行三維斷層影像重建

Three-Dimensional Tomography Reconstruction for  
X-Ray Microscopy



研究生：鄭昌杰

指導教授：荊宇泰 博士

胡宇光 博士

中華民國一百零三年四月

以 X 光顯微鏡進行三維斷層影像重建  
Three-Dimensional Tomography Reconstruction for  
X-Ray Microscopy

研究生：鄭昌杰

Student: Chang-Chieh Cheng

指導教授：荊宇泰 博士

Advisor: Dr. Yu-Tai Ching

胡宇光 博士

Dr. Yeukuang Hwu

國立交通大學資訊學院  
資訊科學與工程研究所  
博士論文

A Dissertation Submitted to  
Institute of Computer Science and Engineering  
College of Computer Science  
National Chiao Tung University  
In Partial Fulfillment of the Requirements for  
the Degree of Doctor of Philosophy  
in Computer and Information Science

1896

April 2014

Hsinchu, Taiwan

中華民國一百零三年四月

# 以 X 光顯微鏡進行三維斷層影像重建

學生：鄭昌杰

指導教授：荊宇泰 博士

胡宇光 博士

國立交通大學資訊學院

資訊科學與工程研究所

## 摘 要

由於高解析度影像在取像上並不是那麼容易，所以從這些影像上取得的資訊是十分有價值的。X 光顯微鏡利用了硬 X 光的短波長與高穿透力的特性來取得厚實標本的奈米解析度影像，這是其他取像技術無法辦到的。一般的 X 光取像設備是以 X 光吸收率來取像，但若以折射率來進行相位取像可有效地取得物體材質間的對比變化並加強之，且這種取像過程不會被某些生物標本的高 X 光吸收結構所影響。這種先進的技術在生醫研究上具有獨特的價值，但也浮現許多問題需要解決。本研究針對其中幾個問題並提出新的影像處理方法來解決。在斷層取像過程中，物體的旋轉會發生影像上的不對稱抖動現象，我們提出一個特徵點對齊法來對齊這些 X 光投影影像，使得斷層掃描影像可以成功的重建出來。為了解決高解析度攝影視野的限制，我們提出一個影像拼貼的技術將數張較小視野的影像合成一張可完整包含物體的大尺寸影像。此方法經由一組模擬資料來驗證其大型物體的三維斷層影像重建之可行性。最後，我們以傅立葉體積描繪法來快速地瀏覽重建完成的三維立體資料，也運用了具備美觀性與準確性的紋理體積描繪法來進行三維立體資料的視覺化。其中，我們為傅立葉體積描繪法提出一個快速的資料分類方法，可有效地改變轉換函數的權重並即時地反映在繪圖結果上。

# **Three-Dimensional Tomography Reconstruction for X-Ray Microscopy**

student : Chang-Chieh Cheng

Advisors : Dr. Yu-Tai Ching  
Dr. Yeukuang Hwu

**Institute of Computer Science and Engineering  
College of Computer Science  
National Chiao Tung University**

## **ABSTRACT**

High resolution images pushing the performance limit are always difficult to acquire and therefore extremely valuable to extract crucial information from them. X-ray microscopy takes advantage of the short wavelength and high penetration characteristic nature of hard-X-rays and demonstrated nanometer level resolution on thick specimens that is not available by other techniques. With devices to obtain contrast from differences of the refractive index, rather than the absorption of X-rays, the phase imaging can effectively enhance the contrast of materials which does not contain high X-ray absorbing structures such as biology specimens. The cutting edge level of performance found unique value in particularly biomedical research but also highlighted several problems which requires attention. This study addresses specifically several issues and tackles them by new image processing approaches. An image alignment method is implemented to eliminate the problem caused by the nonsystematic jitter of the sample rotation mechanism using a feature-recognition based alignment algorithm so a tomography reconstruction can be performed on these aligned projection images. To overcome the limitation of small field of view associated with the high resolution and limited number of imaging pixels, another image alignment method is developed to stitch these small field-of-view images into a larger one covering larger area of the specimen. This method is tested on a large number of simulated projection images of a phantom to demonstrate the possibility of tomography reconstruction in 3D of a large object. Finally, to visualize the tomography reconstructed images, a method based on the Fourier volume rendering (FVR) algorithm is designed to achieve better visualization and precision of the commonly used texture-based volume rendering (TVR) method. Improved processing



efficiency by adjusting the weight of transfer function for FVR drastically reduces the processing time and computation resources required for voxel classification and make it possible for routine application.

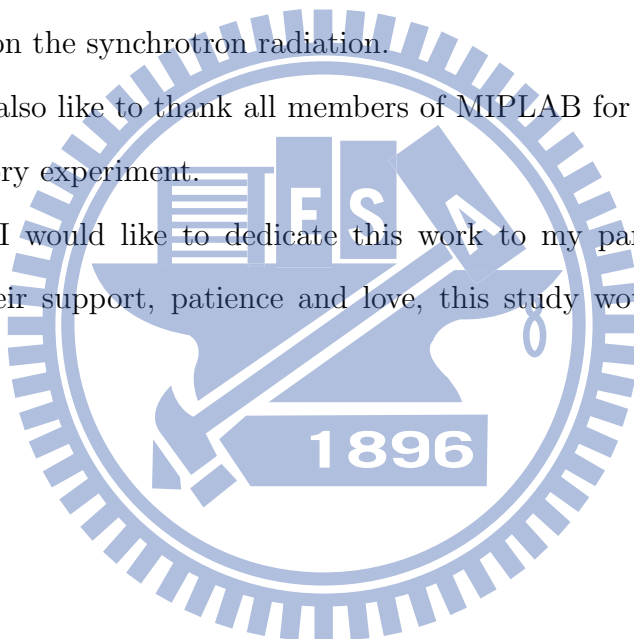


## Acknowledgement

First of all I would like to express my sincere gratitude to my thesis advisor, Prof. Yu-Tai Ching, for his support and guidance throughout my research. My sincere appreciation is extended to Dr. Yeukuang Hwu, for his support and advise on the synchrotron radiation.

I would also like to thank all members of MIPLAB for their assistance in the laboratory experiment.

Finally, I would like to dedicate this work to my parents and brother. Without their support, patience and love, this study would not have been completed.



# Three-Dimensional Tomography Reconstruction for X-Ray Microscopy

Student: Chang-Chieh Cheng<sup>1</sup>

Advisors: Yu-Tai Ching<sup>1</sup>, Yeukuang Hwu<sup>2</sup>

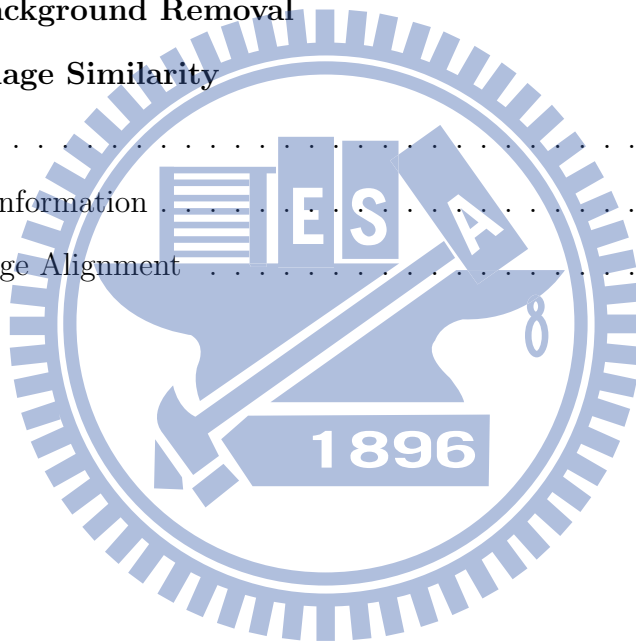
1. Department of Computer Science, National Chiao Tung University,  
Hsinchu, Taiwan.

2. Institute of Physics, Academia Sinica, Taipei, Taiwan.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgement</b>	<b>iv</b>
<b>Chapter 1 Introduction</b>	<b>6</b>
<b>Chapter 2 Tomography Reconstruction</b>	<b>13</b>
2.1 Filtered Back Projection . . . . .	13
2.2 Algebraic Reconstruction Technique . . . . .	20
<b>Chapter 3 Visualization</b>	<b>25</b>
3.1 Texture-Based Volume Rendering . . . . .	25
3.2 Fourier Volume Rendering . . . . .	29
3.2.1 Transfer Function Design for FVR . . . . .	31
3.2.2 Bézier curve Transfer Function . . . . .	32
3.2.3 B-Spline Transfer Function . . . . .	34
3.2.4 Increasing the Control Points . . . . .	35
3.2.5 Results of FVR . . . . .	37
<b>Chapter 4 High-Resolution Volume Reconstruction</b>	<b>46</b>
4.1 Methods . . . . .	47
4.1.1 Sample Preparation and Image Acquisition . . . . .	47
4.1.2 The Alignment Method . . . . .	49
4.1.3 Horizontal Displacement Estimation . . . . .	53
4.2 Results . . . . .	55
4.2.1 The Phantom . . . . .	55
4.2.2 HeLa Cells . . . . .	57

<b>Chapter 5</b>	<b>High-Resolution Large Volume Reconstruction</b>	<b>71</b>
5.1	Stitching . . . . .	72
5.2	Results . . . . .	82
5.3	Discussion . . . . .	85
<b>Chapter 6</b>	<b>Conclusions and Future Work</b>	<b>86</b>
6.1	Summary . . . . .	86
6.2	Future Work . . . . .	87
<b>Appendix</b>		<b>88</b>
<b>Chapter A</b>	<b>Background Removal</b>	<b>89</b>
<b>Chapter B</b>	<b>Image Similarity</b>	<b>92</b>
B.1	Entropy . . . . .	93
B.2	Mutual Information . . . . .	94
B.3	Two-Image Alignment . . . . .	95
<b>Bibliography</b>		<b>96</b>



# List of Figures

1.1	The structure of synchrotron X-ray microscope . . . . .	9
1.2	The synchrotron X-ray microscope in NSRRC . . . . .	10
1.3	The synchrotron X-ray images . . . . .	11
1.4	The stitched image of a cancer cell . . . . .	12
2.1	The tomography reconstruction . . . . .	17
2.2	Fourier slice theorem . . . . .	18
2.3	The FBP algorithm . . . . .	18
2.4	The pixel area covered by ray . . . . .	19
2.5	The results of FBP . . . . .	19
2.6	The theory of ART . . . . .	22
2.7	The tomographic images of FBP and SIRT . . . . .	23
3.1	Texture mapping . . . . .	27
3.2	The proxy polygons in TVR . . . . .	28
3.3	A result of TVR . . . . .	28
3.4	The FVR algorithm . . . . .	31
3.5	A spline with $m$ control points . . . . .	33
3.6	The clustered control points . . . . .	37
3.7	The results of FVR with Bézier curve transfer function(CT chest) . . . . .	40
3.8	The results of FVR with B-spline transfer function(CT chest) . . . . .	41
3.9	The results of FVR with B-spline transfer function(CT chest) . . . . .	42
3.10	The results of FVR with B-spline transfer function(HeLa1) . . . . .	43
3.11	The results of FVR with B-spline transfer function(HeLa2) . . . . .	44

4.1	The misalignment of tomography . . . . .	48
4.2	The feature tracking between two X-ray projection images $I^{i-1}$ and $I^i$ . . .	55
4.3	The simulated X-ray image of the phantom data . . . . .	60
4.4	Feature loci of the phantom data . . . . .	61
4.5	One slice in the phantom data . . . . .	61
4.6	X-ray projection images of HeLa cells . . . . .	62
4.7	The loci of the reliable projected feature points of HeLa1 and HeLa2 . . .	63
4.8	One slice in the reconstructed tomographic images of HeLa1 and HeLa2 . .	64
4.9	The 3D volume rendering of the reconstructed volume . . . . .	65
4.10	Eight examples of successful alignment (1) . . . . .	66
4.11	Eight examples of successful alignment (2) . . . . .	67
4.12	The first example of unsuccessful alignment . . . . .	68
4.13	The second example of unsuccessful alignment . . . . .	69
5.1	The acquisition for a large image . . . . .	76
5.2	The search area for alignment. . . . .	77
5.3	The alignment of multiple references. . . . .	77
5.4	The stitch order for a large image . . . . .	78
5.5	The construction of Gaussian-Laplacian pyramid . . . . .	79
5.6	The reconstruction of Gaussian-Laplacian pyramid . . . . .	80
5.7	The image blending . . . . .	81
5.8	Image stitching for the phantom . . . . .	83
5.9	The tomographic result reconstructed from the stitched images of phantom	84
A.1	The procedure of background removal. . . . .	91
B.1	The image alignment. . . . .	96



# List of Tables

2.1	The time comparisons of FBP and SIRT . . . . .	24
3.1	The computing time of two cases using the B-spline as a transfer function .	45
4.1	The test results of misalignment correction. . . . .	70



# Chapter 1

## Introduction

Analyze high resolution images of any imaging method are of very high value. Particularly for those images obtained at the cutting edge level performance. When an imaging method pushes its performance at its limit, it is not likely be improved with technical advancement. Image processing methods repeatedly can offer invaluable support to advance the imaging technique over barriers which cannot be resolved otherwise. X-ray microscopy is an emerging imaging method takes advantage of the unique short wavelength and high penetration characteristics and offer nanometer resolution at thick, hundreds of  $\mu\text{m}$ , specimens in 3D which is currently not available by other imaging techniques. Its importance in biomedical research is boosted by the high quality X-ray photons generated by synchrotron accelerators[1]. Characteristics of X-rays from synchrotron radiation such as very high intensity, coherence, and tunable in energy are all very valuable to achieve high performance for X-ray imaging [2]. At some energy range, such as soft-X-rays (wavelength longer than 1 nm), no other devices offer photons of sufficient number for useful imaging applications. X-ray microscopy using hard-x-rays suffer from the low efficiency optics and therefore the high intensity X-rays from synchrotron are the only choice to allow high performance imaging. Using nanofabricate Fresnel zone plate (FZP) [3] optics of 20 nm outermost zone with high thickness,  $>500$  nm, of Au (gold), 16.5 nm resolution was demonstrated using synchrotron X-rays. Inserting a ring device placing at the back focal plane, Zernike phase contrast is able enhance images of materials which show no strong X-ray absorption such as biology specimens. Fig. 1.1 shows a schematic of a full-field

X-ray microscope based on FZP and Zernike phase contrast. Three dimensional imaging is achieved by rotating the specimen and image the transmitted X-rays at different angles as the “projection images”. Several devices, such as the capillary condenser, the pinhole, and the beamstop, are required to facilitate the illumination conditions for the FZP. An X-ray microscopy of this type is implemented at the BL01B beamline in National Synchrotron Radiation Research Center (NSRRC, Hsinchu) and provide all the images used in this study. Fig. 1.2 shows the X-ray microscopy in NSRRC. This beamline is capable of performing X-ray imaging at different resolution and specimens size. In addition to the X-ray microscopy, a phase contrast micro-radiation imaging setup can produce X-ray micrographs of live animals at  $\mu\text{m}$  resolution. Fig. 1.3 shows three images taken from these X-ray imaging apparatuses. Specifically, Fig. 1.3(a) shows a high resolution  $1024 \times 1024$  pixels X-ray micrograph of an EMT (Epithelial-Mesenchymal Transition) cell with, Fig. 1.3(b) shows  $1600 \times 1200$  pixels phase contrast X-ray image of the tip of a toothpick, and Fig. 1.3(c) shows a montage image of  $5 \times 7$   $1024 \times 1024$  pixels phase contrast microradiology images of a mouse head. The size of the pixels of these images are 11.78 nm (a),  $2.87 \mu\text{m}$  (b), and  $2.87 \mu\text{m}$  (c). These images showcased the superior performance of the X-ray imaging in resolution and contrast and the size of specimen compare of conventional X-ray imaging and other imaging techniques.

After taking projection images at a large number, typically  $> 100$ , at different rotation angles, ideally cover 180 degree, many processing methods can be used to generate a 3D models from these projection images [4]. For example, a most common method, filtered-back-projection method (FBP), transforms the sinogram, consist of a stack image lines from different projection images, to a reconstructed slice of reconstructed image. Putting these slices together, one can obtain a 3D data metrics of the absorption of each voxel. Visualization tools can then be used to produce a “model” to recreated the 3D models which nevertheless is accurate, in terms of special resolution, to the level of the X-ray microscopy imaging.

The most advanced development in X-ray microscopy, reaching 16.5 nm as mentioned above, however, as a technology is pushed to state-of-the-art, suffers from a number

of technical problems and many of these issues cannot be solved by improvement on the hardware alone. One of the most severe problem due to the extreme high resolution of the imaging, is the lack of sufficient precision in the rotation movement. It is not possible with the current technology to guarantee two consecutive rotated images produced precisely at the pre-assigned rotation axis. Even with the best rotation movement, the axis is wobbling and jittering which moves the rotation axis from its otherwise fixed position  $\mu\text{m}$  away, which is intolerable with the images of 20 nm resolution and 2-5 nm pixels size. Such inaccuracy cannot be calibrated because it is not systematic and therefore it does not only reduce the resolution of the reconstructed model, most likely also cause the complete failure of the tomography reconstruction. Current solution to this problem is by manually recognize specific feature in the individual micrographs and then project their position in the rotation motion and then align the images accordingly. Which is not only time consuming, but difficult to execute in most of the cases.

The limitation due to the matching of pixel size and the resolution, the size of the object are often larger than the field of view of an individual image. Although tomography reconstruction is possible, it loses the references to blank area and therefore the absolute references of the absorption coefficient. This makes the quantitative analysis based on the pixel value difficult, if not impossible. There is therefore a strong desire to obtain image to cover the whole specimen while not losing the resolution. Most common method is to stitch a series of images taken by imaging different location of the specimen. Alignment of these images without a precise recording of the specimen position, again limited by the mechanical movement, would require similar image processing effort. Fig. 1.4 shows a high-resolution image of a cancer cell stitched from a  $3 \times 3$  patch of  $1024 \times 1024$  pixel images. Without obtaining a full set of such stitched images at the full rotation angle, we tested our tomography process with a set of phantom data generated to simulate the large-scale tomography reconstruction.

In this study, we also tackle the issue of visualization which is required to analyze 3D volume data rendered from the X-ray projection images. Volume rendering is a commonly used technique that generates a two-dimensional image from a volume data with given

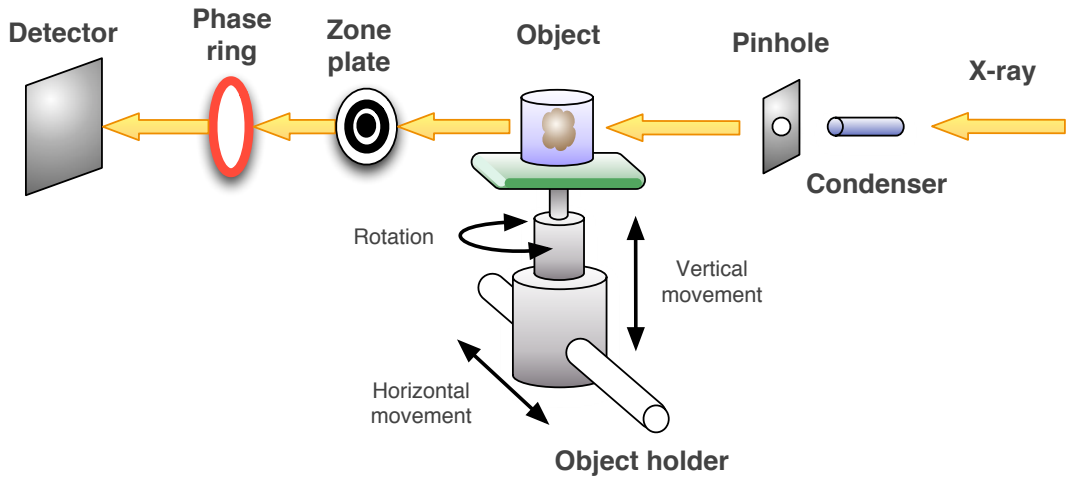


Figure 1.1: **The structure of synchrotron X-ray microscope.**

viewing parameters. Texture-based volume rendering (TVR) [5] using graphics hardware and texture mapping to accelerate the integration of ray casting is the most popular visualization technique in recent years. Fourier volume rendering (FVR) [6], [7], is another real-time visualization method based on Fourier transform. Theoretically, the FVR can be faster than TVR. But in reality, the voxel classification scheme in FVR is much more demanding than in TVR because the rendering is carried out in the frequency domain. This study proposed a method to set a transfer function based on B-spline and demonstrate successfully its much enhanced efficiency in classifying the voxels in FVR.

This thesis is organized as following: In Chapter 2 two tomography reconstruction algorithms used in this study is described in detail and two visualization techniques, TVR and FVR, are presented in Chapter 3. The proposed algorithm for high-resolution volume reconstruction is described in Chapter 4. The proposed method for tomography reconstruction of large high-resolution volume is presented in Chapter 5. Chapter 6 summarizes this thesis with further discussion and proposed future work. In Appendix A, the method of background removal described and the similarity estimation of two images is described in Appendix B.



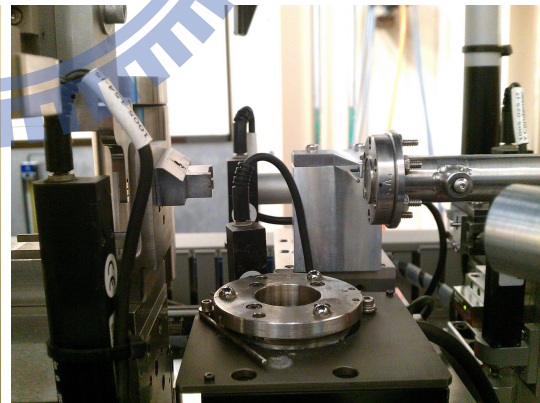
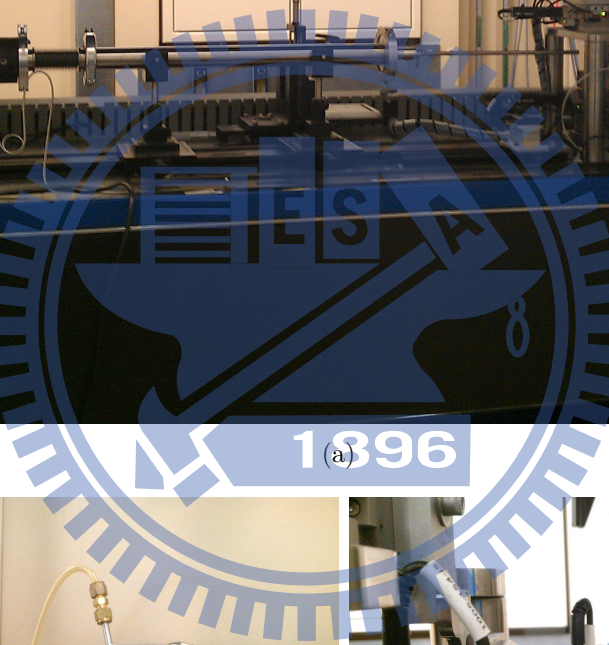
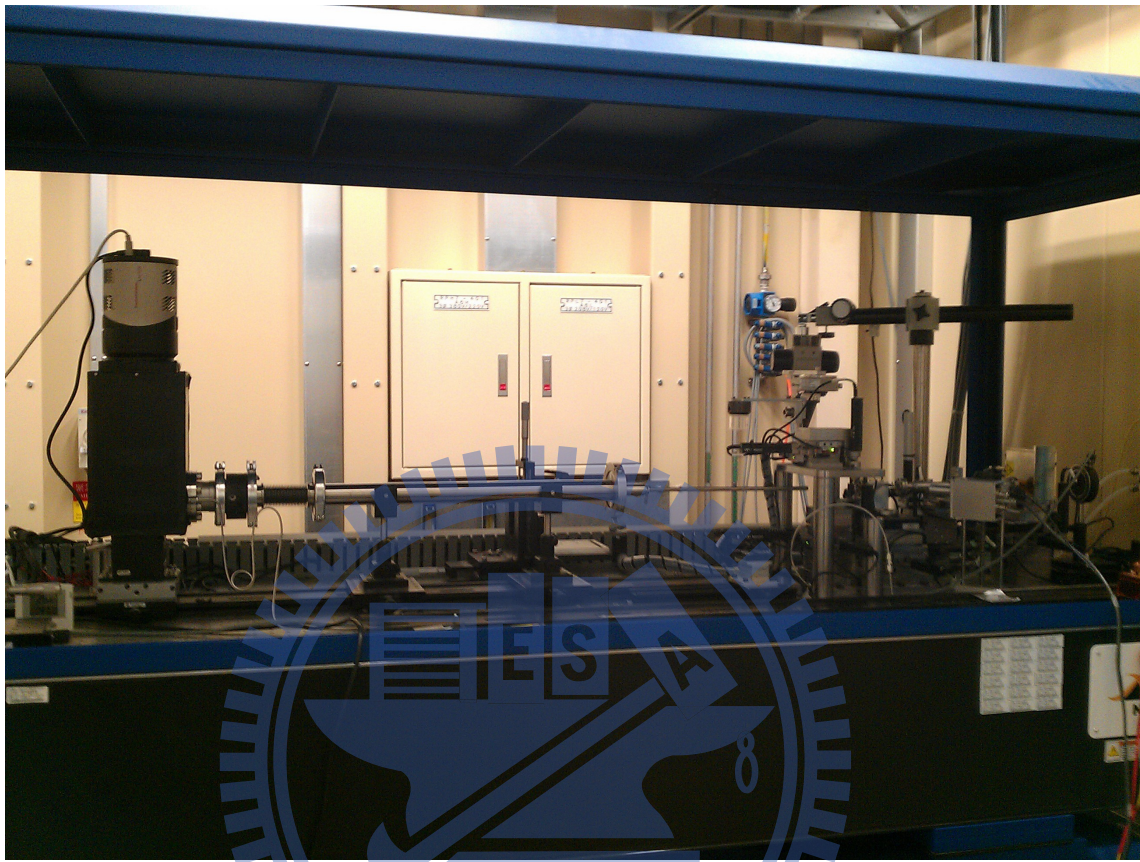
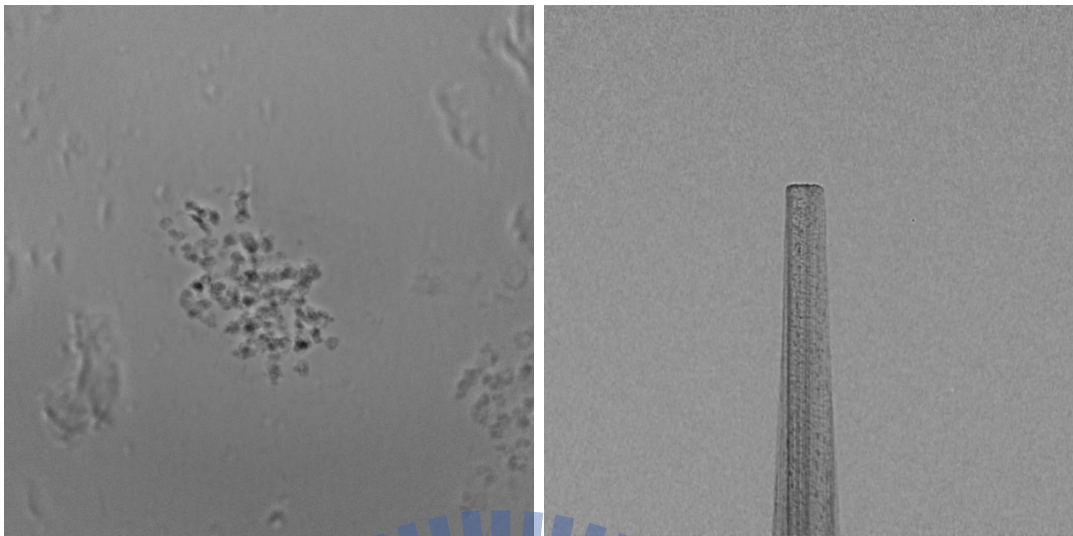


Figure 1.2: **The synchrotron X-ray microscope in NSRRC.** (a) The overview of microscope. (b) The X-ray emitter. (c) The object holder.



(a)

(b)



(c)

Figure 1.3: **The synchrotron X-ray images.** (a) An EMT cell. (b) A toothpick. (c) A mouse head.



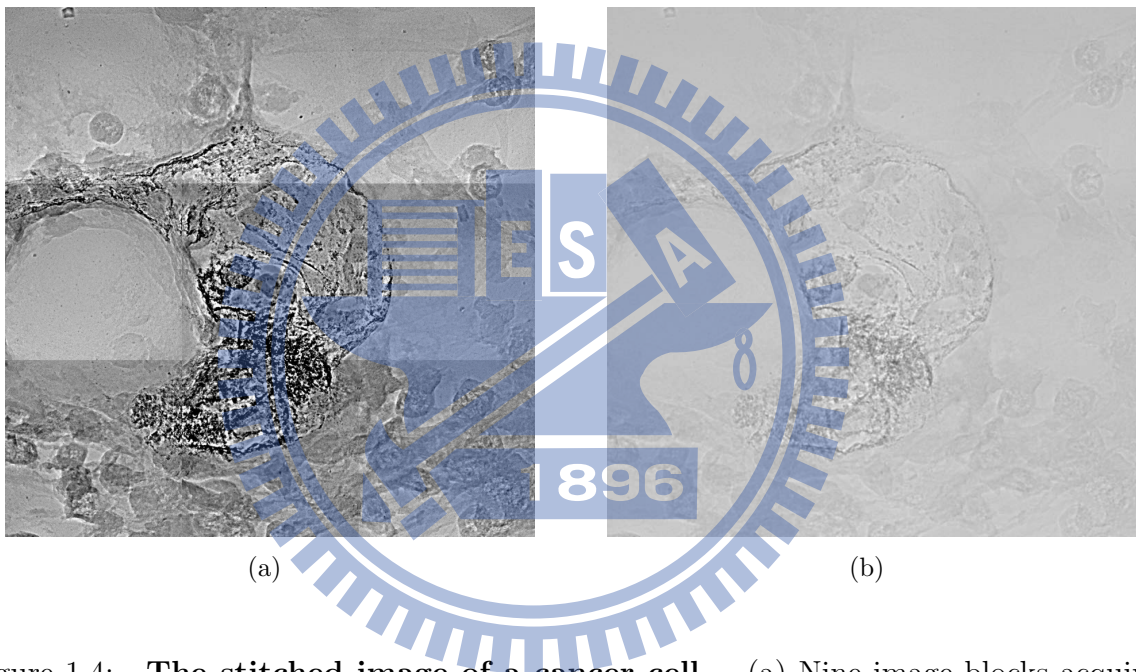


Figure 1.4: **The stitched image of a cancer cell.** (a) Nine image blocks acquired by synchrotron X-ray microscope. (b) The stitched result.

# Chapter 2

## Tomography Reconstruction

It is known that a series of X-ray projections around an object can be used to reconstruct a 3D volume data by using an appropriate reconstruction algorithms [4] (Fig. 2.1). In this study, two tomography reconstruction methods are used to reconstruct three-dimensional volumes. For completeness, the following sections review these two reconstruction methods

### 2.1 Filtered Back Projection

The Radon transform [8] expresses that using an apparatus of parallel-beam projection to gather the intensity of X-ray. Let  $f(x, y)$  be a two-dimensional function to represent a slice of an object. Given a projection angle  $\theta$ , The Radon transform is defined as follows.

$$p_{\theta}(x') = \int_{-\infty}^{\infty} f(y' \sin \theta + x' \cos \theta, -y' \cos \theta + x' \sin \theta) dy'. \quad (2.1)$$

$p_{\theta}(x')$  can be expressed a  $\theta - x$  plane called sinogram. The middle figure of Fig. 2.1 shows an example of sinogram generated from the Shepp-Logan phantom object [9]. The sinogram is an image in the plane of  $\theta$  and  $x$ . Each raw of the sinogram is retrieved from the X-ray images with the same  $y$  coordinate.

The *Filtered Back Projection*(FBP) algorithm based on the Fourier slice theorem [10] can estimate a tomographic image from a sinogram. The Fourier slice theorem in 2D space

is described as follows. Given a 2D function in spatial domain,  $f(x, y)$  (Fig. 2.2(a)), and let  $p_\theta(x')$  be the projection of  $f(x, y)$  in direction  $\theta$  where

$$x' = x \cos \theta + y \sin \theta, \quad (2.2)$$

and

$$y' = -x \sin \theta + y \cos \theta. \quad (2.3)$$

Let  $F(u, v)$  be the Fourier transform of  $f(x, y)$  (Fig. 2.2 (b)), and  $P_\theta(w)$  be the 1D Fourier transform of  $p_\theta(x')$ .  $P_\theta(w)$  is a line segment with orientation  $\theta$  in  $F(u, v)$  passing through the origin. Using the Fourier slice theorem, the projection of  $f(x, y)$  along orientation  $\theta$  can be obtained by taking the inverse Fourier transform of  $P_\theta(w)$ . To apply the fast Fourier transform (FFT) [11], a convention between Cartesian coordinate system ( $x$ - $y$  plane) and polar coordinate system ( $w$ - $\theta$  plane) is required. Let  $\mathcal{F}_2^{-1}$  be the 2D inverse Fourier transform, the relation between  $f(x, y)$  and  $F(u, v)$  can be expressed as follows.

$$\begin{aligned} f(x, y) &= \mathcal{F}_2^{-1}\{F(u, v)\} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{2\pi i (ux + vy)} du dv, \end{aligned} \quad (2.4)$$

where  $i = \sqrt{-1}$ . Let  $u = w \cos \theta$  and  $v = w \sin \theta$  to convert the  $u - v$  coordinate system to  $w$ - $\theta$  coordinate system. Eq. 2.4 can be rewritten as follows.

$$\begin{aligned} f(x, y) &= \int_0^{2\pi} \int_0^{\infty} F(w, \theta) e^{w 2\pi (x \cos \theta + y \sin \theta) i} w dw d\theta \\ &= \int_0^{2\pi} \left( \int_{-\infty}^{\infty} |w| F(w, \theta) e^{w 2\pi (x \cos \theta + y \sin \theta) i} dw \right) d\theta. \end{aligned} \quad (2.5)$$

Define

$$q_\theta(x') = \int_{-\infty}^{\infty} |w| F(w, \theta) e^{w 2\pi x' i} dw. \quad (2.6)$$

$q_\theta(x')$  is the inverse Fourier transform of a filtered line segment. This line segment passes through the origin of frequency domain and its angle from the horizontal axis is  $\theta$ . Then, a ramp waveform,  $|w|$ , is multiplied to this line segment.  $|w|$  is called the ramp filter

because a multiplication in the frequency domain is equal to a filtering process in the spatial domain. Finally, the FBP equation is obtained as follows.

$$f(x, y) = \int_0^\pi q_\theta(x \cos \theta + y \sin \theta) d\theta. \quad (2.7)$$

Eq. 2.7 is the fundamental theory of FBP. The implementation of FBP is described as follows.

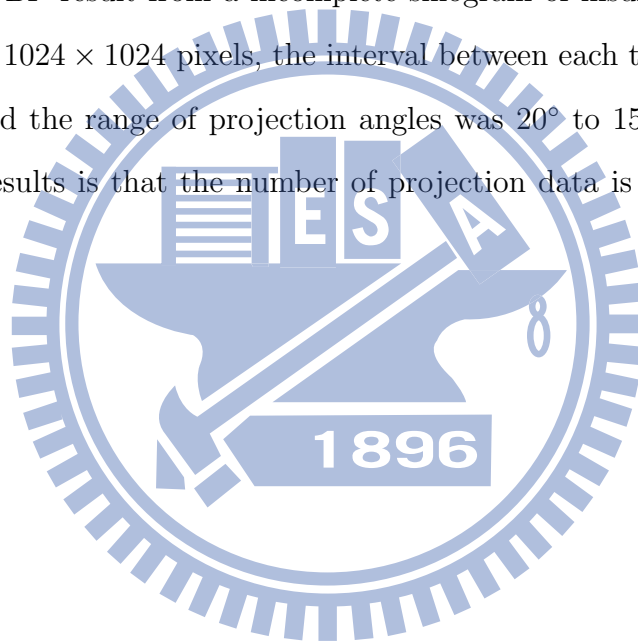
1. Apply 1D FFT,  $\mathcal{F}_1$ , to transform each row of the sinogram  $p_\theta(x')$ ,  
 $P_\theta(w) = \mathcal{F}_1\{p_\theta(x')\}$ .
2. Multiply the ramp filter,  $|w|$ , to  $P_\theta(w)$ .
3. Calculate the filtered projections,  $q_\theta$ , by applying  $\mathcal{F}_1^{-1}$  to the filtered  $P_\theta(w)$ .
4. Back project  $q_\theta$  for all  $\theta$  to the output image.

Fig. 2.3 shows the algorithm of FBP. According to the projection angle  $\theta$ , the back projection in Step 4 projects a ray for each element of  $q_\theta$  to the output image as shown in Fig 2.4(a). If a pixel of the output image is covered by a ray of  $q_\theta(x')$ , estimate the covered area and add the value of  $q_\theta(x')$  multiplied by the covered area to this pixel. Fig 2.4(b) shows the estimation of the pixel area covered by a ray. Given a re-projected ray, and there are several sampling points with uniform interval on this ray. For each pixel  $f(x, y)$ , if its perpendicular distance to the ray,  $d$ , is less than the pixel width, find the nearest two sampling points on the ray,  $R_a$  and  $R_b$ . Two areas of rectangles,  $W_a$  and  $W_b$ , then can be calculated by  $R_a$ ,  $R_b$ ,  $x$  and  $y$ . Therefore, the area covered by ray can be estimated by  $W_a + W_b$ .

FBP can be implemented on a parallel computing machine[12]. In this study, FBP was implemented on the GPU (graphic processing unit) and the multi-core processor. The GPU is a multi-processor chip of SIMD (single instruction multiple data) architecture. The GPU and CPU used in this study respectively are nVidia GTX 760 and i7 3.4 GHz with four cores. In the experiments, the test image was the Shepp-Logan phantom object with  $1024 \times 1024$  pixels as shown in Fig. 2.5(a). The interval between each two successive

projection angles was  $1^\circ$  and the range of projection angles was  $0^\circ$  to  $179^\circ$ . The CPU with single-core mode consumed 13.6 second to produce the reconstructed image. The computational time of CPU with four-core mode was 2.9 second. The computational time of GPU was 0.1 second. These experiments exhibited that the parallel computing can reduce the computation cost of FBP extremely.

If the number of projection data is insufficient, FBP could produce a poor result. Fig. 2.5(b) shows an FBP result from a incomplete sinogram of large angle between projections. The size of image was  $1024 \times 1024$  pixels, the interval between each two successive projection angles was  $10^\circ$ , and the range of the projection angles was  $0^\circ$  to  $179^\circ$ . Fig. 2.5(c) shows an FBP result from a incomplete sinogram of insufficient projections. The size of image was  $1024 \times 1024$  pixels, the interval between each two successive projection angles was  $1^\circ$ , and the range of projection angles was  $20^\circ$  to  $159^\circ$ . The main reason of these two poor results is that the number of projection data is far less than the size of output image.



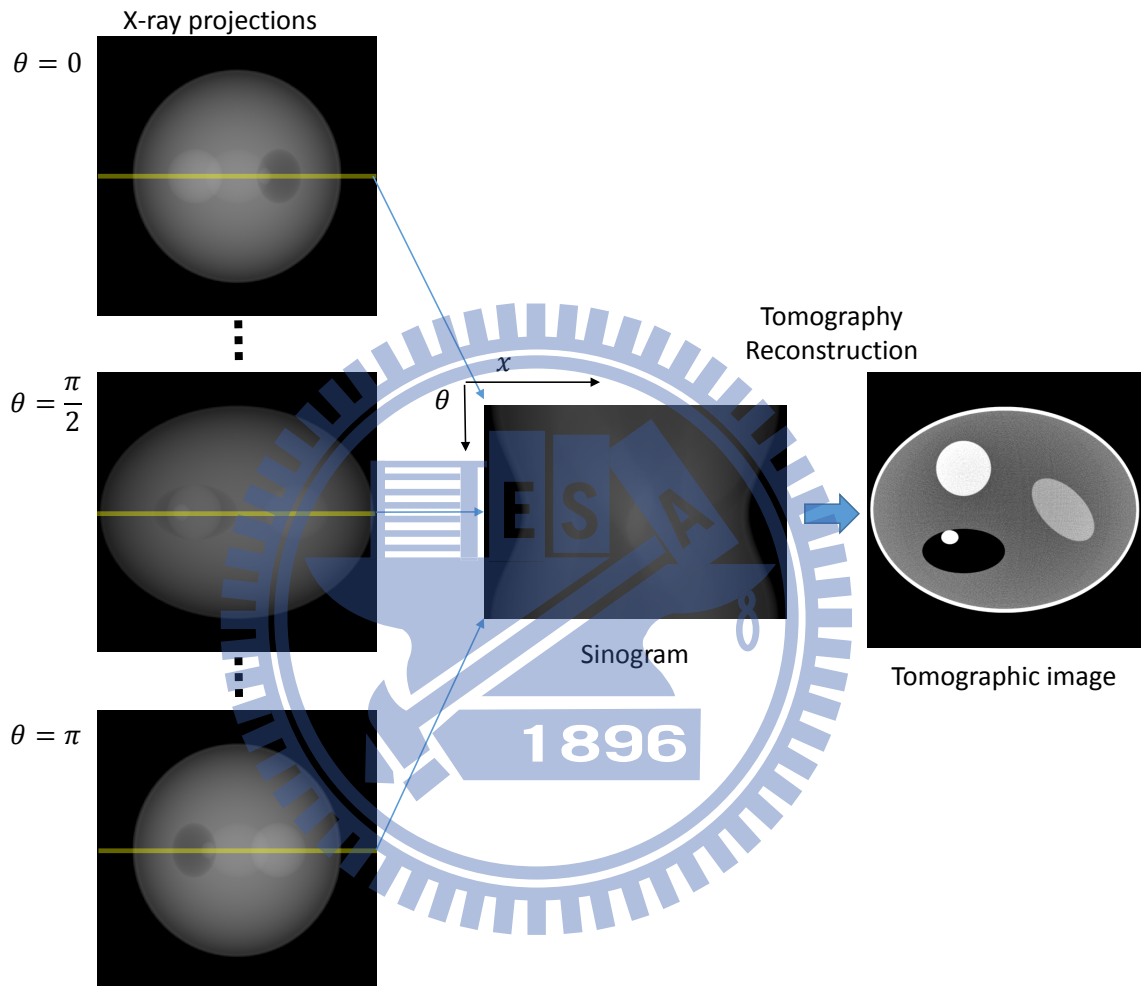


Figure 2.1: **The tomography reconstruction.** The middle figure shows a sinogram created from a series of X-ray projection images, where the object is a Shepp-Logan phantom and the projection range is 0 to  $\pi$ . A tomography reconstruction algorithm then can be applied to generate the tomographic image.

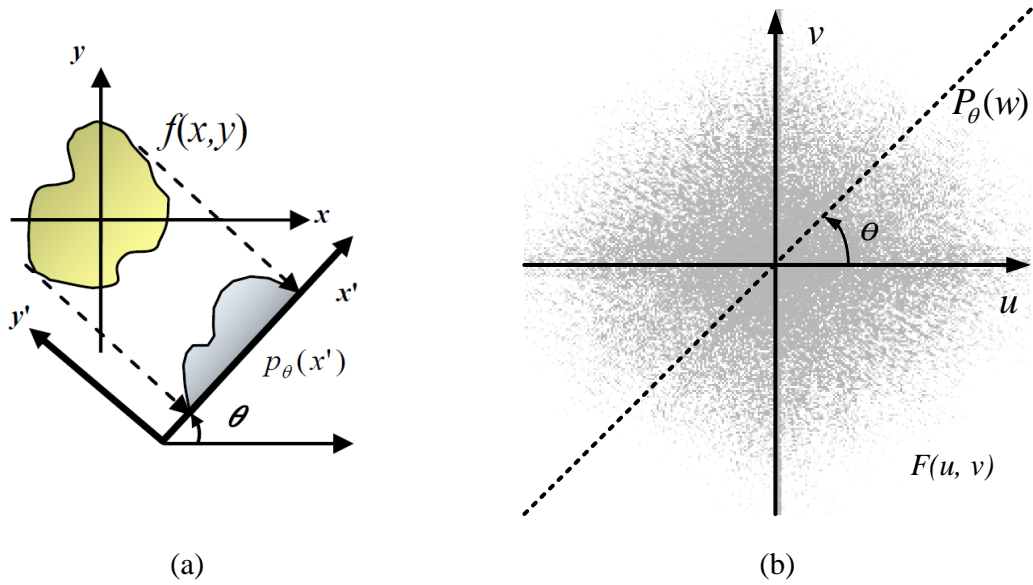


Figure 2.2: **Fourier slice theorem.** (a)  $p_\theta(x')$  is the projection of  $f(x,y)$  along  $\theta + \pi/2$ . (b) In the frequency domain, the line segment,  $P_\theta(w)$ , is the 1D Fourier transform of  $p_\theta(x')$ .

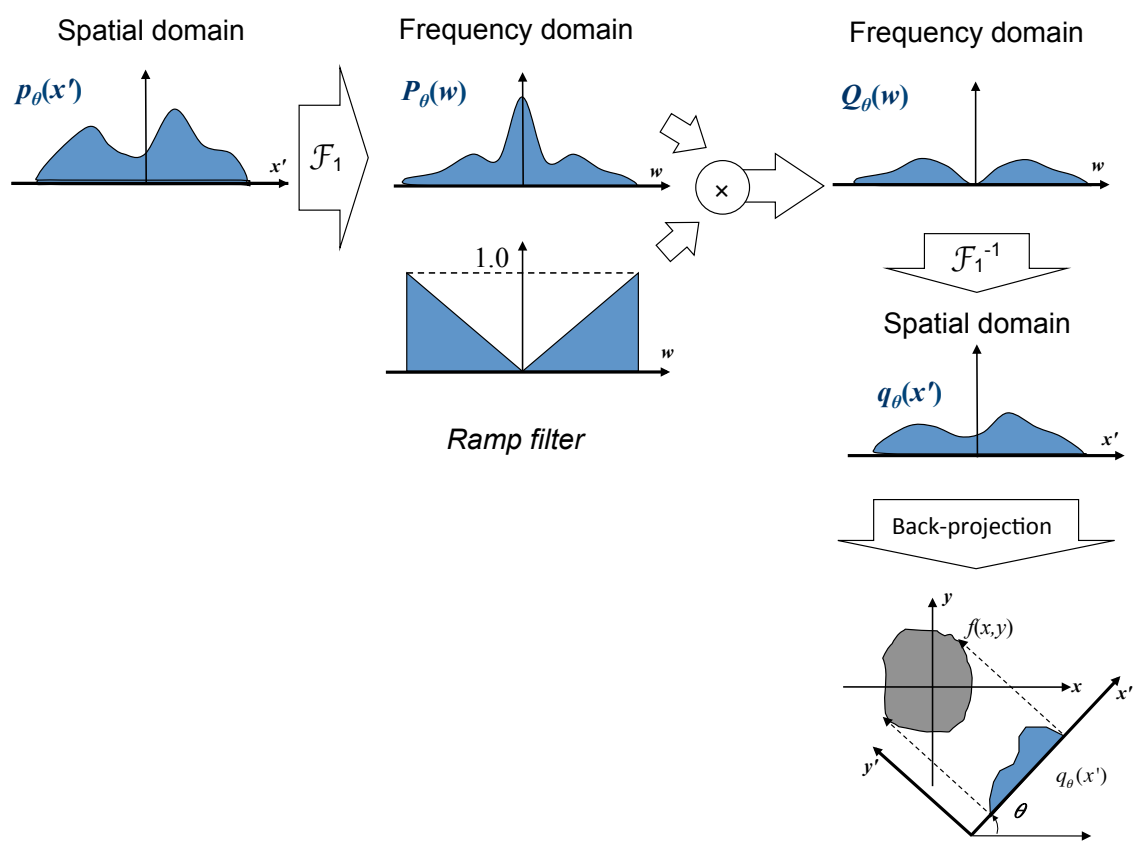


Figure 2.3: **The FBP algorithm.**



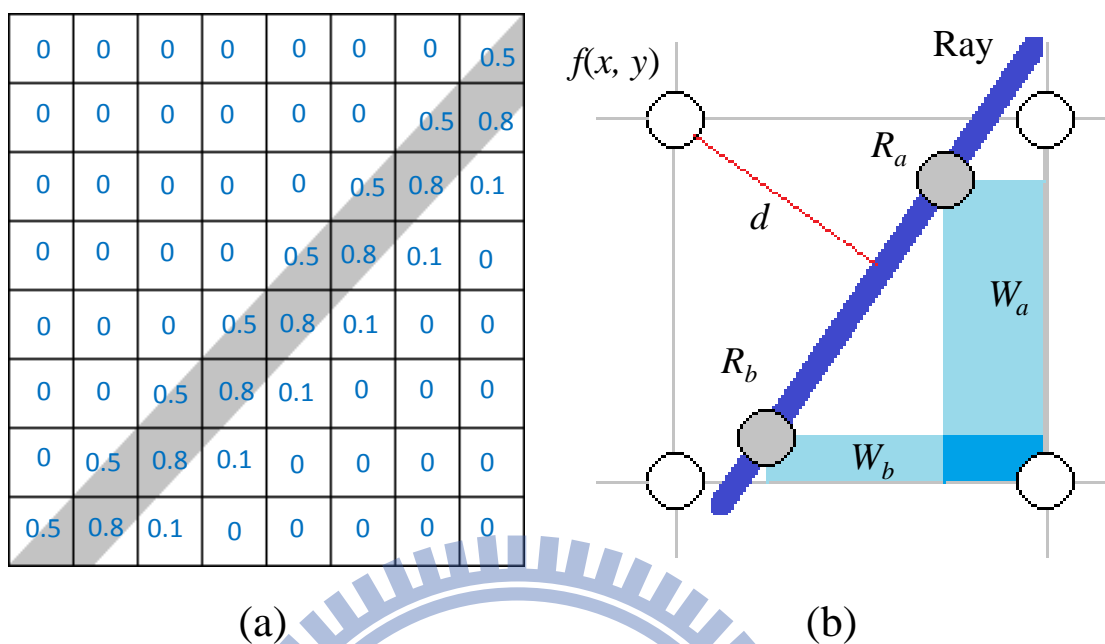


Figure 2.4: **The pixel area covered by ray.** (a) In back projection, each filtered projection is re-projected on the output image. The number in each cell represents the area covered by ray. (b) The approximation of pixel area covered by ray.

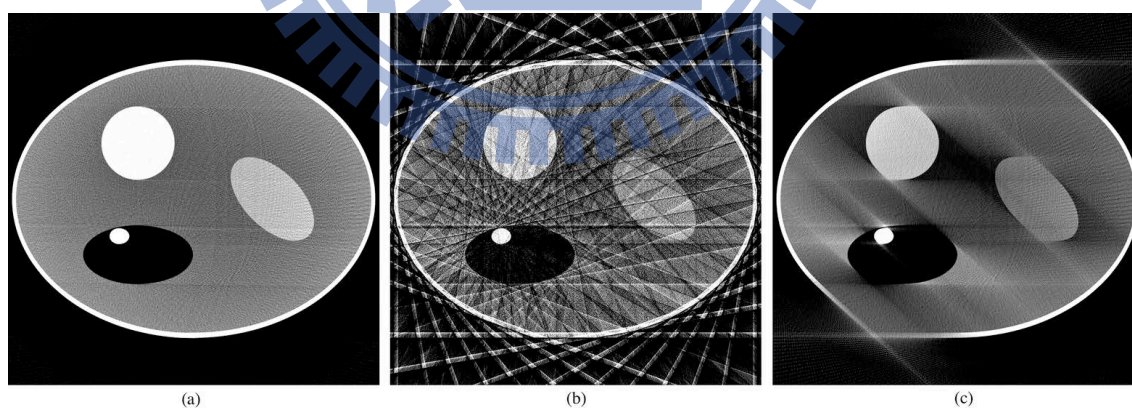


Figure 2.5: **The results of FBP.** The object was the SheppLogan phantom. (a) The quality result reconstructed from the range and interval of projection angles were  $[0^\circ, 179^\circ]$  and  $1^\circ$  respectively. (b) The result of large angle between projections. The range and interval of projection angles were  $[0^\circ, 179^\circ]$  and  $10^\circ$  respectively. (c) The result of insufficient projections. The range and interval of projection angles were  $[20^\circ, 159^\circ]$  and  $1^\circ$  respectively.

## 2.2 Algebraic Reconstruction Technique

The *algebraic reconstruction technique* (ART) is an iterative method of tomography reconstruction [13]. There are two main differences between the ART and FBP. First, all estimation processes of ART is in spatial domain rather than in frequency domain. Second, the ART uses several iterations to correct the errors to reconstruct the image. In most cases, the ART can achieve a better reconstruction than the FBP can get. However, the computational cost of ART is much larger than the FBP.

Fig. 2.6 shows how does the ART work. Assume that the size of output image is  $N$  pixels, and the size of sinogram is  $M$  projections. Let the reconstructed image and the pixel areas covered by  $j$ th ray respectively be two  $N$ -dimensional vector,  $\vec{\mathbf{f}} = [f_1, f_2, \dots, f_N]$  and  $\vec{\mathbf{w}}_j = [w_1^j, w_2^j, \dots, w_N^j]$ . The projection measured from the  $j$ th ray,  $p_j$ , then can be expressed as follows.

$$p_j = \vec{\mathbf{w}}_j \cdot \vec{\mathbf{f}}. \quad (2.8)$$

Assume that  $\vec{\mathbf{f}}_i$  is the result of  $i$ -th iteration. The line  $\overline{OU}$  in Fig. 2.6 is the normalized vector of  $\vec{\mathbf{w}}_j$ ,

$$\overline{OU} = \frac{\vec{\mathbf{w}}_j}{\sqrt{\vec{\mathbf{w}}_j \cdot \vec{\mathbf{w}}_j}}, \quad (2.9)$$

and

$$|\overline{OF}| = \vec{\mathbf{f}}_{i-1} \cdot \overline{OU}. \quad (2.10)$$

The lengths of  $\overline{OA}$  and  $\overline{HG}$  then can be obtained by the following equations.

$$|\overline{OA}| = \overline{OU} \cdot \overline{OC} = \frac{p_j}{\sqrt{\vec{\mathbf{w}}_j \cdot \vec{\mathbf{w}}_j}} \quad (2.11)$$

$$|\overline{HG}| = \overline{OF} - \overline{OA} = \frac{\vec{\mathbf{w}}_j \cdot \vec{\mathbf{f}}_{i-1} - p_j}{\sqrt{\vec{\mathbf{w}}_j \cdot \vec{\mathbf{w}}_j}} \vec{\mathbf{w}}_j. \quad (2.12)$$

Finally, the result of  $i$ -th round,  $\vec{\mathbf{f}}_i$ , then can be obtained by the following equation.

$$\vec{\mathbf{f}}_i = \vec{\mathbf{f}}_{i-1} + \frac{p_i - \vec{\mathbf{w}}_j \cdot \vec{\mathbf{f}}_{i-1}}{\vec{\mathbf{w}}_j \cdot \vec{\mathbf{w}}_j} \vec{\mathbf{w}}_j. \quad (2.13)$$

Eq. 2.13 is the recursion for the ART. The implementation of ART consists of four steps as described as follows. First, calculate the  $\vec{\mathbf{w}}_j$  for the  $j$ -th ray. The second step is called forward-projection that estimates a projection value by inner product of  $\vec{\mathbf{f}}_{i-1}$  and  $\vec{\mathbf{w}}_j$ . The correction then can be obtained by subtraction of  $p_j$  and  $\vec{\mathbf{w}}_j \cdot \vec{\mathbf{f}}_{i-1}$ . Finally, back project this correction to  $\vec{\mathbf{f}}_{i-1}$ . These four steps are iteratively executed until the correction is smaller than a threshold or the number iteration is larger than a limitation.

However, the original method of ART is not practical because it iterates for each single projection such that the computational time is extreme high. Eq. 2.13 can be rewritten by the following equation.

$$\vec{\mathbf{f}}_i = \vec{\mathbf{f}}_{i-1} + \sum_{j=1}^K \frac{p_i - \vec{\mathbf{w}}_j \cdot \vec{\mathbf{f}}_{i-1}}{\vec{\mathbf{w}}_j \cdot \vec{\mathbf{w}}_j} \vec{\mathbf{w}}_j. \quad (2.14)$$

In Eq. 2.14, the output of each iteration is updated by  $K$  rays to achieve better performances than Eq. 2.13. This improved method is called *Simultaneous Iterative Reconstruction Technique* (SIRT). There are many software systems of SIRT have been developed. Tomo3D is a parallel-computing software system of SIRT [14, 15]. Tomo3D can be executed on an environment of multi-core CPU and GPU. This study used tomo3D to reconstruct the large tomographic images. Table 2.1 shows the time comparisons of FBP on GPU and Tomo3D. Two test data sets named HeLa and Kidney that were a HeLa cell and a kidney of mice respectively. In HeLa, the range of projection angle was  $\pm 70^\circ$ , the angle interval was  $1^\circ$ , and the size of output was  $1024 \times 1024$  pixels. In Kidney, the range of projection angle was  $\pm 90^\circ$ , the angle interval was  $1^\circ$ , and the size of output was  $4880 \times 4880$  pixels. Note that Tomo3D used 30 iterations to reconstruct each results. Fig. 2.7 shows the images reconstructed by FBP and Tomo3D. As shown in Fig. 2.7, the quality of SIRT is better than FBP when the image size is large.



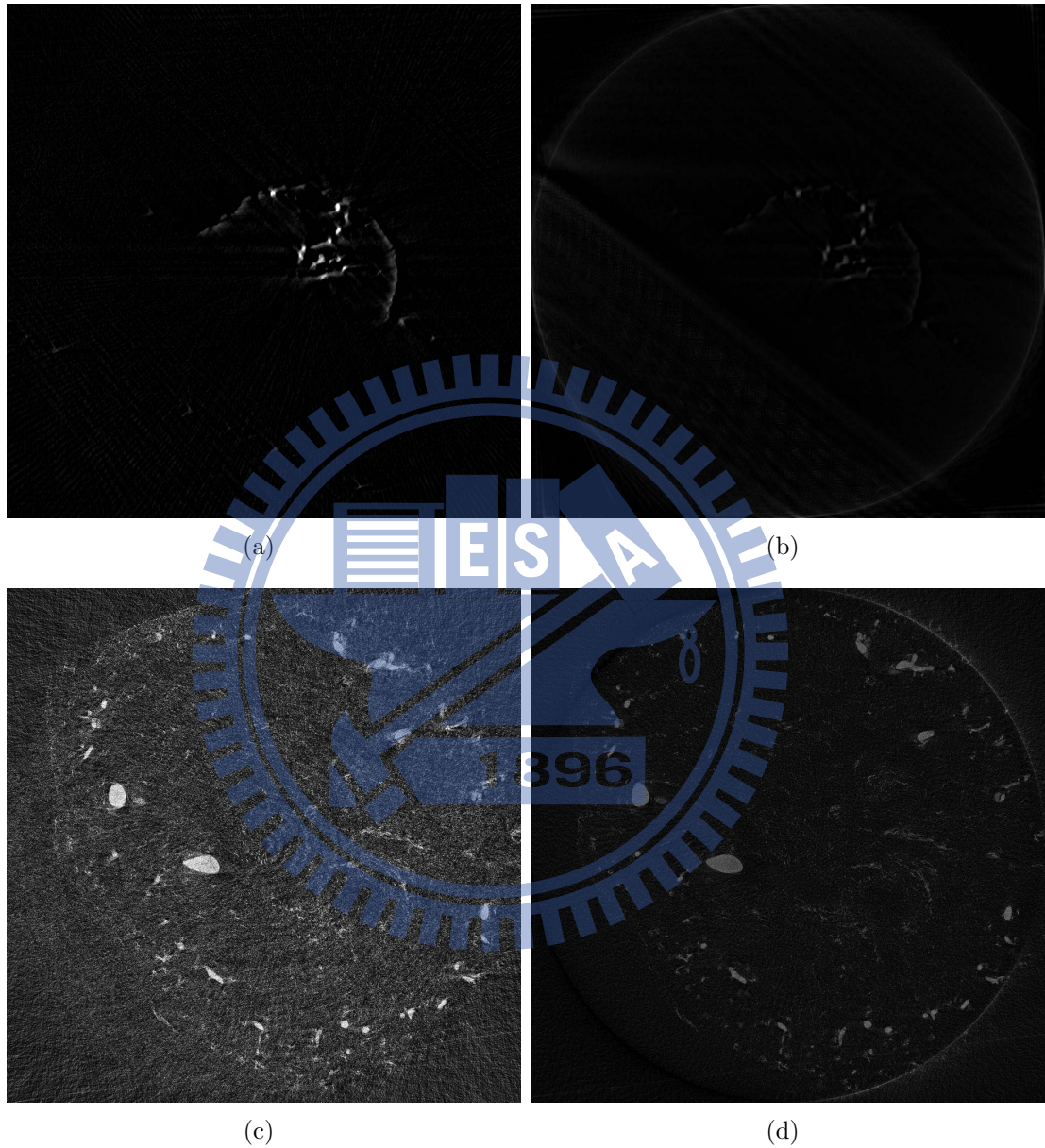


Figure 2.7: **The tomographic images of FBP and SIRT.** (a) and (b) are the images of HeLa cell, and (c) and (d) are the images of mice kidney, respectively. (a) and (c) were reconstructed by FBP on GPU, and (b) and (d) were reconstructed by Tomo3D.

Table 2.1: **The time comparisons of FBP and SIRT.**

Data	Sinogram size (pixels)	Output size (pixels)	Time cost of FBP on GPU (sec)	Time cost of Tomo3D (sec)
HeLa	$1024 \times 140$	$1024 \times 1024$	0.1	0.61
Kidney	$4880 \times 180$	$4880 \times 4880$	9.1	63.6

The run time statistics were obtained by using Intel i7 3.4GHz CPU, 12GB main memory, nVidia GTX 760 GPU, and running Windows 8.1. 30 iterations for SIRT.



# Chapter 3

## Visualization

Visualization is a technique that generates images, animations, or figures, enabling users to understand and analyze a multidimensional data set. Volume visualization techniques can be divided into two types, surface rendering and ray-casting rendering. Surface rendering requires geometric primitives to construct the isosurfaces that represent the set of points of a selected constant value. These isosurfaces are then rendered using a shading algorithm. Lorensen and Cline [16] proposed the marching cube method. In this approach, 15 primitive types of polygons were defined to create isosurfaces from a unit cube.

Unlike surface rendering, ray-casting rendering produces rendered images by imitating X-rays passing through an object. The 2D projections are obtained by integrating the voxel values on the lines along the view direction. The integral is an optical model proposed by Max [17]. This study uses two ray-casting rendering methods, texture-based volume rendering and Fourier volume rendering, to verify the reconstructed volume data.

### 3.1 Texture-Based Volume Rendering

Texture-based volume rendering (TVR) [5] is the most popular visualization technique in recent years. This method uses graphics hardware (GPU) and texture mapping to accelerate the integration of ray casting. A texture is a data set stored in a video memory space of graphics hardware. To use the texture, a proxy polygon is constructed. Each vertex of the proxy polygon is assigned a coordinate of texture. Note that the range of



each component of texture coordinate is in  $[0, 1]$ . The graphics hardware then uses linear interpolation to compute the texture coordinate of each pixel on the drawing area of proxy polygon during the rendering of proxy polygon. According to these interpolated texture coordinates, the texture then can be mapped to the drawing area of proxy polygon. Fig. 3.1 shows the procedure of texture mapping.

Before the volume rendering, a volume must be stored into the video memory as a three-dimensional texture. Given a viewpoint and a bounding cube of the volume, TVR then constructs a set of proxy polygons that parallel the view plane and all vertices are on the edges of bounding cube. Fig. 3.2 shows the proxy polygons for TVR. These proxy polygons are used to sample voxels from the three-dimensional texture during the rendering. In the voxel sampling stage, the color and opacity of each voxel can be decided by a transfer function to enhance the region of interest. Next, sort these polygons by their distances to the viewpoint. As shown in Fig. 3.2, Polygon N is the farthest polygon and Polygon 1 is the nearest polygon. Finally, the rendering result can be generated by using the alpha blending method to draw the proxy polygons from Polygon N to Polygon 1. Fig. 3.3 shows a rendering result of two volumes of  $256^3$  voxels, a tomography of head and a car engine. This result was rendered by TVR with Phong shading [18] and shadowing [19].

To enhance the region of interest, designing an appropriate transfer function is vital. Numerous practical and effective transfer functions have been proposed. For example, Engel et al. [20] proposed the pre-integrated 2D transfer function to improve the rendering quality. Kindlmann et al. [21] used the surface curvature to determine the contour thickness for non-photorealistic volume rendering. Lum and Ma [22] proposed the lighting transfer function to enhance the boundary surfaces of the region of interest. Caban and Rheingans [23] designed a transfer function that considered the texture of a feature. Additionally, Correa and Ma developed a number of multidimensional transfer functions for various purposes [24], [25], [26]. To conveniently modify a transfer function, Wu and Qu [27] devised three operations for combining transfer functions. Zhou and Takatsuka [28] utilized the contour tree and residue flow model to automatically generate a harmonic

color transfer function.

Because TVR samples volume data from the video memory, the original TVR method cannot render a large volume data if the size of volume data is larger than  $1024^3$  voxels. To solve this problem, the large volume is divided to several blocks before the data loading. These blocks are then loaded into the video memory according to their visibilities. There are two methods proposed to construct a data structure for large volume data. Kniss et al. used level of detail (LOD) to construct a multi-resolution data structure for a large volume [29]. Wang et al. used the entropy to divid a large volume to several blocks [30].

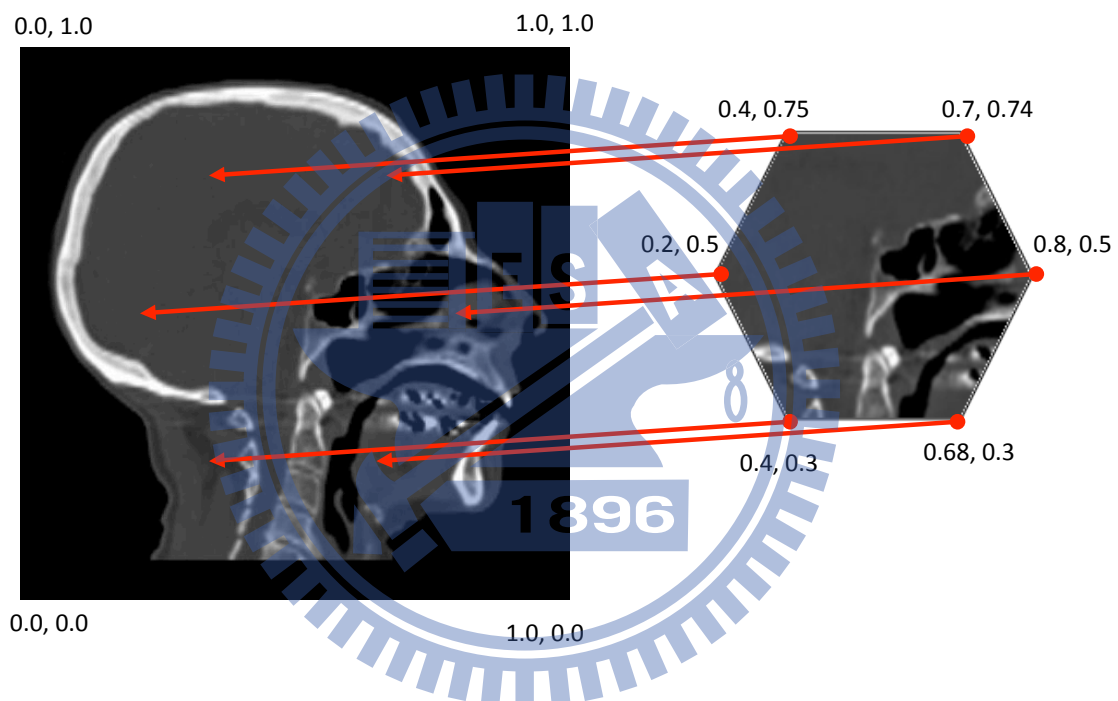


Figure 3.1: **Texture mapping.** The image on left side is a 2D texture of a tomographic head. The image on right side is a proxy polygon. Each vertex of proxy polygon is assigned a texture coordinate. The range of texture coordinate is (0, 0) to (1, 1).

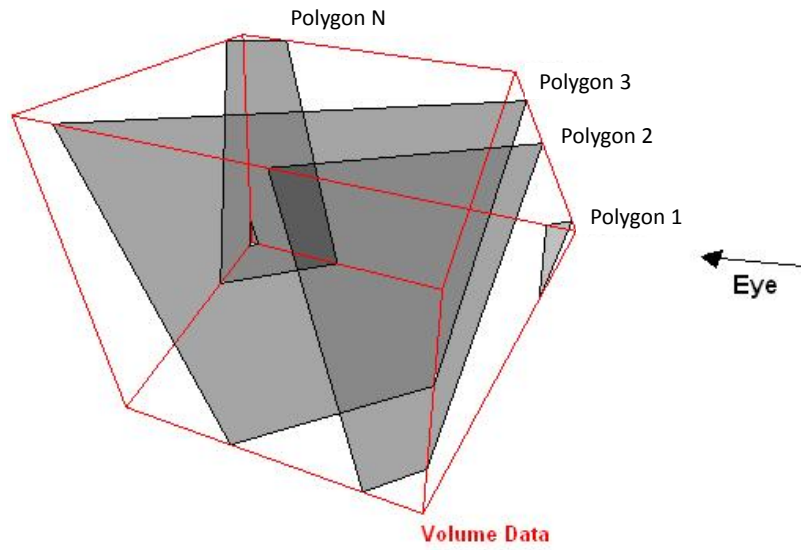


Figure 3.2: **The proxy polygons in TVR** Each proxy polygon parallels the view plane and each vertex is exactly on the edges of bounding cube.

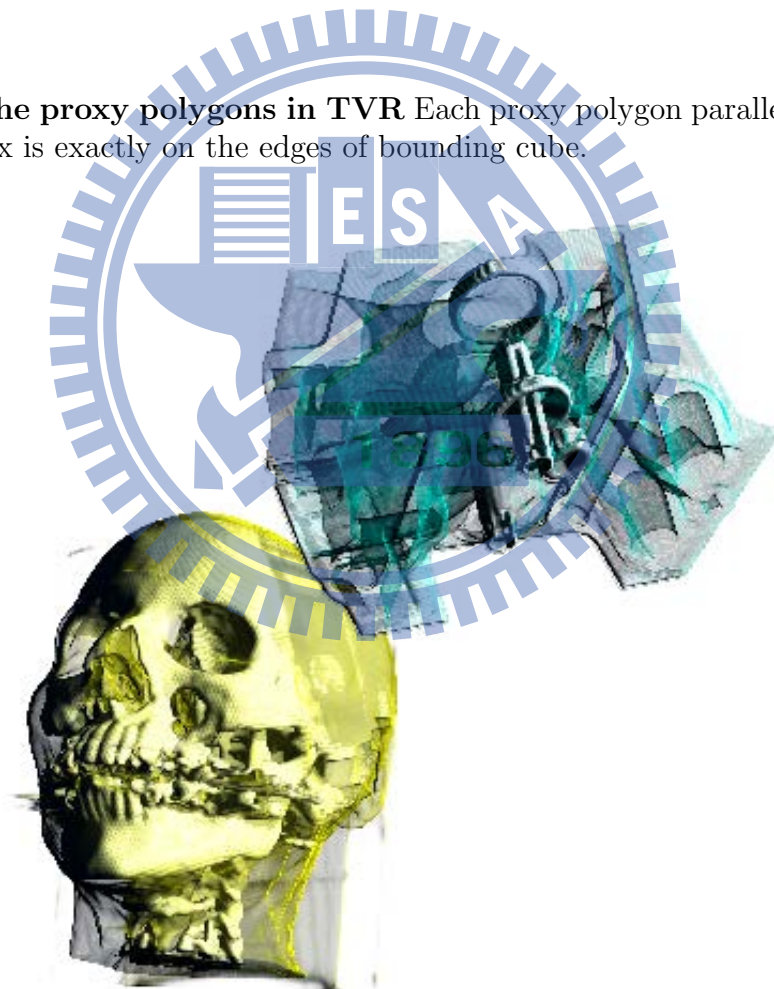


Figure 3.3: **A result of TVR** This result was generated by TVR with Phong shading and shadowing. Two volume of  $256^3$  voxels were rendered, a CT head and car engine.

## 3.2 Fourier Volume Rendering

Fourier Volume Rendering (FVR)[6], [31], [7] is a volume rendering method based on the Fourier slice theorem. With a 3D volume data, the FVR algorithm requires  $O(n^2 \log n)$  time to generate a result. Because it requires time less than  $O(n^3)$  does, FVR is preferred for designing a real-time rendering algorithm with a preprocessing step. The Fourier slice theorem has been described in Section 2.1. The Fourier slice theorem holds in 3D space (Fig. 3.4). Let  $F(u, v, w)$  be the Fourier transform of 3D volume  $f(x, y, z)$ . Given the view direction  $\mathbf{v}$ , the projection  $P_{\mathbf{v}}(x, y)$  is the projection along  $\mathbf{v}$ . Using the Fourier slice theorem,  $P_{\mathbf{v}}(x, y)$  is obtained by taking the inverse Fourier transform of  $P_{\mathbf{v}}(u', v')$ , where  $P_{\mathbf{v}}(u', v')$  is the frequency signals in a 2D plane passing through the origin in  $F(u, v, w)$ .

The FVR method can be summarized in the following equations. Given the volume data  $f(\mathbf{x})$ ,  $\mathbf{x} \in \mathbf{R}^3$ , we define  $\Pi_{\mathbf{v}}$  as an operator that performs FVR from the viewing direction  $\mathbf{v}$ . The FVR of  $f(x)$  from  $\mathbf{v}$  can be presented as:

$$\begin{aligned} I &= \Pi_{\mathbf{v}}(f(\mathbf{x})) \\ &= \mathcal{F}_2^{-1}(\mathcal{F}_3(f(\mathbf{x}))\delta_{\mathbf{v}}). \end{aligned} \tag{3.1}$$

In the first part of Eq. (3.1),  $I$  is the projection of  $f(\mathbf{x})$  from the viewing direction  $\mathbf{v}$ ; in the second part,  $\delta_{\mathbf{v}}$  restricts the spectrum of the 3D Fourier transform,  $\mathcal{F}_3$ , to a plane passing through the origin and perpendicular to  $\mathbf{v}$ . We then take the 2D inverse Fourier transform,  $\mathcal{F}_2^{-1}$ , to obtain the projection  $I$ .

Assuming that the Fourier transform of volume data is available, the inverse Fourier transform of a slice from the frequency domain, which passes through the origin and is perpendicular to the view direction, is the projection of the volume along the view direction. This projection is the exact result of direct volume rendering. Using FVR, because only the inverse Fourier transform of a slice must be calculated, a volume of  $n^3$  voxels can be rendered in  $O(n^2 \log n)$  time, providing the Fourier transform of the volume is available. To accelerate the speed of rendering, Viola et al. [32] presented an implementation of FVR using GPU with the advantage of parallel computing to accelerate

FVR computation.

Because resampling in frequency domain could produce replicas in spatial domain [10], Malzbender designed several filters to reduce the artifacts caused by the resampling in the frequency domain [7]. Levoy [31] presented three shading models for FVR, including depth cueing, directional lighting, and specular reflections. Combined with Levoy’s shading model, Entezari et al. [33] used the spherical harmonic function to approximate cubic illumination shading for FVR.

To enhance the region of interest, designing an appropriate transfer function is vital. Numerous practical and effective transfer functions have been proposed. For example, Engel et al. [20] proposed the pre-integrated 2D transfer function to improve the rendering quality. Kindlmann et al. [21] used the surface curvature to determine the contour thickness for non-photorealistic volume rendering. Lum and Ma [22] proposed the lighting transfer function to enhance the boundary surfaces of the region of interest. Caban and Rheingans [23] designed a transfer function that considered the texture of a feature. Additionally, Correa and Ma developed a number of multidimensional transfer functions for various purposes [24], [25], [26]. To conveniently modify a transfer function, Wu and Qu [27] devised three operations for combining transfer functions. Zhou and Takatsuka [28] utilized the contour tree and residue flow model to automatically generate a harmonic color transfer function.

For any volume rendering algorithm that processes volume data in a spatial domain, the transfer function can be applied to the volume, and calculate the rendered result within the same time bound. However, for FVR with frequency domain data, re-rendering the volume after applying a transfer function within the same  $O(n^2 \log n)$  time bound is challenging. A naive approach would be to employ one of the following two methods: apply the transfer function to the volume data and recompute the Fourier transform, or implement the convolution operation in the frequency domain. The time required for both approaches exceeds  $O(n^2 \log n)$ . To improve the re-rendering time after the application of a transfer function, Nagy et al. [34] proposed designing a binary classification transfer function for FVR. They used the Fourier series of a step function to eliminate unwanted

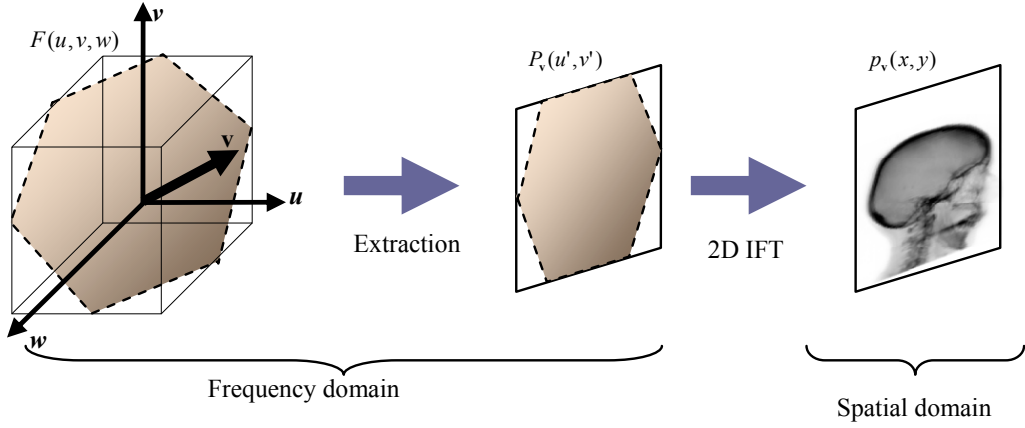


Figure 3.4: **The FVR algorithm.** In the frequency domain, the 2D frequency function  $P_v(u', v')$  passing through the origin of  $F(u, v, w)$  is extracted. The projection  $p_v(x, y)$  is obtained by taking the 2D inverse Fourier transform of  $P_v(u', v')$ .

voxels.

### 3.2.1 Transfer Function Design for FVR

This study proposes a voxel classification method for FVR [35, 36]. This method uses the Bézier curve and B-spline to construct transfer functions such that the voxel classification can be immediately adjusted in rendering time. Let  $f(x) = a \cdot f_1(x) + b \cdot f_2(x)$  where  $f_1(x)$  and  $f_2(x)$  are two integrable functions. The Fourier transform,  $\mathcal{F}$ , possesses the linear combination property as shown in Eq. (3.2),

$$\mathcal{F}(f(x)) = \mathcal{F}(a \cdot f_1(x) + b \cdot f_2(x)) = a \cdot \mathcal{F}(f_1(x)) + b \cdot \mathcal{F}(f_2(x)), \quad (3.2)$$

where  $a$  and  $b \in \mathbf{R}$  are two constants. This property can be used to efficiently implement the shading model and transfer function of FVR.

The shading model of FVR proposed by Levoy [31] employs the linear combination property advantage. Let  $g$  be a shading model applied to the volume data  $f(\mathbf{x})$ . From Eq. (3.1), the FVR of the resulting volume can be presented as

$$I = \Pi_v(g(f(\mathbf{x})))$$

$$= \mathcal{F}_2^{-1}(\mathcal{F}_3(g(f(\mathbf{x})))\delta_{\mathbf{v}}). \quad (3.3)$$

If  $g(f(\mathbf{x}))$  can be decomposed into a linear combination of  $m$  terms, then:

$$\begin{aligned} g(f(\mathbf{x})) &= h_0 \cdot g_0(f(\mathbf{x})) + h_1 \cdot g_1(f(\mathbf{x})) \\ &+ \dots + h_{m-1} \cdot g_{m-1}(f(\mathbf{x})), \end{aligned} \quad (3.4)$$

where  $h_i$  is the shading factor. Substituting Eq. (3.4) into  $g(f(\mathbf{x}))$  in Eq. (3.3), we obtain Eq. (3.5).

$$\begin{aligned} I &= \Pi_{\mathbf{v}}(h_0 \cdot g_0(f(\mathbf{x})) + h_1 \cdot g_1(f(\mathbf{x})) \\ &+ \dots + h_{m-1} \cdot g_{m-1}(f(\mathbf{x}))) \\ &= FT_2^{-1}((h_0 \mathcal{F}_3(g_0(f(\mathbf{x}))) \\ &+ h_1 \cdot \mathcal{F}_3(g_1(f(\mathbf{x}))) \\ &+ \dots + h_{m-1} \cdot \mathcal{F}_3(g_{m-1}(f(\mathbf{x}))))\delta_{\mathbf{v}}) \\ &= h_0 \cdot \Pi_{\mathbf{v}}(g_0(f(\mathbf{x}))) + h_1 \cdot \Pi_{\mathbf{v}}(g_1(f(\mathbf{x}))) \\ &+ \dots + h_{m-1} \cdot \Pi_{\mathbf{v}}(g_{m-1}(f(\mathbf{x}))) \\ &= \sum_{i=0}^{m-1} h_i \cdot \Pi_{\mathbf{v}}(g_i(f(\mathbf{x}))). \end{aligned} \quad (3.5)$$

In Eq. (3.5), the rendered result is the summation of  $m$  weighted FVR results. Therefore, recomputing the 3D Fourier transform is not required if  $\Pi_{\mathbf{v}}g_i(f(\mathbf{x}))$ ,  $i = 0, 1, \dots, m-1$ , in Eq. (3.5) are available.

### 3.2.2 Bézier curve Transfer Function

A Bézier curve defined by  $m$  control points is shown as Eq. (3.6):

$$B(u) = \sum_{i=0}^{m-1} w_i b_{i,m-1}(u), \quad (3.6)$$



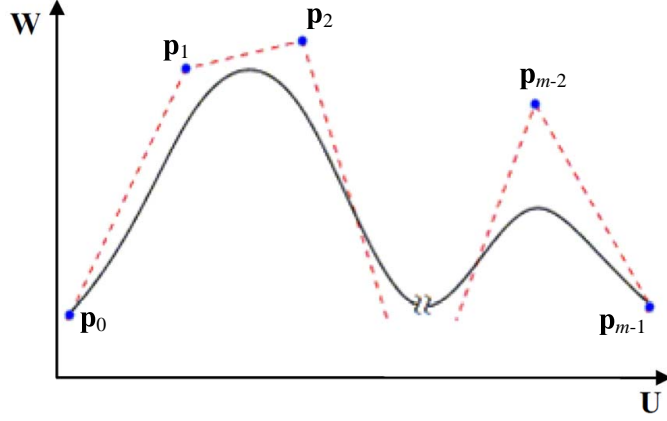


Figure 3.5: **A spline with  $m$  control points can be used to present the transfer function.** The shape of a spline curve can be modified by changing the height of the control points using a graphic user interface system.

where  $u$  and  $w_i \in \mathbf{R}$ , and

$$b_{i,m-1}(u) = \binom{m-1}{i} (1-u)^{m-i-1} u^i. \quad (3.7)$$

As shown in Fig. 3.5, a Bézier curve is defined in the  $\mathbf{U}$ - $\mathbf{W}$  coordinate system, where the  $\mathbf{U}$ -axis corresponds to the voxel value and the  $\mathbf{W}$ -axis corresponds to the shading weight of the voxel value. The Bézier curve consists of  $m$  control points,  $\mathbf{p}_i = (u_i, w_i)$ ,  $i = 0, 1, \dots, m-1$ ,  $u_i = u_0 + ih$  and  $h = \frac{1}{m-1}$ . By adjusting  $w_i$ , we edit the transfer function. Combining Eqs. (3.6), (3.3), and (3.5), FVR with a Bézier curve transfer function is written as

$$\begin{aligned} I &= \mathcal{F}_2^{-1}(\mathcal{F}_3(\sum_{i=0}^{m-1} w_i b_{i,m-1}(f(\mathbf{x})))\delta_{\mathbf{v}}) \\ &= \mathcal{F}_2^{-1}((\sum_{i=0}^{m-1} w_i \mathcal{F}_3(b_{i,m-1}(f(\mathbf{x}))))\delta_{\mathbf{v}}) \\ &= \sum_{i=0}^{m-1} w_i \Pi_{\mathbf{v}}(b_{i,m-1}(f(\mathbf{x}))). \end{aligned} \quad (3.8)$$

According to Eq. (3.8), the FVR of a volume after applying the Bézier curve transfer function of  $m$  control points is the summation of  $m$  weighted FVR results.

### 3.2.3 B-Spline Transfer Function

One disadvantage of the Bézier curve transfer function is its inability to enable good localized control of the curve shape. Modifying a control point may alter the shape of a significant portion of the Bézier curve. Compared to the Bézier curve, a B-spline [37] is more useful in controlling the curve shape.

A  $k$ -degree B-spline is defined in Eq. (3.9):

$$R(u) = \sum_{i=0}^{m-1} w_i r_{i,k}(u), \quad (3.9)$$

where

$$r_{i,k}(u) = \frac{s_{i,k}(u)}{\sum_{j=0}^{m-1} s_{j,k}(u)} \quad (3.10)$$

and  $s_{i,k}(u)$  is the B-spline basis function. Let a non-decreasing sequence  $\mathbf{T} = \{t_i | t_i \leq t_{i+1}, i = 1, \dots, l-1\}$  be the knot vector.  $s_{i,k}(u)$  is recursively defined as Eq. (3.11):

$$s_{i,k}(u) = \frac{u - t_i}{t_{i+k} - t_i} s_{i,k-1}(u) + \frac{t_{i+k+1} - u}{t_{i+k+1} - t_{i+1}} s_{i+1,k-1}(u), \quad (3.11)$$

and

$$s_{i,0} = \begin{cases} 1 & \text{if } t_i \leq u < t_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \quad (3.12)$$

Given the volume data  $f(\mathbf{x})$ , taking Eq. (3.9) as the transfer function, and applying FVR to the resulting volume, we have

$$I = \Pi_{\mathbf{v}}(R(f(\mathbf{x}))). \quad (3.13)$$

Substituting Eq. (3.9) into Eq. (3.13), and according to Eq. (3.5), we have

$$\begin{aligned} I &= \mathcal{F}_2^{-1}(\mathcal{F}_3(\sum_{i=0}^{m-1} w_i r_{i,k}(f(\mathbf{x})))\delta_{\mathbf{v}}) \\ &= \mathcal{F}_2^{-1}((\sum_{i=0}^{m-1} w_i \mathcal{F}_3(r_{i,k}(f(\mathbf{x}))))\delta_{\mathbf{v}}) \end{aligned}$$

$$= \sum_{i=0}^{m-1} w_i \Pi_{\mathbf{v}}(r_{i,k}(f(\mathbf{x}))). \quad (3.14)$$

The FVR of a volume after applying the B-spline transfer function is the summation of the  $m$ -weighted FVR results.

### 3.2.4 Increasing the Control Points

Using additional control points enables the transfer function to be easily shaped to a desired form. Unfortunately, the memory space requirements are linearly proportional to the number of control points. Consequently, the volume data may not fit into the GPU memory. To overcome this problem, we propose to increase the number of control points, but then cluster the control points into groups. Control points in the same group will either share identical shading factors, or their shading factors can be obtained through interpolation. The memory required then depends on the number of groups, which are manageable in size.

Given  $n$  control points in  $q$  clusters, the  $i$ -th cluster contains  $m_i$  control points with the weight  $w_i$ , and the formula of the spline curve is provided as Eq. (3.15):

$$R(u) = \sum_{i=0}^{q-1} \sum_{j=0}^{m_i-1} w_i \alpha_j r_{\hat{i}(\mathbf{p}_j),k}(u), \quad (3.15)$$

where  $\alpha_j$  is a constant and  $\hat{i}(\mathbf{p}_j)$  is the index of the control point  $\mathbf{p}_j$ . Applying Eq. (3.15) to Eq. (3.5), we have:

$$\sum_{i=0}^{q-1} w_i \Pi_{\mathbf{v}} \left( \sum_{j=0}^{m_i-1} \alpha_j r_{\hat{i}(\mathbf{p}_j),k}(f(\mathbf{x})) \right). \quad (3.16)$$

The control points in the same cluster share the same shading weight; thus, only one copy of the volume data is stored in the GPU memory. In Fig. 3.6, the curve consists of eight control points. We can cluster the eight control points into four groups,  $\mathbf{G}_0 = \{\mathbf{p}_0\}$ ,  $\mathbf{G}_1 = \{\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_5\}$ ,  $\mathbf{G}_2 = \{\mathbf{p}_2, \mathbf{p}_4\}$  and  $\mathbf{G}_3 = \{\mathbf{p}_6, \mathbf{p}_7\}$ . In this case, only four copies of the volume data are required.

If the weight of a control point can be obtained by interpolation from the weights of other control points, the control point can be clustered into a group. Given a sequence of

control points  $\mathbf{p}_c$ ,  $0 \leq i \leq c \leq j < m$ , the weight of the control point  $\mathbf{p}_d$ ,  $i < d < j$ , can be obtained through polynomial interpolation

$$w_d = \sum_{c=i, c \neq d}^j w_c l_c(u_d). \quad (3.17)$$

In Eq. (3.17),  $l_c$  is an interpolation basis function, such as linear interpolation, spline interpolation, or Lagrange interpolation. For the ease of explanation, we assume only one control point is obtained through interpolation. However, generalizing to more than one control point is not difficult. Applying the transfer function to a volume, Eq. (3.14) becomes:

$$\begin{aligned} I &= w_d \Pi_{\mathbf{v}}(r_{d,k}(f(\mathbf{x}))) + \sum_{c=0, c \neq d}^{m-1} w_c \Pi_{\mathbf{v}}(r_{c,k}(f(\mathbf{x}))) \\ &= \sum_{c=i, c \neq d}^j w_c l_c(u) \Pi_{\mathbf{v}}(r_{d,k}(f(\mathbf{x}))) + \sum_{c=0, c \neq d}^{m-1} w_c \Pi_{\mathbf{v}}(r_{c,k}(f(\mathbf{x}))) \\ &= \sum_{c=i, c \neq d}^j w_i \Pi_{\mathbf{v}}(l_c(u) r_{d,k}(f(\mathbf{x})) + r_{c,k}(f(\mathbf{x}))) + \\ &\quad \sum_{0 \leq c < i} w_c \Pi_{\mathbf{v}}(r_{c,k}(f(\mathbf{x}))) + \sum_{j < c < m} w_c \Pi_{\mathbf{v}}(r_{c,k}(f(\mathbf{x}))). \end{aligned} \quad (3.18)$$

The simplest polynomial interpolation is linear interpolation, given three control points  $\mathbf{p}_i = (u_i, w_i)$ ,  $\mathbf{p}_j = (u_j, w_j)$  and  $\mathbf{p}_d = (u_d, w_d)$ , where  $u_i < u_d < u_j$ . The weight  $w_d$  can be obtained by

$$w_d = (1 - \beta)w_i + \beta w_j, \quad (3.19)$$

where  $\beta = (u_d - u_i)/(u_j - u_i)$ . Applying the transfer function to the volume, Eq. (3.14) becomes

$$\begin{aligned} I &= w_i \Pi_{\mathbf{v}}(r_{i,k}(f(\mathbf{x}))) + w_d \Pi_{\mathbf{v}}(r_{d,k}(f(\mathbf{x}))) + w_j \Pi_{\mathbf{v}}(r_{j,k}(f(\mathbf{x}))) \\ &= w_i \Pi_{\mathbf{v}}(r_{i,k}(f(\mathbf{x}))) + w_j \Pi_{\mathbf{v}}(r_{j,k}(f(\mathbf{x}))) + \\ &\quad ((1 - \beta)w_i + \beta w_j) \Pi_{\mathbf{v}}(r_{d,k}(f(\mathbf{x}))) \\ &= w_i \Pi_{\mathbf{v}}(r_{i,k}(f(\mathbf{x})) + (1 - \beta)r_{d,k}(f(\mathbf{x}))) + \\ &\quad w_j \Pi_{\mathbf{v}}(r_{j,k}(f(\mathbf{x})) + \beta r_{d,k}(f(\mathbf{x}))). \end{aligned} \quad (3.20)$$

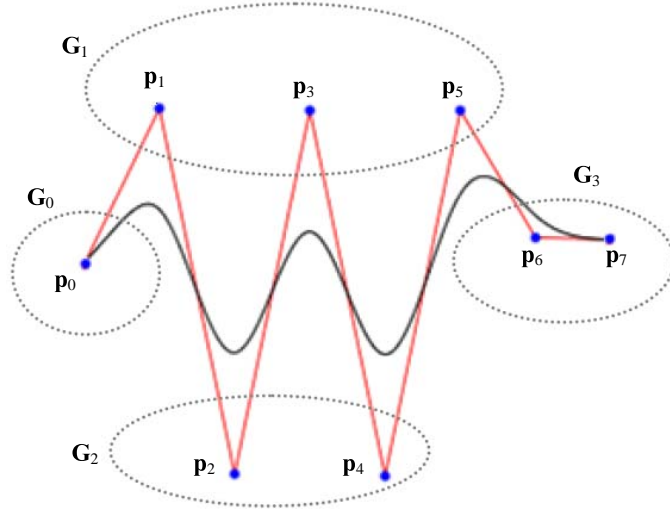


Figure 3.6: **The clustered control points.** Eight control points are clustered into four groups. In this case, only four copies of volume data are required.

Equation (3.20) shows that recomputing the Fourier transform of the volume data is not required if the weights of the control points are modified.

### 3.2.5 Results of FVR

We developed a GUI (Graphical User Interface) software system with the proposed transfer function for FVR. Using the developed software system, users can easily modify the transfer function and change the view direction. Our implementation environment was as follows:

- CPU: Pentium 4, 2.4 GHz,
- Memory: 2 GB,
- GPU: nVidia GeForce 8800 GT,
- Video memory: 768 MB,
- Viewport size:  $512 \times 512$  pixel.

To evaluate the performance, we used two sets of volume data of different size,  $128^3$  and  $256^3$ , with the B-spline transfer functions defined by one, four, and six control points.

We measured the rendering time and the pre-processing time, which included computing a 3D FFT. The performance of different cases is shown in Table 3.1. Because modern GPUs support parallel computing, increasing the volume size and number of control points slightly affect the frame rate. However, the pre-processing time is linearly proportional to the volume size and the number of control points.

This study presents the rendered results using several data sets. The first data set was CT-scan human chest volume data. The volume size was  $256^3$  voxels. The rendered results obtained using the Bézier curve and the B-spline transfer functions defined by six control points are shown in Figs. 3.7 and 3.8. In both Figs. 3.7 and 3.8, the images on the left show the rendered result obtained by applying the transfer functions that are shown on the right. The transfer functions in the first and second rows of both figures were designed to depict lung and bone structure, respectively. As shown in Fig. 3.8, using B-spline as the transfer function can significantly enhance lung capillaries. The result in Fig. 3.9 show that using a transfer function with greater control points can achieve better rendering. This study used a B-spline transfer function with 20 control points clustered into six groups. A comparison of the images shown in Figs. 3.8 and 3.9, the lung capillaries were significantly enhanced.

We also tested two volume data sets of HeLa cells, HeLa1 and HeLa2 (see Section 4.1.3). These data sets were reconstructed from the synchrotron radiation images aligned by the proposed alignment method(see Ch. 4). The size of HeLa1 and HeLa2 is  $256^3$  voxels. Fig. 3.10 shows the results of HeLa1. The results were obtained using the B-spline transfer function of 25 control points. Among these control points, three control points,  $\mathbf{P}_0$  to  $\mathbf{P}_2$ , were used to adjust the weights of membrane voxels. The cluster  $\mathbf{G0}$  contained first two control points to adjust the background voxels. The cluster  $\mathbf{G1}$  contained middle six control points to adjust the weights of the voxels which contain the intensities between membrane and particles. The cluster  $\mathbf{G2}$  contained the remained 11 control points to adjust the weights of particle voxels. In the top row, we mainly enhanced the weights of  $\mathbf{P}_1$  and  $\mathbf{P}_2$  to render the membrane of cell. The bottom row shows the results of the particles. The gray scales of particles are between 0.5 to 1.0, the weight of

$\mathbf{G}_2$  was enhanced to render the particles.

Fig. 3.11 shows the results of HeLa2. The results were obtained using the B-spline transfer function of 50 control points. Among these control points, three control points,  $\mathbf{P}_0$  to  $\mathbf{P}_2$ , were used to adjust the weights of membrane voxels. The cluster  $\mathbf{G}_0$  contained first two control points to adjust the background voxels. The cluster  $\mathbf{G}_1$  contained middle seven control points to adjust the weights of the voxels which contain intensities between membrane and particles. The cluster  $\mathbf{G}_2$  contained the remained 38 control points to adjust the weights of particle voxels. In the top row, we mainly enhanced the weights of  $\mathbf{P}_1$  and  $\mathbf{P}_2$  to render the membrane of cell. The bottom row shows the results of the particles. The gray scales of particles are between 0.5 to 1.0, the weight of  $\mathbf{G}_2$  was enhanced to render the particles.

One disadvantage of the proposed method is the significant amount of memory required. A user can more easily design a curve shape if more control points are used. However, the memory required increases in linear proportion to the number of control points, and the GPU memory is limited. Currently, commercially available GPUs can process a volume of  $256^3$  if six control points are used. When additional control points are required, the control points can be clustered into groups if they possess the same weight or their weight can be obtained through interpolation. Each group requires a copy of the volume to ensure the required memory can be maintained at a manageable size.

One limitation of the proposed method is that transfer functions containing a negative value are not allowed. This issue is the future research. Additionally, another future works is designing an efficient sampling method for extracting a slice of the frequency domain volume because the computing time required for sampling a slice of the frequency domain data is longer than that required to perform the inverse 2D Fourier transform. A more efficient sampling method is required to improve the overall performance.



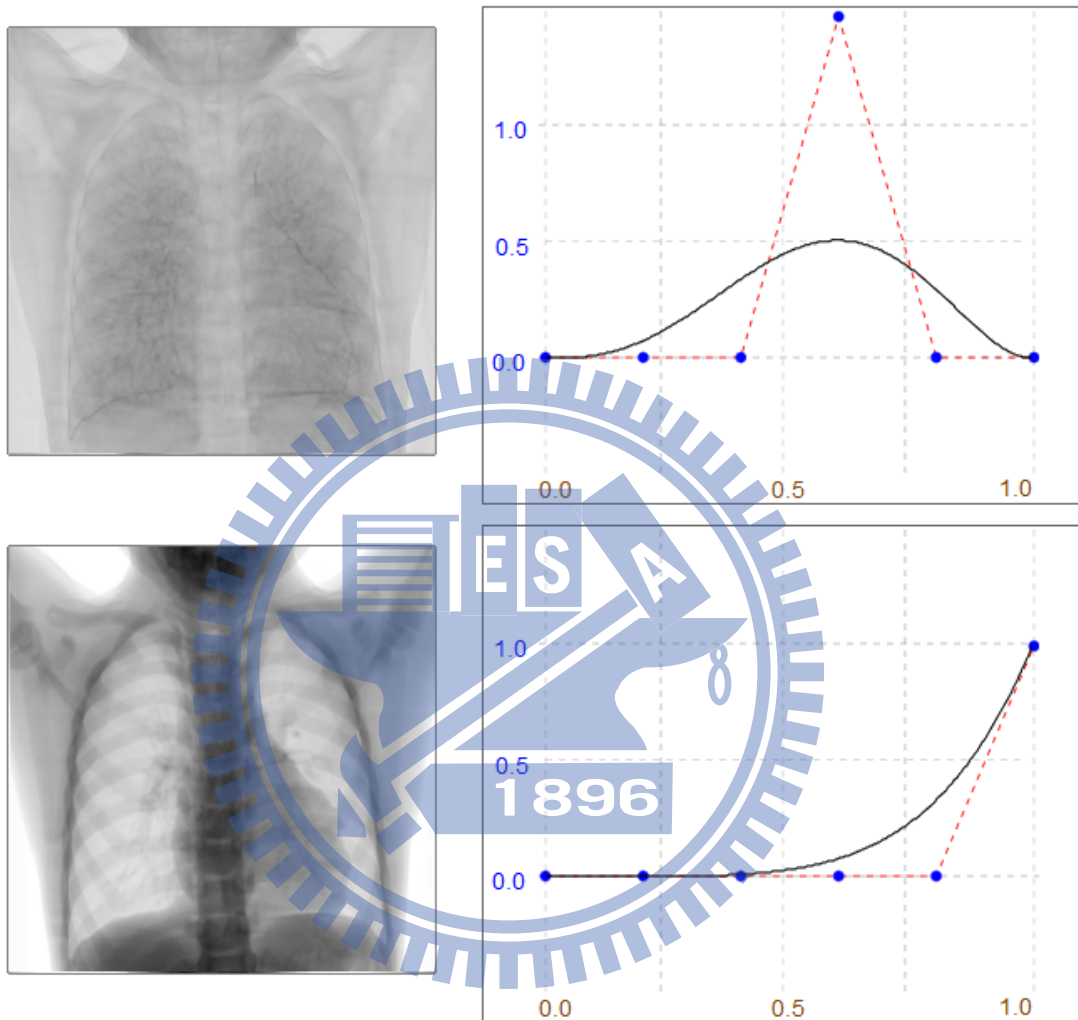


Figure 3.7: **The results of FVR with Bézier curve transfer function(CT chest).** The rendered results that were obtained using the Bézier curve transfer function. The data set was the CT-scan human chest of  $256^3$  voxels. The left image in the first row depicted lung structure. The image in the right shows the transfer function. We tried to enhance the gray scales between 0.4 and 0.65. But the gray scales between 0.25 and 0.8 are also enhanced so that the lung capillaries were blurred. The left image in the second row shows the bone structures through enhancing the gray scales between 0.9 and 1.0.

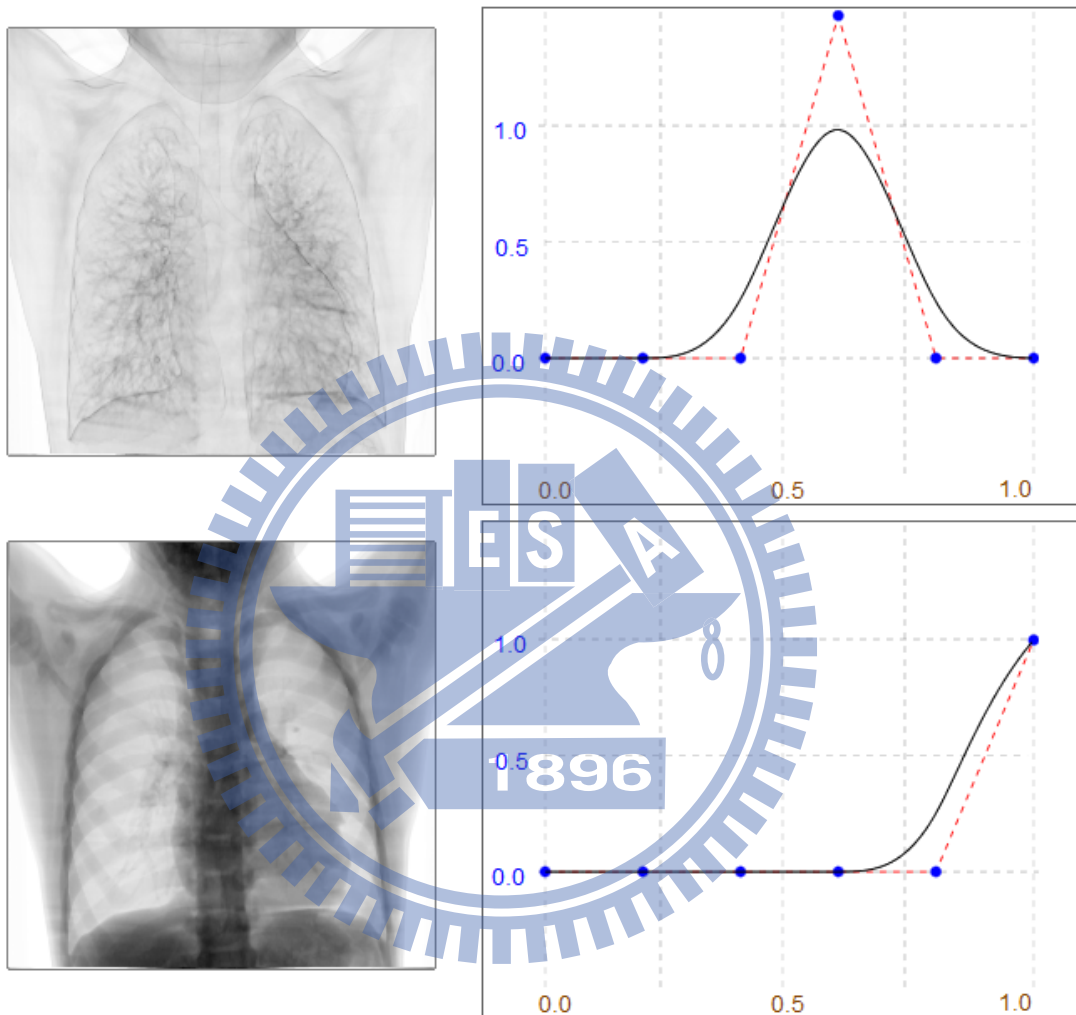


Figure 3.8: **The results of FVR with B-spline transfer function(CT chest).** The same volume data used as Fig. 3.7. The rendered results that were obtained using the B-spline transfer function. The first row shows lung structure and the second row shows the bone structure. Using the B-spline transfer function was easier to control the curve shape. The lung capillaries were more clearly shown in the rendered result compared to the result in Fig. 3.7.

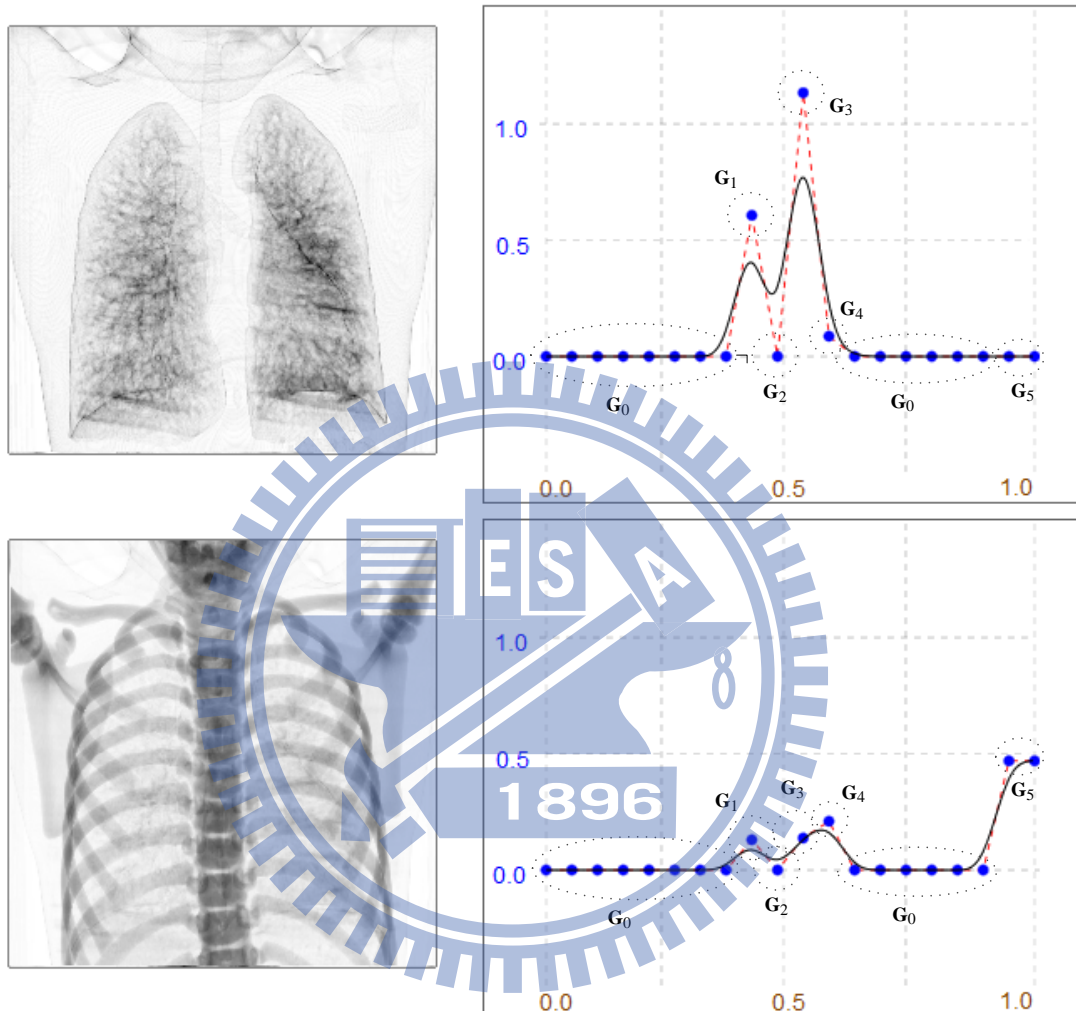


Figure 3.9: **The results of FVR with B-spline transfer function(CT chest).** The same volume data used as Fig. 3.7. The rendered results were obtained using the B-spline transfer function defined by 20 control points. The control points were clustered into six groups,  $G_{0\sim 5}$ . The first row shows lung structure, the gray scales between 0.4 and 0.65 were enhanced. The second row shows bone structure, the gray scales between 0.7 and 1.0 were enhanced. Compared to the results that are shown in Fig. 3.8, the lung capillaries and backbone structure were further enhanced.

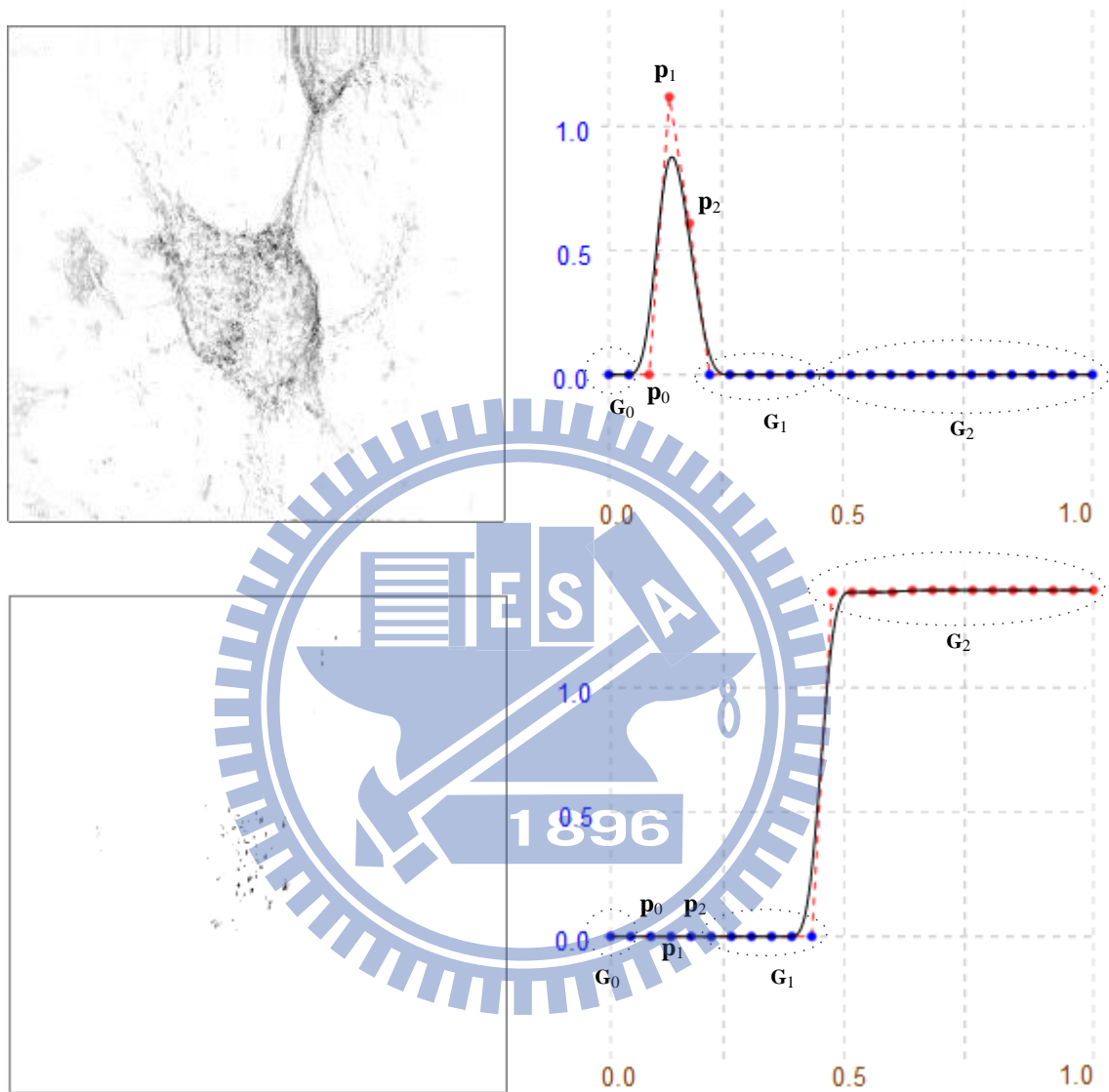


Figure 3.10: **The results of FVR with B-spline transfer function(HeLa1).** This is the volume of HeLa cell with  $256^3$  voxels. The results were obtained using the B-spline transfer function of 25 control points.  $P_0$  to  $P_2$ , were used to adjust the weights of membrane voxels. The cluster  $G_2$  contained the final 11 control points to adjust the weights of particle voxels. In the top row, the weights of  $P_1$  and  $P_2$  were enhanced to render the membrane of cell. In the bottom row, the weights of  $G_2$  is enhanced to render the particles.

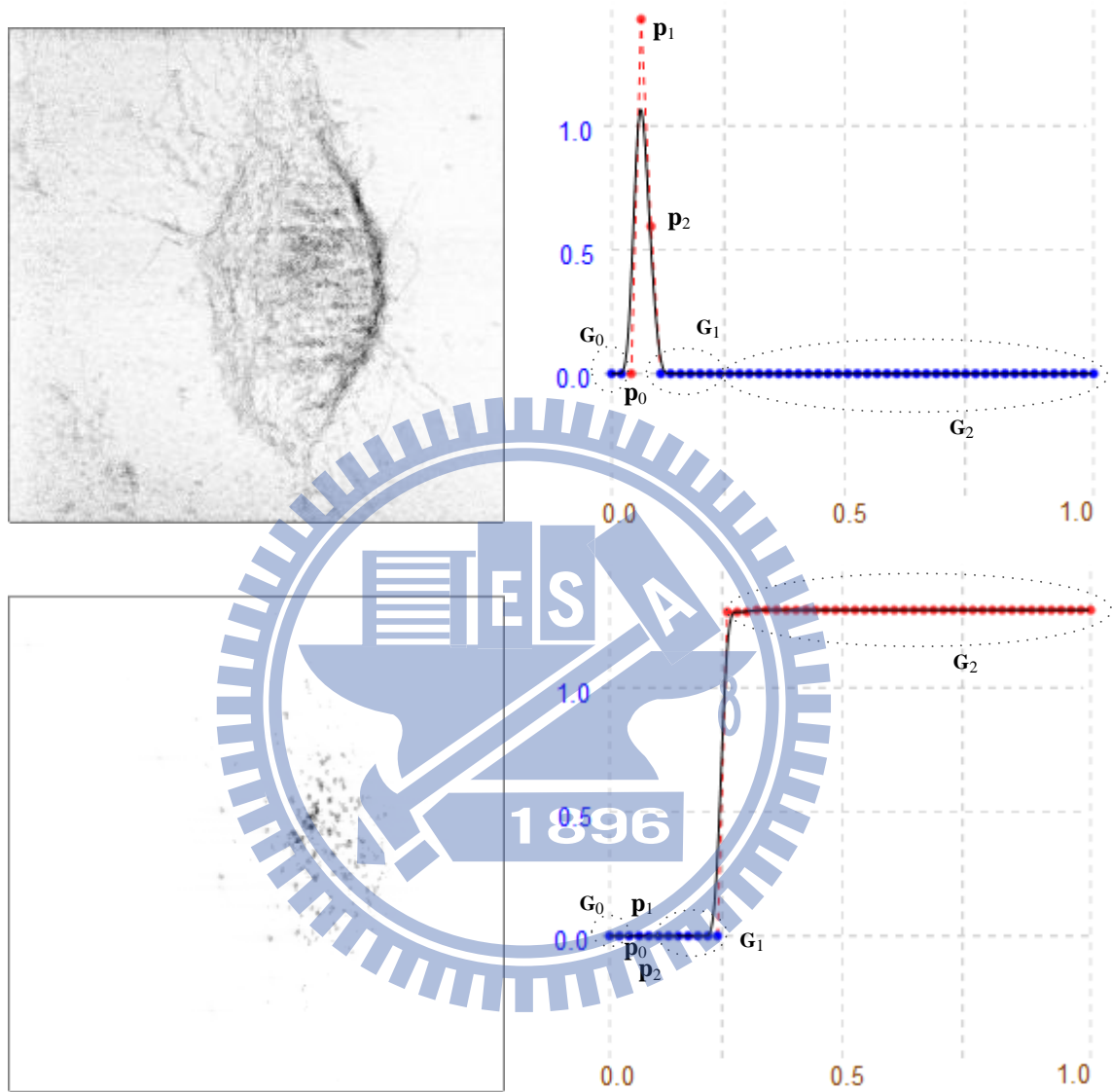


Figure 3.11: **The results of FVR with B-spline transfer function(HeLa2).** This is the volume of HeLa cell with  $256^3$  voxels. The results were obtained using the B-spline transfer function of 50 control points.  $\mathbf{P}_0$  to  $\mathbf{P}_2$ , were used to adjust the weights of membrane voxels. The cluster  $\mathbf{G}_2$  contained the final 11 control points to adjust the weights of particle voxels. In the top row, the weights of  $\mathbf{P}_1$  and  $\mathbf{P}_2$  were enhanced to render the membrane of cell. In the bottom row, the weights of  $\mathbf{G}_2$  is enhanced to render the particles.

Table 3.1: **The computing time of two cases using the B-spline as a transfer function.**

Data size (voxel)	128 <sup>3</sup>			256 <sup>3</sup>		
The number of the control points	6	4	1	6	4	1
Memory required (MB)	96	64	16	768	512	128
Pre-processing (second)	18.1	12.1	3	167.4	109	8.7
Frame rate (FPS)	64.1	65.8	66.46	59.2	62.41	66.26

## Chapter 4

# High-Resolution Volume

## Reconstruction

Theoretically, the volume image can be reconstructed from a sequence of high-resolution projections. This is not true, because the problem of mechanical imprecision arises when the resolution increases to a certain level, such as that required for cell tomography. When the pixel size is approximately 10 nm, a slight mechanical vibration can hinder accurate reconstruction. Fig. 4.1 demonstrates this problem. In Fig. 4.1(a), four X-ray images of a cancer cell are taken from different projection angle. The yellow lines mark the Y-axis positions of a black particle. Because the mechanical vibration, the Y-axis position of each black particle in the images are different. Fig. 4.1(b) shows the sinogram constructed from a row in each image of the 140 X-ray images of a cancer cell. The sinogram is no smooth and the reconstruction is not possible. Fig. 4.1(c) shows the FBP result.

In our cell tomography experience, the pixel size of image is 11.78 nm. Rotating the object holder can cause a 5 to 30 pixel difference in position because of the mechanical instability. Although the position of the object holder can be calibrated during image acquisition, the calibration process can take an unacceptably long time, causing the object to receive excessive X-rays. The TXM controller provided by Xradia (hereinafter called “Xradia”) was designed to solve the misalignment problem. Unfortunately, manual adjustments are generally required to obtain satisfactory tomography reconstruction. A sim-



ilar problem also exists in electron microscopic tomography. Using the cross-correlation function to align the electron microscopic images is a common solution to this problem [38]. A software system, “SPIDER”, was implemented based on the cross-correlation function [39]. However, the cross-correlation function alignment does not consider the projection model, thus limiting the quality of tomography reconstruction.

This study presents a feature-based alignment approach for calibrating the displacement caused by mechanical vibration [40]. Because a synchrotron X-ray is a parallel beam projection, the resulting displacement can be decomposed into vertical and horizontal displacements. The proposed method aligns the images in the vertical direction by direct image alignment. Calibrating the images in the horizontal direction is more complex than that in the vertical direction. In addition to matching feature points, the matched feature points must form sine-wave shaped loci. We propose approximating the loci of the matched feature points in the  $x-\theta$  coordinate system according to sine waves by using the least square curve fitting. The deviation between the loci and the sine waves provides information for horizontal calibration.

The following two sections respectively describe the proposed method and the results.

## 4.1 Methods

### 4.1.1 Sample Preparation and Image Acquisition

HeLa cells were used in this study. The cells were grown on Kapton film, and endocytosis [41] was used to stain HeLa cells by absorbing gold nanoparticles of  $250\mu\text{M}$  (micromolar). The cells were then fixed in a container using a mixture of paraformaldehyde and glutaraldehyde [42].

The synchrotron microscope used in this study was built at the National Synchrotron Radiation Research Center, Hsinchu, Taiwan. The CCD size is  $2048 \times 2048$  pixels, and the field of view is  $24\ \mu\text{m}$ . Each projection image was taken after a  $1^\circ$  rotation. To prevent the object holder from becoming perceptible (occurring when the object holder is nearly parallel to the X-ray), the range of the rotation angle of the object holder was  $\pm 70^\circ$ . Only

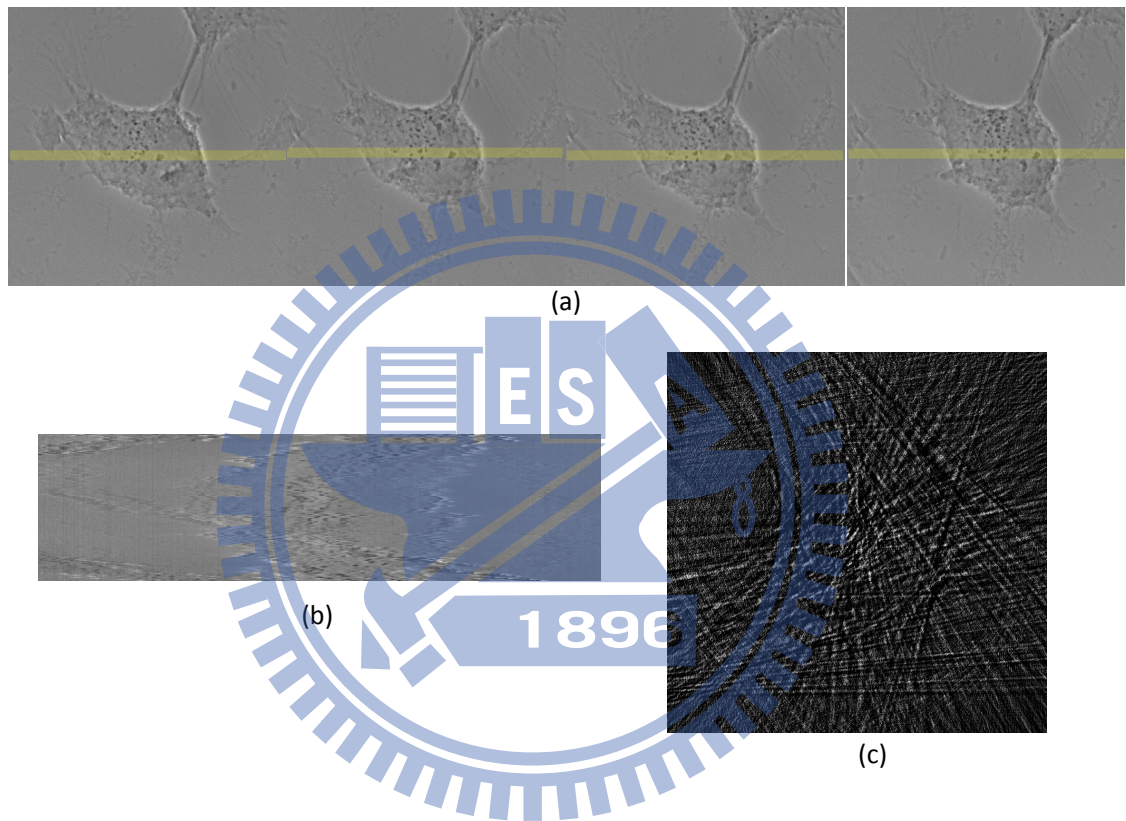


Figure 4.1: **The misalignment in tomography.** (a) Four X-ray images of a cancer cell acquired from different projection angles. The yellow lines mark the Y-axis positions of a black particle. These positions are different due to the mechanical vibration. (b) The sinogram extracted from 140 X-ray images with misalignment. (c) The FBP result of (b).

140 projection images were acquired. The size of each image was  $1024 \times 1024$  pixels, and the pixel size was 11.78 nm. In this study, the exposure time of each image was 1 second.

### 4.1.2 The Alignment Method

A projected feature point is a pronounced mark in an X-ray projection image. Alignment is accomplished by first aligning the projected feature points in the vertical direction and then in the horizontal direction. The projected feature points should be maintained in the vertical direction. Thus, vertically aligning the projected feature points in the second image to the previous image is sufficient. However, the location of the feature points in the horizontal direction varies among projection images. Calculating the horizontal location of the feature points is a more difficult task than alignment in the vertical direction. The following subsections describe these steps.

#### Vertical Direction Alignment

For each pair of projection images, the sum of the intensity values on each row is calculated. The sums of the rows form histograms that should be similar in a pair of consecutive images. The vertical displacement can be calculated by minimizing the difference between the histograms.

Given an  $N \times M$  image  $I(x, y)$ ,  $0 \leq x < N$ ,  $0 \leq y < M$  and  $I(x, y) \in [0, 1]$ . The vertical histogram  $h$  is calculated by

$$h(I, y) = \frac{1}{N} \sum_{x=0}^{N-1} I(x, y). \quad (4.1)$$

Assume that  $I^a$  is the unaligned image and  $I^b$  is the reference image. The vertical correction of  $I^a$  is  $\hat{y}$ , which can be estimated by

$$\hat{y} = \arg \min_{y'} \sum_{y=0}^{M-1} (h(I^a, y + y') - h(I^b, y))^2. \quad (4.2)$$

To achieve the most favorable results, the image is preprocessed to enhance the features. In this experiment, the estimated correction is more accurate when the images are enhanced

by applying the edge detection method[43].

### **Horizontal Direction Calibration**

The horizontal calibration is based on the projected feature points forming a sine-shaped locus in the  $x-\theta$  coordinate system. This calibration involves three steps: detecting projected feature points, matching projected feature points to construct a set of loci from the matched projected feature points, and fitting the loci to sine curves to adjust the horizontal displacement of images.

### **Detecting Projected Feature Points**

Feature point extraction is a fundamental step in image stitching, object recognition, and feature-based image alignment [44]. Researchers have proposed many feature detection methods. The corner detection method proposed by Harris and Stephen [45] is commonly used to extract corner-shape regions in an image. To achieve scale invariance, Kadir and Brady [46] selected the salient region from the image scale-space as the feature that possesses the maximum entropy. Lowe [47] proposed the scale-invariant feature transform (SIFT) algorithm to select local extrema from the differences of a Gaussian (DoG) pyramid in an image. The SIFT algorithm uses the gradient location-orientation histogram as a feature descriptor to achieve rotation invariance and illumination invariance. Researchers have proposed several improved versions of the SIFT algorithm. Bay et al. used the Haar wavelet to expedite feature detection [48]. Rady et al. proposed entropy-based feature detection [49], and Suri et al. combined mutual information alignment with the SIFT algorithm [50].

In this study, a modified SIFT algorithm was employed to extract automatically the projected feature points contained in X-ray images. The typical SIFT implementation involves describing a feature according to its location, size, the orientation of the sampling region, and the image gradient histogram in the sampling region. Because the proposed method matches the projected feature points in two X-ray images based on mutual information [51, 52, 53], each projected feature point in this study contained the entropy

of the sampling region rather than the image gradient histogram. To reduce the noise and the number of low-contrast projected feature points, the entropy of each selected projected feature point must exceed a given threshold. The experiments in this study entailed setting a threshold between 0.5 and 1.0. Because the features in the objects are gold nanoparticles, the size and orientation of the sampling region were fixed in this implementation.

### Matching Projected Feature Points

Let  $\mathbf{F}^i$ ,  $i = 1, \dots, m$  be the sets of projected feature points in  $m$  projection images. The projected feature points are classified into  $k$  groups. In the ideal case, each group is the set of projected feature points, which are the projections of a feature (i.e., gold nanoparticle) in the object from various angles. Because the rotation angle of the object is small, the projected feature points are in proximity and have similar mutual information in two consecutive images. However, the distance between the two matched projected feature points depends on the distance between the feature and the rotation axis of the object. This means that an affine transform cannot match the projected feature points in two images. Therefore, this study presents a greedy method for classifying the projected feature points. For each pair of images, the random sample consensus (RANSAC) method[54] was first applied to compute an initial alignment of the two images, and a tracking method was then employed to match the projected feature points in the next image.

Several feature tracking methods are available [44]. The proposed method is designed based on the Shafique and Shah's method [55], which is a greedy algorithm for tracking moving objects in videos, and the Tang and Tao's method [56], which integrates the hidden Markov model to eliminate unreliable matches.

Given the projected feature points sets  $\mathbf{F}^{i-1}$  and  $\mathbf{F}^i$ , the RANSAC method was applied to compute a translation matrix  $T_{i,i-1}$ , so that a sufficient number of projected feature points  $p$  and  $q$  in  $\mathbf{F}^{i-1}$  and  $\mathbf{F}^i$  respectively,  $|T_{i,i-1}(q) - p|$  is less than a given threshold  $\epsilon$ . Applying translation matrices  $T_{i,i-1}$ ,  $i = 2, \dots, m$  to the consecutive images achieves the initial alignment of the  $m$  projection images. All of the images are aligned based on the

first image.

Given the initially aligned projected feature points  $\mathbf{F}^i, i = 1, \dots, m$ , the following procedures yield a set of possible loci of the projected feature points produced by feature points in the object.

1. Every projected feature point in  $\mathbf{F}^1$  is the starting point of a locus.
2. Iteratively process  $\mathbf{F}^i, i = 2, \dots, m$ ;
  - (a) Let  $L$  be the set of the loci computed. For each locus  $l \in L$ , compute  $T_{i-1,i}(p)$  where  $p$  is the final point of  $l$  and  $T_{i-1,i}$  is the inverse of  $T_{i,i-1}$ . Let  $\Lambda$  be a region in  $I^i$  centered at  $T_{i-1,i}(p)$ . Search in  $\Lambda$  for the projected feature points (Fig. 4.2). If this region contains only one projected feature point  $q$ , then that point  $q$  is selected as the final point of  $l$ . If the region contains more than one projected feature point, then select the  $q$  that has the greatest  $M$ , where  $M$  is the average entropy of  $q$  and the previous  $t$  points on  $l$ . If  $t$  is greater than  $i - 1$ , then  $M$  is the average entropy of  $q$  and the points of the entire locus  $l$ .
  - (b) If  $\mathbf{F}^i$  contains unmatched projected feature points, then each of these points creates a new locus.
3. Reverse the image orders and repeat Step 2, but do not include 2b to backtrack all loci.

The X-ray images used in this study measured  $1024 \times 1024$  pixels, and  $128 \times 32$  pixels was the size of search region  $\Lambda$ , and five previous points for  $t$ .

Because two loci could intersect (i.e., two projected feature points on two loci could overlap or be extremely close), the average entropy must be computed to select the best-matching projected feature point in Step 2a. In this step, some projected feature points with a high entropy in  $\mathbf{F}^i$  are not included in any locus. These significant projected feature points should not be disregarded, and Step 2b entails creating a new locus for each of them.

After Step 2, the forward feature tracking required to construct the set of loci is complete. To verify the correctness of the loci and complete the loci added in the Step 2b, backtrack all loci in the final step.

### 4.1.3 Horizontal Displacement Estimation

Let the set of  $k$  loci collected in the previous step be  $\{l_1, l_2, \dots, l_k\}$ . Each locus,  $l_j$ ,  $1 \leq j \leq k$ , consists of  $m$  projected feature points,  $\langle f_j^1, f_j^2, \dots, f_j^m \rangle$ . These projected feature points are expected to be the projections of a feature,  $p_j$ , in the object from various angles. Assume that these projected feature points are the projections from  $-\pi/2$  and  $\pi/2$ . Recall that  $f_j^i$  is not a point, but a rectangular box. Take the  $X$ -coordinates of the center of the rectangular boxes,  $x$ , and transform the  $j$  to  $\theta$ , the direction of the projections. Then, draw  $(x, \theta)$  for all  $f_j^i$  in the  $x$ - $\theta$  coordinate system. The locus of  $(x, \theta)$  corresponds to a sine curve, (i.e., the sinogram). Given a set of loci of features, the horizontal alignment is conducted by fitting the curves to a set of sine curves and then by computing the deviations.

Consider a locus,  $l_j$ , and assume that the projected feature points on the locus are the projections of the feature point  $p_j$ . This feature point  $p_j$  can be expressed as  $(\gamma_j, \omega_j)$ , where  $\gamma$  is the radial coordinate and  $\omega$  is the angular coordinate. According to the Radon transform [4], the corresponding horizontal position  $x_j$  in the  $i$ th image (rotation angle  $\theta_i$ ),  $1 \leq i \leq m$ , can be written as

$$x_j^i = \gamma_j \sin(\theta_i + \omega_j) \quad (4.3)$$

$$= \gamma_j (\cos \theta_i \sin \omega_j + \sin \theta_i \cos \omega_j), \quad (4.4)$$

or in matrix form as

$$\mathbf{x} = \Theta \mathbf{u}, \quad (4.5)$$



where  $\mathbf{u} = [u_1, u_2]^T = [\gamma_j \sin \omega_j, \gamma_j \cos \omega_j]^T$ ,  $\mathbf{x} = [x_j^1, x_j^2, \dots, x_j^m]^T$ , and

$$\Theta = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 \\ \cos \theta_2 & \sin \theta_2 \\ \vdots & \vdots \\ \cos \theta_m & \sin \theta_m \end{bmatrix}.$$

$u_1$  and  $u_2$  can be solved by the least-squares method,

$$\mathbf{u} = (\Theta^T \Theta)^{-1} \Theta^T \mathbf{x}. \quad (4.6)$$

Finally  $(\gamma_j, \omega_j)$ ,

$$\gamma_j = \sqrt{u_1^2 + u_2^2} \quad (4.7)$$

and

$$\omega_j = \sin^{-1}\left(\frac{u_1}{\gamma_j}\right). \quad (4.8)$$

The horizontal displacement  $\hat{x}_j^i$  is estimated by

$$\hat{x}_j^i = \gamma_j \sin(\theta_i + \omega_j) - x_j^i. \quad (4.9)$$

Because both  $\omega_j$  and  $\pi - \omega_j$  are solutions to (4.8), choose the one that minimizes the sum of errors,  $\sum_{i=1}^m \hat{x}_j^i$ . To determine the horizontal displacement  $c_i$  for the  $i$ th image from  $k$  feature loci, use the average of the  $k$  horizontal corrections:

$$c_i = \frac{1}{k} \sum_{j=1}^k \hat{x}_j^i. \quad (4.10)$$

Some loci are unreliable because of noises, out-of-view projected feature points, or bad projected feature points matching. These loci should be removed. For each locus  $l_j$ , adjust the points on  $l_j$  based on the estimated  $c_i$ ,  $1 \leq i \leq m$ .  $l_j$  is unreliable if the absolute peak distance between a point on the adjusted  $l_j$  and its best-fitting sine wave is greater than a given threshold  $\sigma$ . If there are no unreliable loci, stop the algorithm

and output the aligned results. Otherwise, remove the unreliable loci and repeat the horizontal displacement estimation algorithm.

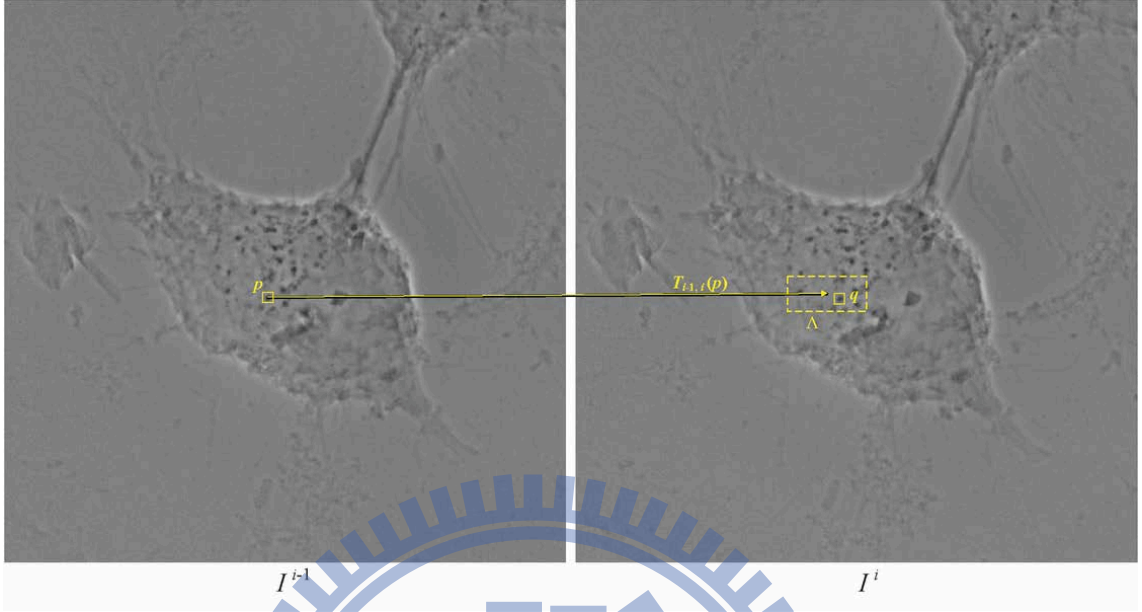


Figure 4.2: **The feature tracking between two X-ray projection images  $I^{i-1}$  and  $I^i$ .** For a point  $p$ , its corresponding projected feature point,  $q$ , can be searched in the area  $\Lambda$  that is determined by applying the affine transformation  $T_{i-1,i}$  to  $p$ .

## 4.2 Results

The proposed algorithm was used to align projection images and then reconstruct the 3D volume data of HeLa cells from X-ray projections. To verify the correctness of the alignment algorithm, a phantom data set was used to simulate the HeLa cell stained using the gold nanoparticles. The following subsections present the construction details and results obtained.

### 4.2.1 The Phantom

A volume datum containing 20 imitative gold nanoparticles was constructed, and X-ray projections of the shifted volume were generated to simulate machine vibrations. Fig. 4.3 shows the simulated X-ray image of the phantom cell, in which 180 projections of  $512 \times 512$  pixels images were generated. The volume rotated is  $1^\circ$  between successive

projections. Before each projection, the volume was shifted in the vertical and horizontal directions to simulate the mechanical imprecision of the object holder. The amount of displacement was determined according to a random number uniformly distributed over the range  $\pm 20$  pixels.

The proposed alignment algorithm was used to calibrate the images, which were first aligned in the vertical direction. The detection and matching methods of projected feature points were then applied to determine 16 feature points. The locus of the horizontal position of a feature point over various angles should form a sine-shaped curve. Fig. 4.4(a) shows the loci of the 16 points. This figure is in the  $x-\theta$  coordinate system: the vertical axis represents the rotation angle, and the horizontal axis represents the horizontal position of the projected feature point. The loci are not smooth because of horizontal displacement. The most suitable sine waves to fit the loci (Fig. 4.4(b)) were then calculated, and the displacement of the feature points in the horizontal position were estimated (Fig. 4.4(c)). The filtered back projection (FBP) algorithm was then used to reconstruct the 3D volume data based on the aligned X-ray images [12].

To verify the accuracy of the proposed alignment method, the positions and diameters of the spherical particles in the reconstructed images were compared with the original volume data, which included 20 spherical particles, and the diameter of each particle was 12 voxels. There were 20 particles found in the reconstructed volume. The average errors of the particle center and diameter were found to be 0.72 and 0.03 voxels, respectively. The reconstructed volume is close to the original volume. Fig. 4.5(a) shows a slice in the original volume data. Fig. 4.5(b) shows the tomography results of the same slice obtained using the proposed alignment method. For comparing the results of the proposed method with the SPIDER, the same phantom image was reconstructed using SPIDER. Fig. 4.5(c) shows the tomography reconstructed results using SPIDER.

The mean squared error (MSE) of the foreground voxels between the original volume data and the reconstructed volume was also calculated. A voxel was classified as a foreground voxel when its intensity exceeded a given intensity threshold. The MSE between the original volume and the reconstructed volume from the unaligned volume was found

to be 0.029. The reconstructed volume obtained from the images aligned using SPIDER has an MSE of 0.002. For the volume reconstructed from the images aligned using the proposed method, the MSE was found to reduce to 0.0005. Because Xradia does not recognize the file format of the phantom images, the proposed method was not compared with Xradia.

### 4.2.2 HeLa Cells

Figs. 4.6(a) and (b) separately show the projection images of two HeLa cells. The first HeLa cell, named HeLa1, contained 84 identified projected feature points. Six reliable feature points were selected from the identified projected feature points. Fig. 4.7(a) shows the loci of the selected points in the  $x-\theta$  coordinate system. Fig. 4.7(b) shows the sine waves that fit the loci most effectively. The projection images of HeLa1 are aligned according to the fitted sine waves, and Fig. 4.7(c) shows the loci of the projected feature points after alignment. In the second HeLa cell, HeLa2, four reliable features were selected from 89 identified projected feature points, and Fig. 4.7(d) shows the loci. Fig. 4.7(e) shows the sine waves that fit the loci. Fig. 4.7(f) shows the loci of the projected feature points after alignment. To compare the different alignment methods, SPIDER and Xradia were also applied to align the projection images of HeLa1 and HeLa2. The same FBP algorithm was then applied to reconstruct the volumes from the aligned images.

Slices in the reconstructed volumes of HeLa1 and HeLa2 are shown in Figs. 4.8(a), (b), and (c) and Figs. 4.8(d), (e), and (f) respectively. Figs. 4.8 (a) and (d) were obtained using the proposed method, Figs. 4.8 (b) and (e) were obtained using SPIDER, and Fig. 4.8 (c, f) were obtained using Xradia. As shown in Fig. 4.8, the results of the proposed method were more favorable than those of SPIDER and Xradia. Comparing the results of the SPIDER and Xradia, the proposed method exhibits most well-defined membrane structures with least amount of artifacts, which is evident from 4.8(a) and (d). The texture-based volume rendering algorithm [5] was used to produce a 3D image of the volume data. Figs. 4.9 (a), (b), and (c) show the volume-rendering results of HeLa1, and Figs. 4.9 (d), (e), and (f) show the volume-rendering results of HeLa2. Figs. 4.9 (a) and

(d) show the results of the proposed method, Figs. 4.9 (b) and (e) show the results of SPIDER, and Figs. 4.9 (c) and (f) show the results of Xradia. The gold nanoparticles in Figs. 4.9 (a) and (d) are clearly shown, thereby enabling the evaluation of the location, the size, and the amount of the particles. The cell membranes can also be visualized in the rendered image, helping the user identify the geometry of the cells.

The proposed algorithm was applied to process 10 other X-ray image sets of HeLa cells. Among these 10 image sets, eight were successfully reconstructed (A1-A8, Figs. 4.10 and 4.11) except the other two (B1 and B2, Fig. 4.12 and 4.13). SPIDER and Xradia were also applied to the same sets of image data. Neither SPIDER nor Xradia could align the images in B1 and B2 for reconstruction. Specifically, SPIDER was effective for A3, A5, and A6; Xradia was effective for A1, A2, A5, A7, and A8. Although the reconstructions could be completed, comparing to our results, the artifacts produced due to the misalignment are more apparent.

Table 4.1 lists the experiments conducted in this study including the phantom, HeLa1, HeLa2, and the other 10 HeLa cells, as well as the computing time required. This table shows that the main factors affecting computational time are the image size, number of images, and number of identified projected features. The experiments in this study show that when the input data contains 180 images of  $1024 \times 1024$  pixels, the alignment can be performed in 10 min.

Fig. 4.12 and 4.13 show 16 frames of unsuccessful cases respectively. The main reason for the unsuccessful cases is the insufficient number of projected feature points in the X-ray images. In B1, most projected features exist in the image series less than 10 frames. Although the membrane of cell appears in each frame, the low contrast and large size cause the membrane cannot be a feature. The occluded projected features also exist in B1 and cause the feature matching failed. The same problem is in B2. No any reliable projected feature can be found in B2.

The most crucial factor affecting the performance of the proposed method is the number of reliable projected feature points. If there are enough reliable projected feature points, even if the projected feature points are not in the field of view in some projec-

tions, the method still works well because it also considers the set of partial loci. The proposed method performs most favorably if the features that produce the projected feature points are close to the rotational axis. Carefully adjusting the rotation axis before images acquisition can improve the quality of the reconstruction.

Considering the shape of the projected feature points, aside from particle objects, the proposed method can manage any shape of object if it contains a sufficient number of distinct projected features. For examples, the corners of a square, the two tips of a rod, and the branch points of a tree-structured object can be used as feature points as long as the features do not deform during image acquisition. If the images satisfy these requirements, then the proposed method can successfully align the images.

A graphical user interface software system for the Windows system and Mac OS X 10.8 has been developed. The software system can be downloaded from the following URL: <http://www.cs.nctu.edu.tw/~chengchc/SCTA> or <http://goo.gl/s4AMx>.

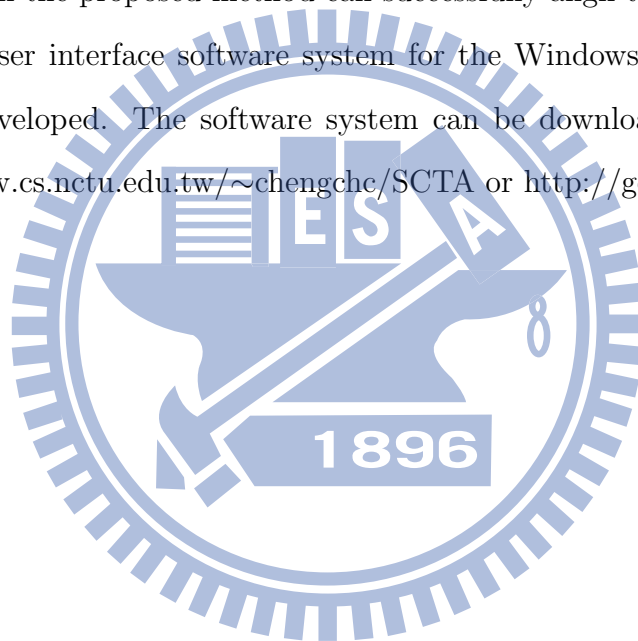




Figure 4.3: **The simulated X-ray image of the phantom data.** 180 emulated X-ray images were generated. The size of each image is  $512 \times 512$  pixels, the rotation angle between two consecutive images is  $1^\circ$ , and the ranges of the vertical and horizontal errors are  $\pm 20$  pixels.

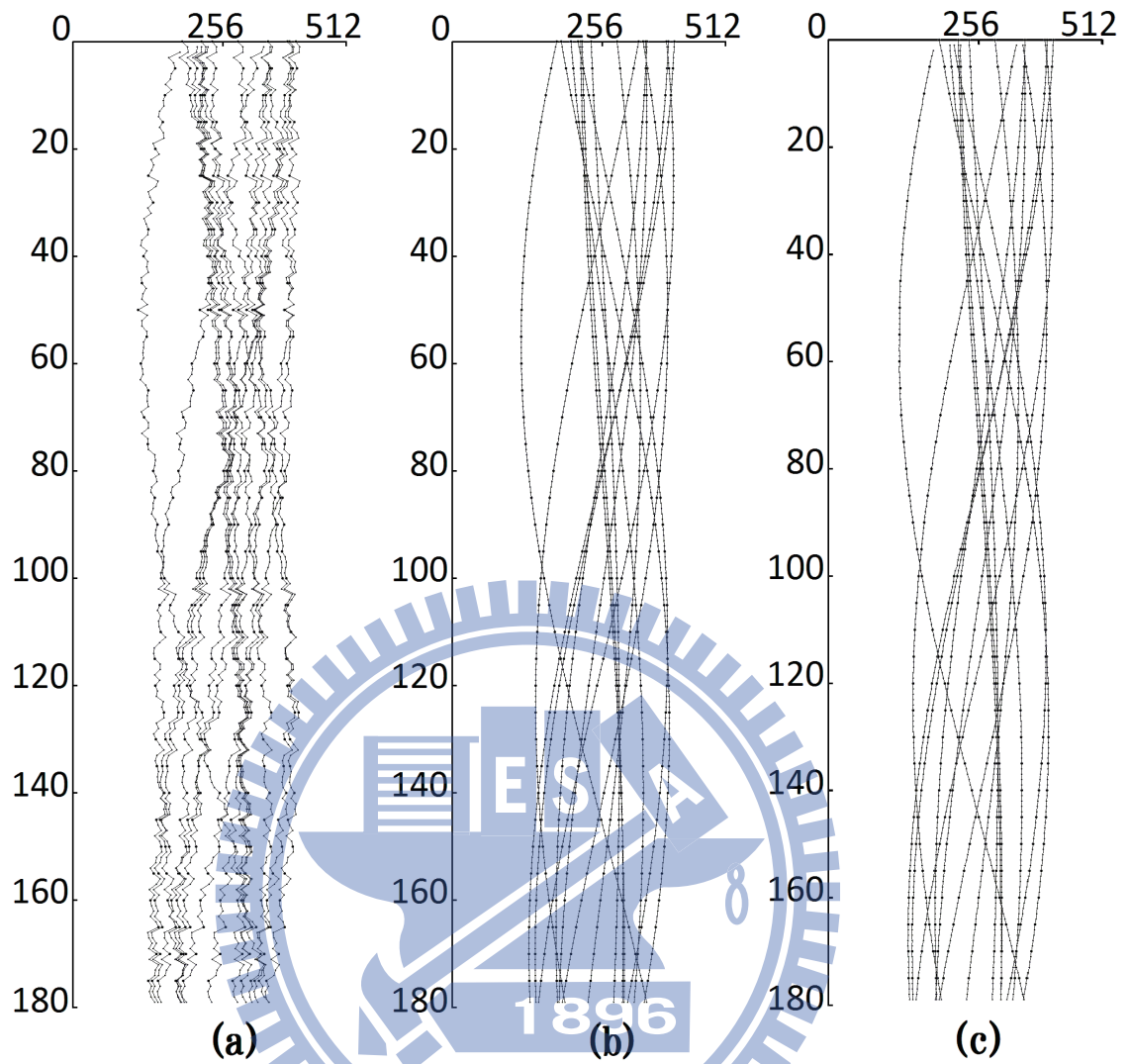


Figure 4.4: **Feature loci of the phantom data:** (a) loci of 16 projected feature points before horizontal alignment; (b) best-fit sine waves; and (c) the aligned loci.

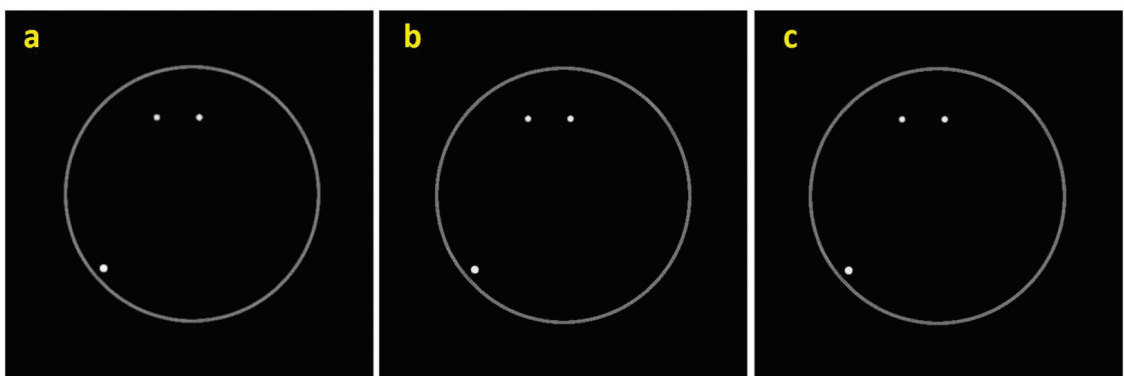


Figure 4.5: **One slice in the phantom data:** (a) original image; (b) result of proposed method; and (c) the result of SPIDER.



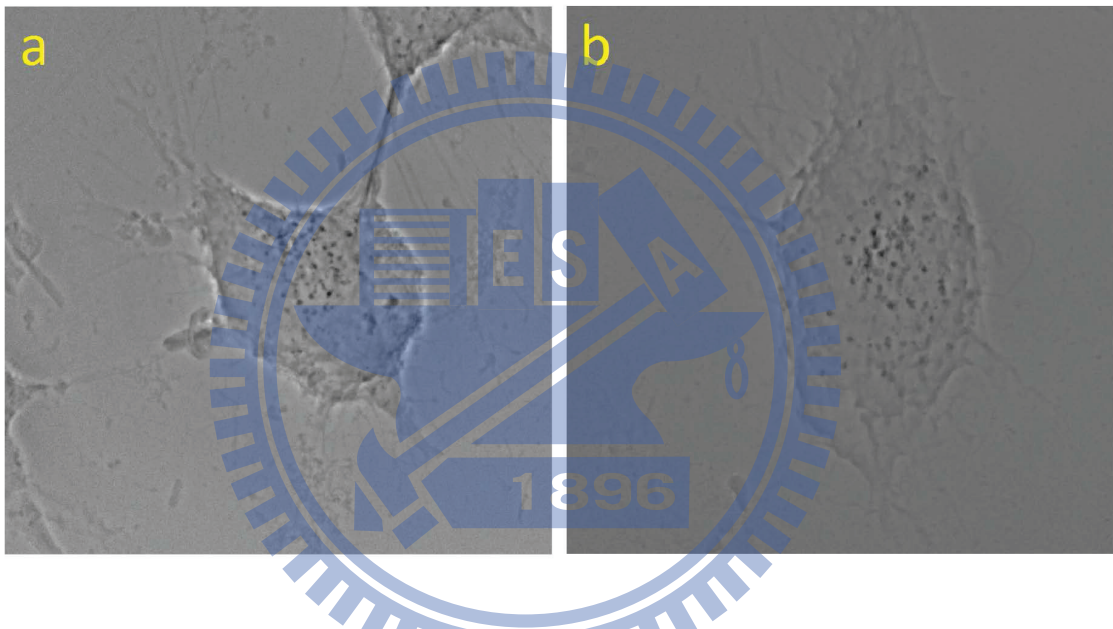


Figure 4.6: **X-ray projection images of HeLa cells.** 140 synchrotron X-ray projection images of  $1024 \times 1024$  pixels were acquired for HeLa1 (a) and HeLa2 (b). The rotation angle between two projection images is  $1^\circ$ .

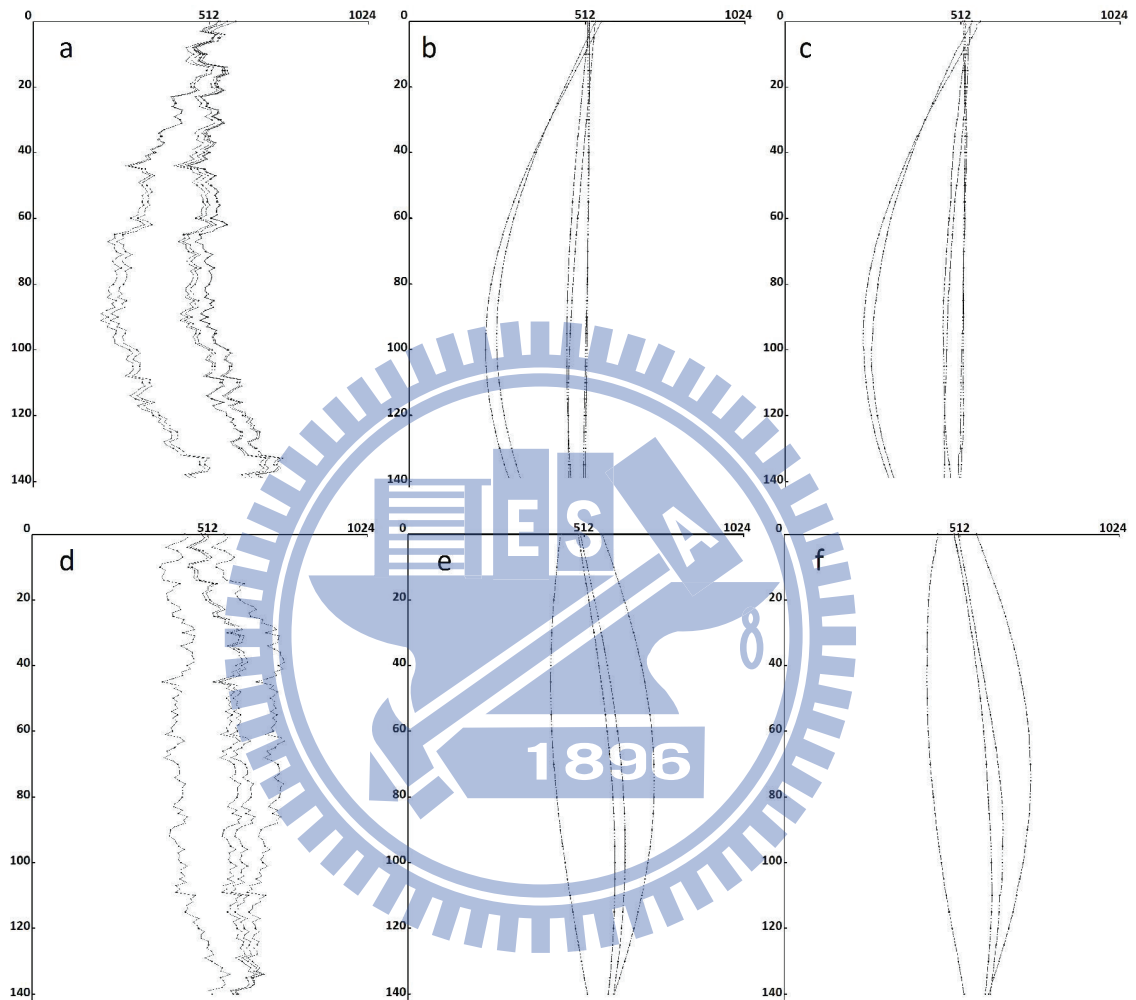


Figure 4.7: **The loci of the reliable projected feature points of HeLa1 and HeLa2.** (a) There were six reliable feature loci found in the acquired images of HeLa1. (b) Most suitable sine waves of HeLa1, and (c) aligned loci of HeLa1. (d) There were four reliable feature loci identified in the acquired images of HeLa2. (e) Best-fit sine waves of HeLa2, and (f) aligned loci of HeLa2.

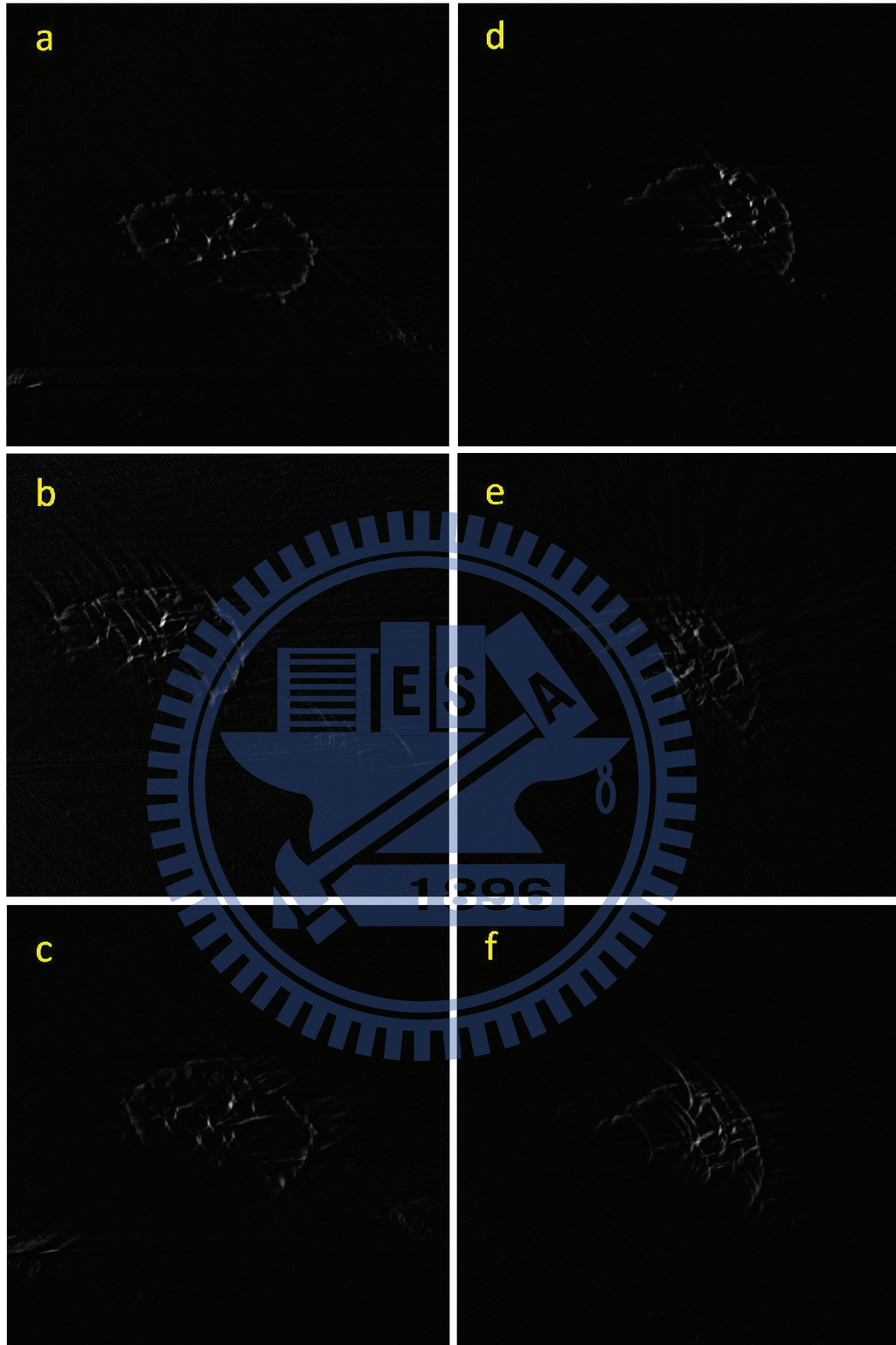


Figure 4.8: **One slice in the reconstructed tomographic images of HeLa1 and HeLa2.** (a), (b), and (c) are the same slice in the tomographic images of HeLa1: (a) result of proposed method; (b) result of SPIDER, and (c) result of Xradia. (d), (e), and (f) are the same slice in the tomographic images of HeLa2: (a) result of proposed method; (b) result of SPIDER, and (c) result of Xradia.

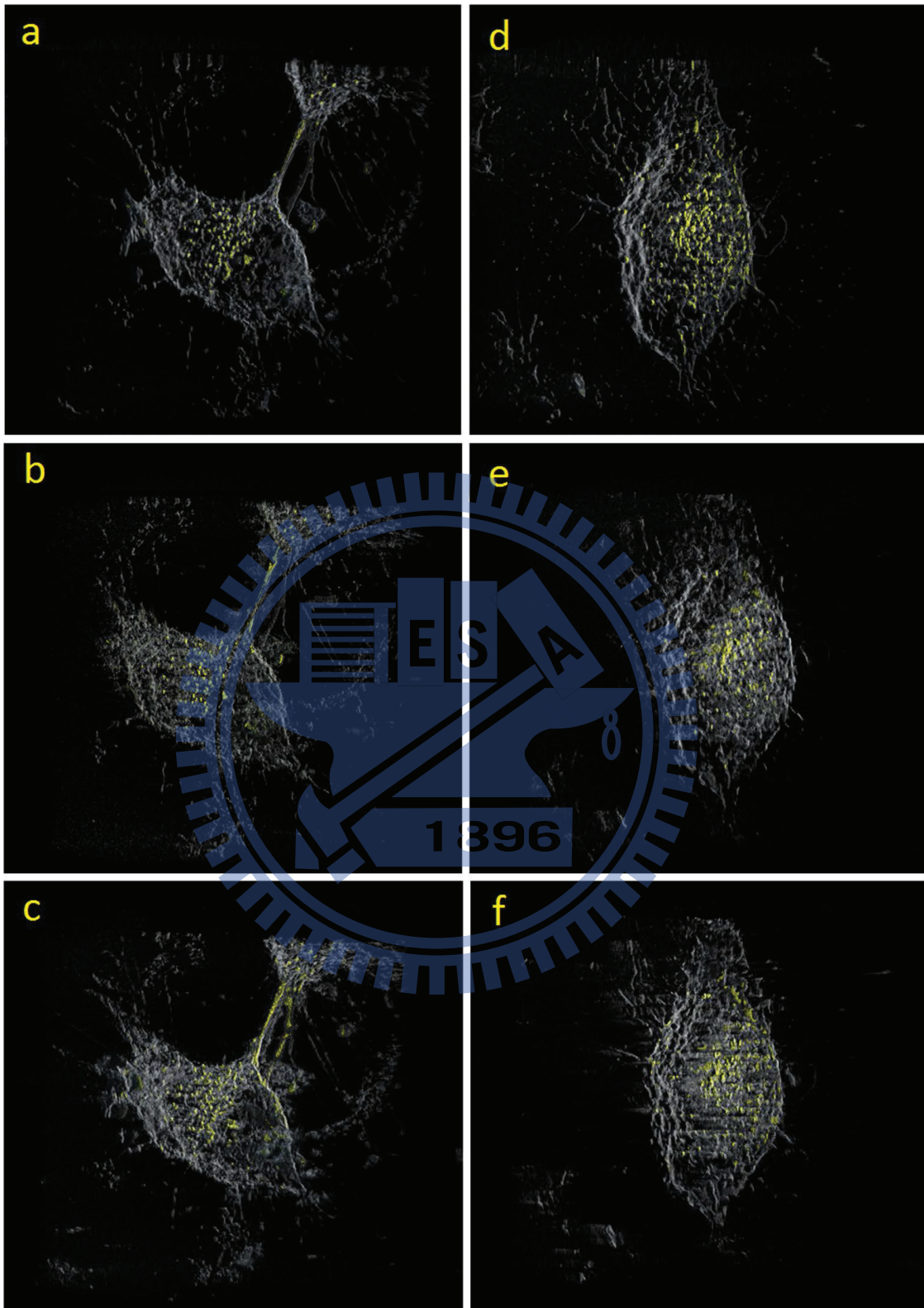


Figure 4.9: **The 3D volume rendering of the reconstructed volume** (a), (b), and (c) are HeLa1; (d), (e), and (f) are HeLa2: (a) and (d) Results of proposed method; (b) and (e) results of SPIDER, and (c) and (f) results of Xradia.



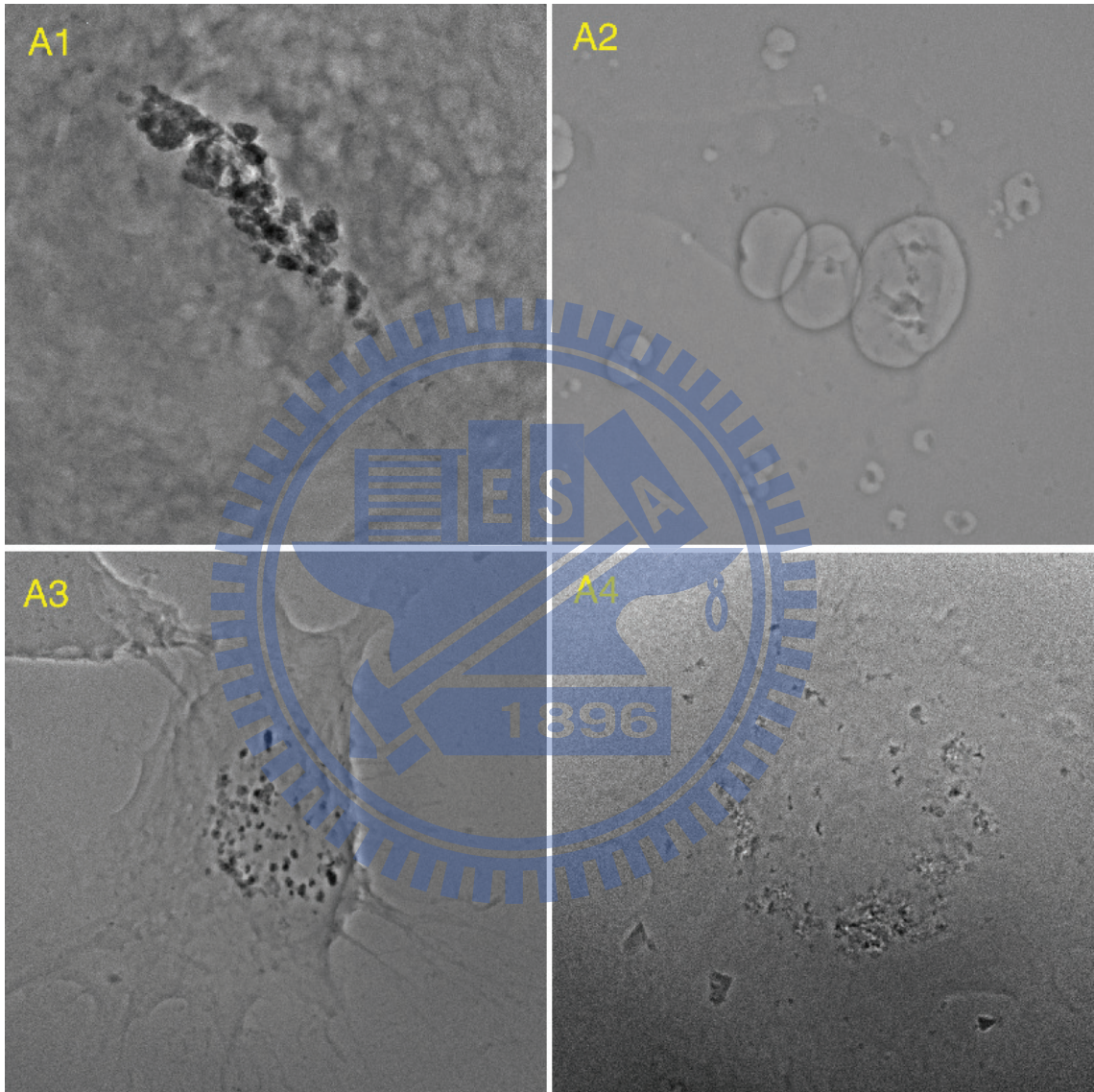


Figure 4.10: **Eight** examples of successful alignment (1).

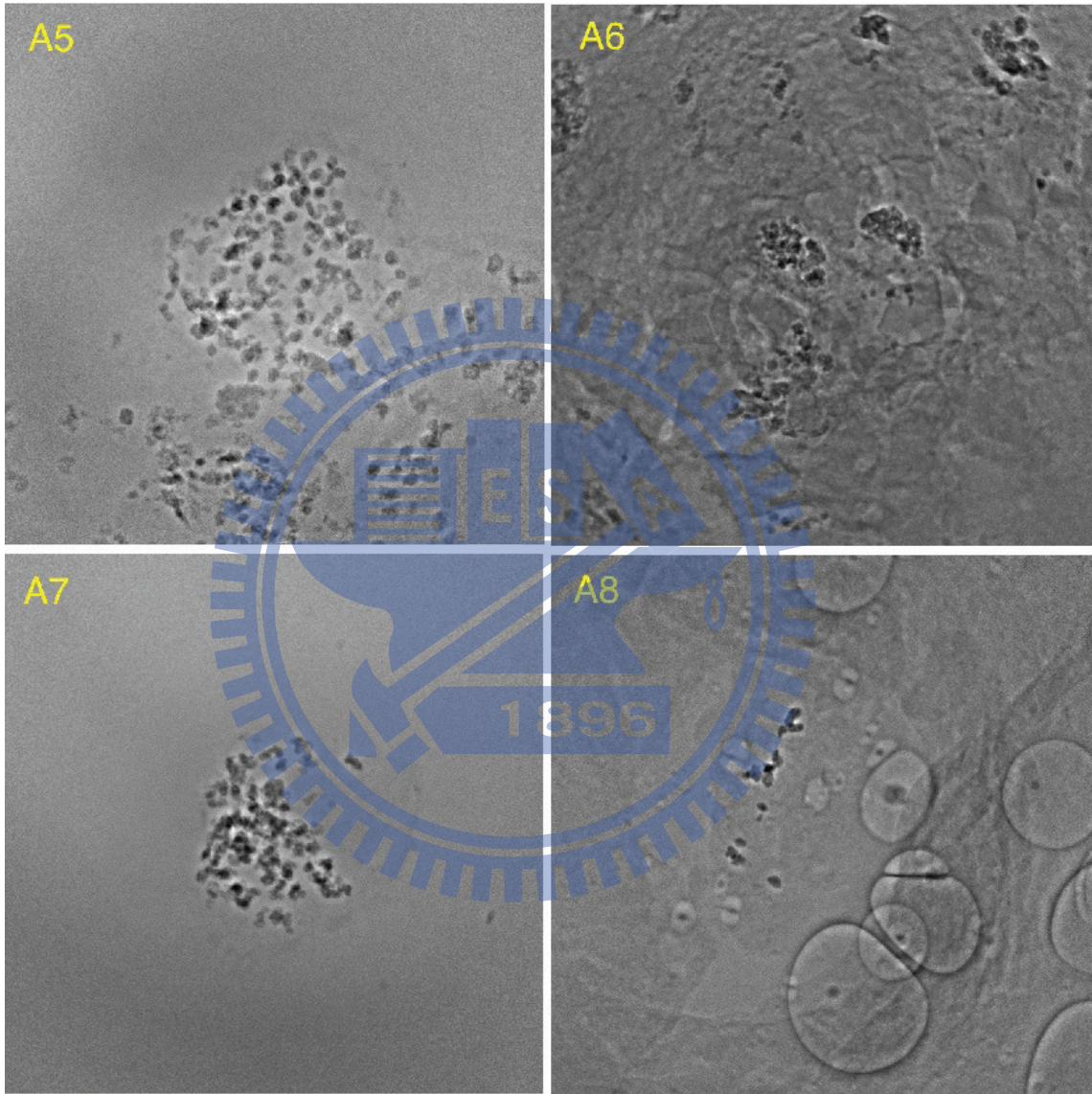


Figure 4.11: **Eight** examples of successful alignment (2).





Figure 4.12: **The first example of unsuccessful alignment, B1.** No one projected feature exists in the image series over 10 frames. The membrane of cell cannot be a feature because the low contrast and large size.

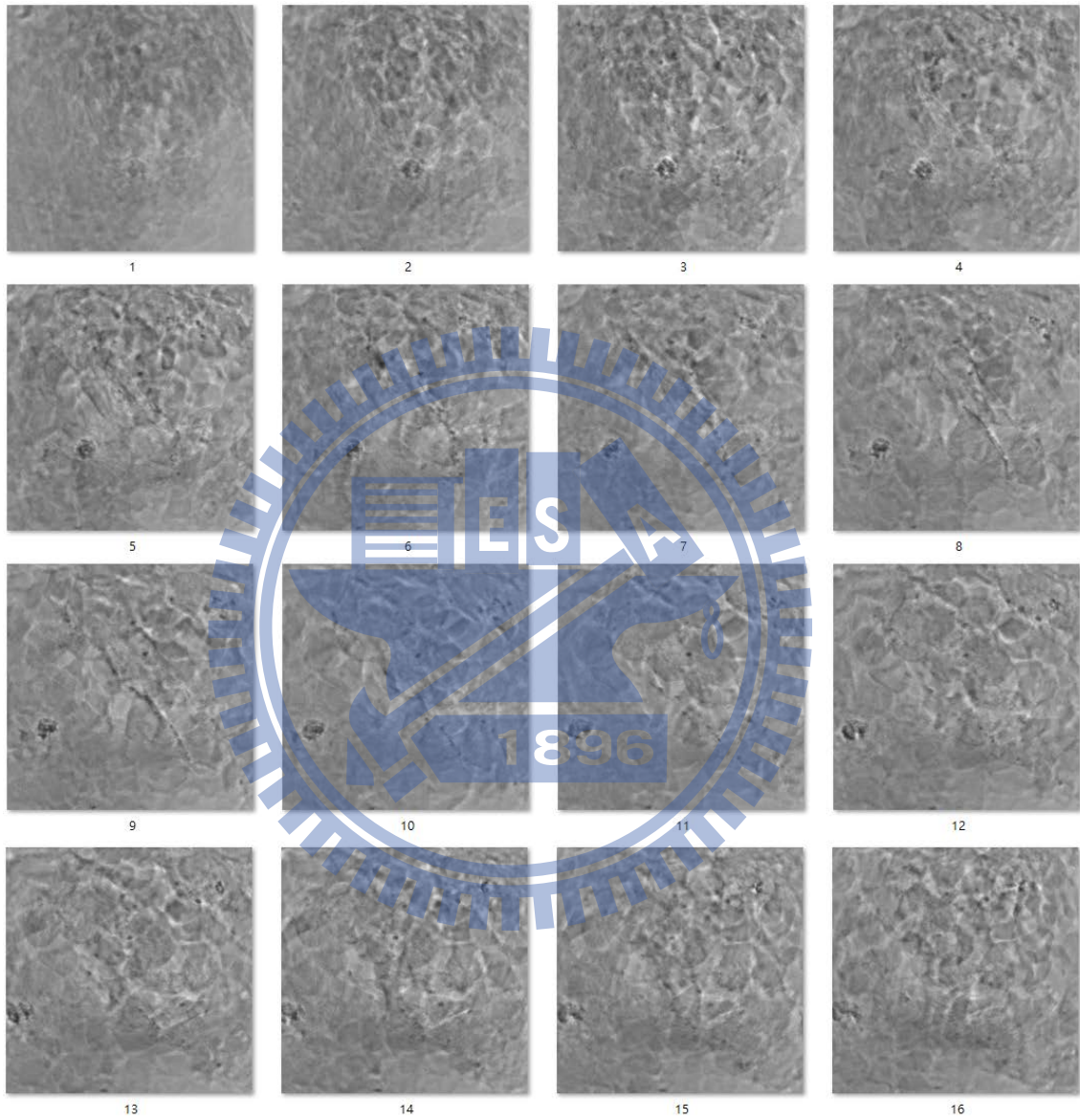


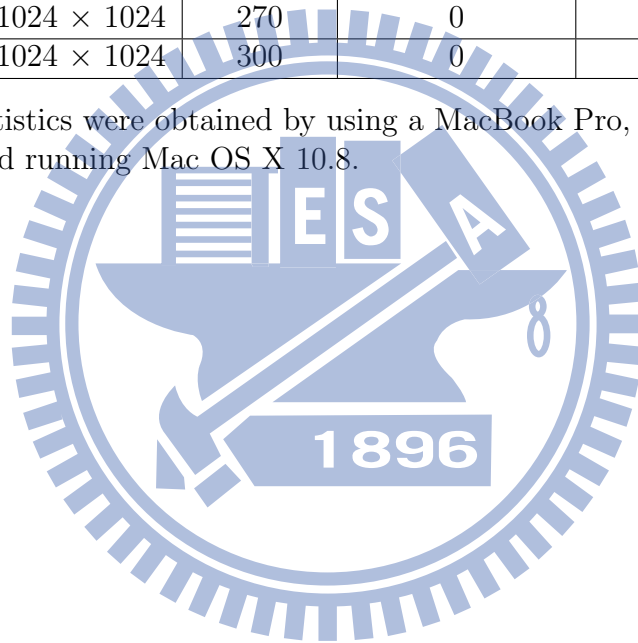
Figure 4.13: **The second example of unsuccessful alignment, B2.** No any reliable projected feature can be found in B2.



Table 4.1: **The test results.**

<b>Name</b>	<b>Size (pixel)</b>	<b>Number of images</b>	<b>Number of identified projected features</b>	<b>Number of reliable features</b>	<b>Time (sec.)</b>
Phantom	$512 \times 512$	180	16	16	102
HeLa1	$1024 \times 1024$	140	84	6	364
HeLa2	$1024 \times 1024$	140	89	4	397.5
A1	$1024 \times 1024$	300	86	3	886
A2	$1024 \times 1024$	150	85	4	393
A3	$1024 \times 1024$	140	104	72	518
A4	$1024 \times 1024$	320	64	8	712
A5	$1024 \times 1024$	140	121	4	454.5
A6	$1024 \times 1024$	280	80	8	784.5
A7	$1024 \times 1024$	140	103	4	406
A8	$1024 \times 1024$	280	26	3	590
B1	$1024 \times 1024$	270	0	0	477
B2	$1024 \times 1024$	300	0	0	490

The run time statistics were obtained by using a MacBook Pro, Intel i7 2.2GHz, 8GB main memory, and running Mac OS X 10.8.



## Chapter 5

# High-Resolution Large Volume

## Reconstruction

A noticeable application of X-ray microscope is to construct a high-resolution tomography of an object without physically slicing the object. If the object is large, because of the field-of-view limitation, a projection of the object has to be taken by several shots, then to stitch the sub-images together. The tomography of the object can then be reconstructed from the stitched projections. We call this the high-resolution large volume, HRLV, problem. To this problem, we propose the “stitch, align, and then reconstruct” approach.

Given a set of images, if the overlap area between two images is sufficiently large, these images can be stitched to a single large image [57, 58, 59]. In this study, because the radiation dosage should be minimized, the number of X-ray image acquisitions is reduced as much as possible. As a consequence, the overlap of images could be small. The projection model of the synchrotron microscopy is regarded as the parallel beam projection model because the frame size is far less than the distance between light source and object. Object tilting during the acquisition also can be ignored if the observed object is fixed in a container. Thus, we only considers the vertical and horizontal displacements for the stitching. In this study, the size of the overlap was 15%-20% of the image size.

The problem with the mechanical stability also exists. The proposed feature-based alignment method in Chapter 4 was applied for the calibration. Finally, the high-

resolution volume data then can be reconstructed by FBP or ART.

In this study, real data sets were not available because the time required for image acquisition is impractically long. A few  $5 \times 4$  projections require a whole day for image acquisition. We used a phantom data set in this study.

## 5.1 Stitching

The procedures of image acquisition and stitching are described as follows. First, the whole image space is divided into a set of uniform grid blocks as shown in Figure 5.1. The size of each overlap area was ranged 15%-20% of the image blocks. The image alignment method used for stitching is stated in Appendix B.3. The result of the image alignment is bounded by a limited range as shown in Fig. 5.2. If the alignment result is out of the limited range, it is replaced by the average from the alignment results of the other image pairs. This situation occurs when there are insufficient features in the overlapped area. Thus taking average does not cause too much bad effects.

In this study, the maximum number of reference images is two, which are the horizontal and vertical neighbours of  $I_a$  (Fig. 5.3). When an image has two reference images, first, we estimate the transformations and similarities between  $I_a$  and all references respectively. Then, these similarities are used as the interpolation ratios to calculate a alignment result by the linear interpolation.

Because the center image usually contains the most significant features, we choose the center image to be the first reference. Given  $m \times n$  image blocks, we select the  $\lfloor \frac{n}{2} \rfloor$ th image of the  $\lfloor \frac{m}{2} \rfloor$ th row as the center image block. All image blocks then can be sequentially aligned to the center image block as shown in Fig. 5.4. The alignment order is described as follows.

1. For the left part of middle row, from right to left, aligning each image to its right-side image.
2. For the right part of middle row, from left to right, aligning each image to its left-side image.

3. For the top part of middle column, from bottom to top, aligning each image to its down-side image.
4. For the bottom part of middle column, from top to bottom, aligning each image to its up-side image.
5. For the top-left part, from right-to-left and then bottom-to-top, aligning each image to its right-side and down-side images.
6. For the top-right part, from left-to-right and then bottom-to-top, aligning each image to its left-side and down-side images.
7. For the bottom-right part, from left-to-right and then top-to-bottom, aligning each image to its left-side and up-side images.
8. For the bottom-left part, from right-to-left and then top-to-bottom, aligning each image to its right-side and up-side images.

The multi-band image blending method [60] proposed by Burt and Adelson is applied to blend the overlaps. For completeness, we briefly state the blending algorithm. First, a Gaussian-Laplacian pyramid is constructed for each input image such that the image can be decomposed into a set of sub-band signals. Define two signal sampling operators,  $D_2$  and  $U_2$ .  $D_2$  is the downsampling operator that halves the sampling rate of a signal, and  $U_2$  is the upsampling operator that doubles the sampling rate of a signal. Given an image  $I$ , the  $m$ -level Gaussian-Laplacian pyramid of  $I$ ,  $\mathbf{L} = \{L_1, L_2, \dots, L_m\}$ , is constructed by the following equations.

$$L_i = I_i - I'_i, i = 1, 2, \dots, m - 1, \quad (5.1)$$

$$L_m = I_m, \quad (5.2)$$

where

$$I_1 = I, \quad (5.3)$$

$$I_i = D_2(I_{i-1}), i = 2, 3, \dots, m, \quad (5.4)$$

$$I'_i = G(I_i), \quad (5.5)$$

where  $G$  is the Gaussian operator. Fig. 5.1 shows the construction of Gaussian-Laplacian pyramid.

The image  $I$  can be reconstructed by  $\mathbf{L}$  as follows:

$$I_i = I''_i + L_i, i = 1, 2, \dots, m - 1, \quad (5.6)$$

$$I = I_1, \quad (5.7)$$

where

$$I''_i = G(U_2(I_{i-1})) \quad (5.8)$$

Fig. 5.1 shows the reconstruction of Gaussian-Laplacian pyramid.

Given two images,  $I_A$  and  $I_B$ , and their Gaussian-Laplacian pyramids,  $\mathbf{L}_A = \{L_1^A, L_2^A, \dots, L_m^A\}$  and  $\mathbf{L}_B = \{L_1^B, L_2^B, \dots, L_m^B\}$ . The result image  $I_{AB}$  can be blended by the following algorithm.

1. Given an image,  $I_W$ , to store the blending weight of each pixel,  $I_W(x, y) = 1$  if the location of  $(x, y)$  is in the area of  $I_A$ , otherwise  $I_W(x, y) = 0$ .
2. Create a pyramid,  $\mathbf{W} = \{W_1, W_2, \dots, W_m\}$ , by the following equations.

$$W_1 = I_W, \quad (5.9)$$

$$W_i = D_2(G(W_{i-1})), i = 1, 2, \dots, m. \quad (5.10)$$

3. Construct a Gaussian-Laplacian pyramid  $\mathbf{L}_{AB} = \{L_1^{AB}, L_2^{AB}, \dots, L_m^{AB}\}$ ,

$$L_i^{AB}(x, y) = L_i^A(x, y)W_i(x, y) + (1 - W_i(x, y))L_i^B(x, y), \quad (5.11)$$

where  $i = 1, 2, \dots, m$ .

4. Reconstruct  $\mathbf{L}_{AB}$  to obtain the result image  $I_{AB}$ .

Fig. 5.1 shows an example of image blending. In the figure, the left image,  $I_A$ , was transformed such the the similarity of overlap between  $I_A$  and  $I_B$  is maximum. The area bounded by yellow lines represents the area of  $I_A$ . Then, a pyramid,  $\mathbf{W}$ , was constructed to store the blending weights. Finally, the result,  $I_{AB}$ , can be obtain by the blending algorithm.



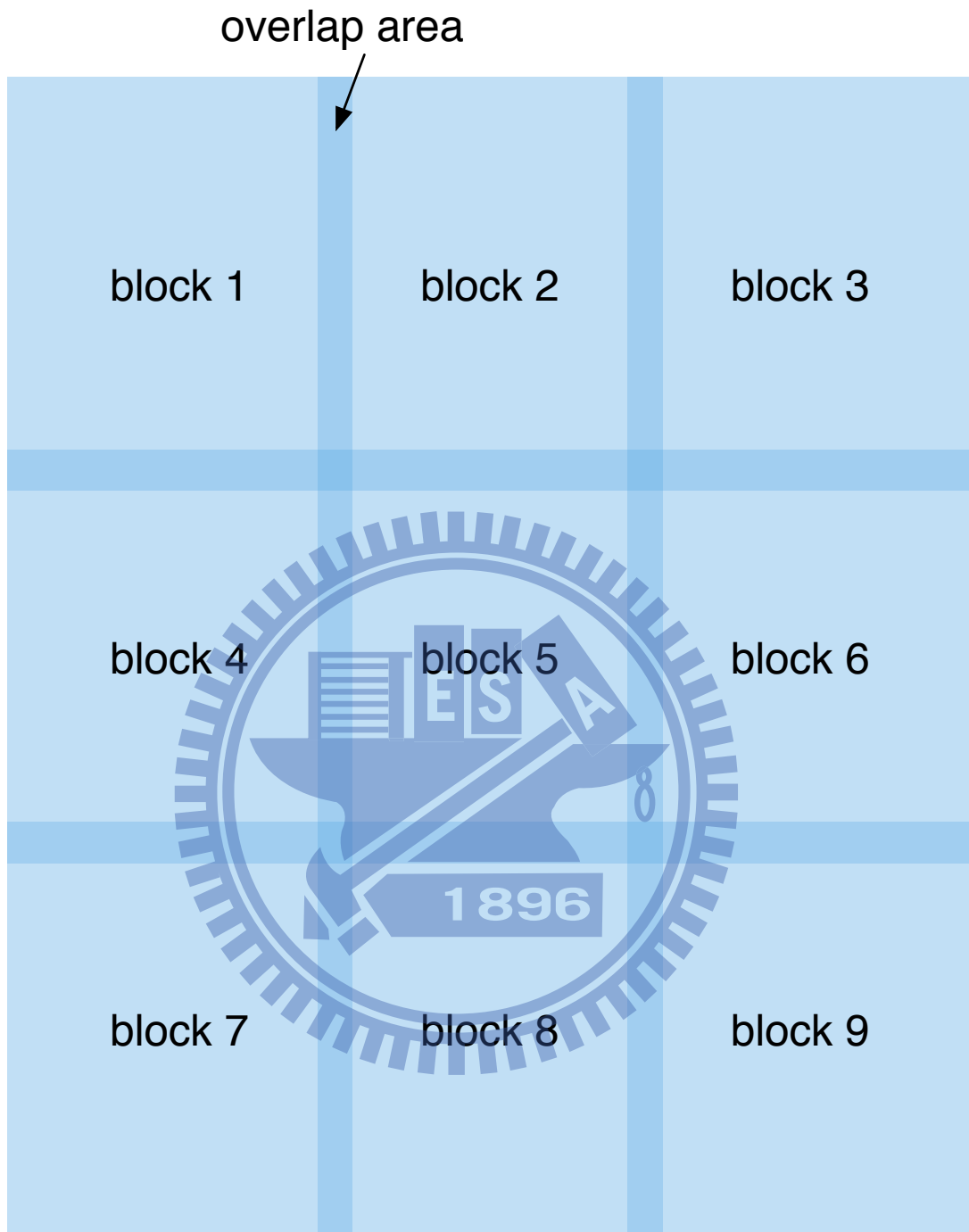


Figure 5.1: **The acquisition for a large image.** An example of  $3 \times 3$  uniform image blocks. The label of each block represents the acquisition order. There is a small overlap (15% to 20%) between each adjacent blocks for alignment and stitching.



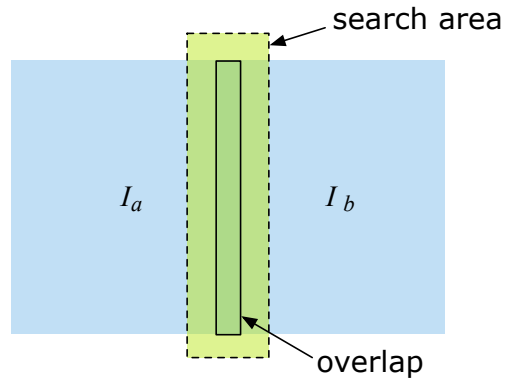


Figure 5.2: **The search area for alignment.** The alignment is bounded in a search area to avoid the divergent solution.

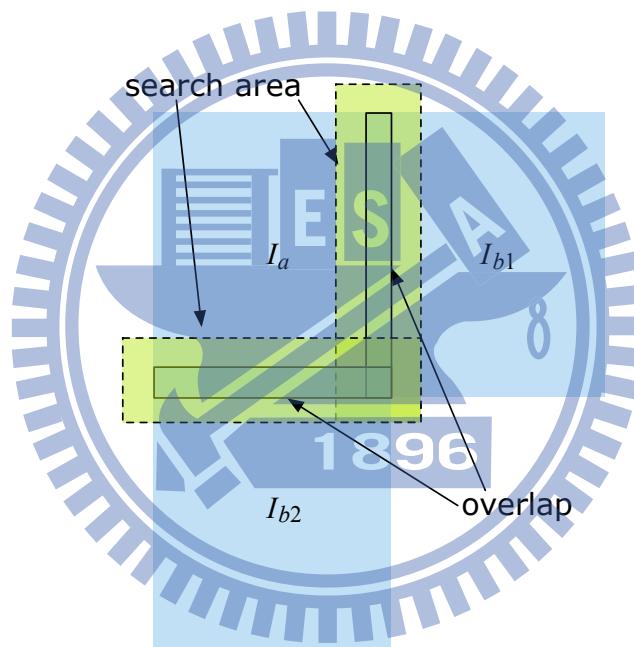


Figure 5.3: **The alignment of multiple references.** The transformation of  $I_a$  is estimated by linear interpolation from the transformations between  $I_a$  and all references ( $I_{b1}$  and  $I_{b2}$ ).

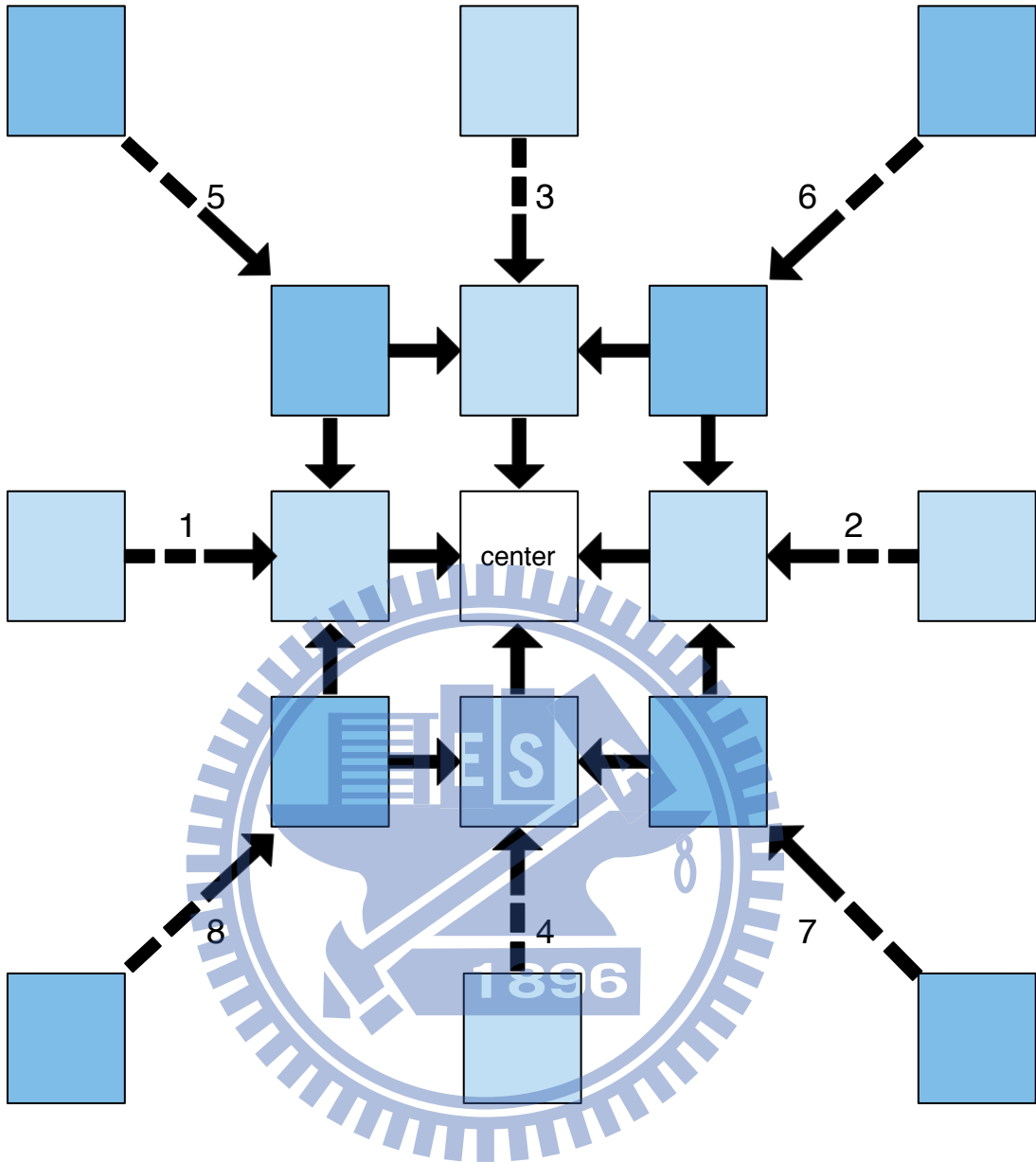


Figure 5.4: **The stitch order for a large image.** The numerical labels mark the stitch order. This order enables that all images are aligned to the center image.

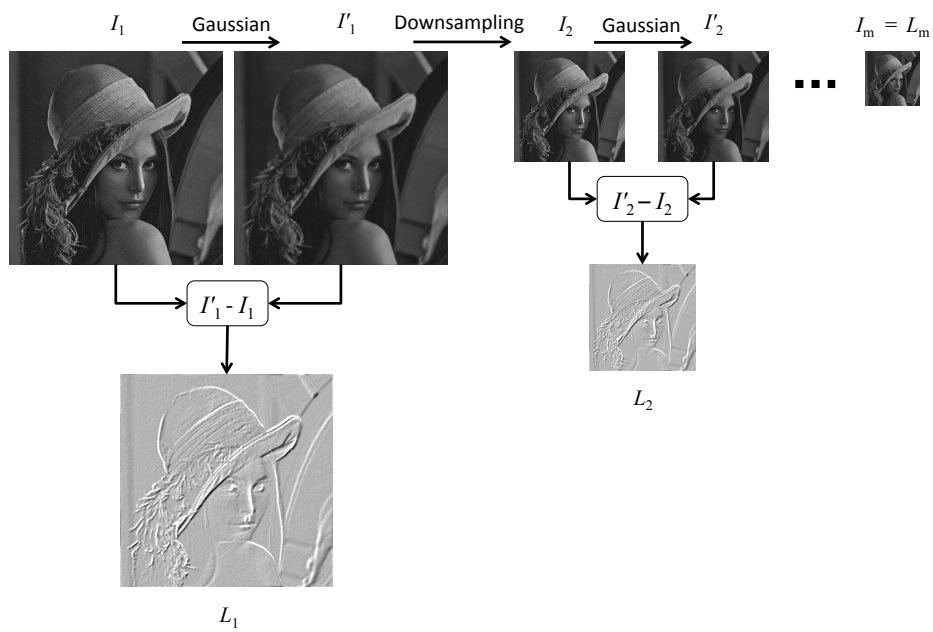


Figure 5.5: **The construction of Gaussian-Laplacian pyramid.** A  $m$ -level Gaussian-Laplacian pyramid,  $\mathbf{L} = \{L_1, L_2, \dots, L_m\}$ , is constructed from the input image  $I_1$ .

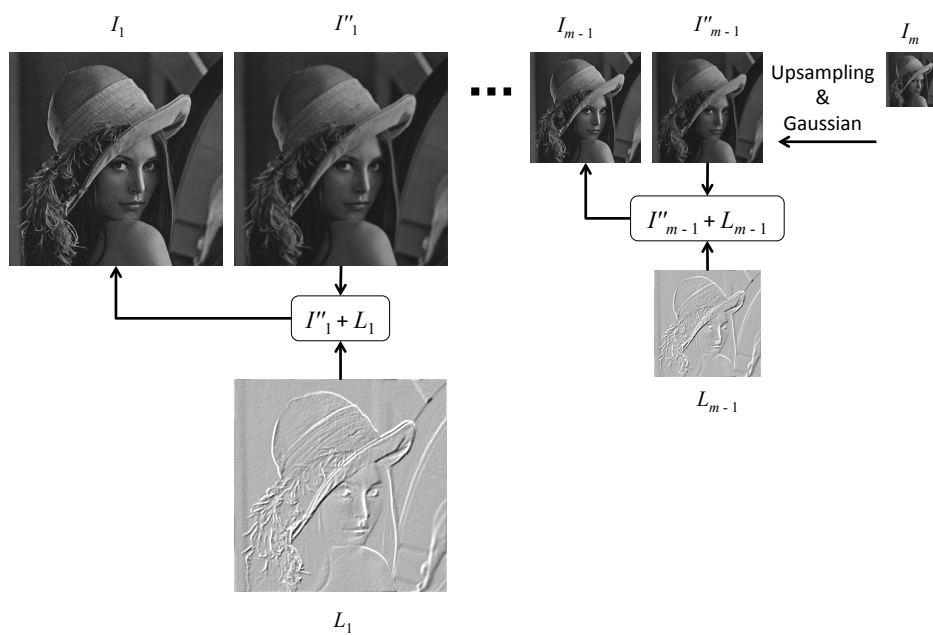


Figure 5.6: **The reconstruction of Gaussian-Laplacian pyramid.** The original image  $I_1$  can be reconstructed from its  $m$ -level Gaussian-Laplacian pyramid,  $\mathbf{L} = \{L_1, L_2, \dots, L_m\}$ .

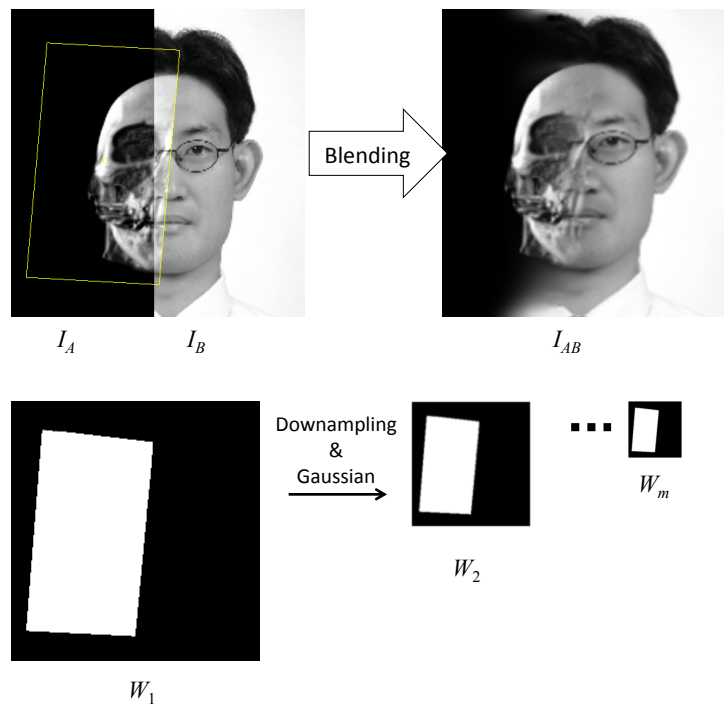


Figure 5.7: **The image blending.**  $I_A$  was transformed such the the similarity of overlap between  $I_A$  and  $I_B$  is maximum. The area bounded by yellow lines represents the area of  $I_A$ . Then, a pyramid,  $\mathbf{W} = \{W_1, W_2, \dots, W_m\}$ , was constructed to store the blending weights. Finally, the result,  $I_{AB}$ , can be obtain by the blending algorithm.

## 5.2 Results

A volume data of phantom containing 512 imitative gold nanoparticles was constructed, and X-ray projections of the shifted volume were generated to simulate the machine vibration and the field-of-view limitation. Fig. 4.3 shows the simulated X-ray image of the phantom, in which 180 projections of  $7 \times 7$  images of  $512 \times 512$  pixels were generated. The rotation angle is  $1^\circ$  between successive projections and the ratio of overlaps was 20%. The amount of displacement of machine vibration was determined according to a random number uniformly distributed over the range  $\pm 10$  pixels. Fig. 5.8(a) shows the image blocks of the first projection. The proposed stitching algorithm was used to combine the image blocks of each projection. Fig. 5.8(b) shows the stitch result of Fig. 5.8(a). The size of each stitched image was  $3072 \times 2866$  pixels. The alignment algorithm of Ch. 4) was then used to correct the misalignment. Finally, the tomography reconstruction algorithm is applied to reconstruct the stitched images. The size of each reconstructed image was  $3072 \times 3072$  pixels, and 2866 images were reconstructed. Fig. 5.9(a) shows the tomography result reconstructed by Tomo3D(SIRT). Fig. 5.9(b) shows the tomography result reconstructed by FBP.

The following describes the run time statistics that were obtained by using Intel i7 3.4GHz, 12GB main memory, nVidia GTX760, and running Windows 8.1. The image stitching consumed 114 seconds per projection angle. The misalignment correction consumed 7023 seconds. The tomography reconstruction consumed 63.6 seconds per image. The total time was 11.5 hours.



(b)

Figure 5.8: **Image stitching for the phantom.** (a)  $7 \times 7$  image blocks of  $512 \times 512$  pixels. (b) The stitched result.





(b)

Figure 5.9: **The tomographic result reconstructed from the stitched images of phantom.** (a) A slice of volume data reconstructed by Tomo3D(SIRT). (b) A slice of volume data reconstructed by FBP

### 5.3 Discussion

In HRLV, the object holder is required to accurately move to a specified position. However, the current hardware system is unable to support the proposed method when it is used to acquire the tomographic images. Because its control system of object holder is designed for single-image acquisition of one rotation, the position on X and Y axes cannot be moved at each projection angle. Thus, a multiple-image acquisition of one rotation requires several manual operations and an unacceptable amount of time. For example, the acquisition of  $5 \times 5$  images requires 2-3 hours. The manual operations also could lead a large position error. So far, the propose method is not applied to a real data since the deficiency of hardware ability.

Another problem is that the reconstruction of outer ring of volume cannot be done well. Both FBP and ART cannot successfully reconstruct the outer ring. This problem can be observed in the phantom data (Fig. 5.9(a) and 5.9(b)). There were several particles evenly distributed in the volume. The particles near the center can be reconstructed successfully. However, the particles near the boundary of volume cannot be reconstructed successfully. This problem is caused by the alignment error and large angle between projections. The proposed alignment method is required to be more accurate. The reconstruction algorithm is required to deal with the large projection angle.

# Chapter 6

## Conclusions and Future Work

### 6.1 Summary

This study used image processing techniques to enable the three-dimensional tomography by using the synchrotron X-ray microscope. The vibration of object holder may cause the misalignment problem of tomography. The tomography reconstruction is unsuccessful if misalignment exists in a serial of X-ray images obtained by different viewing direction. This study proposed a feature-based alignment method to efficiently correct the misalignment problem. The proposed method identifies projected feature points and classifies the projected feature points into a set of loci. The key idea of this method is that the projection of a point in the object should be a sine wave in the  $x$ - $\theta$  coordinate system. Thus, fitting the set of loci to a set of sine waves can compute the parameters required for alignment.

A high-resolution image of an object could not be taken in one image acquisition because the field-of-view limitation. The image space is divided to several image blocks are taken in different view position. This study proposed a method of seamless image stitching that can combine these image blocks to produce a large high-resolution image. Because the synchrotron radiation is a parallel beam traversal and the object is fixed in a holder, the alignment variables in stitching can be reduced to the translation in vertical and horizontal directions. Therefore, the size of overlap between any two image blocks can be reduced to 15%-20% of the size of image block. After all image blocks are aligned such

that the similarity of overlap between any two image blocks is maximum, the multi-band image fusion algorithm then be applied to mix these image blocks.

The reconstructed volume data of nanoscale object can be verified by a visualization technique. This study used two real-time interactive techniques of volume rendering for volume data visualization, TVR and FVR. TVR that is a popular visualization technique uses a graphics hardware to simulate the integration of ray-casting projection. TVR and its transfer functions can be easily implemented with lighting and shadowing to generate a colorful image. Another visualization technique is FVR, a frequency-based method of volume rendering. Because the voxel classification in FVR equals the convolution in frequency domain, the transfer function design in FVR is more difficult than in TVR. This study proposed a method that using B-spline to design transfer functions for the voxel classification in FVR. The user can adjust the control points of B-spline to immediately change the transfer function in rendering time.

All of proposed methods have been implemented a GUI (graphical user interface ) software system by C++ with QT library, a cross-platform library for GUI programming . This software can be executed on Windows 8.1 and Mac OS X 10.9.

## 6.2 Future Work

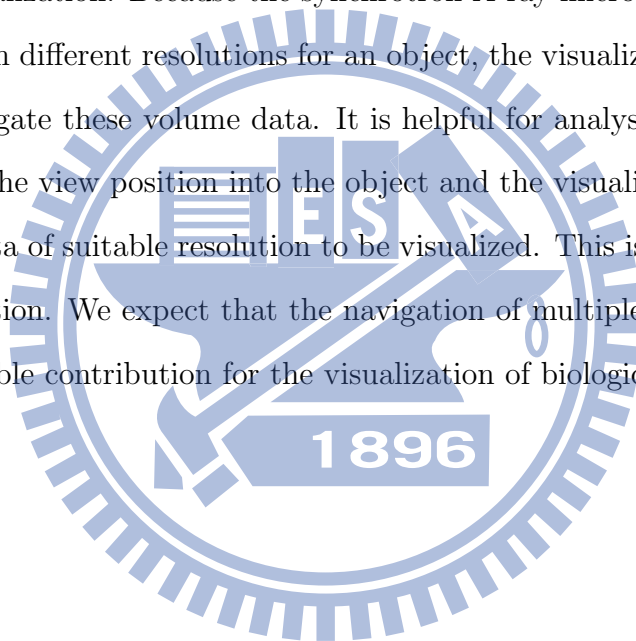
In the proposed feature-based alignment method, two main reasons causes the alignment failed, the insufficient number of features and low-contrast features. The first problem can be solved by using the loci tracked from partial frames to calculate the correction value. For the second problem, the enhancement of feature extraction is required for the detection of low-contrast features.

The proposed feature-based alignment method requires that the object is fixed in a holder during the image acquisition. However, the object could shrink or swell during the image acquisition in the in-vivo experiments. Therefore, the modification of feature matching in alignment is required for the tracking of a deformed feature.

The intensity range of all image blocks could not be the same because each image block is taken in the different imaging condition including the intensity of radiation,

exposure time, and ambient illumination. This problem is also existed in the background signal removal. If the intensity ranges of the background image and object image are different, the division operation could generate more unwanted signal. There are many intensity calibration methods have been proposed including histogram equalization [43], radiometric calibration [61], and camera response functions [62]. However, these methods are suitable for photographic images rather than synchrotron X-ray images. The intensity calibration is required to consider the physical optics of synchrotron X-ray.

The visualization for large-scale volume data is the next research work. An efficient level-of-detail method will be applied to our visualization software system for the large volume data visualization. Because the synchrotron X-ray microscope can acquire multiple volume data in different resolutions for an object, the visualization system is required to efficiently navigate these volume data. It is helpful for analysis and verification if the user can control the view position into the object and the visualization system can alternate a volume data of suitable resolution to be visualized. This is an unexplored research topic in visualization. We expect that the navigation of multiple-resolution volume data can bring a valuable contribution for the visualization of biological data.



# Appendix A

## Background Removal

The synchrotron X-ray images exist a lot of background signals caused by noises, dirty camera lens, and uneven illumination. There are many efficient image processing techniques that can be used to eliminate the unwanted signals for photographic images [43]. The subtraction is the most used operator for photographic images. However, the division operator is used to remove the background signals for X-ray images. According to the optics model of X-ray [63], an X-ray projection can be formulated as follows.

$$I_d = I_0 \cdot I_1 \cdot I_2, \quad (\text{A.1})$$

where  $I_d$  is the intensity of X-ray received by radiation detector,  $I_0$  is the initial intensity of X-ray,  $I_1$  represents the attenuation of intensity caused by noise and background signal, and  $I_2$  represents the attenuation of intensity caused by object. Let  $I_b = I_0 \cdot I_1$ , and  $I_b$  can be acquired from an empty object holder. The image of object,  $I_2$ , then can be calculated by the following equation.

$$I_2 = \frac{I_d}{I_b}. \quad (\text{A.2})$$

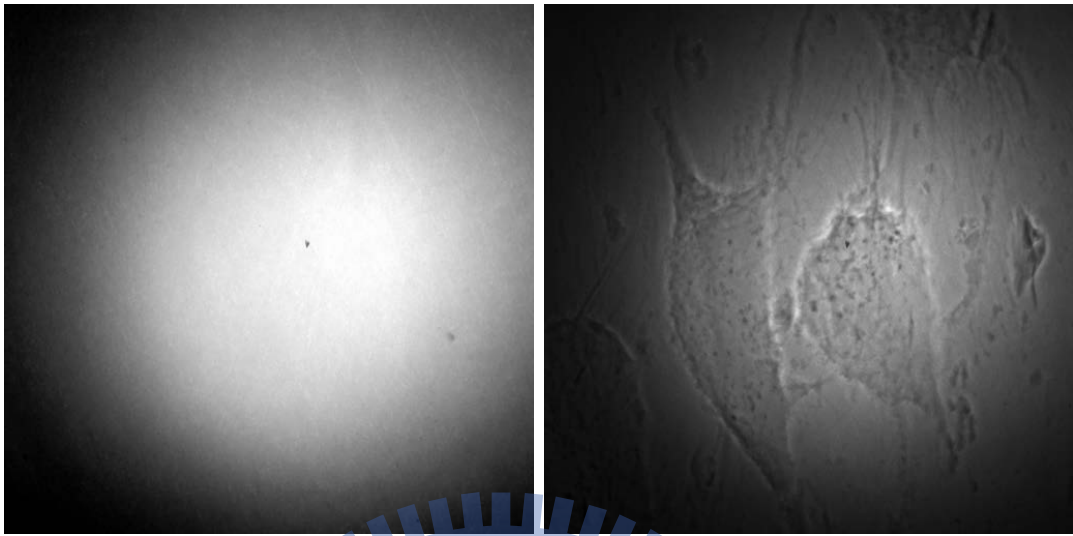
Fig. A.1 shows the background removal procedure. The sample object was a HeLa cell, the image size was  $1024 \times 1024$  pixels, and the pixel size was 11.78 nm. Fig. A.1(a) shows the background image,  $I_b$ , and A.1(b)) shows the raw image,  $I_d$ . The pixel-wise

division operator then can be applied to obtain the result of background removal, Fig. A.1(c).

This background removal method requires that  $I_b$  and  $I_d$  must be acquired in the same imaging conditions including the energy of radiation and exposure time. Otherwise, some background signals could not be removed, or the intensities of background signals could be emphasized.

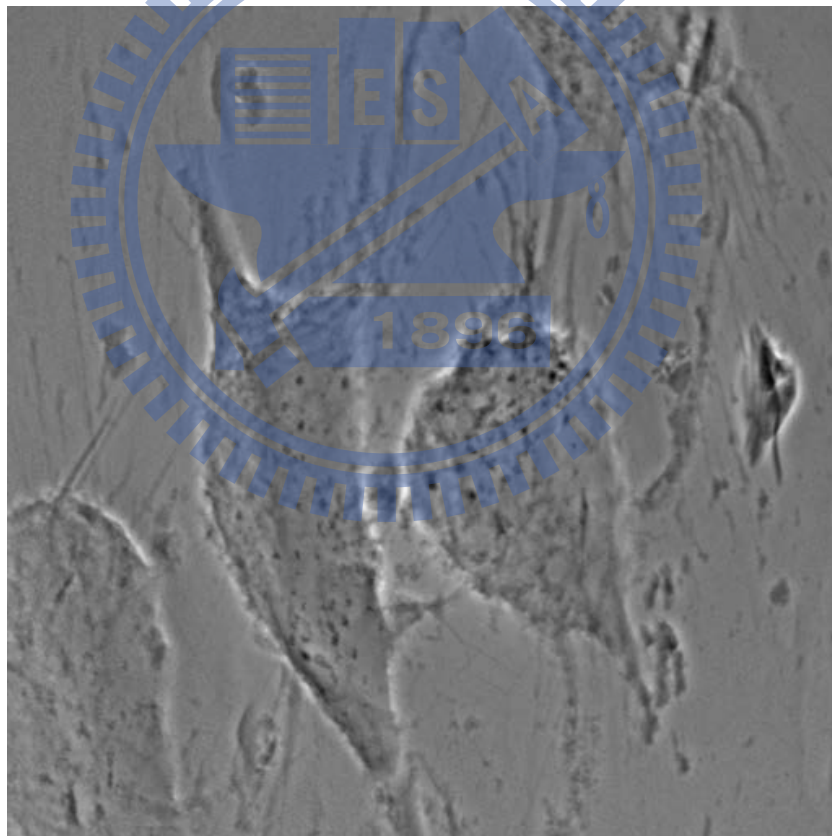






(a)

(b)



(c)

Figure A.1: **The procedure of background removal.** (a) The background image. (b) The raw image. (c) The result of background removal.

# Appendix B

## Image Similarity

Two methods of similarity estimation between two images, sum of squared error and mutual information, are described in this section. These two methods are commonly used to quantize the degree of similarity between two sets of discrete data in information theory.

**Sum of Squared Error** The first method of similarity estimation is sum-squared error, SSE. Assume that  $\mathbf{A}$  and  $\mathbf{B}$  are two discrete data sets of  $n$  elements,  $\mathbf{A} = \{a_1, a_2, \dots, a_n\}$ , and  $\mathbf{B} = \{b_1, b_2, \dots, b_n\}$ . The SSE between  $\mathbf{A}$  and  $\mathbf{B}$ ,  $S(\mathbf{A}, \mathbf{B})$  can be computed by the following equation.

$$S(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^n (a_i - b_i)^2, \quad (\text{B.1})$$

SSE is commonly used to quantize the difference between two photographic images. [64]. Let  $I_A(\mathbf{x})$  and  $I_B(\mathbf{x})$  represent two images, where  $\mathbf{x}$  is the location of each pixel, and the gray intensity of each pixel is normalized to the range of 0 and 1. The similarity of  $I_A$  and  $I_B$  can be estimated by SSE:

$$S(I_A, I_B) = \sum_{\forall \mathbf{x}} (I_A(\mathbf{x}) - I_B(\mathbf{x}))^2, \quad (\text{B.2})$$

To prevent the interference of noise, the images can be processed by a noise-removal filtering [43] before the SSE estimation. If the intensity ranges of two images are different, the similarity estimated by SSE could not accurate. Using the differentiated images to

calculate SSE can solve this problem.

$$S(I'_A, I'_B) = \sum_{\forall \mathbf{x}} (I'_A(\mathbf{x}) - I'_B(\mathbf{x}))^2, \quad (\text{B.3})$$

Further, the equation of SSE can be expanded as follows:

$$S(I_A, I_B) = \sum_{\forall \mathbf{x}} I_A(\mathbf{x})^2 - \sum_{\forall \mathbf{x}} I_B(\mathbf{x})^2 - 2 \sum_{\forall \mathbf{x}} I_A(\mathbf{x})I_B(\mathbf{x}), \quad (\text{B.4})$$

The final term in Eq. B.4 can be used to measure the similarity between  $I_A$  and  $I_B$ . If the pixel location of  $I_A$  can be transformed by  $\delta$  such that  $I_A(\mathbf{x} + \delta)$  is similar to  $I_B(\mathbf{x})$ , the cross-correlation,  $C$ , is defined as follows.

$$C(I_A, I_B) = \int_{-\infty}^{\infty} \sum_{\forall \mathbf{x}} I_A(\mathbf{x} + \delta)I_B(\mathbf{x})d\delta. \quad (\text{B.5})$$

The cross-correlation is commonly used to measure the similarity of two waveforms in signal processing [65]. The frequency-based alignment uses the cross-correlation to align two images [38].

Although SSE can be easily implemented, the similarity estimated by SSE could not accurate if the number of pixels is insufficient. A more accurate method of similarity estimation is needed for the images of small size.

## B.1 Entropy

The entropy can quantize the complexity of a discrete data set in information theory [66]. Let  $I(x)$  represents a radiation intensity at a location on image and  $\mathbf{x} \in \mathbf{R}^2$ . All intensity values of  $I(x)$  are then classified to  $k$  bins, the width of each bin  $w = \frac{\max(I) - \min(I)}{k}$ , and the range of the  $j$ th bin  $\mathbf{b}_j = [(j - 1)w, jw]$ ,  $j = 0, 1, \dots, k$ . The intensity histogram of  $I(x)$  in  $\mathbf{b}_j$ ,  $H(I, \mathbf{b}_j)$ , is defined by the following equation.

$$H(I, \mathbf{b}_j) = \frac{1}{n} \sum_{\forall \mathbf{x}} \psi(I(\mathbf{x}), \mathbf{b}_j), \quad (\text{B.6})$$

where

$$\psi(I(\mathbf{x}), \mathbf{b}_j) = \begin{cases} 1 & \text{if } I(\mathbf{x}) \in \mathbf{b}_j, \\ 0 & \text{otherwise,} \end{cases} \quad (\text{B.7})$$

and  $n = \sum_{j=1}^k \psi(\mathbf{b}_j, I(\mathbf{x}))$ , which represents the total number of pixels in the image  $I$ .

Based on the histogram, the Shannon entropy of an image,  $E$ , is defined as follows:

$$E(I) = - \sum_{j=1}^k H(I, \mathbf{b}_j) \ln H(I, \mathbf{b}_j), \quad (\text{B.8})$$

Because the range of  $H$  is 0 to 1, the value of entropy must be positive. If an image contains a simple pattern, for example, an image filled by a single color, its entropy must be close to zero. Otherwise, a high entropy is estimated if the image contains complex information. Therefore, the entropy is a reasonable method to determine the degree of complexity for an image.

## B.2 Mutual Information

Given two images of  $n$  pixels,  $I_A$  and  $I_B$ , and their histograms estimated by Eq. B.6, the mutual information of  $I_A$  and  $I_B$  is defined as the following equation:

$$M(I_A, I_B) = \sum_i^k \sum_j^k H'(I_A, I_B, \mathbf{b}_i, \mathbf{b}_j) \ln \frac{H'(I_A, I_B, \mathbf{b}_i, \mathbf{b}_j)}{H(I_A, \mathbf{b}_i)H(I_B, \mathbf{b}_j)}, \quad (\text{B.9})$$

where  $k$  is the number of bins and  $H'$  is the joint histogram function defined as follows:

$$H'(I_A, I_B, \mathbf{b}_i, \mathbf{b}_j) = \frac{1}{n} \sum_{\forall \mathbf{x}} \psi(I_A(\mathbf{x}), \mathbf{b}_i) \psi(I_B(\mathbf{x}), \mathbf{b}_j). \quad (\text{B.10})$$

Note that the mutual information can be calculated by entropies of images. Eq. B.9 can be rewritten as follows:

$$M(I_A, I_B) = E(I_A) + E(I_B) - E'(I_A, I_B), \quad (\text{B.11})$$

where  $E'(I_A, I_B)$  that is called the joint entropy of  $I_A$  and  $I_B$  is defined as follows:

$$E'(I_A, I_B) = - \sum_i^k \sum_j^k H'(I_A, I_B, \mathbf{b}_i, \mathbf{b}_j) \ln H'(I_A, I_B, \mathbf{b}_i, \mathbf{b}_j). \quad (\text{B.12})$$

According Eq. B.11, there is a relation between the entropies of two images and their mutual information:

$$0 \leq M(I_A, I_B) \leq \max(E(I_A), E(I_B)). \quad (\text{B.13})$$

If  $I_A$  and  $I_B$  are the same image, their mutual information must be equal to either  $E(I_A)$  or  $E(I_B)$ . Otherwise, the mutual information close to zero if  $I_A$  and  $I_B$  are extremely dissimilar. The mutual information can measure the degree of similarity between two images, even the images are acquired by different imaging modalities, for example, two images are respectively taken by X-ray camera and MRI (magnetic resonance imaging) apparatus. Therefore, the mutual information is a well objective criterion to quantize the similarity between two images acquired by synchrotron-radiation microscope.

### B.3 Two-Image Alignment

There are many applications of image alignment for photographic images, for example, object tracing, image stitching, and panoramic photography [64]. Fig . B.1 shows an example of the stitching of two images.

Given an image  $I$ , and a transformation function  $T$  with a parameter vector  $\mathbf{u}$ , the transformed image  $I'$  is derived by the following equation.

$$I'(\mathbf{x}) = I(T(\mathbf{x})). \quad (\text{B.14})$$

Given two images,  $I_A$  and  $I_B$ , the two-image alignment problem is that searching a  $\mathbf{u}$  to transform the location of all pixels in  $I_A$ , such that the similarity of  $I_A$  and  $I_B$  is maximum. The simplest solution for two-image alignment problem is the brute force method that tests all possible  $\mathbf{u}$ . However, this method is impracticable when the images are large or the dimension of  $\mathbf{u}$  is greater than two, for example, a transformation includes



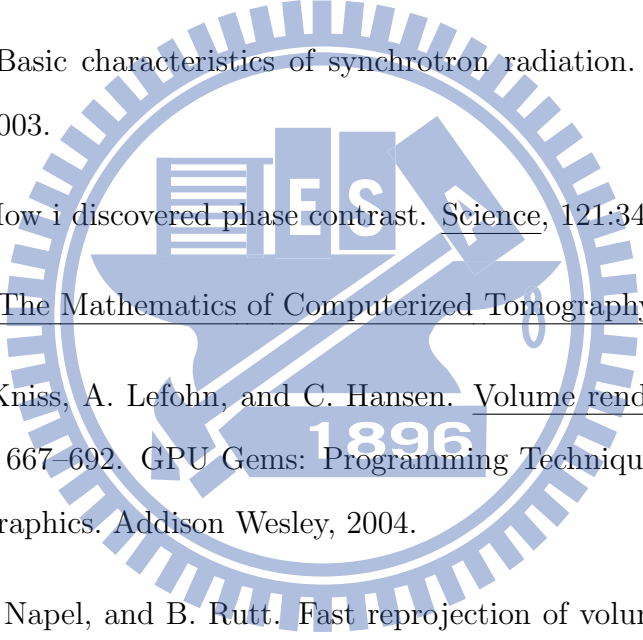
Figure B.1: **The image alignment.** (a)  $I_A$ . (b)  $I_B$ . (c) The alignment result of  $I_A$  and  $I_B$ .

shift, rotation, and scaling. The level-of-detail method (LOD) is a more efficient method than the brute force method. However, LOD could find a false solution although it can be easily implemented. The most popular alignment method is the LucasKanade method (LKM) [67]. The fundamental of LKM is that it iteratively find the increase of  $\mathbf{u}$  by the LevenbergMarquardt algorithm such that the SSE between two images is minimum. However, because the LKM is based on SSE, the alignment results of LKM could be failed if the images are taken from different imaging environments.

The mutual information can also solve the two-image alignment problem. Viola [51] proposed an alignment method by maximizing the mutual information of two images. Dowson and Bowden combined the mutual information and LKM to devise an efficient alignment algorithm.

Another alignment method based on the analysis in frequency domain has been proposed. Frank used the cross-correlation (Eq. B.5) to estimate the transformation from the frequency domain of images [38]. As the LKM, this method could not align the images taken by different imaging modalities.

# Bibliography

- 
- [1] R. Meuli, Y. Hwu, J. H. Je, and G. Margaritondo. Synchrotron radiation in radiology: radiology techniques based on synchrotron sources. European Radiology, 14, Issue 9:1550–1560, 2004.
- [2] G. Shenoy. Basic characteristics of synchrotron radiation. Structural Chemistry, 14(1):3–14, 2003.
- [3] F. Zernike. How i discovered phase contrast. Science, 121:345–349, 1955.
- [4] F. Natterer. The Mathematics of Computerized Tomography. SIAM, 2001.
- [5] M. Ikits, J. Kniss, A. Lefohn, and C. Hansen. Volume rendering techniques, chapter 39, pages 667–692. GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics. Addison Wesley, 2004.
- [6] S. Dunne, S. Napel, and B. Rutt. Fast reprojection of volume data. In Conference on Visualization in Biomedical Computing, pages 11–18, 1990.
- [7] T. Malzbender. Fourier volume rendering. ACM Transactions on Graphics, 12(3):233–250, 1993.
- [8] J. Radon. On the determination of functions from their integral values along certain manifolds. Medical Imaging, IEEE Transactions on, 5(4):170–176, 1986.
- [9] L.A. Shepp and B.F. Logan. The fourier reconstruction of a head section. Nuclear Science, IEEE Transactions on, 21(3):21–43, 1974.



- [10] R. N. Bracewell. The Fourier transform and its applications, 3rd edition. McGraw-Hill, 1999.
- [11] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. Mathematics of Computation, 19(90):297–301, 1965.
- [12] F. Xu and K. Mueller. Accelerating popular tomographic reconstruction algorithms on commodity pc graphics hardware. IEEE Transactions on Nuclear Science, 52(3):654–663, 2005.
- [13] A. C. Kak and M. Slaney. Principles of computerized tomographic imaging. Society for Industrial and Applied Mathematics, 2001.
- [14] J. I. Agulleiro and J. J. Fernandez. Fast tomographic reconstruction on multicore computers. Bioinformatics, 27(4):582–583, 2011.
- [15] J. I. Agulleiro and J. J. Fernandez. Evaluation of a multicore-optimized implementation for tomographic reconstruction. PLoS ONE, 7(11), 2012.
- [16] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In Proceedings of ACM SIGGRAPH Computer Graphics '87, volume 21, pages 163–169, 1987.
- [17] N. Max. Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics, 1(2):99–108, 1995.
- [18] B. T. Phong. Illumination for computer generated pictures. Commun. ACM, 18(6):311–317, June 1975.
- [19] J. Kniss, C. Hansen, P. Shirley, and A. Mcpherson. A model for volume lighting and modeling. IEEE Transactions on Visualization and Computer Graphics, 9:150–162, 2003.
- [20] K. Engel, M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In Proceedings of Eurographics/SIGGRAPH Workshop on Graphics Hardware, pages 9–16, 2001.

- [21] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Mller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In Proceedings IEEE Visualization 2003, pages 513–520, 2003.
- [22] Eric B. Lum and Kwan-Liu Ma. Lighting transfer functions using gradient aligned sampling. In Proceedings IEEE Visualization 2004, pages 289–296, 2004.
- [23] J. J. Caban and P. Rheingans. Texture-based transfer functions for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics, 14(6):1364–1371, 2008.
- [24] C. D. Correa and K. L. Ma. Size-based transfer functions: A new volume exploration technique. IEEE Transactions on Visualization and Computer Graphics, 14(6):1380–1387, 2008.
- [25] C. D. Correa and K. L. Ma. The occlusion spectrum for volume classification and visualization. IEEE Transactions on Visualization and Computer Graphics, 15(6):1465–1472, 2009.
- [26] C. D. Correa and K. L. Ma. Visibility histograms and visibility-driven transfer functions. IEEE Transactions on Visualization and Computer Graphics, 17(2):192–204, 2011.
- [27] Y. Wu and H. Qu. Interactive transfer function design based on editing direct volume rendered images. IEEE Transactions on Visualization and Computer Graphics, 13(5):1027–1040, 2007.
- [28] J. Zhou and M. Takatsuka. Automatic transfer function generation using contour tree controlled residue flow model and color harmonics. IEEE Transactions on Visualization and Computer Graphics, 15(6):1481–1488, 2009.
- [29] E. LaMar, B. Hamann, and K. I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In Proceedings of the Conference on Visualization '99: Celebrating Ten Years, VIS '99, pages 355–361, 1999.

- [30] C. Wang, S. Member, and H. W. Shen. Lod map, a visual interface for navigating multiresolution volume visualization. IEEE Transactions on Visualization and Computer Graphics, 12:1029–1036, 2006.
- [31] M. Levoy. Volume rendering using the fourier projection-slice theorem. In Proceedings of Graphics Interface, pages 61–69, 1992.
- [32] I. Viola, A. Kanitsar, and M. E. Gröller. Gpu-based frequency domain volume rendering. In Proceedings of the 20th spring conference on Computer graphics, pages 55–64, 2004.
- [33] A. Entezari, R. Scoggins, T. Moller, and R. Machiraju. Shading for fourier volume rendering. In Proceedings of the 2002 IEEE symposium on Volume visualization and graphics, pages 131–138, 2002.
- [34] Z. Nagy, G. Müller, and R. Klein. Classification for fourier volume rendering. In Proceedings of the Computer Graphics and Applications, volume 0, pages 51–58, 2004.
- [35] C. C. Cheng and Y. T. Ching. Transfer function design for fourier volume rendering and implementation using gpu. In Proceedings of SPIE Medical Imaging, 2008.
- [36] C. C. Cheng and Y. T. Ching. Real-time adjustment of transfer function for fourier volume rendering. Journal of Electronic Imaging, 20(4), 2011.
- [37] L. Piegl and W. Tiller. The NURBS book, 2nd edition. Springer-Verlag, 1997.
- [38] J. Frank. Three-dimensional electron microscopy of macromolecular assemblies: visualization of biological molecules in their native state. Oxford University Press, 2006.
- [39] T. R. Shaikh, H. Gao, W. T. Baxter, F. J. Asturias, N. Boisset, A. Leith, and J. Frank. Spider image processing for single-particle reconstruction of biological macromolecules from electron micrographs. Nature Protocols, 3(12):1941–1974, 2008.

- [40] C. C. Cheng, C. C. Chien, H. H. Chen, Y. Hwu, and Y. T. Ching. Image alignment for tomography reconstruction from synchrotron x-ray microscopic images. PLoS ONE, 9(1), 2014.
- [41] H. H. Chen, C. C. Chien, C. Petibois, C. L. Wang, Y. S. Chu, S. F. Lai, T. E. Hua, Y. Y. Chen, X. Cai, I. M. Kempson, Y. Hwu, and G. Margaritondo. Quantitative analysis of nanoparticle internalization in mammalian cells by high resolution x-ray microscopy. Journal of Nanobiotechnology, 9(14), Apr. 2011.
- [42] H. H. Chen, C. C. Chien, C. Petibois, C. L. Wang, Y. S. Chu, S. F. Lai, T. E. Hua, Y. Y. Chen, X. Cai, I. M. Kempson, Y. Hwu, and G. Margaritondo. Quantitative analysis of nanoparticle internalization in mammalian cells by high resolution x-ray microscopy. Journal of Nanobiotechnology, 9(1):14, 2011.
- [43] R. C. Gonzalez and R. E. Woods. Digital image processing, 3rd edition. Prentice Hall, 2007.
- [44] R. Szeliski. Computer vision: algorithms and applications. Springer, Nov. 2010.
- [45] C. Harris and M. Stephens. A combined corner and edge detector. In Proceedings of the 4th Alvey Vision Conference, Manchester, Aug. 1988.
- [46] T. Kadir and M. Brady. Saliency, scale and image description. International Journal of Computer Vision, 45(2):83–105, Nov. 2001.
- [47] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91–110, Nov. 2004.
- [48] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In 9th European Conference on Computer Vision, Graz, Austria, May 2006.
- [49] S. Rady, A. Wagner, and E. Badreddin. Entropy-based features for robust place recognition. In IEEE International Conference on Systems, Man and Cybernetics, 2008, Singapore, 2008.

- [50] S. Suri, P. Schwind, P. Reinartz, and J. Uhl. Combining mutual information and scale invariant feature transform for fast and robust multisensor sar image registration. In American Society of Photogrammetry and Remote Sensing . 75th Annual ASPRS Conference, Baltimore, MD, USA, 2009.
- [51] P. Viola. Alignment by maximization of mutual information. International Journal of Computer Vision, 24(2):137–154, 1997.
- [52] J. P. W. Pluim, J. B. Antoine M., and M. A. Viergever. Mutual information based registration of medical images: a survey. IEEE Transactions on Medical Imaging, 22(8):986–1004, Jul. 2003.
- [53] N. Dowson and R. Bowden. Mutual information for lucas-kanade tracking: an inverse compositional formulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(1):180–185, Jan. 2008.
- [54] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24, Issue 6:381–395, 1981.
- [55] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(1):51–65, Jan. 2005.
- [56] F. T. and H. Tao. Probabilistic object tracking with dynamic attributed relational feature graph. IEEE Transactions on Circuits and Systems for Video Technology, 18(8):1064–1074, Aug. 2008.
- [57] R. Szeliski and H. Y. Shum. Creating full view panoramic image mosaics and environment maps. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pages 251–258, New York, NY, USA, 1997.
- [58] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. International Journal of Computer Vision, 74(1):59–73, Aug. 2007.

- [59] J. Kopf, M. Uyttendaele, O. Deussen, and M. F. Cohen. Capturing and viewing gigapixel images. ACM Transactions on Graphics (TOG), 26(3):93–1–93–10, July 2007.
- [60] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. ACM Transaction on Graphics, 2:217–236, October 1983.
- [61] S. Lin, J. Gu, S. Yamazaki, and H. Y. Shum. Radiometric calibration from a single image. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages 938–945, 2004.
- [62] M.D. Grossberg and S.K. Nayar. Modeling the space of camera response functions. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26(10):1272–1282, 2004.
- [63] P. Goebel, N. Belbachir, and M. Truppe. Blind background subtraction in dental panoramic x-ray images: An application approach. 3663:434–441, 2005.
- [64] R. Szeliski. Image alignment and stitching: A tutorial. Foundations and Trends in Computer Graphics and Computer Vision, 2 Issue 1:1–104, Dec. 2006.
- [65] A. V. Oppenheim and R. W. Schaffer. Discrete-time signal processing. Prentice Hall Press, 3rd edition, 2009.
- [66] C. E. Shannon. A mathematical theory of communication. Bell System Technical Journal, 27(3):379–423, Jul. 1948.
- [67] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proceedings of Imaging Understanding Workshop 1981, pages 121–131, 1981.