Chapter 17
Computation in Medicine: Medical Image Analysis and Visualization

Adekunle Micheal Adeshina

Abstract Computation in medicine has recently revolutionized those ideal procedures for translating fundamentally proven mathematical concepts in medical imaging and analysis into relevant routines of algorithms. Modern computational techniques, such as CUDA, a parallel computing platform, enabling direct access to the GPU instruction and parallel processing capability, are currently providing flexibility in the use of high-performance computational approaches. Similarly are the other software optimization procedures that assure low-cost and high-throughput visualization of medical datasets. Without mincing words, significant impact of such hardware and software optimization algorithms in medical image analysis and visualization cannot be overemphasized. In the same vein, acquisition of appropriate clinical datasets plays a great role in the accurate diagnosis of diseases and therapy management. The use of appropriate datasets and suitable image modalities are both important in order to successfully prove the effectiveness of any applied computational approaches in medical image analysis and visualization. Moreover, data reconstruction and representation from 2-D to 3-D usually follow notable mathematical approaches such as Euclidean plane, projective plane, and Cartesian coordinate systems and involve other interactive properties such as rotation, scaling, and translation which are also relying on various renderable concepts of data representation. This chapter documents some of the image procedures for acquiring morphological and functional information of patients with more emphasis on mathematical computations of commonly used techniques, such as X-ray, computed tomography (CT), and magnetic resonance imaging (MRI). Interestingly, a typical framework for medical imaging and visualization has been conceptualized in the course of this documentation. Relevant approaches to medical data representation, restructuring, and modeling procedures such as volume segmentation, classification, shading, gradient computation, interpolation, and resampling are presented along with all the significant processes required before generating informative composition of images. In order to facilitate better

A.M. Adeshina (✉)
Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
e-mail: codedengineer@yahoo.com

D.-Q. Wei et al. (eds.), *Translational Bioinformatics and Its Application*, Translational Medicine Research, DOI 10.1007/978-94-024-1045-7_17

understanding of some of the concepts introduced in this chapter, real-world examples of CT and MRI datasets in 2-D and in their 3-D correspondence are showcased to depict the significance of the mapped structures in the 2-D.

Keywords CUDA • GPU • Image analysis • Medical imaging • Visualization

17.1 Introduction

Recent evolvement of translational informatics has been a strong driving force for translation of laboratories' data. The term "translation" is seen to involve correlating genotype with phenotype, which often requires dealing with information at all structural levels, ranging from molecules and cells to tissues and organs and from individuals to populations (Chen et al. 2013). A relatively new concept was coined with translational bioinformatics, introducing profound changes which include the identification of conviction biology as an informational science; the application of high-throughput genomic and proteomic platforms for global analyses; the requirement to bring computer science, mathematics, and statistics into biology; the use of model organisms as *Rosetta Stones* for deciphering biological complexity; and also the power of comparative genomics for coming to understanding the logic of life (Hood 2003). Apparently, such conceptual analysis opens up a new dawn in medicine. Translational bioinformatics involves the development and the use of computational methods that can reason over the enormous amounts of life science data being collected and stored for the purpose of creating new tools for medicine (Butte 2008). This field has been identified as a revolutional domain addressing some of the hindering computational challenges in medicine. Translational bioinformatics is seen as an emerging field addressing the computational challenges in biomedical research and the analysis of the vast amount of clinical data generated from it (Butte 2008). Technically, the term "computational" involves certain specific procedures for translating those ideal and fundamentally proven mathematical concepts into routines of algorithms. However, all the accurate diagnosis, surgical treatment, and assessment of response to treatment depend on the ability to see through the affected tissues or organs (Aldrich et al. 2012), and this brings medical image analysis and visualization forward into play in translational bioinformatics, thereby forming both combinatory and an integral part of the revolutionary processes of translational bioinformatics. Those actions requiring the use of scientific mathematics and execution of algorithms in order to attain significant and more precise results in medical analysis are deeply rooted in the word *computation in medicine*.

Computational approaches in medical image analysis have also gain more attention due to the recent overwhelming rate of generation of biomolecular data. This accumulated information explosion is being driven by the development of low-cost, high-throughput experimental technologies in genomics, proteomics, and molecular imaging, among others, tying the anticipated success in the life sciences

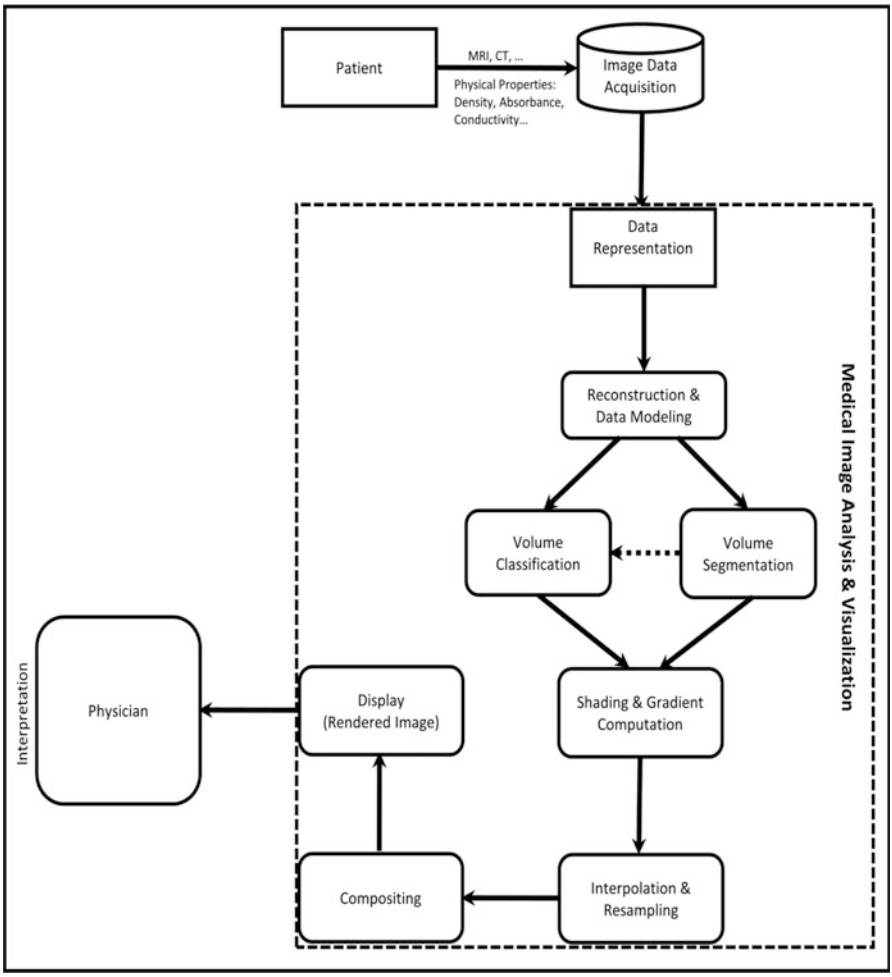

Fig. 17.1 Medical imaging and visualization architecture

to our ability to rationally interpret these large-scale, high-dimensional datasets into clinically understandable and useful information, which in turn requires us to adopt advances in informatics (Chen et al. 2013). Overview of medical imaging and visualization architecture is presented in Fig. 17.1. However, with current alignment of computational medicine with high-performance computation, computer models and efficient software could be leveraged in figuring out, within a considerable interactive speed, how diseases develop and how to thwart it. Invariably, paramedical research and computational approaches seem inseparable. This chapter introduces computation in medicine. Medical image acquisition techniques, their numerical computations, structuring, and data visualization procedures are presented.

17.2 Acquisition of Medical Image Data

In medical diagnosis and disease therapy management, acquisition of medical image data is a crucial process immediately after the diagnosis of the concerned patients. However, in certain circumstances, acquisition of patient images may be considered a priority, overriding the usual medical doctors' preexamination and interpretation of the health situation of patients. Such cases could be in the case of emergency situation either as a result of severe injury especially when handling unconscious patients. Nevertheless, in clinical practices, medical image data could be acquired for diagnosis, therapy planning, intraoperative navigation, or postoperative monitoring (Preim and Bartz 2007). According to Dhawan et al. (2008), medical imaging could be seen as a process of collecting information about a specific physiological structure (an organ or tissue) using a predefined characteristic property that is displayed in the form of an image. Such predefined characteristic property may be physical properties such density, absorbance, or conductivity. Image acquisition technique required in any case largely depends on the intended information from patient medical examination. Image modalities such X-ray, computed tomography (CT), and magnetic resonance imaging (MRI) are useful in extracting "morphological information" from the patient. Other specialized MRI techniques include MR spectroscopy, MR angiography, and MR microscopy. However, in order to obtain physiological or functional information from patients, positron emission tomography (PET) and single-photon emission computerized tomography (SPECT) are appropriate. Diffusion tensor imaging (DTI) also plays significant roles in diagnosis procedures that require measuring of the diffusion of water and in tracking of the brain's nerve fibers, the white matter. Apparently, suitability of image modalities solemnly depends on the required medical examination, and thus, image modalities are seen to be complementary to each other in the medical diagnosis and disease and therapy management procedures (Adeshina et al. 2012). This section briefly discusses the X-ray, computed tomography, and magnetic resonance imaging being the most commonly used image modalities.

17.2.1 X-Ray

In 1895, Wilhelm Conrad Röntgen (or "Roentgen" in anglicized typography) discovered X-ray (Roentgen 1898) as a high-voltage discharge between electrodes in a gas at very low pressure producing a penetrating radiation which causes certain materials to fluoresce visible light. X-ray is a medical imaging technique that utilizes the radiation that is partly transmitted and partly absorbed through irradiated objects. The X-ray photons are a form of electromagnetic radiation produced following the ejection of an inner orbital electron and subsequent transition of atomic orbital electrons from states of high to low energy (Jenkins 2000). X-ray is widely used in projection of images based on absorption and scattering with very

high spatial resolution, and it has been seen to be greatly useful in imaging fractured bones, such as a broken arm or wrist, often used by surgeons during therapeutic procedures, such as a coronary angioplasty, to help guide equipment to the area being treated and in highlighting a lung infection, such as pneumonia. Meanwhile, X-rays can only produce 2-D images; it exposes patients to radiation and not suitable for imaging soft tissues. According to Bingham (1998), considering the following assumptions, the inside structures of an object could be investigated:

1. There exists an object with n-dimensional space where n is fixed as 2.
2. $f(x)$ is the X-ray attenuation coefficient at point $X \in \mathbb{R}$ where the attenuation coefficinet depends on the material through which the ray passes. Therefore, f is expected to give information about object.
3. Suppose the object is contained in a ball of radius $\mathcal{R}$ with the center at the origin and that the X-ray attenuation coefficient f is zero outside the object.
4. If the object is x-rayed in a direction $\theta \epsilon S^{n-1}$ from a point $a \epsilon A := S^{n-1}(0, R)$, the attenuation of the X-ray intensity I at each point $a + t\theta$, $t \geq 0$.

According to assumption (iii), we can have Eq. (17.1):

$$\text{supp}\, f \subset B(0, R) \tag{17.1}$$

and then Eq. (17.2):

$$\Leftrightarrow dI = f(a + t\theta)\, I dt \tag{17.2}$$

By solving this differential equation, we see that the intensity of the X-ray measured by a detector situated behind the object is as Eq. (17.3):

$$I_{\text{meas}} = I_0 \exp\left(\Leftrightarrow \int_0^{\infty} f\,(a + t\theta)\, dt\right) \tag{17.3}$$

Therefore, we can derive formulae for reconstructing $f(x)$ from the measurements I_{meas} or equivalently from Eq. (17.4):

$$\int_0^{\infty} f\,(a + t\theta) dt = \ln\left(\frac{I_O}{I_{\text{meas}}}\right) \tag{17.4}$$

The above equation could be seen as cases of having different combinations of $a \epsilon A$ *and* $\theta \in S^{n-1}$.

17.2.2 Computed Tomography

X-ray imaging techniques follow the same scenario of allowing radiation to pass through different parts of the patients' body. Such passage of X-rays is dependent

on the amount of X-rays that could be absorbed or exit the body of the patients, which in turn determines the radiation dose of the patient. Computed tomography (CT) is not exceptional to this; however, in CT multiple X-ray images are taken from different directions producing cross-sectional images or "slices" of patients' anatomy. The cross-sectional images could be used in medical diagnosis and disease therapy. CT entails the reconstruction of a function f from a finite number of line integrals f (Faridani and Ritman 2000). With such understanding, it becomes apparent that the goal of CT is to recover an approximation to $f(x)$ from CT datasets over a finite number of lines.

X-rays from a located source travel and pass through the patient. However, some energy of rays are attenuated, and rays with less energy eventually reach the detector. Rays of CT are able to produce a map of gray values representing a close resemblance of the insides of the patient. This situation can be understood either through a monochromatic beam or polychromatic beam considering the intensity I_{in} at distance x; I_{out}, the intensity at the detector's end; and μ, the attenuation coefficient or absorption coefficient. If we consider a situation whereby the radiation passes through a body with the same property at every point, a homogeneous body, we expect the intensity of radiation passing through the body to decrease exponentially with distance; hence, we have Eq. (17.5):

$$I_{(x)} = I_{in}\exp(-\mu x) \tag{17.5}$$

Therefore, if we differentiate Eq. (17.5),

$$\frac{dI}{dx} = -\mu I \tag{17.6}$$

However, for a nonhomogeneous body where the absorption coefficient varies with distance x,

$$I_{(x)} = I_{in}\exp\left(-\int u dx\right) \tag{17.7}$$

Similarly, we can consider a specific interval a, b where a *and* b have values between 0 *and* n in order to get a more specific approximation, thus Eq. (17.8):

$$I_n = I(n) = I_{in}\exp\left(-\int_a^b \mu dx\right) \tag{17.8}$$

If we know I_n, the total absorption, A_t, could be calculated:

$$A_t = \int_a^b \mu dx = -\log\left(\frac{I_n}{I_{in}}\right) = \log\frac{I_{in}}{I_n} \tag{17.9}$$

However, even if we know I_{in} and I_n, we still cannot clearly say the distribution of the material within the interval a, b as being illustrated. Resolution of this was

first attempted by Radon (1917). Similarly, the absorption coefficient could be analyzed following a related approach based on a number of assumptions established for computed tomography (Faridani 2003):

1. $f(x)$ = density of the cross-section at $x \in \mathbb{R}^2$
2. L = the line of X-rays
3. $I(x)$ = the intensity of X-rays at $x \in L$

Apparently, in theoretical physics, $I(x)$ decreases proportional to $f(x)$; thereby, we can have Eqs. (17.10) and (17.11):

$$dI/dx = -f(x)I(x) \tag{17.10}$$

$$dI/I = -f(x)dx \tag{17.11}$$

Therefore we can have the measured data calculated as,

$$\text{meas} = \frac{I_{in}}{I_{out}} = e^{RF(L)} \tag{17.12}$$

where the total attenuation along L,

$$RF(L) = \int_{x \in L} f(x)ds \tag{17.13}$$

Radon (1917) referred to the expression in Eq. (17.8) as the total "material" along L. To construct the absorption coefficient μ (x, y) as a function of position using Radon approach, we assume:

1. Projection is a line integral.
2. Projection p $(s, \varnothing)$ at angle $\varnothing$, s is coordinate on detector.

The Radon transform (RT) of a distribution f(x; y) is given by Eq. (17.14):

$$p(s, \varnothing) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y).\delta(x\cos\varnothing + y\sin\varnothing - s)dxdy \tag{17.14}$$

where δ is the Dirac delta function and x, y, $\varnothing$, and s are respective coordinates. The Radon transform of an off − center point source is a sinusoid; hence, the function $p(s, \varnothing)$ is usually being referred to as a sinogram.

17.2.3 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is an application of nuclear magnetic resonance (NMR) which is a subtle quantum mechanical phenomenon that has played a major role in medical imaging revolution over the last 30 years. Hydrogen in water molecules possesses an inherent ability referred to as *spin* which gives it potential to act as magnet. Nuclear magnetic resonance is a phenomenon which occurs when the nuclei of certain atoms are immersed in a static magnetic field and exposed to second oscillating magnetic field (Hornak 1997). The *spin* property in proton makes the nucleus that produces NMR signal. Mathematical description of NMR could be better presented using 2-D Fourier transform, a standard Fourier transformation of two variables $f(x, y)$, wave forms $e^{2\pi i(k_x \ x + k_y y)}$ and k − space $\left(k_x k_y\right)$.For $f \in L^2$ $\left(R^2\right)$, the Fourier transformation of f is presented in Eq. (17.15):

$$F(f)\left(k_x, k_y\right) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi\left(k_x x + k_y y\right)} \, dxdy \tag{17.15}$$

In the same vein, we can re-represent Eq. (17.15) to portray a reverse approach to Fourier transformation as in Eq. (17.16):

$$F^{-1}(f)(x,) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi\left(k_x x + k_y y\right)} \, dk_x dk_y \tag{17.16}$$

MRI is the most suitable and widely used imaging technique for brain and other soft tissues. It is capable of producing detail image of patients in any plane. MRI is highly flexible to use and it provides better spatial resolution with higher discrimination, making it very relevance in contrasting soft tissue. Moreover, unlike X-ray and CT, MRI has no ionizing radiation.

17.3 Medical Data Visualization

17.3.1 Reconstruction and Data Representation

Projective plane is seen as a geometric structure with extended concept of a plane. However, in ordinary Euclidean plane, unless the line crosses each other and intersects, parallel lines do not intersect. Meanwhile a projective plane with any two lines intersect in one and only one point called vanishing point, a point where parallel lines that are not parallel to the image plane appear to converge, which could be better interpreted with projective plane.

In the development of medical imaging and visualization framework, using homogeneous coordinate principle, a point x, y of $2-\mathrm{D}$ slice in the Euclidean plane is represented in the projective plane $(3-\mathrm{D})$ by adding a third

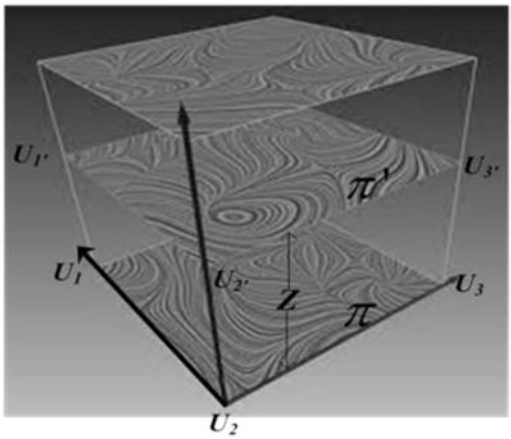

Fig. 17.2 Point estimation of 2-D slices

coordinate 1 at the end, x, y, 1 . This is based on the fundamental Euclidean theorem which states that a point in an $n-$dimensional Euclidean space is represented as appoint in an $(n+1)$-dimensional projective space. However, overall scaling is not important.

The MRI slices are abstractly represented as stack of images as in Fig. 17.2. It is assumed that there are points U_i with $U_i = 1, 2, 3, 4$ arranged parallel in line with plane π . Since there exist such many slices, we assume the slices are moved up a distance Z as shown in Fig. 17.2. With such moved distance of the slices, there will be a formation of new sets of points $U_{i'}$ with $i' = 1, 2, 3, 4$ leaning on a new plane π'. The first issue to address is estimation of the new points $U_{i'}$ automatically which can be done by estimating directly from the first plane π.

At this point, it can be assumed that U_1 and U_3 are known ; hence, U_2 and U_4 can be estimated and computed by applying intrinsic properties of the vanishing points. Figure 17.2 shows the point estimation of 2-D slices. The vanishing point of the parallel lines leaning on plane π could be computed as in Eq. (17.17):

$$V = (U_1 \times U_2) \times (U_3 \times U_4) \tag{17.17}$$

However, based on projective geometry, which describes the physical characteristics of the virtual camera and the relationships between the images, the projection of a point X_w in the object space to a point U_i in the image space using projective camera is expressed in terms of a direct linear mapping in homogeneous coordinates as in Eq. (17.18):

$$\lambda U_i = PX_w = [P_1 \quad P_2 \quad P_3 \quad P_4] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{17.18}$$

where λ is the scale factor due to projective equivalency of $(k_x\,;k_y\,;k) = (x; y; 1)$, P is a 3*4 camera projection matrix, and P_i is the *ith* column of P.

As earlier discussed, with homogeneous coordinate representation, value 1 in the last row of the vector denotes that the defined point leans on the image plane. However, if the point in the object space leans on ground plane $Z = 0$, hence the linear mapping will change to Eq. (17.19):

$$sU = HX'_w = [P_1 \quad P_2 \quad P_4] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{17.19}$$

H is the homography matrix mapping points lying on a plane in the object space across different images ; s introduced scaling factor in the mapping equation stems from setting Z *to* 0.

In order to establish relationship between U_i and U'_i, we can restate Eq. (17.18) above as (17.20):

$$\lambda U_i \; = [P_1 \quad P_2 \quad P_4] \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} + P_3 Z \tag{17.20}$$

where P_3 corresponds to the vanishing point in the direction of Z axis or the normal of the ground plane.

The main target is to project the lines and points that made up the 2-D slices in 3-D. The Euclidean formula for s line is $ax + ay + c = 0$; this is regarded as nonzero scaling factor, and since the equation is unaffected by scaling, we can however arrive at the following:

$$qX + rY + sZ = 0$$

where q, r, and s are the homogeneous coordinates of points (x, y) in the line:

$$t^T P = P^T t = 0$$

$$t = [q, \quad r, \quad s]^T \text{ representing the line}$$

$$p = [X \quad Y \quad Z]^T \text{representing the point}$$

Substituting V_z for P_3 in Eq. (17.20) and combining the result with Eq. (17.19), we have

$$\lambda_i U_i = s_i U_i \; + V_z Z \tag{17.21}$$

λ_i and s_i are the unknowns from this equation, though they were both defined earlier in Eq. (17.18) and Eq. (17.19). We can estimate the respective values by using Eq. (17.22):

$$\begin{bmatrix} \lambda_i \\ s_i \end{bmatrix} = \left(A_j{}^T A_i\right)^{-1} A^T{}_i b_i \tag{17.22}$$

$$A_i = [U_i \mid -U_i] \tag{17.23}$$

$$b_i = V_z Z \text{ and } V_z = P_3$$

Since s_i is estimated, we can continue setting different values for Z in order to estimate any other image point along the lines. Hence, we can equally estimate U_2 and U_4 as follows:

$$U_2 = (U_2 \times V_z) \times (U_1 \times V) \tag{17.24}$$

$$U_4 = (U_4 \times V_z) \times (U_3 \times V) \tag{17.25}$$

17.3.1.1 Renderable Representation

Memory system architecture of medical image analysis and visualization framework is typically concerned with making the data available for its optimal architectural use. It is at this stage that series of the original data slices are stacked, shaped, and positioned for flow. Properties such as *rotation*, *scaling*, and *translation* are likewise necessary in the data for better *value distribution*. Coordinate system is greatly useful for the success of data preparation; hence, medical image analysis and visualization framework is usually developed to use the *model*, *world*, *view*, and *display coordinate systems*.

The *model coordinate system* is typically a local *Cartesian coordinate system*. As the name *model* implies, it is the coordinate system in which model is defined. This type of coordinate system is locally defined by the modeler. We can refer to this as an *inherent coordinate system* based on the decision of the person that generates it. The units used in its definition may be *meters*, *inches*, or *feet* and its axis might be arbitrary; these are based on discretion of the modeler.

The *world coordinate system* is the *3-D space* where *actors* are positioned. Unlike model coordinate system, which is a typical local Cartesian coordinate system, world coordinate system is the only standard coordinate system where all actors *locally defined coordinate systems* are converted to. The world coordinate system is the coordinate system where all the actors are *scaled*, *rotated*, and *translated into*. Moreover, the position and orientation of cameras and light are specified in the world coordinate system.

The *view coordinate system* is directly referenced to the *camera*; it represents what is *visible* to the *camera*. It consists of *x*, *y*, *z values*. The *x* and *y* specify location of the *image plane* and it ranges from − 1, 1, while *z* is the *depth coordinate* that represents the distance or ranges from the camera. In order to convert from the *world coordinates* to *view coordinates*, a four by four (4*4) *coordinate transformation* matrix is applied, introducing the perspective effects of a camera.

The usual way to represent element in three dimensions is through *Cartesian vector* x, y, z. However, in order to project $2-\mathrm{D}$ image to $3-\mathrm{D}$ plane, vanishing point must be included in the projection; hence, homogeneous coordinate system is needed. Unlike *cartesian vecto*r with three (3)elements x, y, z, homogeneous coordinate has four *element vectors* represented as X, Y, Z, W as earlier explained in the previous section. The conversion from Cartesian coordinates to homogeneous coordinates is presented in Eqs. (17.26), (17.27), and (17.28):

$$x = \frac{X}{W} \tag{17.26}$$

$$y = \frac{Y}{W} \tag{17.27}$$

$$z = \frac{Z}{W} \tag{17.28}$$

Four by four (4^*4)matrix is used for the performance of translation, scaling, and rotation through repeated multiplication of matrix. We can create a transformation of matrix that translates a point x, y, z in Cartesian space by vector t_x, t_y, t_z as in Eq. (17.29). Figure 17.3 illustrates translation:

$$T_T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17.29}$$

where T_T is the matrix for translation.

The created translated matrix needs to be post-multiplied with homogeneous coordinate X, Y, Z, W. Meanwhile, we have to construct the homogeneous coordinate from the Cartesian coordinate before such multiplication; hence, if we set $W = 1$ representing *finite point*, X, Y, Z will yield X, Y, Z, 1. In the same vein, we pre$-$multiply the current position by the transformation matrix T_T in order to determine the translated point $X^{'}$, $Y^{'}$, $Z^{'}$ for yielding the translated coordinate. Hence, we have Eq. (17.30):

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & t_z \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{17.30}$$

Using the general pattern of conversion back to Cartesian coordinates as in Eqs. (17.26), (17.27), and (17.28), we have Eqs. (17.31), (17.32), and (17.33):

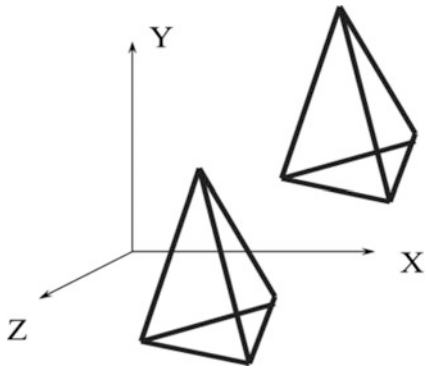

Fig. 17.3 Translation

$$x' = x + t_x \tag{17.31}$$

$$y' = y + t_y \tag{17.32}$$

$$z' = z + t_z \tag{17.33}$$

Equations (17.32) and (17.33) are the procedure to translate an object. Similar procedure can be employed for scaling or rotating of an object. Using the transformation matrix as Eq. (17.34) where T_s is the transformation matrix for scaling, s_x, s_y, s_z represent the scale factors along x, y, z axes, respectively. Figure 17.4 illustrates scaling about the origin:

$$T_s = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17.34}$$

In the same vein, we can do rotation around x, y, *and* z *axes* by angle θ as illustrated in Figs. 17.5, 17.6, and 17.7 to produce T_{R_x}, T_{R_y}, and T_{R_z}, respectively.

$$R_x(\theta): y' = y\cos\theta - z\sin\theta$$
$$z' = y\sin\theta + z\cos\theta$$
$$x' = x$$

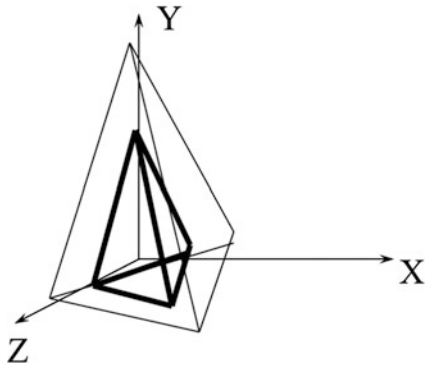

Fig. 17.4 Scaling about the origin

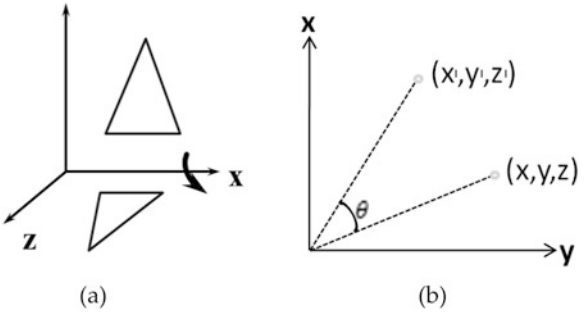

Fig. 17.5 Rotation about x axis

$$T_{R_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17.35}$$

Illustration of rotation about y axis is given in Fig. 17.6.

$$R_y(\theta): z' = z\cos\theta - x\sin\theta$$
$$x' = z\sin\theta + x\cos\theta$$
$$\mathrm{y}' = \mathrm{y}$$
$$T_{R_y} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17.36}$$

Similarly, Fig. 17.7 illustrates rotation about z axis producing T_{R_z}.

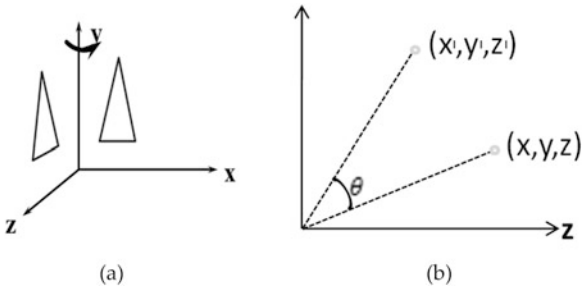

Fig. 17.6 Rotation about y axis

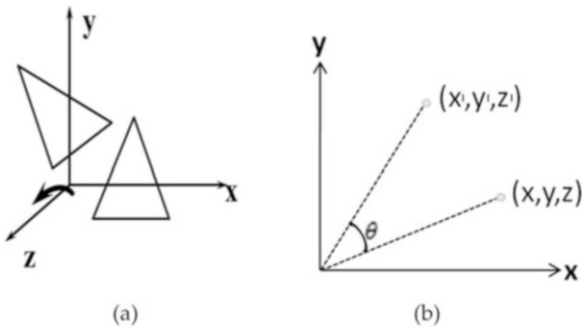

Fig. 17.7 Rotation about z axis

$$R_z(\theta): x = x' \cos\theta - y \sin\theta$$
$$y' = x \sin\theta + y \cos\theta$$
$$\mathrm{z}' = \mathrm{z}$$

$$T_{R_z} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17.37}$$

However, during the rotation of the object, we might need to transform the object from one coordinate axes to another, from $x - y - z$ to $x^{'} - y^{'} - z^{'}$. In order to do this, we need to first derive a transformation matrix by assuming the following:

1. The unit $x^{'}$axis makes the angle $\theta x^{'}x$, $\theta x^{'}\mathrm{y}$, $\theta x^{'}z$ around $x - y - z$ axis.
2. The unit $y^{'}$axis makes the angle $\theta y^{'}x$, $\theta y^{'}y$, $\theta y^{'}\mathrm{z}$ around $x - y - z$ axis.
3. The unit $z^{'}$axis makes the angle $\theta z^{'}x$, $\theta z^{'}y$, $\theta z^{'}z$ around $x - y - z$ axis.

where ($\theta x^{'}x$, $\theta x^{'}\mathrm{y}$, $\theta x^{'}z$), ($\theta y^{'}x$, $\theta y^{'}y$, $\theta y^{'}z$), *and* ($\theta z^{'}x$, $\theta z^{'}y$, $\theta z^{'}z$) are the directional cosines.

Hence, placing the directional cosines along the rows of the transformation matrix will produce Eq. (17.38) which is referred to as the resulting rotation matrix T_R:

$$T_R = \begin{bmatrix} \cos\theta x'x & \cos\theta x'y & \cos\theta x'z & 0 \\ \cos\theta y'x & \cos\theta y'y & \cos\theta y'z & 0 \\ \cos\theta z'x & \cos\theta z'y & \cos\theta z'z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17.38}$$

To *rotate* around the *center of the object*, which is usually more convenient, we must first *translate* from the *center of the object* to the *origin*, and then we apply *rotations* followed by *translating the object back to its center*. However, in order to achieve the *translation*, *rotation*, and *scaling* of the object using the transformation matrix, the order of the multiplication is important.

In *display coordinate system*, the coordinates are actual x, y pixel locations on the image plane, though display coordinate uses the same basis as view coordinates except it does not use -1, 1 range. The *view coordinates* determine *window size* and *view point*. *Display coordinates* determine how the negative one – to – one $(-1, 1)$ of *view coordinates* is *mapped* into *pixel locations of display*. With view port, it is possible to divide the port which ranges from 0, 1 for x and y axes and depth value representation with z axis. This is particularly useful in cases where one needs to render two different scenes but display them in the same window. The analysis and justifications for the preparation of datasets in its more suitable renderable form are also in line with the explanations of Schroeder et al. (2002).

17.3.2 Data Restructuring and Modeling

The prepared dataset has to be filtered thoroughly in order to enhance its pixel intensities. Similarly, the specified focused data should be geometrically mapped for better image quality. Modeling of the camera focus point in medical image analysis and visualization framework is likewise significant, often aligned with the physical laws of optics which could be better described by modeling the transport theory of light with specific attention on geometrical optics lights. Meanwhile, factors such as the wave character of light, possible light polarization states, diffraction, and interference are usually neglected.

If x is the radiant field at any point in the direction of the radiant energy n and around υ, the radiant energy could be defined as $R(x\,.\,n\,.\,\upsilon)$. Therefore, if θ is the angle between the direction n and the normal on da for time dt, the traveling radiant energy δE can be respresented in Eq. (39) provided that there is a specified frequency interval dυ around υ through a solid angle dΩ:

$$\delta E = R(x.n.\upsilon)\cos\theta da d\Omega d\upsilon dt \tag{17.39}$$

However, we can also define radiant energy using photon number density $\psi(x\,.\,n\,.\,\upsilon)$. If x denotes the position of the photons per unit volume, $d\upsilon$ represents the frequency interval around υ along the direction n and travels into an element of solid angle $d\Omega$. Then, the number of photon $\check{\mathrm{N}}$ per unit volume could be represented in Eq. (17.40).

$$\check{\mathrm{N}} = \psi(x.n.\upsilon) d\Omega d\upsilon \tag{17.40}$$

Equation (17.40) could be extended for calculation of the number of photons $\check{\mathrm{N}}$ by representing surface da with time dt and traveling velocity c in Eq. (17.41):

$$\check{\mathrm{N}} = \psi(\cos\theta da)(cdt)(d\Omega d\upsilon) \tag{17.41}$$

Nevertheless, if the energy carried by each photon is considered as $h\upsilon$ in accordance to the constant expressed in Planck – Einstein relation where h is the Planck's constant, hence, a new relationship could be established for radiant energy using photon number density:

$$\delta E = ch\upsilon\psi(x.n.\upsilon)\cos\theta da d\Omega d\upsilon dt \tag{17.42}$$

Apparently, since we have clearly defined radiant energy in Eqs. (17.39) and (17.40), we can therefore equate these equations:

$$R(x.n.\upsilon)\cos\theta da d\Omega d\upsilon dt = ch\upsilon\psi(x.n.\upsilon)(\cos\theta da d\Omega d\upsilon dt)$$

$$R(x.n.\upsilon) = ch\upsilon\psi(x.n.\upsilon) \tag{17.43}$$

Equation (17.43) shows clear similarity between radiance and photon number density as in Eq. (17.43). Therefore, in order to record all the focused points in an image, it becomes reasonable if we compute $R(x\,.\,n\,.\,\upsilon)$ for all the focused points. Some of these concepts are documented in Adeshina et al. (2012) and more elaborately in Hege et al. (1996).

17.3.3 Volume Segmentation and Classification

Volume or image segmentation entails partitioning of image or volume into meaningful region representation. Segmentation improves the analysis of an image establishing a reasonable correspondence between the image pixel properties and the type of tissue to facilitate the successful manipulation of data for medical visualization, while classification focuses on labeling the pixels of an image

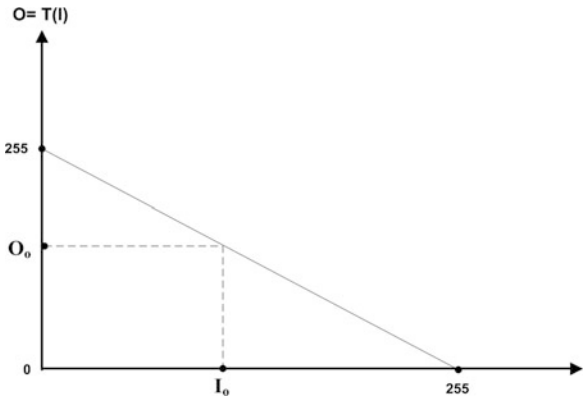

Fig. 17.8 Image point processing approach

corresponding to a specific type of tissue or anatomical structure usually with color and opacity. Apparently at the end of a successful volume segmentation and classification, specific objects within the image would be separated, and regions that have similar pixel properties would be identified along a specified predetermined boundaries. All these create rooms for a more detail image or volume analysis. In volume rendering, depicting region of interest based on color and transparency mappings of respective scalar values to the corresponding regions of volume is achieved using transfer function. Image point processing scale is presented in Fig. 17.8.

Transfer function could be established using image point ranging from 0 to 255 scale. *Point processing* image enhancement techniques is based on the intensity of individual pixels in the image. Hence, based on Eq. (17.44), intensity transfer function could be represented through *255 output pixels* and *255 input pixels* as in Fig. 17.8:

$$O = T(I) \tag{17.44}$$

where O represents the *output pixel*, T is the *transform*, and I is the *input pixel*.

Feature enhancement is extremely important in order to distinguish normal tissues distinctly from abnormal tissues especially when intensities of abnormal tissues match with the intensities of normal ones. Despite the fact that brain tumor might sometimes be large, space occupying, it could still exist in the same intensity as the normal tissues making it difficult to distinguish.

Transfer function was utilized in mapping data value to "renderable quantities" as the output value. The two (2) main transfer functions usually designed are the *opacity transfer function* and the *color transfer function*. The *opacity transfer function* maps intensities of volume elements (voxels) in the data sample to the corresponding opacity value based on the framework intensity scale and selectively makes some voxels transparent enough to be seen through the assigned opacity value in order to show the interior of the data sample. Meanwhile, *color transfer function* uses coloration for its classification procedures. It maps intensities of

voxels to corresponding color values using lookup table and likewise does selective painting of voxels with different colors such that voxels of different intensity values are presented with appropriate corresponding color variances. However, in order to have better clarities of the output images, contrast enhancement transfer function, referred to as the *contrast transfer function (CTF), could be* applied in 3-D reconstruction procedures.

17.3.4 Shading and Gradient Computation

The *ambient coefficient*, the *diffuse coefficient*, and the *specular coefficient* are the three parameters that are usually modeled for illumination. Ambient lighting, the background illumination, is represented in equation (17.45):

$$R_c = L_c O_c \tag{17.45}$$

where R_c is the resulting intensity curve, L_c is the light intensity curve, and O_c is the color curve of object.

Ambient light has no direction and is independent of light position, orientation of the object, and observer's position. With this in mind, ambient is simply seen as the approximate contributions of light to the scene which is irrespective of the location of object and light. Figure 17.9 illustrates that.

Diffuse lighting is the non-shiny illumination and shadows. It has no dependence on camera angle. Diffuse lighting is illustrated in Fig. 17.10 and represented as Eq. (17.46). In order to determine diffuse's contribution to the surface, surface normal and the direction of the incoming rays are important:

$$R_c = L_c O_c \cos\theta \tag{17.46}$$

where L_c is the light color, O_c is the object color, and $\cos\theta$ is the product of the vector of light source (a negative value) and the vector of surface normal value to the object.

Specular lighting is the bright and shiny reflections which has no dependence on object color. Specular lighting is represented as Eq. (17.47) and illustrated in Fig. 17.11.

$$R_c = L_s K_s \cos(\alpha)^n \tag{17.47}$$

L_c represents the light color, K_s is the reflection constant, and R_c *is* the *color curve* . The product of the vector of light source, which is a negative value, and the vector of surface normal value to the object is $\cos\alpha$. However, specular power is denoted as n resulting from different n values of specular light. Equation (17.48) presents the integration of the three parameters that are usually modeled for illumination:

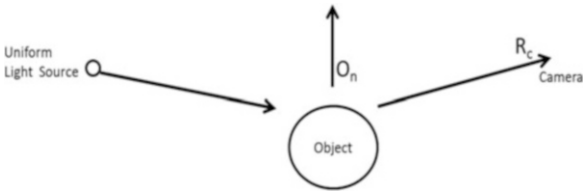

Fig. 17.9 Ambient lighting

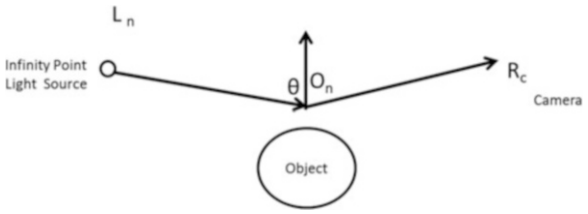

Fig. 17.10 Diffuse lighting

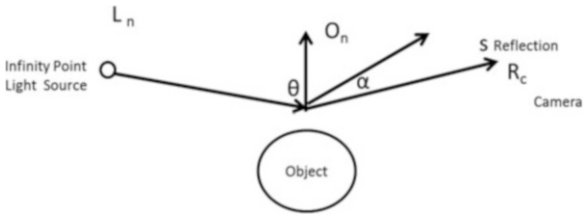

Fig. 17.11 Specular lighting

$$R_c = W_a + W_d + W_s \tag{17.48}$$

where W_a, W_d, and W_s are the relative weights of *ambient*, *diffuse*, and *specular*, respectively.

Moreover, in order to achieve quality image output, medical image analysis and visualization framework could be configured to select either *flat*, *Gouraud*, *Phong shading*, or their combination for better shading of images with respect to the level of pixels in the datasets. Flat shading is the earliest shading method which requires shading the polygons in the data samples with single color. However, because sometimes resulting interpolation color could be needed during shading to have a better image coloration, Gouraud shading was introduced. With Gouraud shading, polygons are shaded by interpolating color that are computed at the vertices of the image. Unfortunately, Gouraud shading usually produces *specular highlights*, a bright spot of light that appears on shining objects when illuminated. Phong shading produces better shading results compared to Gouraud shading by fixing the issue of specular highlights. However, despite the shortcomings in flat and Gouraud shading, using all in combination will contribute to obtaining better-shaded image.

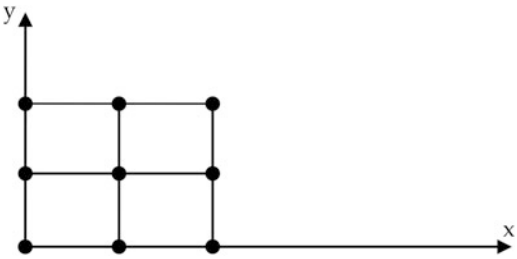

Fig. 17.12 A 3 by 3 image sample

17.3.5 Interpolation and Resampling

Interpolation is very important in medical visualization. Interpolation and resampling usually become necessary particularly whenever we perform scaling operations on digital images, for instance, a 3 by 3 image with its pixels represented in *xy* coordinates as illustrated in Fig. 17.12.

$$\begin{bmatrix} \widehat{x} \\ \widehat{y} \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{17.49}$$

In the scaling operation presented in Eq. (17.49), $(S_x, \ 0)$ and $\left(0, S_y\right)$ respresent the transformation matrix, - *xy* is the coordinate of the pixel in the original image, and the $\widehat{x}$ and $\widehat{y}$ are - the coordinates of the pixels in the transform.The original 3 - *by* 3 image has 3 pixels in the horizontal direction- and 3 pixels in the vertical direction.- However, after performing scaling operation by - a factor of 3 in both axes, the final image size has 9 pixels in the horizontal and- 9 pixels in the vertical direction, leaving many pixels not- filled up. If we have a 1 dimensional signal- $f(t)$ and sampled signal as $f_s(t)$, we could therefore see the sample values $f_s(t)$ represented accordingly at discrete locations in the Fig. 17.13. However, since we do not have information of the positions at the intermediate locations as in Fig. 17.13, we need to do interpolation for all the values of *t* and subsequently do resampling in order to fill up those positions.

It is important to ensure that the interpolation operations follow the following properties:

1. The interpolation function should have a finite region of support, i.e., the interpolation operations should be carried out based on local information of the sample values and not on global information.
2. The interpolation function should be very smooth without introducing any discontinuity in the signal.
3. The interpolation function should be shift invariant; when the signal is shifted through operations such as translation, the same operation should be performed.

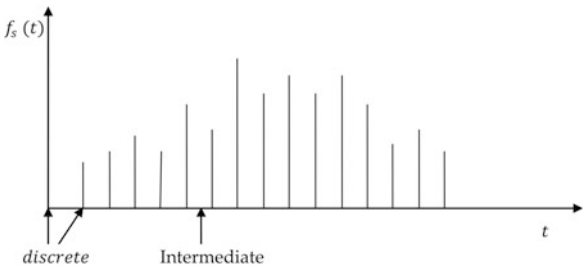

Fig. 17.13 Signals in discrete and intermediate locations

The stated properties are commonly satisfied with B-spline function (Prochazkova 2005). B-spline function is represented in Eq. (17.50):

$$x(t) = \sum_{i=0}^{n} P_i B_{i,k}(t) \tag{17.50}$$

where $n+1$ is the number of approximated samples, P_i are the control points that determine the smooth curve in $\mathrm{B}-$spline functions, and $B_{i,\ k}$ is the normalized $\mathrm{B}-$spline of order of k.

In order to produce smoother images with less artifacts, *trilinear interpolation* approach is usually being considered. Meanwhile this also comes with other overheads as computation of trilinear interpolation usually takes longer time. Therefore, optimization procedures should also be designed to reduce the computational overheads associated with interpolation and resampling procedures.

17.4 Compositing and Algorithm Performances

There are a number of notable volume rendering techniques (also referred to as the direct volume rendering) such as splatting, shear warping, texture mapping, and the ray casting, the Levoy's historic method of rendering. The commonly used compositing functions are the maximum intensity projection (MIP) and the local maximum intensity projection (LMIP). Apparently, LMIP is an extension of MIP. The image in MIP is created by selecting the maximum value along an optical ray that corresponds to each pixel of the 2-D MIP image, while the image of LMIP is created by tracing an optical ray traversing 3-D data from the viewpoint in the viewing direction and then selecting the first maximum value encountered that is larger than a preselected threshold value (Sato et al. 1998). Hence, MIP deals with maximum sampled values, while LMIP involves first local maximum above prescribed threshold and thus approximates occlusion. LMIP is considered faster and therefore better than MIP.

Due to the sensitivity nature and huge data cases in medical visualization, a robust, quality, high-fidelity, and high-performance rendering algorithm is important. Meanwhile, with the advent of high-performance computing architectures,

integrating medical application into fast and parallel hardware has been seen as a viable alternative. Various acceleration approaches have been previously proposed for image composting and medical image visualization at large in order to reduce the usual associated computational cost. Apart from the huge number of graphic processing units (GPU) recently available, Compute Unified Device Architecture (CUDA) framework has been lately seen as a heel of high-performance computing which has been leveraged in many circumstances from the clinical data acquisition phase to the results analysis. With a firm design of algorithm, the computational complexities in some of the processes are handled by the high-performance graphic components. Compositing procedures and some of the previously proposed algorithms, although most of them mainly rely on different acceleration approaches, are intensively documented in Cabral et al. (1995), Fang and Chen (2000), Röttger et al. (2000), Engel et al. (2001), Aluru and Jammula (2014), and Leeser et al. (2014) and specifically with CUDA (Adeshina et al. (2012, 2013, 2014), Liu et al. (2014), Adeshina and Hashim (2015), and Kalms (2015). Sample *2-D* slices of brain MRI, *2-D* CT slices of human pelvic region, and the obtained *3-D* correspondents after a series of translational and visualization procedures are presented in Fig. 17.14.

17.5 Conclusion

Modern medicine is greatly benefiting from the fundamentals of mathematics and algorithmic approaches. The evolvement of high-performance algorithms also opened up more growth opportunities in traditional medicine, revolutionalizing the way medical image analysis and visualization are carried out for effective disease diagnosis and therapy management.

X-ray, CT, MRI, PET, SPECT, and other related techniques are used for acquiring morphological or functional information of patients. Apparently, each of the image modalities has its peculiar advantages over another making them somehow complementary rather than being a complete replacement; hence, each of the techniques may be used in various appropriate circumstances. Moreover, in certain decisions such as consideration on the level of exposure to radiation and in certain circumstances, acquisition time could also be considered by the physician while deciding the suitable image acquisition modalities to engage.

Data representation plays a significant role in achieving a reliable visualization results. In some cases, cross-sectional 2-D images in form of slices might need to be stacked. The stacked data needs to be properly enhanced in order to improve its pixel accuracy for effective segmentation and classifications. This chapter has presented some significant stages in data representation, data reconstruction, and modeling.

Compositing approaches and most of the stages in medical visualization could be accelerated using computational techniques such as CUDA, a parallel computing platform allowing programmers to have direct access to the GPU instruction and parallel computational elements. Such acceleration procedures drastically reduce

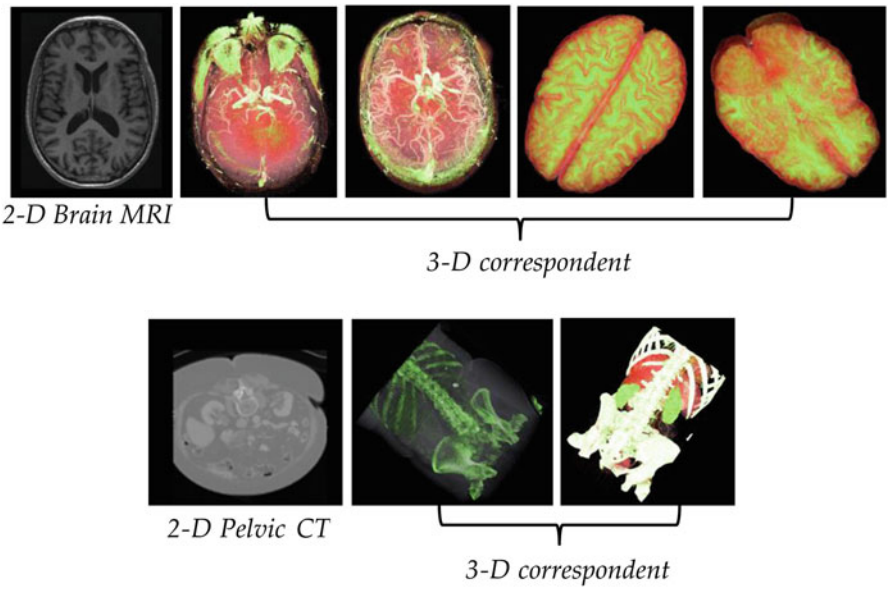

Fig. 17.14 Sample *2-D* slices, as being obtained from *MRI* and *CT* machines, and the correspondent *3-D* reconstructions (Adeshina et al. 2012, 2013)

computational overheads, thereby saving some of the associated computational cost. In the same vein, potential users (doctors) could spend less time in the disease and diagnosis procedures, thereby saving more lives.

References

Adeshina AM, Hashim R. ConnectViz: accelerated approach for brain structural connectivity using Delaunay triangulation. Interdisc Sci Comput Life Sci. 20151–13.

Adeshina AM, Hashim R, Khalid NEA, Abidin SZZ. Locating abnormalities in brain blood vessels using parallel computing architecture. Interdiscip Sci Comput Life Sci. 2012;4(3):161–72.

Adeshina AM, Hashim R, Khalid NEA, Abidin SZZ. Multimodal 3-D reconstruction of human anatomical structures using surlens visualization system. Interdiscip Sci Comput Life Sci. 2013;5(1):23–36.

Adeshina AM, Hashim R, Khalid NEA. CAHECA: Computer Aided Hepatocellular Carcinoma therapy planning. Interdiscip Sci Comput Life Sci. 2014;6(3):222–34.

Aldrich MB, Marshall MV, Sevick-Muraca EM, Lanza G, Kotyk J, Culver J, Wang LV, Uddin J, Crews BC, Marnett LJ, Liao JC, Contag C, Crawford JM, Wang K, Reisdorph B, Appelman H, Turgeon DK, Meyer C, Wang T. 2012. Biomedical optics express; 2012. p. 764–776.

Aluru S, Jammula N. A review of hardware acceleration for computational genomics. Hardware Acceleration in Computational Biology. IEEE Publications; 2014.

Bingham K. Mathematics of local X-ray tomography. Master's thesis. Helsinki University of Technology; 1998.

Butte AJ. Translational bioinformatics: coming of age. J Am Med Inform Assoc. 2008;15:709–14.

Cabral B, Cam N, Foran J. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. IEEEl; 1995. p. 91–131.

Chen J, Qian F, Yan W, Shen D. Translational biomedical informatics in the cloud: present and future. BioMed Res Int. 2013;2013:1–8.

Dhawan AP, Huang HK, Kim D-S. Principles and advanced methods in medical imaging and image analysis. Hackensack: World Scientific Publishing; 2008.

Engel K, Kraus M, Ertl, T. High-quality pre-integrated volume rendering Using hardware-accelerated pixel shading. ACM 2001 1-58113-407-X; 2001. p. 9–14.

Fang S, Chen H. Hardware accelerated voxelization. Comput Graph. 2000;24(3):433–42.

Faridani A. Introduction to the mathematics of computed tomography. Inside Out Inverse Prob Appl. 2003;47:1–46.

Faridani A, Ritman EL. High-resolution computed tomography from efficient sampling. Inverse Prob. 2000;16(3):635.

Hege HC, Höllerer T, Stalling D. Volume rendering – mathematicals models and algorithmic aspects. In: Nagel W (Hrsg.) PartielleDifferentialgleichungen, Numerik und Anwendungen. Konferenzen des ForschungszentrumsJülich GmbH, S; 1996. pp 227–255.

Hood L. Systems biology: integrating technology, biology, and computation. Mech Ageing Dev. 2003;124:9–16.

Hornak JP. The basics of NMR. Rochester: Department of Chemistry, Rochester Institute of Technology; 1997.

Jenkins R. X-ray techniques: overview. Encyclopedia of analytical chemistry; 2000. p. 13269–88.

Kalms M. High-performance particle simulation using CUDA. Sweden: Linköping University; 2015.

Leeser M, Mukherjee S, Brock J. Fast reconstruction of 3D volumes from 2D CT projection data with GPUs. BMC Res Notes. 2014;7:582.

Liu S, Chen G, Ma C, Han Y. GPGPU acceleration for skeletal animation-comparing Opencl with CUDA and GLSL. J Comput Inf Syst. 2014;10(16):7043–51.

Preim B, Bartz D. Visualization in medicine theory, algorithms, and applications. Amsterdam: Morgan Kaufmann Publishers, Elsevier Inc; 2007.

Prochazkova J. Derivative of B-Spline function. In: Proceedings of the 25th conference on geometry and computer graphics. Prague; 2005.

Radon J. Über die Bestimmung von FunktionendurchihreIntegralwerte längsgewisserMannigfaltigkeiten. Ber Verh Sächs Akad WissLeipzig Math Nat Kl. 1917;69 (1917):262–77.

Roentgen WC. On A New Kind of Rays. Ann Phys Chem. 1898;64:1–11.

Röttger S., Kraus M, Ertl T. Hardware-accelerated volume and isosurface Rendering based on cell-projection. In: Proceedings of the conference on visualization. IEEE Computer Society Press; 2000. p. 109–16.

Sato Y, Shiraga N, Nakajima S, Tamura S, Kikinis R. Local Maximum Intensity Projection (LMIP): a new rendering method for vascular visualization. J Comput Assist Tomogr. 1998;22(6):912–7.

Schroeder W, Martin K, Lorensen B. The visualization toolkit, an object-oriented approach to 3D graphics. 3rd ed. Clifton Park: Pearson Education, Inc.; 2002.