A Virtual Reality Based Surgical Simulation as an Alternative of Halal Surgical Trainings and Better Surgical Planning

Sugeng Rianto
Fisika FMIPA UB
*priantos.ub.ac.id

Abstract. The growing of medical industries need cadaver as a part of full filling better understanding of the components of surgical competency in teaching surgical technical skills in safe and pedagogically efficient environment. However, majority of Muslim scholars argue that using cadaver for surgical trainings are not permissible (haram), thus to find alternative halal procedures replacing cadaver in surgical training are urgent. Moreover, the old approach of 'see one do one, teach one' is no longer acceptable to either the surgical profession or to the well-informed and demanding public. New tools have been developed for teaching and assessing technical skills outside the operating room using virtual reality simulation, which has been applied for many years with great success in many industries including aviation and the military. With software development, the simulators now enable users to perform complete procedures with the added simulation of rare anatomical variations and various pathological conditions. The interface of these high-fidelity systems enables the surgeon to 'feel' the tissue (haptic feedback). Although, the realism of these simulated procedures is still suboptimal and the high cost of virtual reality simulators, the advantages of the high-fidelity, high-cost systems have not yet been demonstrated and our studies have established the potential advantages of procedure-specific simulation. The research using phantom-omny haptic to enable surgeon to “feel, touch, and interact” with tissue, fluid, and bone during surgical procedures using PC based simulator.

1. Introduction
Cadavers or corpses take many important roles in the medical industries. Cadavers can either be used as autopsy material for medical research or as a tool for practicum for novice doctors. Moreover Indonesia, with 280 million population, is actually easy to find corpses and the price can be cheap, even corpses can be obtained without having to buy because there are many unidentified corpses in hospitals. Bodies used for medical practice are usually not identified or are victims of accidents that have been severely damaged, so it is rare in Indonesia to find corpses for surgical practice in perfect condition. In contrast to Indonesia, in developed countries many people are willing to donate bodies for surgical practice even with a very high code of ethics, including keeping corpses in good condition[1].

On the contrary, based on the view of Islamic law, using corpses as research material or practical tools is still an area of dispute between whether or not they are allowed. The stronger opinion (rajih) forbids any types of autopsy, including autopsy in the context of practicum for medical students[2][3]. There are also an authentic traditions that prohibit violating the honor of a corpse, such as chopping, slicing or breaking the bones, particularly forbidden autopsy for Muslim corpses[4].
In addition, medical simulator can be an alternative to replace corpse that is widely used in the medical industry. A remarkable advances in surgical techniques and technology in recent decades, disparities in access to necessary surgical care have dramatically increased [5]. Consistent with other global health care challenges, developing countries and rural regions carry the bulk of the burden. The ability to receive surgical care is dependent on two factors, the accessibility of surgical centers and the availability of qualified healthcare professionals to deliver that care[5]. However, current techniques have proven to be adequate in meeting the increasing demand for specialized surgical care. In addition to necessary investments to facilities and underlying infrastructure, innovations in surgical education may play a crucial role in ameliorating the world-wide shortage of surgeons, while diminishing gaps in knowledge, and ultimately increase access to care. One of the most promising strategies to achieve these goals derives from an advanced application of virtual reality simulator[6], known as surgical simulator[7].

It is shown that a medical simulator reduce the use of resources. These Emerging technologies can train new procedures, determine the level of competence and create the best planning before operations[8]. Thus, in the near future physicians and surgeons will be trained using simulation, virtual reality, and Web-based electronic learning[9], and It could also be derived as halal surgical procedure.

2. Literature Review

Virtual reality (VR) is known as a computer simulation of a real or imagined environment that allows the users experience a generated sensory stimulation from the simulated system. For instance, the user can have a visual experience in the three dimensions of width, height, and depth, and may have additional interactive experiences in full real-time motion with sound and possibly with tactile and other forms of feedbacks [10]. Formerly VR could only be run on a special dedicated expensive workstation; however, the developments of Graphics Processor Unit (GPU) make it possible to run VR on a standard Personal Computer (PC) with GPU.

In VR the developers often build a computer model to simulate or generate a virtual experience. Sometimes virtual reality is also known as virtual worlds or virtual environments (VEs). These are three dimensional computer generated environments that the user is able to not only experience interactively, but also manipulate in real time [11]. The way humans interact with their physical environments are artificially imitated in VEs. Beside, VEs provide natural interface between humans and computers. The key component of successful VR generation is the provisions of interactions that permit the user to have a sense of touch, as well as the ability to manipulate the virtual world to some extent. To support the interaction between humans and machines, VEs usually include interfacing devices connected through an output of sensory information and input of commands. Several examples of such input and output devices are 3D space mice, data gloves, 3D navigation devices, and Phantom haptic device.

Virtual reality (VR) has been widely used as a training tool in science, engineering, and industry [12][13]. In medical science for instance, VR simulators provide the potential for a realistic, safe, controllable environment for medical students or novice doctors to practice surgical operations, allowing them to make mistakes without serious consequences [13], [14]. Using surgical simulators, a procedure can also be performed for preoperative evaluations and beforehand practices for surgeons. The procedure can be recreated or repeated in a virtual environment without harming patients or placing the patient at risk of trauma or injury.

One of the significant examples is the laparoscopy simulator [15], [16]. This is a virtual reality application in surgery simulation for minimally invasive surgery. The technique provides training systems for surgery that avoid serious damage to the surrounding tissue of the actual area of medical significance. The simulation resembles an operation with a fibre optic camera and surgical tools through a small number of portals in the skin without cutting through muscles and other tissues lying in the way of the organs or joints of interest. The simulator has a synthetic human torso into which can be placed surgical procedures to allow trainees to introduce surgical like devices with force feedback.
and practice suture skills, gallbladder removal, and other procedures on latex simulated organ, while watching their work on a typical video endoscopic screen.

Despite the recent successful development of many surgical simulators, there are still many research problems remained to be answered in the field in order to provide a truly realistic surgical simulation which includes simulating natural phenomenon involved in real surgery.

3. Methodology

3.1. Volume rendering

Volume rendering is a term that is usually used for a set of techniques to display a series two dimensional (2D) images projection of a 3D discretely sampled data set. Volumetric data are very commonly presented in two ways: direct volume rendering and 3D surface rendering. The direct volume rendering scheme was firstly developed by [10], followed by [11] and [12]. On the other side, researchers such as [13], [14], and [14] have more work on 3D surface rendering. There are also limited research effort that combines both surface and volume rendering together.

In medical visualization, volumetric datasets are gathered from patient images in several ways. Three common devices producing volumetric medical datasets include computed tomography (CT), magnetic resonance imaging (MRI), and single-photon-emission-computed-tomography (SPECT). These devices generate a volume by scanning a series of cross-sections of a human body. Volume graphics is also popularly used in other engineering and science fields for data presentation and visualization. Volume rendering has also been used in haptic visualization especially for exploring scalar and vector fields of various objects [16], [17].

A volume usually consists of many particles that emit and absorb light. Several effects such as iso-surfaces and semi-transparent objects can be achieved by simply manipulating the mapped values of the light emission and absorption coefficients of the original volume data. For haptic visualization, the user will have the potential advantages of the volume rendering when they can naturally view and have sense of touch with the 3D volume in real-time through force feedback of the haptic according to the properties of the real object. However volumetric modelling requires large storage space, and is computationally expensive.

Direct volume rendering does not explicitly extract the geometric of the bounding surface. This method requires every sample value to be mapped to opacity and a color. Geometry extraction usually is undertaken by iso-surfaces visualizations. Volume data can be stored directly in special-purposed graphics memory as a stack of two-dimensional (2D) texture slices or as a single 3D texture object [17], where the texture values represent voxel (volume element) density values of an object. When 3D texture objects are used, the volume rendering process is manageable only by a graphic processor unit (GPU) [18] without the need of assistance from a central processor (CPU). A GPU based volume rendering integrates the value of a continuous volume function reconstructed from discrete sampling points along selected projectors to produce images [19]. Using GPU as a volume processor leads to faster computation and will drastically reduce the computation time.

In this project, a novel combination of surface and direct volume rendering is adapted to take full advantage of all aspects of a volume data representation. Geometry information of objects in immersive virtual environments, depth perception, 3D direct interaction, and manageable collision detection, are all issues that need to be addressed for the objectives of surgical operations and biomedical simulations.

The volumetric object is created by a combination of 2D texture and transfer functions to represent a volumetric entity in a rendering scene and inherits functions to volume’s position, orientation and origin. This volumetric data then is stored as 3D texture or is render to the screen.

3.2. Volume Segmentations

A digital image can be split into multiple segments (sets of pixels, also known as super-pixels). This partitioning process usually is defined as Segmentation. The segmentation results a simple
representation of an image into something that is more meaningful and easier to analyze [20]. In special case, volume segmentation is typically used to locate scalar field data and boundaries (surface) in volume data set and assign a label to every voxel with the same label share certain visual characteristics.

Although, there are many floating methods on volume segmentation, this project only focuses on two segmentation methods: marching cubes and color transfer functions. Marching cube refer to the work of [21] that is usually utilized for extracting a polygonal mesh of an iso-surface from a three-dimensional scalar field (sometimes called voxels). In two dimensional (2D) data application similar method is called the marching squares algorithm. Marching cubes algorithm proceeds through the scalar field, taking eight neighbor locations at a time (this forming an imaginary cube), then determining the polygon(s) needed to represent the part of the iso-surface that passes through this cube. The individual polygons are then fused into the desired surface.

The gradient of the scalar field at each grid point is also the normal vector of a hypothetical iso-surface passing from that point. When we interpolate to these normal along the edges of each cube, the normal of the generated vertices can be created. These normal are essential for shading the resulting mesh with some illumination model. The Marching cubes application mainly concern with medical visualizations including CT and MRI scan data images, and special effects or 3-D modelling with what is usually called meta-balls or other meta-surfaces.

The color transfer function method on the other hand, has role in emphasizing features in the data mapping values and other data measures to optical properties. The simplest and most widely used transfer functions (TF) are one dimensional (1D). When 1D transfer function is applied, the ranges of data values is mapped into color and opacity and usually is implemented with 1D texture lookup tables. When the lookup table is created, color and opacity are usually assigned separately by the transfer function. For correct rendering, the color components need to be multiplied by the opacity, because the color approximates both the emission and the absorption within a ray segment (opacity-weighted color)[22]. Since in 1D TF utilize only data value measurements to control the assignment of color and opacity, the effectiveness of classifying features in the data may be limited. To generate better visualization and control to the images the gradient magnitude should be included. When we use a combination of data value measurement and gradient magnitude to compose transfer function, the result is two dimensional (2D) transfer functions.

The transfer function design is not simple as in marching cubes method. Beside we need a difficult iterative procedure that requires significant insight into the underlying data set, other information, such as the histogram of data values indicating which ranges of values should be emphasized, should be provided. In the 2D transfer function, the presence of material boundaries can be identified by arches within the value and gradient magnitude distribution. To assign the resulting color and opacity to voxels with the corresponding ranges of data values and gradient magnitudes, a set of brushes are used to paint into the 2D transfer function dependent texture [23].

3.3. Hardware accelerations

Interactive direct volume rendering particularly for haptic rendering has been restricted to high-end graphics workstations and special-purpose hardware, due to the large amount of tri-linear interpolations that are required for high image quality. Implementations using the 2D texture capabilities of standard PC hardware usually render object-aligned slices, in order to substitute tri-linear interpolations by bilinear interpolations. Consequently the resulting images often contain visual artefacts caused by the lack of spatial interpolation.

The advancement of graphics hardware and the creation of its new feature of programmability have inspired people to develop algorithms transferring much of the processing needs from CPU to GPU on common PCs [24][17][23]. The intrinsic parallelism and efficient communication features of a GPU mean it can perform calculations much faster than a CPU. Furthermore, the power of GPUs is currently increasing at a much faster rate compared to that of CPUs, and the algorithm realized on GPU can have great potentials in the future. To achieve fast volume rendering, this project implement
an algorithm that takes advantage of FX Quadro. They hold some new characteristics, such as reading
texture data in vertex programs, supporting dynamic shift command in pixel programs and even longer
lengths for shader programs. A rendering technique is proposed in this project that significantly
improves both the performance and the image quality of 2D texture-based rendering. The multi-
texturing capabilities of GeForce Quadro are exploited to enable interactive high quality volume
visualization to demonstrate hardware based rendering to get efficiently render shaded iso-surfaces
and to compute diffuse illumination for semi-transparent volume rendering at interactive frame rates.

The traditional slice-based 3D texture volume rendering is performed by slicing the volume in back-
to-front or front-to-back order with planes oriented parallel to the view plane [18]. Each fragment
program gets the sampled color from texture by tri-linear interpolation and then blends them with the
current value in the color buffer using the proper blending functions, as described in Eq.1 and 2.

\[
C_d = (1 - A_s)(C_d + A_s C_s) \\
C_d = (1 - A_s)(C_d + A_d C_s)
\]

And

\[A_d = A_d + (1 - A_d) A_s\]

\(C_d, A_s\) and \(C_s\), are accumulated color from the viewing ray, opacity absorption, and the color opacity
weighted emission of the incoming fragment, respectively. In the front-to-back order, the accumulated
opacity will be stored in an A-buffer and the A-test should be enabled.

The volume data is stored using 3D texture and resample them in pixel processing. To reduce
computing time, the volume bounding cube is not sliced in CPU, but is ray casted in GPU. CPU only
produces a polygon parallel to the view plane as the basic ray, and prepares some necessary
parameters for GPU. To further speed up the intersection calculations, the slabs are used instead of
slices [25], together with pre-integrated classification. A slab is the space between two adjacent slices,
which can be rendered as a slice with its immediately neighboring slice, either in the back or in the
front, projected onto it.

3.4. Haptic Rendering

Originally haptic(Greek) refers to the sense of touch and comes from the Greek verb haptesthai,
meaning to contact or to touch. Currently, haptic refers to haptic technology that mean a tactile
feedback technology that takes advantage of a user's sense of touch by applying forces, vibrations, or
motions to the user [26]. This mechanical stimulation may be used to assist in the creation of virtual
objects (objects existing only in a computer simulation), the control of such virtual objects, and the
enhancement of the remote control for machines and devices (tele-operators). More generally, haptic
is normally used today to concern to the science of touch in real and virtual environments. This would
include not only the study of touch capabilities in different organisms, including humans, but also the
development of engineering systems to create haptic virtual environments.

The concepts of the haptic rendering infrastructure refers to the most common haptic action or
rendering surfaces to provides extensive surface rendering functionality in addition to the general
force model. Surface rendering restricts itself to the task of producing force renderings of mostly solid
objects that have a well-defined surface. This object may be somewhat soft and spongy. We
implements a surface-rendering force model at a low level that records state information about the
surfaces being touched and the point at which they are being touched. There are two points in space
that are important in describing the surface rendering: the proxy and the finger. The proxy is a small
sphere that is constrained to remain outside of all surfaces in the scene, whereas the finger is the actual
haptic device position which is not as constrained. Visually, proxy and finger are one object that is
called as a tip. When the tip touches the surface and the user push further then the proxy and finger
will be split out. The proxy remains on the surface of the object and finger a bit immerse from the
surface (see Fig.1). Thus the user fell like touches the spongy surface or fell have repulsion from
spring. But we see only the proxy on the surface unless we program to show both proxy and finger
with transparent surface to show the finger. The stylus is visual representation of the haptic in virtual environment. The stylus is composed of handle and the tip (see Fig.1). In the normal haptic, stylus handle have no collision detection to the surface, so that it could be immersed inside the object.

![Figure 1](image)

**Figure 1.** (a) The proxy and finger representing haptic tip when touching virtual surface. One model of visually haptic representations. (a) Haptic with visual stylus, handle, and Tip. (b) Haptic no.1 touch the surface from the top object, Haptic no. 2 touch from the side with handle immerse in the object, and haptic no.3 in space with no touch.

A haptic rendering is the process of how a virtual environment system allows user to touch and sense virtual objects in a simulated environment. The implementation of this process is designing force computations, so that we can decide what force should be exerted and how we will deliver the forces to users. A force computation generating feedback into a haptic is known as a force model. This force model has a means of mapping from the haptic device position to a force vector [27]. It should also deliver constrain or reaction forces that are sensed by the user through the haptic device. Although a force model is generally updated in a scene-graph loop at about 60 Hz, this force needs to be used extensively by the real-time loop at about 1000 Hz by repeating the call to its evaluate function. To keeping the stability and quality of the haptic rendering the loop is maintained in the high frequency processing, whereas the visual aspects can have frame rates only in the order of 30 Hz [28], [29].

In this project haptic rendering algorithm refers to *Reachin API engine* that execute two parallel event loops. One loop is execute once per millisecond and is used to interface with the haptic rendering interface and hardware. This loop is called as real-time loop and is keeping in millisecond interval to preserve the stability and quality of the haptic rendering. The real-time loop does not interface directly to the scene graph or field network. It works with temporary, simplified and localized objects which are derived from the scene graph representation. At the concurrent time, in a separate loop is running the scene graph loop. The scene graph loop simply runs as fast as possible given the CPU constraints of the real-time loop. This loop responsible for interfacing with, and interpreting, the scene graph data structures (via traversals).

Haptic rendering apply a collision traversal that refer to detecting collisions of the haptic device with various object in the scene. The collision traversal and visual rendering that occurs in scene graph loop are detected in real-time loop for the calculation of the force reaction and this force then is delivered back to the user in real-time loop. Although the haptic and graphics rendering are split into different loops, they remain synchronized by sharing the same data set. When each loop has separate data set, then, manual synchronization should be done. The synchronization is particularly useful to avoid issues concerned with artefacts that come up when the haptic rendering loop appear faster or slower than the real-time loop. For example, when simulating tissue cutting, changes in geometry require frequent updates in both graphic and haptic dataset in real-time. While haptic loop is writing to the shared memory, graphic loop is reading, and then the artefact may appear. To overcome this situation the synchronization to be done to avoid conflict between loops.

The haptic rendering goal is to create a 3D model that will be recognized by the device (haptic). The model may be similar to the visual object but the may have different resolution to keep high frame rate interaction with the device. The object that is established on the display is not directly generating
object that can be recognized by haptic unless the haptic rendering is undertaken in a separate thread. More detail explanation regarding force model design and its interaction with object in virtual environment (VE) will be explained in the next sections.

3.5. Surface and volume Haptics
The establishment of a surgical simulator requires developing and integrating several components consist of defining the virtual three-dimensional (3D) environment, determining the properties of the objects in the VE, managing the interface to the 3D interaction devices (haptic, space mouse) including force feedback, and design of the visual displays. Traditionally force model algorithms rely on surface modelling [30],[31], and deal with the problem of interacting force with various surface representations. The surface can be a form of geometric roughness, object collisions, an incremental surface tracing, and probe-surface contact forces interactions (shears, torques, viscosities etc.). There two main visual renderings to be considered during force model design include surface and volume rendering. Surface renderings are believe to be computationally intensive particularly during triangulation or polygon build-up process from the vertices [32]. The other important issue in interactive environment is collision detection between surface and device to generate force. The number of contact points in surface modelling is related to the number of vertices on the contacting surfaces, in the range of tens to hundreds contact points per collision instance. On the other hand volumetric objects can have the thousands for relatively small contact surfaces of volume elements [33], [34].

The iso-surface that has been discussed in previous section means nothing for haptic force-feedback unless we define the force model to generate feedback during touching haptic over the surface. The force model that is applicable for surface rendering usually is defined as surface haptic. The surface haptic should provide several properties of different forces generating feedback such as surface roughness, plastic or membrane like surfaces, deformed surface, magnetic or sticky surface, or slippery surface. These types of surface haptic can be rendered based on texture bump mapping or scalars based values and are used to send the force feedback through haptic device to user hand according the real physical properties of the object. For instance, bone should be feel hard, fat is soft and sticky, and skin is flexible as a membrane.

The surface haptic is not too complex to be calculated, however, every data that is not a part of the surface is unrepresented in the surface haptic. On the other hand, the volume data is defined as a vector valued function to produce force feedback solely from the data around the haptic tip and the velocity of the tip. The possible force could be generated based on local gradient vector depicts the orientation as well as the magnitude changes in the scalar data to generate viscosities like forces. This force may applicable to represent fluidic contents in the volume.

Force models for volume haptic usually are approximated from a proxy point that is executed at every scene graph loop in the servo motor. The volume datasets are measured based on the gradient or curvature information from scalar values of a volume object. In volume haptic, scalar value and gradient magnitude of the images have analogy into material (type of objects), surface penetrability and surface distinctness (see Table 1).

| Visual Volume Rendering | Haptic Volume Rendering |
|-------------------------|-------------------------|
| Scalar value→colour     | Scalar value→material  |
| Scalar value→opacity    | Scalar value→surface penetrability |
| Gradient magnitude→opacity | Gradient magnitude→surface distinctness |

3.6. Deformations
A deformation are created when the body suffering stress. Stress is internal forces acting within deformable body and measures the average forces per unit area of a surface within the body on which internal forces act. These internal forces are a reaction to external forces applied on the body. In the case of haptic rendering, the deformation happens as a surface reaction to a colliding device/tool. The pre-deformation state of an organ surface exhibits the viscoelastic behaviors of the tissues on and
underneath the surface [35]. This kind of behavior is controlled by elastic deformation, where each layer is considered as a linear, isotropic elastic material. By using Hooke’s law, strains are measured along the principal axes; the principal axes are found by using only a few geometric operations aligning the original and deformed triangle and computing the oriented bounding box or ellipse to the triangle. The major and the minor axes therefore provide the two principal axes. By measuring strains along the principal axes, the shear-strains do not need to be explicitly computed. The principal stresses are computed using Hooke’s law [36] as in Eq.3.

$$\sigma_1 = \frac{E}{(1-\nu^2)}(\varepsilon_1 + \nu \varepsilon_2) \quad \text{and} \quad \sigma_2 = \frac{E}{(1-\nu^2)}(\varepsilon_2 + \nu \varepsilon_1)$$

where $E$, $\nu$ and $\varepsilon$ are the elastic modulus or the Young’s modulus, the Poisson ratio, and the principal strains respectively. Once the in-plane stress has been computed, it is then converted to the forces at the vertices (nodes) of the triangle. Assume that mass is constant and distributed over the triangles. The vertices acceleration is determined by implicit integration method to obtain their new position of the triangle vertices.

3.7. Force Cuttings

The force-feedback uses a method of local model based on reaction force as the blade is pressed into skin and constrained motion in a volume. There are two issues to consider governing the force, motion, and cutting. Firstly, reaction forces (force model) that correspond to the tissue resistance felt as the scalpel pressed deeper into the cut (it is called as axial force) and the blade is constrained by the surrounding tissue to move in the cut direction. Secondly is visual modelling during blade’s motion.

There are two thread were applied during a cutting process during surgical cutting simulation. When the tissue is cut the surface topology are change and the material on the cutting path to be removed. The first issue related to graphical modelling using mesh triangulation and the second using volume removal.

In 3D case, elastic behavior was simulated with damped mass-spring systems instead of the FEM. Assuming a line-shaped scalpel were used, the re-meshing process around an active node can be done for curved instead of flat surfaces,. Then the scalpel can intersect a curved 3D surface in multiple points, so a consistent model of cutting allows multiple incisions, each with an active node. These active nodes can interact: incisions may meet, leading to annihilations, or incisions may hit folds, leading branches. We assume that the surface is given as a triangle mesh with a boundary, and that no further information on the surface shape is known. The virtual scalpel is a line segment, and its movement is given by sampled positions of its endpoints. The endpoints are assumed to move with constant velocity between the samples. During a cut, active nodes are part of the boundary of the surface. An incision is defined by the two boundary edges incident with an active node again are called cut edges. This process is useful to reflect the topology change of the surface triangle mesh after cutting.

Material removal, on the other hand, is also approximated according to the trajectory of the cutting tool. A voxel will be removed from the volume of the tissue with two conditions: when the contact force is bigger than the threshold of the tissue cutting resistance force and when the voxel exist over the grooving triangle. According to the properties of brittle material, material removal will only occur within the local area around the action point of the contact force. Therefore, the material is removed along the negative direction of the normal vector of each related voxel in the tissue volume.

4. Implementation and Result

This section describes the implementation of 3D modelling and haptic rendering, and shows some of the experimental results. The results presented here concentrate on visual and haptic renderings, and the discussions are split into human organ modelling and surgical tool modellings.

4.1. Human Organ Modelling
This section presents three human organ models generated from a CT scanner according to algorithms as presented in previous sections. The raw images were collected from Visible Human data set (VHD) [37]. The established models can be modified in certain ways such as deformation, volume clipping, and volume and surface cutting. These models are results of image reconstruction and segmentations.

The objects are segmented using two dimensional (2D) color transfer functions. This color transfer function can segment more than seven distinct objects. For instance, the head can be segmented into skin, connective tissue, ape-neurosis, loose areolar tissue, brain, pericardium and bone. The iso-surfaces can be generated by choosing one of the segmented values from the color transfer function. Besides using the color transfer function, the segmentation of the organ can also be determined according to the CT number. A Normalized CT number that ranges from 0 to 2000 is used in the project, where 0 value is associated to -1000 Hounsfield Unit (HU) of original CT Number and 2000 is 1000 HU. CT number assign the density of a voxel in CT scan on an arbitrary scale on which air has density -1000, water 0, and compact bone +1000. Similarly, Hounsfield unit is the numeric information contained in each pixel of a CT image. HU is related to the composition and nature of the tissue imaged and is used to represent the density of tissue. CT number and HU can be interchanged each other. The original image slices are coded based on gray scale values. The gray scale is also associated with the HU or CT Number. In this project, two or more colors and two transfer functions are used to segment the objects of interest.

![Figure 2. Volume and isosurfaces rendering for the human head from visible human data set; (a) One 2D transfer function of the volume rendering; (b) Two 2D transfer function of the volume rendering; (c) Three 2D transfer function of the volume rendering with volume clipping; (d) Volume clipping to show the combination of the volume and isosurfaces rendering.](image)

The results shown in Fig.2 show various manipulations of the volume rendering techniques. The first image shows the full volume when only one two dimensional (2D) color transfer function (CTF) is used. In this image all the gray scale images that represent the original CT number are coded based on the combination of two user defined colors mapping (green, orange). In the second image two 2D CTFs are used to show the separation between the skin and the bone. In the third image three 2D CTFs are employed to describe a more detailed separation between organs of interest. The last image shows the combination of volume and iso-surfaces rendering. The volume is represented by the CTF and the iso-surfaces by indexed face set based polygonal surface that is generated from segmentation using 2D transfer function.

4.2. Deformation and Cutting

Several examples of volume haptic and their interactions with organ models are shown in Fig.3. The skin may be deformed and penetrated by a tip, when the user applies a force greater than the boundary.
stiffness threshold of the skin. However, a bone cannot be penetrated unless the skin as a boundary layer is occluded (see Fig.3).

![Segmented objects for organ of interest: (a) Head; (b) Skull; (c) Heart; (d) Hand; (e) Boundary occlusion when the pressing force is applied greater than the surface stiffness; (f). An example of an object model deformation when the haptic tip interacts with the model.](image)

Figure 3. Segmented objects for organ of interest: (a) Head; (b) Skull; (c) Heart; (d) Hand; (e) Boundary occlusion when the pressing force is applied greater than the surface stiffness; (f). An example of an object model deformation when the haptic tip interacts with the model.

The model deformation examples are shown in Fig.3ef. The deformed coordinates are used for graphics rendering whereas the un-deformed coordinates are used for haptic rendering. The haptic interaction is used to determine the extent of the deformation. Since the graphic and haptic renderings are separated, the different geometry substitution for the purposes of haptic rendering to be develop to keep the high frame rate. For example a high-resolution mesh should be used to provide smooth looking graphical deformation while a coarse mesh can be used to calculate the haptic interaction and object deformation.

The quality of deformation is controlled by the stiffness field. This field controls the extent of the deformation as a function of the distance from the point of contact. The deformation is generated nicely when the triangles in the object surface are not larger than the size of the proxy. Larger triangles often cause the haptic tip to end up on the wrong side of the surface.

Two types of deformations are studied in this project: elastic and clay deformations. The elastic deformation is a local geometry deformation that only occurs when the force is applied to the object surface. When the forces no longer exist on the surface the geometry will return to its initial condition. The clay deformations, on the other hand, have properties like clay where the geometry will remain in the last form when the forces are applied.

The cutting algorithm is an extension of the deformation algorithms. While deformation only changes the geometries of the meshes and the volume to some extents, cutting can result into surface and volume removal and new surface additions on top of geometric changes.

5. Discussion

Several techniques to approximate the formulation of constrained and reaction forces with 3D volumetric model have been developed. There are three main haptic interactions that have been developed in this project include: surface haptic, volume haptic, and proxy-based volume haptic. Each type of the surface haptic calculate forces to touch virtual objects based on explicit surface descriptions that represent iso-values of scalar data. The explicit surfaces can be polygons that prevent the haptic tip from penetrating the object. The surfaces will return force feedback into the user’s hand and push the tip away from the object surface.

Volume-based haptic rendering, on the other hand, has no surface representation, thus the standard proxy-based method cannot be implemented. There are procedures to create the direct volume haptic have been explained in [38][39]. These methods propose a haptic feedback as a computation of vector function. The computation is formulated using the scalar data around the tip and the current tip.
velocity. A force to feel the boundary surface is obtained by the gradient vector representing the variation of the orientation and the magnitude in the scalar data. In this project, we employed local or global surfaces that is adapted from [40] to compute proxy-based surface haptic. Moreover, to get a haptic interaction with a volume data, a direct volume haptic has been generated. This work refer to [41] and [16] with some modifications including force model design and hardware acceleration in computation and visualization. The residual force and the new proxy position were not calculated using residual force minimization based on adaptive step-length numerical solver as in [16], in this project implement our adapted CIP that I consider have more efficient computation than its predecessor, especially when predicting forces through the closest of two boundary surfaces stiffness values.

Furthermore, along with haptic penetration, the boundary surface is removed when a pressure is forced over a surface greater than its predefined threshold. Adaptive surface re-meshing method is employed and the voxels underneath are removed. This approach splits different layers not purely by visibility of objects of interest, careful analysis of ray profiles and boundary values need to be carried out at possible feature transition areas.

6. Conclusion
A method to effectively calculate the forces for interactive haptic-volume rendering in immersive virtual environment for surgical simulation is developed. Overall, this study has achieved high quality visualizations and force feedbacks for haptic on a reasonable frame rate. The volume size of 512x512x220 and 256 x 256 x 128 can be visualized at about 20 fps without haptic jerking. The speed slows down to about 5 fps during large deformations and generally will drop to below 10 fps when the pressure force is applied over the boundary surface for an un-penetrable object (bone), the frame rates keep at above 15 fps when the GPU computations were employed.

In addition, since all surgical related simulations in this project are not related or used any organ, skin or bone of the human, it could be simplified that the surgical procedure derived from this simulation can be revealed as halal surgical procedures.

References
[1] Prasetya Online, “Sugeng Prianto: Lebih Presisi dan Efisien Dengan Bedah Virtual,” Prasetya, 2011. [Online]. Available: https://prasetya.ut.ac.id/berita/Sugeng-Prianto-Lebih-Presisi-dan-Efisien-Dengan-Bedah-Virtual-3606-id.html. [Accessed: 01-Oct-2019].
[2] Farid Ma’ruf, “Hukum Praktikum Bedah Mayat (Otopsi) « Konsultasi Islam,” Konsultasi.com, 2009. [Online]. Available: https://konsultasi.wordpress.com/2009/04/24/hukum-praktikum-bedah-mayat-otopsi/. [Accessed: 22-Sep-2019].
[3] B. S. Atiyeh, M. Kadry, S. N. Hayek, and R. S. Musharafieh, “Aesthetic Surgery and Religion: Islamic Law Perspective,” Aesthetic Plast. Surg., vol. 32, no. 1, pp. 1–10, Jan. 2008.
[4] Shiddiq Al Jawi, “Otopsi Dalam Pandangan Syariah Islam - mozaik www.inilah.com,” Mozaik.com, 2014. [Online]. Available: https://mozaik.inilah.com/read/detail/2165268/otopsi-dalam-pandangan-syariah-islam. [Accessed: 23-Sep-2019].
[5] J. Dankelman and N. Di Lorenzo, “Surgical training and simulation,” Minim. Invasive Ther. Allied Technol., vol. 14, no. 4–5, pp. 211–213, Jan. 2005.
[6] Sugeng Rianto and Rachel Li, “Fluid dynamic visualisations of cuttings-bleeding for virtual reality heart beating surgery simulation,” in Conference: Computer Science 2010, Thirty-Third Australasian Computer Science Conference (ACSC 2010), 2010.
[7] Tempo Interaktif, “Dosen Brawijaya Temukan Alat Pengganti Mayat - Nasional Tempo.co,” Tempo, 2011. [Online]. Available: https://nasional.tempo.co/read/327151/dosen-brawijaya-temukan-alat-pengganti-mayat/full&view=ok. [Accessed: 01-Oct-2019].
[8] Nicolaks Davis, “Cutting-edge theatre: world’s first virtual reality operation goes live | Science | The Guardian,” The Guardian, 2016. [Online]. Available: https://www.theguardian.com/technology/2016/apr/14/cutting-edge-theatre-worlds-first-virtual-
reality-operation-goes-live. [Accessed: 01-Oct-2019].

[9] Sam Daley, “10 Companies Using AR And VR In Surgery | Built In,” Buildin, 2019. [Online]. Available: https://builtin.com/healthcare-technology/augmented-virtual-reality-surgery. [Accessed: 01-Oct-2019].

[10] C. J. Morris and D. Ebert, “Direct Volume Rendering of Photographic Volumes Using Multi-Dimensional Color-Based Transfer Functions,” in Proceedings of the symposium on Data Visualisation 2002, Barcelona, Spain: Eurographics Association, 2002, p. 115–ff.

[11] Z. Wang, C.-K. Chui, Y. Cai, and C.-H. Ang, “Multidimensional volume visualization for PC-based microsurgical simulation system,” in Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry - VRCAI ’04, 2004, p. 309.

[12] L. Kim and S. H. Park, “A Haptic Sculpting Technique Based on Volumetric Representation,” in Lecture Notes in Computer Science, Perales FJ and D. BA, Eds. Springer, Berlin, Heidelberg, 2004, pp. 14–25.

[13] H. Zhang, S. Payandeh, and J. Dill, “Simulation of Progressive Cutting on Surface Mesh Model.” DRAFT6–08 Sept02, 2002.

[14] S. Choi and H. Z. Tan, “Toward realistic haptic rendering of surface textures,” in ACM SIGGRAPH 2005 Courses on - SIGGRAPH ’05, 2005, p. 125.

[15] C. Basdogan, C. Basdogan, and A. A. Srinivasan, “Haptic rendering in virtual environments,” STANNEY, K. (ED.), Handb. VIRTUAL Environ. LAWRENCE ERLBAUM, INC, vol. 2002, pp. 117–134, 2002.

[16] K. Lundin, B. Gudmundsson, and A. Ynnerman, “General Proxy-Based Haptics for Volume Visualization,” in First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 557–560.

[17] Wei Li, K. Mueller, and A. Kaufman, “Empty space skipping and occlusion clipping for texture-based volume rendering,” in IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, pp. 317–324.

[18] J. Kruger and R. Westermann, “Acceleration Techniques for GPU-based Volume Rendering,” Proc. 14th IEEE Vis. 2003, p. 38, 2003.

[19] M. Ikits, J. Kniss, A. Lefohn, and C. Hansen, “Volume Rendering Techniques,” in GPU Gems, R. Fernando, Ed. Addison-Wesley, 2004, pp. 667–732.

[20] S. De, S. Bhattacharyya, and S. Chakrabarty, “True color image segmentation by an optimized multilevel activation function,” 2010 IEEE Int. Conf. Comput. Intell. Comput. Res., pp. 1–4, Dec. 2010.

[21] W. E. Lorenzen, H. E. Cline, W. E. Lorenzen, and H. E. Cline, “Marching cubes: A high resolution 3D surface construction algorithm,” in Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’87, 1987, vol. 21, no. 4, pp. 163–169.

[22] C. M. Wittenbrink, T. Malzbender, and M. E. Goss, “Opacity-weighted color interpolation for volume sampling,” in IEEE Symposium on Volume Visualization (Cat. No.989EX300), p. 135–142.

[23] Milan Ikits, Joe Kniss, University of Utah, Aaron Lefohn, and Charles Hansen, “Volume Rendering Techniques,” in GPU Gems -, 2004.

[24] Moreland and K. Dean, “Fast high accuracy volume rendering,” 2004.

[25] Christof Rezk Salama, “Volume Rendering Techniques for General Purpose Graphics Hardware (Deutsche Kurzfassung),” (Deutsche Kurzfassung), 2003.

[26] S. D. Laycock1 and A. M. Dayl, “The Haptic Rendering of Polygonal Models involving Deformable Tools.”

[27] Anonimous, ReachinAPI 4 Programmer’s Guide, Revision N. Reachin Technology AB, 2005.

[28] A. Liu, F. Tendick, K. Cleary, and C. Kaufmann, “A Survey of Surgical Simulation: Applications, Technology, and Education,” Presence Teleoperators Virtual Environ., vol. 12, no. 6, pp. 599–614, 2003.

[29] M. Agus, A. Giachetti, E. Gobbetti, G. Zanetti, and A. Zorcolo,”Real-time haptic and visual
simulation of bone dissection,” *PRESENCE Teleoperators Virtual Environ.*, vol. 12, no. February, pp. 110–122, 2003.

[30] S. Choi and H. Z. Tan, “Toward realistic haptic rendering of surface textures,” *Comput. Graph. Appl. IEEE*, vol. 24, no. 2, pp. 40–47, 2004.

[31] K.-W. Chen, P.-A. Heng, and H. Sun, “Direct haptic rendering of isosurface by intermediate representation,” in *Proceedings of the ACM symposium on Virtual reality software and technology*, Seoul, Korea: ACM Press, 2000, pp. 188–194.

[32] J. Kim, S. De, and M. A. Srinivasan, “Computationally efficient techniques for real time surgical simulation with force feedback,” in *Proceedings. 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002*, 2002, pp. 51–57.

[33] D. Marchal, F. Aubert, and C. Chaillou, “Collision between deformable objects using fast-marching on tetrahedral models,” in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Grenoble, France: ACM Press, 2004, pp. 121–129.

[34] C. Wagner, M. A. Schill, and R. Manner, “Collision detection and tissue modeling in a VR-simulator for eye surgery,” in *Proceedings of the workshop on Virtual environments 2002*, Barcelona, Spain: Eurographics Association, 2002, pp. 27–36.

[35] F. Khatyr, C. Imberdis, P. Vescovo, D. Varchon, and J.-M. Lagarde, “Model of the viscoelastic behaviour of skin in vivo and study of anisotropy,” *Ski. Res. Technol.*, vol. 10, no. 2, pp. 96–103, 2004.

[36] N. Magnenat-Thalmann, P. Kalra, J. L. Lévêque, R. Bazin, D. Batisse, and B. Querleux, “A computational skin model: fold and wrinkle formation,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 6, no. 4, pp. 317–23, Dec. 2002.

[37] G. A. H. A. J. Bordner, “Boundary element solution of the linear Poisson-Boltzmann equation and a multipole method for the rapid calculation of forces on macromolecules in solution,” *J. Comput. Chem.*, vol. 24, no. 3, pp. 353–367, 2003.

[38] S. Gibson *et al.*, “Simulating Arthroscopic Knee Surgery using Volumetric Object Representations, Real-Time Volume Rendering and Haptic Feedback,” in *Lecture Notes in Computer Science*, Springer, Berlin, 1997, pp. 369–378.

[39] R. S. Avila and L. M. Sobierajski, “A haptic interaction method for volume visualization,” in *Visualization ’96. Proceedings.*, 1996, p. 197–204,485.

[40] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha, “Six-Degree-of-Freedom Haptic Display Using Localized Contact Computations,” 2002.

[41] K. Lundin, A. Ynnerman, and B. Gudmundsson, “Proxy-based Haptic Feedback from Volumetric Density Data,” in *Proceedings of Eurohaptics 2002*, 2002, pp. 104–109.