
Soft tissues deformation and removal simulation modelling for virtual surgery

K. Jayasudha*

VTU-RRC Belgaum,
Presidency University,
Bangalore, Karnataka, 560064, India
Email: jayanataraja@gmail.com
*Corresponding author

Mohan G. Kabadi

Presidency University,
Bangalore, Karnataka, 560064, India
Email: mohankg@presidencyuniversity.in

Abstract: Major advances in the area of virtual reality have paved the way to an important application called surgical simulators. These are safe methods to carry out surgical planning and training. Surgical simulators are expected to replace conventional surgery training methods in the near future. Achieving soft tissue deformations in real time is a challenging task in virtual surgery. The most commonly used methods for deformation simulation are finite element method and mass spring method. The proposed method makes use of Delaunay triangulated mesh model to depict multiple layers of skin. This paper presents simple method of 3D soft tissue deformation and removal simulation using visualisation toolkit. The presented framework is able to simulate: collision detection, deformation and removal of soft tissues for real time computation. Multilayered model of human skin using Delaunay triangulated approach is developed as a pre-process step. The same interactive model is considered for deformation and removal simulation approach. Necessary meshing algorithms are used based on Delaunay criteria to obtain qualitative results.

Keywords: collision detection; deformation; skin; soft tissue; scalpel.

Reference to this paper should be made as follows: Jayasudha, K. and Kabadi, M.G. (2020) 'Soft tissues deformation and removal simulation modelling for virtual surgery', *Int. J. Intelligence and Sustainable Computing*, Vol. 1, No. 1, pp.83–100.

Biographical notes: K. Jayasudha obtained her MTech degree in Visvesvaraya Technological University, Belgaum. She also pursuing her PhD in the same university and has 15 years of teaching experience. Presently, she is working as an Assistant Professor in Presidency University, Bangalore, Karnataka. Her membership in professional bodies are life time membership in CSI, life time membership in Cryptology Research Society of India (CRSI) and professional membership in Institute for Engineering Research and Publication (IFERP). She presented ten technical articles in national and international journals. She also presented six papers in national and international conferences.

Mohan G. Kabadi received her PhD degree in Computer Architecture, CEG Anna University, Chennai, India. She has 35 years of teaching experience. Presently, she is working as a Professor and Head of CSE Department, Presidency University, Bangalore. She is a Fellow of Institute of Engineers (FIE), India, member of Indian Society of Technical Education (MISTE). Her research area includes energy efficient processor subsystems, high performance computing and WSN.

This paper is a revised and expanded version of a paper entitled 'Soft tissues deformation and removal simulation modeling for virtual surgery' presented at National Conference on Cognitive Computing, Presidency University, Bangalore, 26 April 2019.

1 Introduction

Virtual surgery procedure allows medical practitioner to have look and feel of 3D graphic images of real patient data. This would help the surgeon or medical practitioner to visualise and understand the complex data to the closest extent. This not only reduces the cost but also reduces time to carry out actual surgery with less number of mistakes. Achieving deformation in soft tissue modelling is a real challenging task in conventional mesh modelling techniques. Traditional approach is to use tetrahedral meshes for soft tissue modelling. Meshes formed with tetrahedrons are quite complex and takes more computation time. To get high accurate results realistic simulation is gaining popularity. This framework attempts to provide realistic and real time simulation modelling of soft tissues with respect to deformation and removal of mesh cells. Initial step is to achieve collision detection between soft tissue model and the scalpel. Deformation occurs after collision detection. Necessary algorithms are used to show the deformation effect. Next deals with removal of cells, if deformation is continued for a prolonged period of time. The proposed framework attempts to show deformation effect on these meshes using deformation algorithm and also remove those intersecting cells using removal algorithm. All these are done with the required computation time in real time as accurate as possible. Next section deals with, reviews of techniques are briefed for modelling of soft tissues, collision detection process, surgical instrument called scalpel, volumetric deformation methods and real time simulation modelling.

2 Related work

2.1 Modelling of soft tissues

The soft tissues present in the human skin are being modelled for simulation. Under this many works has been carried out. Finite element method (FEM) is one of the most popular method for modelling of soft tissues with mathematical background. Soft tissues that are Surface based are modelled using FEM. Yan et al. (2007), Guan et al. (2014) and Nebel (2001) uses FEM method for simulation and stress-strain analysis of soft tissues. Menciassi et al. (2001) discusses a model used to estimate mechanical properties. Crichton et al. (2011) uses nano indentation technique to show visco elastic properties in

different layers of skin. Bouten et al. (2011) explains tissue development of arteries and heart valves which means of cardio vascular tissue engineering. d'Eon et al. (2007) demonstrates rendering of human skin in different shades using Gaussian sums whereas Visscher (2010) proposes skin imaging and early detection of tissue injury which includes visual assessment and photographic image collection. Genzer and Groenewold (2006) measures mechanical properties of skin. Mentions how wrinkles can help to study various physical phenomena because wave length of wrinkles depends only on material properties of skin. Wang et al. (2006) explains simultaneously growing two skins with identical connectivity over two skeleton models. They use duplicate skin algorithm and the resultant skin meshes are comparable. Varkey et al. (2015) discusses how to reestablish the protective barriers of skin when loss of skin is extensive. Shah and Gupta (2013) presents cubic spring mesh model on nonlinear springs i.e., modelling biological tissues. The above mentioned papers use various complex techniques like Gaussian sums, skin imaging, FEM, governing equations, continuum model etc. The proposed method uses simple Delaunay triangulated mesh model for modelling soft tissues.

2.2 Collision detection process

Collision detection process is the interaction taking place between the virtual deformable organ models with a virtual rigid tool called scalpel controlled by the user. It is an important step before deformation process. A survey is done on various collision detection procedures. Kockara et al. (2007) mentions a survey of broad-phase and narrow-phase collision detection techniques. Broad-phase technique incorporates three algorithms called all-pair-test, sweep and prune and hierarchical hash table. Narrow-phase technique incorporates four algorithms called feature based, volume based, simplex based and spatial data structure. Compared to both narrow phase gives more detailed information. Fares and Hamam (2005) once again presents the same phases but with different algorithms. Narrow phase includes linear programming, medical axis, I-collide algorithm etc., whereas broad phase includes sphere trees, c-trees, AABB trees etc. Bruyns et al. (2002) provides a survey of various techniques for interactive mesh cutting. Nienhuys and van der Stappen (2000) also explains interactive mesh cutting using FEM without requiring extensive matrix updates or pre-computation whereas Bielser and Markus (2000) explains mesh cutting using axis aligned bounding box hierarchy. Zhang and Kim (2007) presents a fast interactive collision detection using streaming AABB's algorithm. He and Kaufman (1997) discusses collision detection between two volumes octree and sphere tree by finding intersecting regions and calculating collision detection probabilities. As per Teschner et al. (2005), a variety of deformable collision detection approaches are explained based on bounding volume hierarchies, spatial positioning, image space technique and stochastic methods. He concludes that bounding volume hierarchies are very efficient in general but spatial partitioning is appropriate for rigid bodies. Ganovelli et al. (2000) proposes octree associated with axis aligned bounding box, a general approach to reduce cost of collision detection. Frisken-Gibson (1999) present set of algorithms for performing collision detection, deformation and cutting, carving and joining using linked volumes. Basdogan et al. (2001) and Gregory et al. (2005, 2000) explains fast and accurate collision detection for haptic interaction. Basdogan et al. (2001) develops a training system to simulate laparoscopic procedures. Whereas, Gregory et al. (2005, 2000) uses H-COLLIDE, a framework that supports six degree of freedom (DOF) haptic devices for collision tests

between pair of 3D objects and flexible surfaces. All these papers use complex algorithms, but the proposed method uses cell search algorithm, a simple line data set function for collision detection in real time.

2.3 Surgical instrument called scalpel

A surgical instrument called as scalpel is used for interactive mesh cutting in a virtual surgery environment. Surgical instrument plays a vital role in simulating cutting procedures. It depends on several factors such as thickness of needle, sharpness of scalpel at which skin ruptures, and force applied on scalpel for piercing the skin. Duan et al. (2013) explains tracking and positioning apparatus of surgical instruments in a virtual surgery training system by providing six DOF with stereoscopic vision. Okamura et al. (2004) presents experimental procedure for acquiring data from soft tissues with the help of force model for needle insertion. Henry et al. (1998) presents transdermal drug delivery that enhances transport of molecules across the skin, uses a micro fabrication technique to make arrays of micron-size needles into slices for piercing the skin. Mazura and Seifert (1997) presents a method of virtual 3D cutting operations in 3D tomographic data whereas Nienhuys and van der Stappen (2004) explains a method for producing cuts in triangulated surfaces. Nguyen et al. (2013) uses a prototype to validate new surgical instrument in which user tests the virtual insertion of rod in 1, 2 and 3 screws. Cheng et al. (2017) presents interactive cutting simulation model for soft tissue. For this nonlinearity and visco elasticity equations are used to incorporate physical and mechanical properties of skin. All these are applied in a linear simulation cutting system with haptic feedback. Each paper encompasses surgical instrument using either robotics, stereoscopic vision, CAD etc. This paper uses simple scalpel made out of an assembly of 3D objects using visualisation tool kit.

2.4 Volumetric deformation methods

A volumetric deformation method incorporates topological modifications like stretching, skinning and transformations on the virtual organ. It is the process of resizing the intersected vertices of mesh model to undergo topology modifications. Ghali (2008) and Nienhuys (2001) uses FEM to carry out deformation and interactive cutting simulation. Ghali (2008) carries out large deformations by incorporating stress-strain measurement whereas Delingette et al. (1999) uses three physical models for the same, Nienhuys (2001) uses iterative algorithm. Gibson et al. (1997) presents survey of deformable modelling namely: mass-spring model, FE model, continuum model, low DOF models whereas Meier et al. (2005) presents a survey with comparison of different deformable models. Peng et al. (2013) proposes 3D node snapping algorithm to modify surface topology of objects using haptic device. Milliron et al. (2002) presents a frame work for mesh warp with conceptual and mathematical foundation. Kim et al. (2012) proposes deformable mesh data for volume removal of virtual gall bladder using laparoscopic surgery training. The current work describes volumetric deformation using simple algorithm such that when scalpel intersects the cube, its intersection points are known and vertices are resized.

2.5 Real time simulation modelling

Real-time simulation modelling is the process of interaction taking place between modelled mesh and virtual scalpel in time or at the same time with probable outcome. Lombardo et al. (1999) and Cotin et al. (1999) describes interpenetration tested on liver geometry and rigid tool. They use OBB trees whereas papers of Cotin et al. (1999), Zhang et al. (2005), Bielser et al. (1999) and Duan et al. (2013) describes real time simulation and deformation using mass spring techniques. Zhang et al. (2005) uses balloon segmentation technique whereas Bielser et al. (1999) uses Runge-Kutta iteration. Seiler et al. (2011) presents adaptive octree based approach for interactive cutting of deformable objects. Abellan et al. (2015) describes interstitial fluid flow propagation to evaluate mechanical properties of different skin layers by using an indentation device. Möller (1997) explains a method for testing intersected triangles in real time using modified algorithm. Pan et al. (2018) describes real time dissection approach to organ geometry consisting of inner metaballs and surface mesh with texture information. This makes use of VR-based laparoscopic surgery simulator with a haptic interface. The work describes a simple rendering algorithm for visualising the changes occurring to the skin prototype to simulate realistic and real time implementation.

All these above mentioned papers gives us an idea about how tissue modelling, detecting collisions with respect to surgical instrument, also to carry out these simulations in real time. Although this paper does not use haptic feedback system but an attempt is made to prove the concept of soft tissue modelling and deformation of human skin using visualisation technique particularly relevant to virtual surgery. The next section discusses the methodology to carry out the real time simulations.

3 Methodology

A schematic representation of methodology consisting of system input along with process involved in obtaining the system output is shown in Figure 1. This section briefly describes the methodology used in building up the prototype based on the following requirements:

- Prototype development of soft tissue layers based on triangulated meshes.
- Scalpel development and adding collision detection function to perform skin cutting, and should be able to detect collision when scalpel comes in contact with the skin prototype.
- Algorithm development for finding the deformation of only intersecting cells. That is, as the incision of scalpel into the skin proceeds, the skin should change the shape accordingly.
- Algorithm development to remove intersecting cells. That means when the piercing of scalpel into the skin is continued, after the threshold limit the skin gets cut.
- Algorithm development for visualising the changes occurring to the skin prototype (rendering).

Figure 1 System model (see online version for colours)

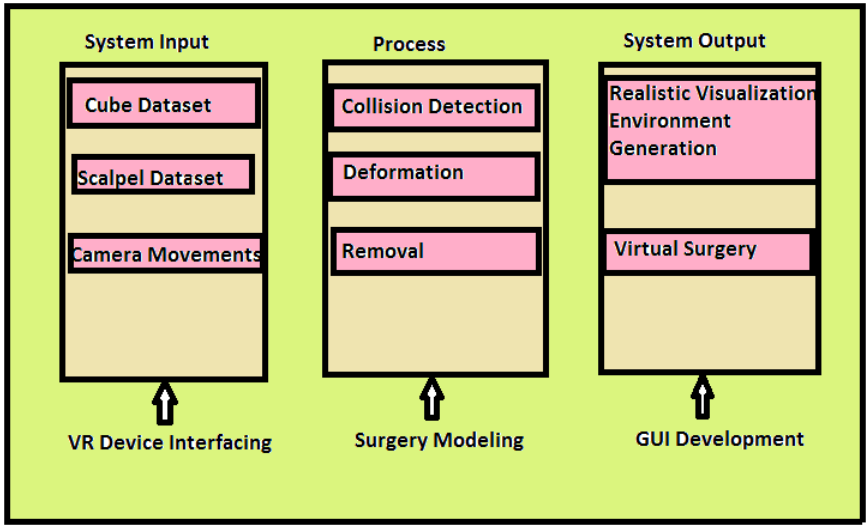
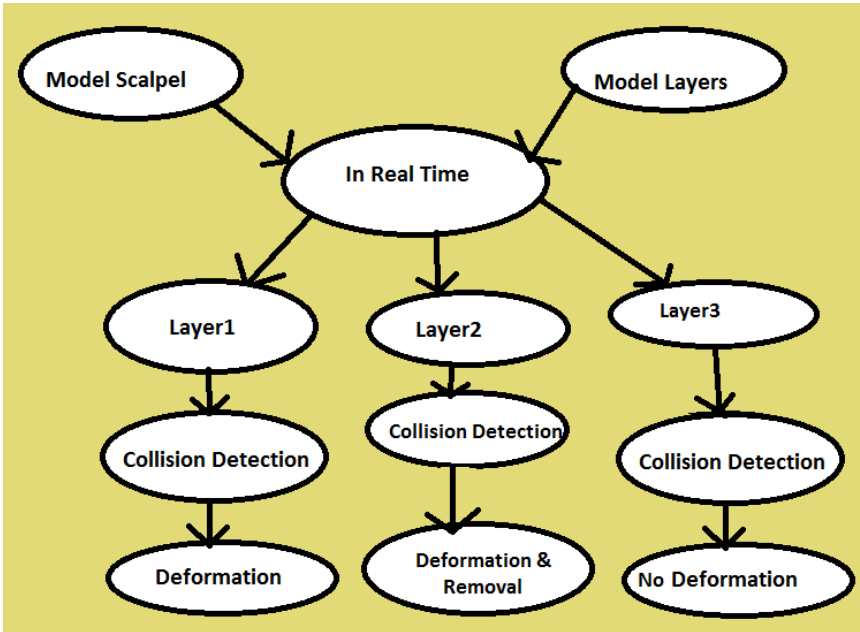


Figure 2 depicts simulation flow chart in the form of data flow diagram of the entire framework. The scalpel and soft tissue layers (cube dataset) are modelled as an initial step. A framework is suggested to formulate three different layers of skin based on Delaunay triangulated mesh using cube data structure. The generated triangulated mesh is multilayered structure depicting human skin layers and is aligned to 3D textured mapping as a pre-process step (Jayasudha et al., 2018). Various changes occur at each layer of the soft tissues when scalpel intersects the multilayered skin model in real time. The collision detection procedures detect the interaction between the scalpel and the tissue structures that can occur only at layer1 and layer2. This is because the first two layers (epidermis and dermis) are spongy in nature and are deformable according to Nebel (2001). The third layer (subcutaneous) is connected to muscles which are very hard and not movable by themselves. Therefore, it is assumed that muscles cannot be deformed by non-destructive external forces and therefore these interactions have not been computed. Hence, even though collision detection takes place at layer3 no deformation can be observed. Incision modelling involves geometry updates to be carried out enabling modification of triangulated mesh. This leads to next stage called deformation. To compute deformation on the tissue structure, a deformation algorithm is run. Even to compute removal of intersected cells, a removal algorithm is run. Deformation cannot take place at third layer even though there is collision detection as it leads to bone marrow. Hence, there is no point of deformation taking place. The soft tissue to be modelled is a three layered structure which depicts the multiple layers of human skin. This is generated using triangulated mesh based on Delaunay criteria in a pre-process (Jayasudha et al., 2018).

Figure 2 Data flow diagram (see online version for colours)



The Delaunay criterion states that ‘for n-dimensional cases a circumsphere of each simplex within the mesh contains only the n+1 defining points of the simplex’. Since the prototype is formulated using a Delaunay triangulated cube, each cube is composed of 8 vertices, 18 edges and 12 triangles. A nonlinear cubic spring mesh model (Cerdeira and Mahadevan, 2003) is studied for modelling soft tissues in real time simulation shown in Figure 3. Since the prototype is formulated using a Delaunay triangulated cube, each cube is composed of 18 edge-springs with linear spring constant k_1 and cubic spring constant a_1 and 18 diagonal-springs with linear spring constant k_2 and cubic spring constant a_2 is as shown in equations (1), (2), (3) and (4).

$$C_{edge} = K_1 x + a_1 x^3 \tag{1}$$

$$C_{diagonal} = K_2 x + a_2 x^3 \tag{2}$$

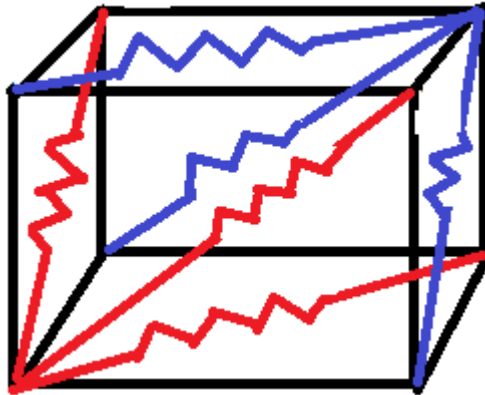
$$K_1 = \frac{(1-3\nu)E}{4(1+\nu)(1-2\nu)} \tag{3}$$

$$K_2 = \nu E / (2(1+\nu)(1-2\nu)) \tag{4}$$

The coefficients of linear terms k_1 and k_2 are based on equations of linear elasticity and equilibrium conditions as given below where E is young’s modulus and ν is poisons ratio. According to Nebel (2001), human skin is composed of three layers: epidermis, dermis and hypodermis/subcutaneous. Epidermis is the topmost layer that acts as a basic protective structure and is little thin. Dermis is the middle layer which is soft, thick and provides cushioning effect. Hypodermis/subcutaneous is the last layer which is very thick and closely connected to muscles. Varkey et al. (2015) describes about the thickness of each layer and their inner contents. According to this epidermis is about 50–100 μm thick and is made up of two components namely stratum corneum (dead cell layer) and

keratinising epithelial cells (living epidermis). Dermis is 300–400 μm thickness, composed of two components called collagen and elastin fibres. Genzer and Groenewold (2006) defines the mechanism of wrinkling in human skin with respect to the geometry of wrinkles. According to author the wavelength of the wrinkle depends on material properties of skin and foundation thickness.

Figure 3 Cubic spring model (see online version for colours)



Source: Cerda and Mahadevan (2003)

According to Nebel (2001), human skin is composed of three layers: epidermis, dermis and hypodermis/subcutaneous. Epidermis is the topmost layer that acts as a basic protective structure and is little thin. Dermis is the middle layer which is soft, thick and provides cushioning effect. Hypodermis/subcutaneous is the last layer which is very thick and closely connected to muscles. Varkey et al. (2015) describes about the thickness of each layer and their inner contents. According to this epidermis is about 50–100 μm thick and is made up of two components namely stratum corneum (dead cell layer) and keratinising epithelial cells (living epidermis). Dermis is 300–400 μm thickness, composed of two components called collagen and elastin fibres. Genzer and Groenewold (2006) defines the mechanism of wrinkling in human skin with respect to the geometry of wrinkles. According to author the wavelength of the wrinkle depends on material properties of skin and foundation thickness.

According to Cerda and Mahadevan (2003), skin gets deformed only up on outside mechanical force or up on muscle contraction. As per their theory, dermis is 10 times thicker than epidermis in terms of micrometres (μm) i.e., $(E_s / E_f) \approx 1,000 \mu\text{m}$, therefore $\lambda \sim H_f$.

E_s elastic modulus of epidermis;

E_f elastic modulus of dermis

Λ sinusoidal deflection profile of skin;

H_f thickness of dermis.

That means when there is skin deflection, its effect is almost seen in dermis layer because of its thickness. The mechanism of wrinkling of skin is beautifully explained by them through simple understanding of many natural phenomenons like geometry, mechanics,

physics and biology. Once the geometric model of skin prototype is ready next step is to determine the line object coordinates in real-time. A line object is used as a preliminary virtual tool for detecting collision with the dataset. The line object is set with a viewable radius for easy interaction. The end points of the line object are determined in real time and checked if any part of the data set is collided. A virtual scalpel is modelled by coupling the 3D objects from the available resources. The two objects such as cone and cylinder are combined in an assembly and then translated, rotated and scaled using visualisation tool kit (VTK) so as to be rendered in the scene. The virtual scalpel model is shown in Figure 4.

Figure 4 Virtual scalpel



The scalpel coordinates are then used to detect collision with dataset. The next step is to simulate the operations that can be performed on the skin i.e., the true skin behaviour like collision detection function which detects the intersection point of scalpel with the skin prototype. Next is to implement deformation of cells such that whenever the scalpel is inserted into the skin, the cells get deformed or transformation of cells takes place, down the line if more pressure is applied to the scalpel it leads to skin cut i.e., removal of cells at later stage. At last describes rendering function such that only those cells which are towards the camera are rendered. These algorithms are described below.

3.1 Collision detection function

The collision detection occurs when scalpel comes in contact with skin prototype. A line object is used as a preliminary virtual tool for detecting collision with the dataset. The following function shown in Figure 5 is run when the scalpel is brought in contact with the cube dataset. Collision detection function parameters include:

- p0 & p1: extremes of intersecting lines
- t: is the tolerance (a value between 0 and 1)
- x: is intersection point in data coordinates
- pcoords: parametric coordinates within the polygon
- Subid: “sub cell” within the cell.

Figure 5 Collision detection function

```
isCollided = locator.intersectWithLine ( p0, p1, 0, t, x, pcoords, subid )
```

3.2 Deformation algorithm

Cell deformation occurs by changing its volume of cells. After collision detection, next step is deformation stage which works according to the algorithm. Deformation algorithm

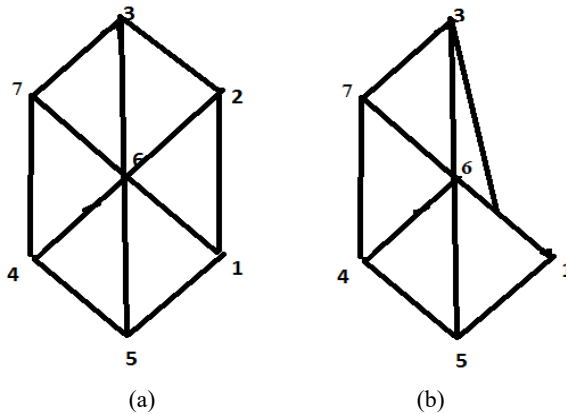
is developed such that whenever the scalpel intersects the cube its intersection point is known, adjacent edges can be traced, and corresponding vertices are resized with displacement of 1 or 2 units. The pseudo code for deformation algorithm is as shown in Figure 6. Here, x is the point of line of intersection. Each Delaunay cube contains 1 ... 8 vertices, the distance from one vertex to other is 10. That means the distance from vertex0 to vertex1 is 10, the distance from vertex0 to vertex2 is 20, the distance from vertex0 to vertex3 is 30 and so on. So each vertex is taken as 10 mesh points, therefore the total control mesh points are 10 ... 80. If intersection point (x) is greater than or equal to 10, then cell vertices of epidermis layer at vertex1 are resized along x - y - z axis and if intersection point (x) is greater than or equal to 20, then cell vertices of epidermis layer at vertex2 are resized along x - y - z axis and if intersection point (x) is greater than or equal to 30, then cell vertices of epidermis layer at vertex3 are resized along x - y - z axis so on. The same process is repeated up to 8 vertices of Delaunay cube to achieve realistic deformation of cells. Figure 7 clearly shows the pictorial representation of a cube consisting of 8 vertices along with deformation of cell taking place at vertex2. As the displacement increases, the deformation of skin also increases. Deformation occurs up to certain level (threshold value), after which performs removal algorithm.

Figure 6 Deformation algorithm

```

Deformation Algorithm ( )
for all the vertices of each Delaunay triangulated cube [1..8] {
    for all the values of each Delaunay triangulated cube [10..80] {
        if ( i <= 10 ) // where i is the intersection point
            deform epidermal cells by resizing vertices at vertex1
        if ( ( i > 10 ) && ( i <= 20 ) )
            deform epidermal cells by resizing vertices at vertex2
        if ( ( i > 20 ) && ( i <= 30 ) )
            deform epidermal cells by resizing vertices at vertex3
        .....
    }
}
    
```

Figure 7 Deformation of vertex2, (a) before and (b) after



3.3 Removal algorithm

Removal algorithm is developed such that after resizing vertices, those cells are removed from the cube, so as to depict that if incision of the scalpel in to the skin is continued, then skin gets cut. Cell removal is based on the logic that that if more pressure is applied to the scalpel, the skin gets cut, i.e., some cells are removed from skin prototype. The pseudo code for removal algorithm is shown in Figure 8. There will be eight vertices in each cube, and 80 intersection points. Removal of cells takes place in epidermis and dermis layers. This is because scalpel is piercing the skin model vertically touching both the layers.

Figure 8 Removal algorithm

```

Removal Algorithm ( )
for all the vertices of each Delaunay triangulated cube [1..8] {
    for all the values of each Delaunay triangulated cube [10..80] {
        if ( i <= 10 ) // where i is the intersection point
            remove epidermal and dermal cells by resizing vertices at vertex1
        if ( ( i > 10 ) && ( i <= 20 ) )
            remove epidermal and dermal cells by resizing vertices at vertex2
        if ( ( i > 20 ) && ( i <= 30 ) )
            remove epidermal and dermal cells by resizing vertices at vertex3
        .....
    }
}

```

When the point of intersection(x) is greater than or equal to 10 then control mesh points at vertex1 are removed from epidermis and dermis layers of skin prototype along x-y-z axis till intersection point becomes zero. If point of intersection (x)is greater than or equal to 20 then control mesh points at vertex2 are removed from epidermis and dermis layers of skin prototype along x-y-z axis till intersection point becomes zero. Similarly if point of intersection (x)is greater than or equal to 30 then control mesh points at vertex3 are removed from epidermis and dermis layers of skin prototype along x-y-z axis until intersection point turns to zero and so on. The same process is repeated up to 8 vertices of Delaunay cube to achieve realistic removal of cells.

3.4 Rendering function

Rendering function is developed for visualising the cubes at periodic intervals like rendering initial position of the cube, rendering the cube after it intersects with the scalpel, rendering after deformations occurs, and rendering during removal of cells. Rendering of Delaunay triangulated mesh is achieved using VTK function 'vtkrender ()'.

4 Results

The software used for the development of this application is in two folds. One is for visualisation and the other one for developing the code. Visualisation software is VTK 6.0 called Visualisation Tool Kit and python is used for code development. Minimum hardware requirement is pentium4 processor with 4GB ram and 512MB graphics card.

The following are the results obtained based on the algorithms explained above. The results were taken at each stage namely deformation and removal of cells using respective algorithms.

4.1 Results of deformation of cells

Whenever the scalp intersects the cube data structure, point of intersection is known, its adjacent vertices can be traced and corresponding vertices are deformed by setting control mesh points at that particular vertex. Then those cells are modified and updated only at the topmost layer i.e., epidermis layer for visual appearance. Figure 9(a) shows deformation of soft tissue prototype in solid model and Figure 9(b) shows the same result in wireframe model. Solid model renders the 3D objects in the form of solids. Wireframe model renders the 3D object in the form of lines and curves. One can clearly see the bifurcation of all the three layers that are texture mapped accordingly. Epidermal layer is texture mapped with skin colour, dermal layer is shown as flesh like and sub-cutis layer with bone flesh like structures. Also to be noted about the thickness that each layer is shown. The outermost layer is usually thin and is represented as single epidermal layer compared to other two layers which are little thicker and represented as double layers.

Figure 9 Deformation of soft tissues, (a) solid model (b) wireframe model (see online version for colours)

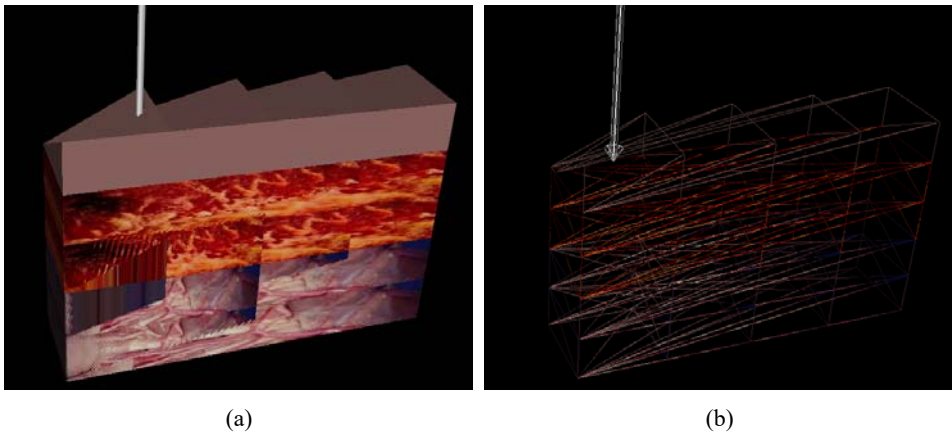


Figure 10 shows the result of deformation of cells from the cube data structure in an event action window. In the event action window, the modifications of time and array bounds are clearly visible to show exactly at what point deformation of cells have occurred. These observations reveal that modelling and deformations on human soft tissues can be carried out even on triangulated mesh.

Figure 10 Event action window of deformation (see online version for colours)

```

Command Prompt
Scalpel is collided at points P0 and P1
LineP0:
13.0899543364 38.7095066199 7.26012466854
LineP1:
9.08995433636 3.20950661986 3.45589850849

points of Intersection
{9.85508035102153, 10.0, 4.1835766086706231}
Deformation Details:

vtkPoints (000000002CC29D0)
  Debug: Off
  Modified Time: 310453
  Reference Count: 4
  Registered Events: (none)
  Data: 000000002D122B0
  Data Array Name: Points
  Number Of Points: 8
  Bounds:
    Xmin,Xmax: (0, 20)
    Ymin,Ymax: (0, 10)
    Zmin,Zmax: (0, 20)

vtkPoints (000000002CC3240)
  Debug: Off
  Modified Time: 310454
  Reference Count: 4
  Registered Events: (none)
  Data: 000000002D13E40
  Data Array Name: Points
  Number Of Points: 8
  Bounds:
    Xmin,Xmax: (0, 40)
    Ymin,Ymax: (0, 10)
    Zmin,Zmax: (0, 20)

vtkPoints (000000002CF1B60)
  Debug: Off
  Modified Time: 310455
  Reference Count: 4
  Registered Events: (none)
  
```

4.2 Results of removal of cells

The next process after deformation is the removal of cells. It is intended to show that if deformations occur for a longer period of time it leads to removal of cells. During deformation stage some vertices of the soft tissue prototype at the point of intersection gets updated. After updating the vertices those cells are removed from the cube again by using point of intersection. The point at which the scalpel intersects the triangle, those triangles are removed at layer1 and layer2 by setting control mesh points at that particular position. Then these cells are modified and updated. This procedure is carried out for first and second layers of skin prototype for visual appearance.

Figure 11 Removal of soft tissues, (a) solid model (b) wireframe model (see online version for colours)

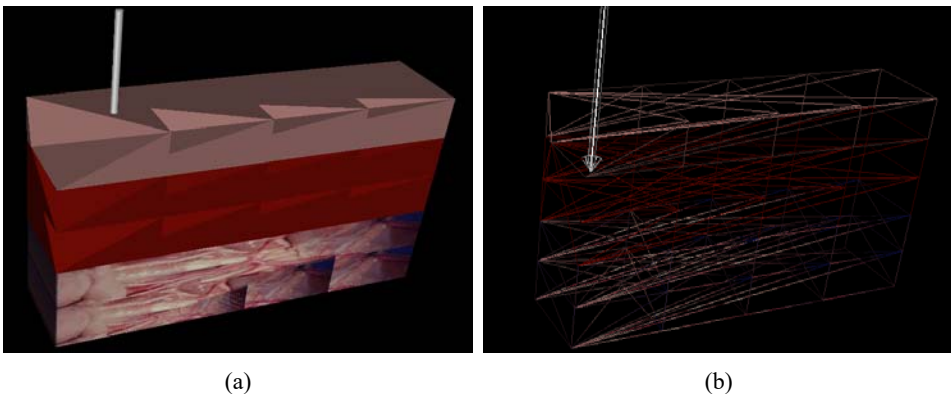


Table 1 Deformation and removal details

<i>Process name</i>	<i>Layer name</i>	<i>Array bounds</i>		<i>Memory location</i>	<i>Execution time (sec)</i>
		<i>(X, Y, Z) minimum</i>	<i>(X, Y, Z) maximum</i>		
Deformation of cells	Epidermis layer	(0, 0, 0)	(20, 10, 20)	3A4FE10	0.208204
		(0, 0, 0)	(40, 10, 20)	3A50680	0.208205
		(0, 0, 0)	(60, 10, 20)	3A50E00	0.208206
		(0, 0, 0)	(80, 10, 20)	3A51580	0.208207
Removal of cells	Hypodermis layer	(0, -10, 0)	(20, 0, 20)	3A369F0	0.208208
		(0, -10, 0)	(40, 0, 20)	3BA9D10	0.208209
		(0, -10, 0)	(60, 0, 20)	3BAA580	0.20821
		(0, -10, 0)	(80, 0, 20)	3BAAD00	0.208211
	Deep dermis layer	(0, -20, 0)	(20, 0, 20)	3C54C70	0.208212
		(0, -20, 0)	(40, 0, 20)	3C555D0	0.208213
		(0, -20, 0)	(60, 0, 20)	3C55C60	0.208214
		(0, -20, 0)	(80, 0, 20)	3D112E0	0.208215

Figures 11(a) and (11b) shows removal of soft tissue prototype cells in solid model and wireframe model. The solid model of Figure 11(a) clearly shows the removal of cells i.e., triangles in the epidermal layer. Also can observe slight deviation of cells i.e., expansion of skin outwardly in the dermal layer. This is to show that dermal layer also will have the impact when removal of cells takes place in the upper layer. The event action window is similar to Figure 10 which also shows the modified time and array bounds at which removal of cells have occurred. The following table I shows the time recorded during visual executions of deformation and removal of soft tissues. This table includes process name, layer name, array bounds of cube data structure, memory location and execution time in seconds required to complete the process. Array bounds gives the 3D co-ordinates of Delaunay cubes that forms the layers of skin. Memory location indicates the location memory space in which these cubes are stored. Execution time gives the time taken for simulation these cubes. Note that deformation of cells takes place only at the epidermis layer. Removal of cells occurs only at dermis layers (deep dermis and superficial dermis). Note from the table that there is gradual increase in execution time, this is because the scalpel has to cross only one layer initially where deformation takes place and then scalpel has to come across two layers representing cell removal process. That means to simulate that scalpel has to pierce the skin in linear vertical direction with gradual increase in time. The performance metrics of the simulated model is evaluated based on time complexity of simulation algorithms which is $O(\log n)$. Also simulation of entire mesh model is 0.04374 milliseconds.

5 Conclusions

The actual surgery operations in reality require full involvement of the medical practitioner or surgeon in the scene. So, it is essential that complex scenes are to be displayed as real as possible in surgical simulation applications. Therefore an attempt has been done to

establish the logic of deformation and removal operations that can be performed on multilayer model of skin prototype. The current work simulates the realistic scenes when scalpel comes in contact with triangulated mesh. This means to depict the effect of real time collision detection in a virtual surgery environment. When the scalpel is being pierced in to the skin realistic scenes of skin prototype like collision detection, deformation of cells, removal of cells and also rendering the scenes at all stages are studied. Appropriate algorithms are implemented to achieve this say: collision detection algorithm, deformation and removal algorithms. Realistic visualisation of skin deformation and skin removal are demonstrated, hence the system proves the concept of virtual surgery. Future enhancements can be made to extending the scope by including the following:

- to include force feedback system through haptic rendering
- to incorporate original shape regaining process of skin when scalpel is taken out
- to simulate flow of blood when skin ruptures.

References

- Abellan, M-A., Bergheau, J-M. and Zahouani, H. (2015) 'Modeling hydro-mechanical couplings between interstitial fluids and human skin soft tissues in vivo', *S13 Comportement biomécanique multi-échelles et multi-physiques*, Vol. 1, No. 1, pp.1–22.
- Basdogan, C., Ho, C-H. and Srinivasan, M.A. (2001) 'Virtual environments for medical training: Graphical and haptic simulation of laparoscopic common bile duct exploration', *IEEE/Asme Transactions on Mechatronics*, Vol. 6, No. 3, pp.269–285.
- Bielser, D. and Gross, M.H. (2000) 'Interactive simulation of surgical cuts', *The Eighth Pacific Conference on Computer Graphics and Applications, 2000. Proceedings*, pp.116–442, IEEE.
- Bielser, D., Maiwald, V.A. and Gross, M.H. (1999) 'Interactive cuts through 3-dimensional soft tissue', *Computer Graphics Forum*, Vol. 18, No. 3, pp.31–38, Blackwell Publishers Ltd.
- Bouten, C.V.C., Dankers, P.YW., Driessen-Mol, A., Pedron, S., Brizard, A.M.A. and Baaijens, F.P.T. (2011) 'Substrates for cardiovascular tissue engineering', *Advanced Drug Delivery Reviews*, Vol. 63, No. 4, pp.221–241.
- Bruyns, C.D., Senger, S., Menon, A., Montgomery, K., Wildermuth, S. and Boyle, R. (2002) 'A survey of interactive mesh cutting techniques and a new method for implementing generalized interactive mesh cutting using virtual tools', *Computer Animation and Virtual Worlds*, Vol. 13, No. 1, pp.21–42.
- Cerda, E. and Mahadevan, L. (2003) 'Geometry and physics of wrinkling', *Physical Review Letters*, Vol. 90, No. 7, p.074302.
- Cheng, Q.Q. et al. (2017) 'An interactive meshless cutting model for nonlinear viscoelastic soft tissue in surgical simulators', *IEEE Access*, Vol. 5, No. 1, pp.16359–16371.
- Cotin, S., Delingette, H. and Ayache, N. (1999) 'Real-time elastic deformations of soft tissues for surgery simulation', *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No.1, pp.62–73.
- Crichton, M.L., Donose, B.C., Chen, X., Raphael, A.P., Huang, H. and Kendall, M.A.F. (2011) 'The viscoelastic, hyperelastic and scale dependent behaviour of freshly excised individual skin layers', *Biomaterials*, Vol. 32, No. 20, pp.4670–4681.
- d'Eon, E., Luebke, D. and Enderton, E. (2007) 'Efficient rendering of human skin', *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pp.147–157, Eurographics Association.

- Delingette, H., Cotin, S. and Ayache, N. (1999) 'A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation', *Computer Animation, 1999. Proceedings*, IEEE, pp.70–81.
- Duan, Y., Huang, W., Chang, H., Chen, W., Toe, K.K., Zhou, J., Yang, T. et al. (2013) 'Modeling and simulation of soft tissue deformation', *International MICCAI Workshop on Computational and Clinical Challenges in Abdominal Imaging*, pp.221–230, Springer, Berlin, Heidelberg.
- Duan, Y., Huang, W., Chang, H., Chen, W., Toe, K.K., Zhou, J., Yang, T. et al. (2013) 'Modeling and simulation of soft tissue deformation', *International MICCAI Workshop on Computational and Clinical Challenges in Abdominal Imaging*, pp.221–230, Springer, Berlin, Heidelberg.
- Fares, C. and Hamam, Y. (2005) 'Collision detection for rigid bodies: a state of the art review', *Int. Conf. Computer Graphics and Applications (GraphiCon'2005)*, Vol. 1, No. 1, pp.1–10.
- Friskin-Gibson, S.F. (1999) 'Using linked volumes to model object collisions, deformation, cutting, carving, and joining', *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No. 4, pp.333–348.
- Ganovelli, F., Dingliana, J. and O'Sullivan, C. (2000) 'Buckettree: improving collision detection between deformable objects', *Proc. of Spring Conference on Computer Graphics SCCG'00*, Vol. 11.
- Genzer, J. and Groenewold, J. (2006) 'Soft matter with hard skin: From skin wrinkles to templating and material characterization', *Soft Matter*, Vol. 2, No. 4, pp.310–323.
- Ghali, B. (2008) *Algorithms for Nonlinear Finite Element-Based Modeling of Soft-Tissue Deformation and Cutting*, PhD diss., McMaster University.
- Gibson, S.F.F. and Mirtich, B. (1997) *A Survey of Deformable Modeling in Computer Graphics*, Technical Report, Mitsubishi Electric Research Laboratories.
- Gregory, A., Lin, M.C., Gottschalk, S. and Taylor, R. (200) 'Fast and accurate collision detection for haptic interaction using a three degree-of-freedom force-feedback device', *Computational Geometry*, Vol. 1, No. 3, pp.69–89.
- Gregory, A., Lin, M.C., Gottschalk, S. and Taylor, R. (2005) 'A framework for fast and accurate collision detection for haptic interaction', *ACM SIGGRAPH 2005 Courses*, ACM, p.34.
- Guan, Q., Du, X., Shao, Y., Lin, L. and Chen, S. (2014) 'Three-dimensional simulation of scalp soft tissue expansion using finite element method', *Computational and Mathematical Methods in Medicine*, Vol. 1, No. 1, pp.1–9.
- He, T. and Kaufman, A. (1997) 'Collision detection for volumetric objects', *Proceedings of the 8th Conference on Visualization'97*, 27pp, IEEE Computer Society Press.
- Henry, S., McAllister, D.V., Allen, M.G. and Prausnitz, M.R. (1998) 'Microfabricated microneedles: a novel approach to transdermal drug delivery', *Journal of Pharmaceutical Sciences*, Vol. 87, No. 8, pp.922–925.
- Jayasudha, K., Kabadi, M.G. and Chacko, B. (2018) 'Multilayered model of soft tissues for surgery simulation', *Integrated Intelligent Computing, Communication and Security (IICC&S), Studies in Computational Intelligence*, Springer, 15 September, Vol. 771, No. 2.7, pp.557–564, Singapore, ISBN: 978-981-10-8796-7,.
- Kim, Y. et al. (2012) 'Carving mesh with deformation for soft tissue removal simulation', *Workshop on Mesh Processing in Medical Image Analysis*, Springer, Berlin, Heidelberg.
- Kockara, S., Halic, T., Iqbal, K., Bayrak, C. and Rowe, R. (2007) 'Collision detection: a survey', *IEEE International Conference on Systems, Man and Cybernetics, 2007. ISIC*, IEEE, pp.4046–4051.
- Lombardo, J-C., Cani, M.P. and Neyret, F. (1999) 'Real-time collision detection for virtual surgery', *Computer Animation, Proceedings*, IEEE, pp.82–90.

- Mazura, A. and Seifert, S. (1997) 'Virtual cutting in medical data', *Studies in Health Technology and Informatics*, Proceedings of Medicine Meets Virtual Reality, IOS Press and Ohmsha, pp.420–429.
- Meier, U., López, O., Monserrat, C., Juan, M.C. and Alcaniz, M. (2005) 'Real-time deformable models for surgery simulation: a survey', *Computer Methods and Programs in Biomedicine*, Vol. 77, No. 3, pp.183–197.
- Menciassi, A., Scalari, G., Eisenberg, A., Anticoli, C., Francabandiera, P., Carrozza, M.C. and Dario, P. (2001) 'An instrumented probe for mechanical characterization of soft tissues', *Biomedical Microdevices*, Vol. 3, No. 2, pp.149–156.
- Milliron, T., Jensen, R.J., Barzel, R. and Finkelstein, A (2002) 'A framework for geometric warps and deformations', *ACM Transactions on Graphics (TOG)*, Vol. 21, No. 1, pp.20–51.
- Möller, T. (1997) 'A fast triangle-triangle intersection test', *Journal of Graphics Tools*, Vol. 2, No. 2, pp.25–30.
- Nebel, J-C. (2001) 'Soft tissue modelling from 3D scanned data', *Deformable Avatars*, pp.85–97, Springer, US.
- Nguyen, D.M.P., Tonetti, J. and Thomann, G. (2013) 'Evaluation of innovative surgical instrument using haptic interface in virtual reality', *The Second Vietnam Conference on Control and Automation*.
- Nienhuys, H-W. and van der Stappen, A.F. (2000) 'Combining finite element deformation with cutting for surgery simulations', *EuroGraphics Short Presentations*, pp.43–52.
- Nienhuys, H. (2001) 'Supporting cuts and finite element deformation in interactive surgery 914 simulation', *Proc. MICCAI*, Vol. 1, No. 1, pp.145–152.
- Nienhuys, H-W. and van der Stappen, A.F. (2004) 'A Delaunay approach to interactive cutting in triangulated surfaces', *Algorithmic Foundations of Robotics V*, pp.113–129, Springer Berlin Heidelberg.
- Okamura, A.M., Simone, C. and O'leary, M.D. (2004) 'Force modeling for needle insertion into soft tissue', *IEEE Transactions on Biomedical Engineering*, Vol. 51, No. 10, pp.1707–1716.
- Pan, J. et al. (2018) 'Real-time dissection of organs via hybrid coupling of geometric metaballs and physics-centric mesh-free method', *The Visual Computer*, Vol. 34, No. 1, pp.105–116.
- Peng, J., Li, L. and Squelch, A. (2013) 'Hybrid surgery cutting using snapping algorithm, volume deformation and haptic interaction', *J. Man Machine Technol.*, Vol. 2, No. 1, pp.35–46.
- Seiler, M., Denis Steinemann, J.S. and Harders, M. (2011) 'Robust interactive cutting based on an adaptive octree simulation mesh', *The Visual Computer*, Vol. 27, No. 6, pp.519–529.
- Shah, R. and Gupta, A. (2013) 'Non-linear cubic spring-mesh model for simulating biological tissues', *Transactions of Japanese Society for Medical and Biological Engineering*, Vol. 51, No. 1, p.U-12.
- Teschner, M., Kimmerle, S., Heidelberger, B., Zachmann, G., Raghupathi, L., Fuhrmann, A., Cani, M.P. et al. (2005) 'Collision detection for deformable objects', *Computer Graphics Forum*, Vol. 24, No. 1, pp.61–81, Blackwell Publishing Ltd.
- Varkey, M., Ding, J. and Tredget, E.E. (2015) 'Advances in skin substitutes – potential of tissue engineered skin for facilitating anti-fibrotic healing', *Journal of Functional Biomaterials*, Vol. 6, No. 3, pp.547–563.
- Visscher, M.O. (2010) 'Imaging skin: past, present and future perspectives', *Giornale italiano di dermatologia e venereologia: organo ufficiale, Società italiana di dermatologia e sifilografia*, Vol. 145, No. 1, pp.11–27.
- Wang, Y., Wang, C.C.L. and Yuen, M.M.F. (2006) 'Duplicate-skins for compatible mesh modelling', *Proceedings of the 2006 ACM Symposium on Solid and Physical Modeling*, ACM, pp.207–217.

- Yan, Z., Gu, L., Huang, P., Lv, S., Yu, X. and Kong, X. (2007) 'Soft tissue deformation simulation in virtual surgery using nonlinear finite element method', *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007*, IEEE, pp.3642–3645.
- Zhang, S., Gu, L., Huang, P. and Xu, J. (2005) 'Real-time simulation of deformable soft tissue based on mass-spring and medial representation', *Computer Vision for Biomedical Image Applications*, Vol. 1, No. 1, pp.419–426.
- Zhang, X. and Kim, Y.J. (2007) 'Interactive collision detection for deformable models using streaming AABBs', *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, No. 2, pp.318–329.