Interactive shape editing techniques for 3D point models using an electronic glove

AKANKSH VASHISTH

A THESIS

IN

THE DEPARTMENT

OF

COMPUTER SCIENCE AND SOFTWARE ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

CONCORDIA UNIVERSITY

MONTREAL, QUEBEC, CANADA

AUGUST 2008

Your file Votre référence
ISBN: 978-0-494-45716-0
Our file Notre référence
ISBN: 978-0-494-45716-0

Abstract

INTERACTIVE SHAPE EDITING TECHNIQUES FOR 3D POINT MODELS USING AN ELECTRONIC GLOVE

AKANKSH VASHISTH

The development of simple and intuitive interactive deformation techniques for 3D point-based models is essential if they have to find wide-spread application in different domains. In this thesis we describe an interactive technique for editing the surface of a point-based model by adapting a physically-based mesh-free shape deformation formulation to work in conjunction with data input from an electronic glove. Each finger tip of the glove forms an interaction point in 3D space, whose movement into/away from the surface is used as a directed force applied on the surface for deforming the model in an intuitive yet computationally stable manner. After the glove is spatially registered with the point-based model, one or more fingers can be simultaneously used for deforming the model. For editing large 3D point based models, we have proposed two heuristic techniques which yield interactive response times for both global and locally detailed deformations on a large model. The heuristics use a localized deformable region or use a reduced model set with offline transfer of shape modification to the larger model. The proposed technique is simple to implement and its effectiveness has been demonstrated through a number of illustrative examples and videos.

Acknowledgements

I am very grateful for the advice and support of my supervisor, Dr. Sudhir P. Mudur. Thanks to Dr. Mudur for his enthusiasm, inspiration, and great efforts to explain things clearly and simply. Throughout my thesis period, he provided encouragement, sound advice, good teaching, and lots of good ideas. His intuitive insights were my guiding light in the blind alleys of research. I further extend a special thanks to Dr. Mudur for giving me the opportunity to present a paper, which we co-authored, at a conference, which was an experience in itself.

I also wish to thank all my colleagues in the graphics and visualization lab and the database labs at Concordia, for sharing their ideas, and for their valuable suggestions. The time spent discussing science and specifically computer science with them was certainly well spent.

Last but not the least; I would like to thank my parents for having faith in my academic and professional capabilities. They have always been the bed rock of my existence and I sincerely appreciate their efforts in trying to mould me into a complete and considerate human being.

Table of Contents

List of Figures

Chapter 1

Introduction

Point-based models have seen an increased interest in research over the last ten years, particularly with the exponential increase in the computational capabilities of modern 3D graphics hardware, and the ease with which point sampled data can be acquired. Rapid advances in scanning technologies have made acquisition of 3D point sampled models of complex objects relatively easy. The widespread availability of these 3D scanners and high-end computing resources has led to a rise in demand for high resolution 3D models and objects. However, point-based 3D surface models have yet to find wide-spread adoption in main-stream application domains such as gaming, cinema, CAD/CAM, etc. Much effort is spent on creating a mesh for the acquired data or in fitting an algebraic surface representation to the scattered point data. This effort, although presently needed, would not actually be required if point based models could be computationally processed and handled, directly without the necessity for the surface topology (point connectivity) to be explicitly represented. Much of recent research on point based models has therefore concentrated on the development of fast and accurate rendering techniques and the mathematical representation of the underlying surface definition needed, mainly due to the absence of explicit surface topological information. We believe that among others, intuitive and interactive editing facilities for such acquired 3D point sampled models are also needed, if such point data models have to find wider use.

Polygonal mesh based 3D geometries have seen a long history of sustained research directed towards the algorithms and mathematics involved in editing them. Consequently, many interactive surface sculpting techniques have evolved for polygonal mesh-based surface models and also for free form surface models such as NURBS based representations. These techniques rely on surface continuity and hence cannot be conveniently adapted to the process of editing point-based models. In this thesis we present a new technique targeted towards adapting a mesh-less deformation technique in order to interactively sculpt/edit a point-based model, directly acquired from a device such as a 3D scanner with minimal or no pre processing of the 3D scanned data.

Further in the rest of this chapter, we introduce some of the underlying concepts and ideas involved in the thesis and study their applicability and integration with interactive editing techniques for point based models.

1.1 Point-Based Models

Drawing from the ideas of Csuri et al [1], Blinn [2], Reeves [3] and Smith [4] in using point primitives to model classes of objects that could not be modeled using classical geometries (e.g., smoke, clouds, fire, and trees), Levoy and Whitted [5] had introduced the possibility of using point primitives for shape modeling. Point based 3D representations can be used to represent 3D models with very high resolution scales for extensive surface, as well as volumetric, detail with minimal overhead, as there is no vertex connectivity information to be explicitly stored. Typically, a single object requires

a large number of scans to capture the object's surface from all angles and generating a single mesh from these multiple scan data sets can be a laborious and skilled task. Point-based representations have the advantage that scan data sets can be used with minimal post-processing. This allows the reduction in over all computational requirements and also assists in enhanced resolution representation.

Techniques for the conversion of geometric models to point-based data, and rendering them, were presented by Grossman and Dally [6]. Pfister et al [7] further developed this work by introducing the surfels technique, which was later used in their system called Pointshop3D [8] for handling 3D point-sampled geometry. Also of interest in the domain of point based geometry representation is the QSplat technique, introduced by Rusinkiewicz and Levoy [9] which uses a hierarchy of spheres to approximate and display high resolution models. M. Alexa et al. use the framework of moving least square (MLS) projection [10,11] to approximate a smooth surface defined by a set of points, and introduce some associated techniques to resample the surface and generate an adequate surface representation. The development of techniques for surface reconstruction using resampling and various methods to render point sampled geometries has led to an increasing interest in the field in order to ascertain the possibilities and feasibilities of the geometric surface formulations for point based models.

1.2 Physically based Deformation

Primarily, any physically based modeling/deformation technique in computer graphics targets the minimization of user/animator effort in terms of creating a required 3D structure. The physically based nature of the deformation technique is incorporated in the editing/sculpting concept in order to allow the modifications performed to be intuitive to a user/animator as he/she is comfortable with physical responses to forces and objects in the real world. Modeling deformations have been studied rigorously in the fields of mechanical engineering and computer animation. While the constraints of accuracy and computational efficiency in current deformation techniques are fairly optimized, the human intervention required to obtain desired results is still non-trivial.

Techniques such as Finite-Element Methods (FEM) and Mass-Spring Systems (MSS) which are currently popular choices as deformation methods are based on the fundamental representation schemes used in mesh-based models. These aforementioned techniques require connectivity information between the vertices in order to generate deformation solutions. With regards to high resolution models, this process becomes computationally expensive, coupled with the fact that the already undertaken process of meshing the model has also increased the overall computational requirements. Hence, these techniques are not applicable to point-based models due to the inherent differences in their representation of surface information. Thus, using a physically based mesh-less deformation technique appears to be a feasible alternative to achieve smooth deformations of 3D point-based data models.

1.3 Virtual object manipulation devices

The commonly used devices for 3D surface editing are the key board and the mouse. The key board is used when numerical input values are needed for specifying the desired surface deformation. The mouse is used to point or select, and move the part of the surface that has to undergo deformation, usually based on numerically supplied inputs or through screen based devices such as displayed valuators. While these are the interaction techniques commonly supported, it is clear that they are not the most convenient or intuitive. Haptic devices such as force feedback electronic gloves, 3D wands, etc. have been in existence for quite some time, and are beginning to gain popularity through their use in video games. A haptic device is a human wearable device which captures/applies the forces, vibrations and/or motions applied/felt by the wearer. This mechanical stimulation allows the user to interact intuitively with the virtual objects in 3D graphics environments and thus allows for a more realistic experience of interaction and visualization of cause-and-effect. With properly designed interaction techniques, an electronic glove could provide a highly intuitive surface manipulation interface for point based surface editing/sculpting.

1.4 Objectives and Contributions

Our primary aim in this research has been to develop an interactive, adaptable and intuitively simple physically based sculpting technique that works directly on the point data set, yielding again a physically appropriate deformed point model, without having to

go through the expensive process of meshing the data or fitting a surface over the point data set. This target technique should also allow the use of various user input devices to assist the user in editing/sculpting a 3D point based model. In particular, the technique should be conveniently adapted to be used in conjunction with motion capture devices as well as haptic devices which can provide the user with force feed back information. This adaptability of the proposed technique along with an analysis of various speed-up heuristics and physically approximate solutions suggest that the technique proposed and developed in this research could be an important step towards the introduction of high resolution point based geometries in the commercial as well as academic 3D graphics pipelines.

The novel approach of embedding a point based 3D model into an infinite solid in order to enable the application of elasticity solutions based on physical mathematics allows us to derive a physically based technique that gives physically approximate results while meeting user specified conditions (if any). This also implies that the forces of the real world can be simultaneously mapped on the virtual model and thus the application of multiple forces on the model can also be conveniently displayed. Deformation of the model under multiple stresses obtained from the finger tip data on a motion capture glove suggests the usability and applicability of this technique in more complex systems of sculpting and editing. The surface force solutions under stress also allow the technique's adaptability for haptic devices which provide force feedback.

We have also studied the integration of the technique with a motion capture device and obtained results portraying the capabilities of the technique. The process of adapting the motion capture device data to the technique has been studied in depth and the issues and concerns have been resolved in order to develop a stable system for using motion capture data to edit 3D point based models. The system was studied with forces acting on the point based model from a single point, as well as multiple fingers could act on the model at the same time.

As part of this research we also studied the possibility of enhancing speed in order to provide real time feedback on deformation using the following heuristics:

1) Reduced Model Set:

If a reduced model of the large model, say of lower resolution, is available, either pre-computed or computed on-the-fly using lower level of detail (LOD) computation technique, then the interactive sculpting technique can be applied to this reduced model and its deformation can be presented to the user at interactive frame rates. At the same time, the interaction data received from the haptic device can be recorded. Once the interaction is completed, then the recorded data can then be used to deform the original full scale data set to achieve the desired results with high level of detail.

2) Localization of Deformation region:

The second approach restricts the deformation to a subset of the original data, the subset being a region surrounding the point on the object where detailed editing/sculpting needs

to be undertaken. This approach can be used in cases where localized but detailed modifications need to be undertaken in a large data set. Although requiring a certain amount of preprocessing in order to setup a large data set for deformation, the availability of this method provided the user with the option to decide whether the effects of the deformation require observation of details during interaction.

1.5 Outline of Thesis

Following this introduction, the next chapter is devoted to discuss and analyze the related work that has been undertaken in the various domains. This chapter helps build a feeling for the topic and this flow is then carried into the subsequent chapters which initially build the mathematical basis of the modeling technique and then study its adaptation to motion capture and haptic based inputs. Finally, in the concluding chapters we cover implementation results, concluding remarks and future extensions.

Chapter 2

Related Work

In the graphics community, the idea of being able to edit/sculpt a 3D model in virtual space using a haptic interface and a head mounted display thus allowing the complete immersion of the artist in a manner such that all the sensory experiences of the artist mimic reality can be seen as the holy grail of human computer interface. Various steps have been taken towards this goal over the entire history of research in computer science and specifically research in computer graphics. However, only some of the techniques which have evolved or have been developed over this duration are suitable to meet the constraints of the final goal. The techniques have to meet various criteria such as adaptability, time/space complexity, intuitive results, stability etc. In this chapter we present a survey of the research that has been undertaken regarding the various sub-fields that are related to the work being presented in this thesis. The discussion of prior work can broadly be classified into the following categories:

1) Research related to 3D point based models.
2) Discussion regarding the various physically based deformation techniques.
3) Work related to the interactive use of Haptic/Motion capture devices.

Also, any research work related to any two or more of the above categories shall be highly relevant to the work which shall be presented here, and those research topics shall

be discussed in higher detail in comparison to any research involved solely in one of the above mentioned categories.

The ability to achieve object deformation is one of the major requirements for a system that targets an eventual evolution towards virtual sculpting capabilities. In the field of computer graphics, 3D virtual object deformation has been successfully implemented using a variety of object models. However, many of these methods have yet to see wide spread use, mainly due to the difficulty in providing intuitive and interactive manipulation capabilities, i.e. the techniques are not adaptable to different types of interaction models via which object deformation can be undertaken. Most of the object deformation techniques also suffer when applied to rendering along with haptic interfaces because of their inability to provide force feedback information, while a few which do possess this capability lack the computational efficiency to do so at interactive rates. A very comprehensive and detailed discussion regarding physically based deformation simulations using various techniques such as finite element/difference/volume methods, mass-spring systems, meshfree methods, coupled particle systems and reduced deformable models based on modal analysis have been presented by Nealen et al. [12]. This survey also discusses the applications of such techniques to the simulation of fluids, gases and melting objects, furthermore high level applications of the techniques such as elastoplastic deformation and fracture, cloth and hair animation, virtual surgery simulation, interactive entertainment and fluid/smoke animation are also presented in considerable depth.

With regards to simulating deformation in mesh-based models starting out with the work of Terzopoulos et al. [13] which employed the application of elasticity theory to construct differential equations that modeled the behavior of non-rigid curves, surfaces, and solids as a function of time. The realistic and dynamic animations were then achieved by numerically solving the underlying differential equations. Regarding physically-based animations using this method of finite difference discretization, a number of offline and real-time methods have been implemented in computer graphics. For example, the mass spring systems technique was used to physically deform cloth [14] by coupling the enforcing of constraints on individual cloth particles with an implicit integration method. The mass spring system has also been vastly applied in the study of deformable objects [15] where a large area of research has been the optimization of the numerical computations involved in the implicit integration needed to solve the system. Various predictor-corrector methods have been employed to alleviate the problem of solving a linear system of equations at each step. Also methods which incorporate the conservation of momentum in order to reduce the computations required at each iteration have also been studied.

The Finite Element Method (FEM) technique has been used in various physically-based simulations of deformable objects [16, 17]. The FEM has found a great deal of interest in the domain of simulating the deformation of mesh based 3D models. It has been applied in a vast range of applications ranging from biomechanical modeling, mechanical engineering visualizations, wave propagation etc.

Similarly the Boundary Element Method (BEM) has also been used to achieve the physically based deformation of mesh based objects. ArtDefo [18] describes the boundary integral equation formulation of static linear elasticity along with the related Boundary Element Method discretization technique. Also presented therein is the process of exploiting the coherence of typical interactions in order to achieve low latency; the boundary formulation presented in ArtDefo lends itself well to a fast update method when boundary conditions are sufficiently static. Both FEM and BEM have found a large range of applications in the industry but however they lack the adaptability of providing intuitive manipulation handles, which are of extreme importance if such techniques are to gain acceptance from non-technical users such as artists. Also, the inherent limitation of requiring a continuous surface definition limits the applicability of the techniques to high supervised situations where continuity is monitored or, generally, enforced.

When it comes to interfacing of haptic devices with surface sculpting, free-form solid deformation [19, 20] is the popular approach, though it is not physically based as it is fundamentally based on trivariate Bernstein polynomials to achieve relative deformations. Even though the free-form solid deformation technique is capable of achieving deformation in surface primitives of any type or degree (planes, quadrics, parametric surface patches, and implicitly defined surfaces) it is inherently depended on the continuity of the surface representation and discretization of the surface cannot be handled by this approach. However, it is capable of imposing any level of derivative continuity to local deformations. Free-form solid deformation occupies itself with providing the user the capabilities of using haptic device driven handles in order to

achieve arbitrarily shaped bumps and design surfaces which be bent along arbitrarily shaped curves.

An interesting haptic device based volumetric method suitable for both visualization and modeling applications is presented in [21]. They compute point contact forces directly from the volume data and the forces are consistent with the isosurface and volume rendering methods. This provides a strong correspondence between visual and haptic feedback. Virtual tools are simulated by applying three-dimensional filters to some properties of the data within the extent of the tool, and interactive visual feedback rates are obtained by using an accelerated ray casting method. While, the use of haptic devices enhances the user interactivity, the technique presented here does not target physically based deformations (local or global) of the 3D object. The technique presented therein is mainly geared towards surface visualization modifications, such as changing color, visibility etc. of the 3D region where the haptic device driven tool is pointed by the user. This leads to a more virtual painting based application rather than a technique which would be a predecessor to virtual sculpting applications.

Frisken and Perry [22] in their research have interestingly explored the use of adaptively sampled distance fields to unify the representation of shapes that integrate numerous concepts in computer graphics such as the representation of geometry and volume data and which support a wide range of processing operations such as rendering, sculpting, level-of-detail management, surface offsetting, collision detection, and color gamut correction. The technique presented by Frisken and Perry consists of a structure which is

uncomplicated and direct and is especially effective for quality reconstruction of complex shapes and modeling soft body contact. However, the non physically based nature of the technique renders the technique as useful only is a subset of the virtual sculpting environments where a physically realistic interaction with the 3D model is not needed or desired, as in the case of using hard edge changes to modify the object (chisels, mallet etc.).

Non-Uniform Rational Basis Spline (NURBS) based curves and surfaces have had a great impact on the visualization capabilities of 3D virtual models with a large reason for the same being their dominance in being CAD/CAM based applications in the industry for both mechanical and industrial engineering visualization. Following their wide spread acceptance, the NURBS surfaces have also been used in a variety of haptics-based applications. Thompson et al. [23] presented an algorithm supporting haptic rendering of NURBS surfaces without the use of intermediate representational schemes. The technique presented by them has been used in conjunction with algorithms for surface proximity testing and surface transitions in order to develop a haptic rendering system. This also describe a method for measuring the quality of the tracking component of the haptic rendering separately from the haptic device and force computation. Similar to earlier models concerning haptic interfacing of surface representations, this scheme does not meet the criterion of being physically based in its deformation mechanism, thus not allowing a "real world" interactivity with the 3D model. Also [25] developed an implicit to B-spline surface haptic interface and presented a haptic sculpting system for B-spline surfaces driven via a shaped implicit surface probe/tool in order to simulate the physical

world where people touch or sculpt with their fingers or tools, instead of just manipulating data points in 3D dimensional spaces. In their research the shaped virtual haptic probe/tool helped the users to relate the virtual deformation process to their real life experience. Similar to the previous approach [24] presents a haptic interaction approach for the direct manipulation of physics-based B-spline surfaces using sculpting probes. The method presented in [24] allows the users of the technique to interactively sculpt virtual 3D B-spline based models with a standard haptic device, and feel the physically realistic presence of virtual B-spline objects with force feedback throughout the design process. This research work also presents techniques which allow for point, normal, and curvature constraints to be specified interactively and modified naturally using virtual forces. The integration of haptics with traditional geometric B-spline based modeling will increase the bandwidth of human-computer interaction, and thus shorten the time-consuming design cycle. However, as in the case of the last two research models, the implicit necessity of the surface continuity criterion was the bottleneck to achieve adaptability to various types of interactivity models and under all deformation cases it was imperative that the surface remain continuous therefore not making itself available to the deformation and haptic interactivity needs of sculpting 3D point based models and/or volumetric data manipulation.

Terzopoulos and Qin [26] have undertaken research towards dynamic NURBS (D-NURBS); a physics based generalization of NURBS derived through the application of Lagrangian mechanics and implemented using FEM. Due to the implementations of NURBS which require the user to interactively adjust many degrees of freedom (DOFs) -

- control points and associated weights -- to achieve the desired 3D model representations the D-NURBS incorporate mass distributions, internal energies, force and other physical quantities into the NURBS geometric substrate. Their dynamic behavior which is a result of numerical integration of a set of nonlinear differential equations provides physically realistic and hence intuitive shape deformations. The use of Lagrangian mechanics is made to formulate the equations of motion for the D-NURBS curves, tensor-product D-NURBS surfaces, swung D-NURBS surfaces and triangular D-NURBS surfaces. Following this finite element analysis is applied to reduce the equations to optimize the numerical computations. Also, Opalach et al. [27] proposed a simple method for local deformations of implicit surface during collisions. Implicit surfaces because of their continuous behavior are well suited towards modeling organic bodies consisting of an internal skeleton and deformable flesh smoothly blended around it. An implicit surface can represent such an object's geometric "skin" that deforms according to the motion of the skeleton. The applications of the method presented in [27] can be undertaken to generate deformable object simulation, character animation and interactive sculpture.

Raviv and Elber [28] presented a direct rendering paradigm of trivariate B-spline functions that is able to incrementally update complex volumetric data sets. This incremental rendering scheme can hence be employed in modeling sessions of volumetric trivariate functions, offering interactive volumetric sculpting capabilities. This technique however, is neither physics based in its deformation nor does it have the capability of supporting discretized 3D surface models such as 3D point based models or volumetric data models.

Regarding physically-based mesh-free deformations, a vast variety of techniques have been proposed to achieve intuitive and realistic results. Techniques such as Element-free Galerkin [29] and Moving Least Squares approximation of the gradient of the displacement vector field [30] have been successfully implemented towards this goal. Muller et al. [30] present a method for modeling and animating a wide spectrum of volumetric objects, with material properties. Both the volume and the surface representation are point based, which allows arbitrarily large deviations form the original shape. In each step, the spatial derivatives are computed for the discrete displacement field using a Moving Least Squares (MLS) procedure. From these derivatives the strains, stresses and elastic forces at each simulated point are obtained. Also the solutions to the equations of motion based on these forces, with both explicit and implicit integration schemes are presented. They also suggest a technique for modeling and animating a point-sampled surface that dynamically adapts to deformations of the underlying volumetric model.

An excellent review of mesh-free methods can be found in [31] with an in-depth study of the majorly used techniques in mesh-free deformation models such as EFGs, MLS etc. A physically based deformation technique for point clouds has been developed in [32] based on the solution to the Kelvin's problem in elasticity theory. The study in [32] proposes a mesh-free deformation technique where only unconnected points are involved. The idea being the development of an approximate analytical solution using the Kelvin solution. Due to the fact that no mesh is involved, deforming a complex shape is as straightforward as deforming a simple one. Also using this technique, the trade-off

between efficiency and accuracy is easy to achieve by redistributing the points concerned both density and in number.

Specifically, deformation techniques for point-based models have also been extensively researched in recent years. Pointshop 3D [8], an open source system for point-based models includes interactive editing of shape and appearance. Their approach uses a point cloud parameterization and a dynamic resampling scheme based on a continuous reconstruction of the surface, with surface editing being limited to normal displacement. A physics-based framework based on particles for deforming implicit surfaces was presented in [33]. They present a technique for modeling elastic surfaces based on interacting particle systems, which can be used to split, join, or extend surfaces without the need for manual intervention. The particles used in their technique have long-range attraction forces and short-range repulsion forces and follow Newtonian dynamics, much like the computational models of fluids. This enables the particles to model surface elements instead of point masses or volume elements; also an orientation to each particle's state is added. Also introduced are the interaction potentials for the oriented particles which favor locally planar or spherical arrangements. Techniques for adding new particles automatically, which enables our surfaces to stretch and grow have also been studied. While on the other hand [34] presents a particle-based approach to sampling and controlling implicit surfaces. The technique implements a method where a simple constraint locks a set of particles onto a surface while the particles and the surface move. Then the constraints are used to make surfaces follow particles, and to make particles follow surfaces. Control points are also used for direct manipulation by

specifying particle motions, then solving for surface motion that maintains the constraint. For the purpose of sampling and rendering, the constraints are made to run in the other direction, creating floater particles that roam freely over the surface. Also, local repulsion is used to make floaters spread evenly across the surface. By varying the radius of repulsion dynamically, and implementing lifetimes for particles based on the local density, good sampling distributions can be achieved and maintained in case of deformations and changes in surface topology.

M. Pauly et al. [35] extend the multiresolution editing concepts, including geometry smoothing, simplification, and offset computation, to point clouds. [35] designs a hierarchical representation for point-based modeling. An editing method along with dynamic resampling techniques facilitates modeling operations at different scales and enables the user to effectively handle very large model deformations. More recently, M. Pauly et al. [36] presented a free-form shape modeling framework for point sampled geometry using the implicit MLS surface. By combining unstructured point clouds with the implicit surface definition of the moving least squares approximation, a hybrid geometry representation is generated. Thus allowing the capability to utilize the advantages of implicit and parametric surface models. Based on this they introduce a shape modeling system that enables the user to perform large constrained deformations as well as Boolean operations on arbitrarily shaped objects.

Guo et al. [37] present a haptic-based volumetric modeling framework, which is founded upon distance functions and certain physics-based modeling. By extending the idea of the

local reference domain in the moving least square (MLS) surface model to the construction of a local and global surface distance field, the incorporation of dynamic implicit volumetric model into deformation of the point-based geometry is implemented, which not only facilitates topology change but also affords dynamic sculpting and deformation. Along with that, integration is presented of the haptics-based dynamic sculpting and Boolean operations into the surface-modeling framework.

Our work in interactive physically based deformation of point-based 3D models using a haptic interface draws from previous work in mesh-free deformations [32] which uses an analytically approximate solution to a concentrated force acting in the interior of an infinite solid (Kelvin's problem [38][39]) This technique differs from earlier techniques for point-based models as it does not mandate an underlying surface definition, nor does it require distance field functions. It also seems to lend itself conveniently to interactive use in terms of coupling it with a haptic interface in order to achieve virtual sculpting using a motion-capture device such as an electronic glove to deform/modify a 3D point-based model.

Chapter 3

Background

We present here the mathematical background underlying the deformation procedure undertaken in the process of interactive deformation of 3D point-based models using an electronic glove or a haptic device. The analytical solutions are based on the solution of the Kelvin problem from elasticity. Deformation as well as analysis of surface force under stress is considered in order to provide suitable feedback information in case of using a haptic device. The mathematical analysis of deformation shall be studied under two contexts, namely, deformation with respect to the application of a single force on the object and deformation under the presence of multiple force points. This distinction is undertaken in order to suggest how the single force point based process is extended in order to incorporate multiple force points, as well as to realize the major differences in the two systems and their inherent computational requirements. Also, in the two scenarios, the implementation of physical techniques is varied; hence the description warrants an individualized study of the two cases. In the case of surface force, similar changes are applied, thus a glance at the procedure of achieving a surface force profile for the object under stress from a single force and from a set of forces has been undertaken in a single description. The adaptation of the mathematical and numerical procedures discussed in this chapter with respect to their use with a motion capture or haptic device shall be studied in chapter 4. This chapter solely presents the relevant work done prior to this thesis, and out adaptation of this work shall be discussed in the following chapters.

3.1 Computation of deformation (Single force)

The computation of physical deformation of an elastic object under the effect of a single force is undertaken by the use of the solution of the Kelvin problem. The Kelvin problem is a physically based analytical solution and hence its usage in the procedure for deformation provides with a foundation for intuitive deformation. There are, however, certain points which need to be modified and/or considered in order to adapt the solution to the deformation of 3D point based models, these points shall be discussed under the appropriate contexts in order to stress their importance in adapting the mechanism.

3.1.1 The Kelvin problem

A Kelvin problem can be stated as the problem of deformation of the interior of an infinite solid under the effect of a single concentrated force acting at a point within the solid. Discussed here is the Kelvin problem with reference to its specific adaptation and implementation in the context of this thesis.

Figure 1 shows a sphere object embedded within an infinite solid and being acted upon by a concentrated force "R" at the origin "O" in a random direction. Assuming that "Q" is a chosen point lying within this embedded sphere, the solution to the Kelvin problem

shall provide us with the displacement at point "Q" as a consequence of the force "R" at point "O" acting in a certain direction.

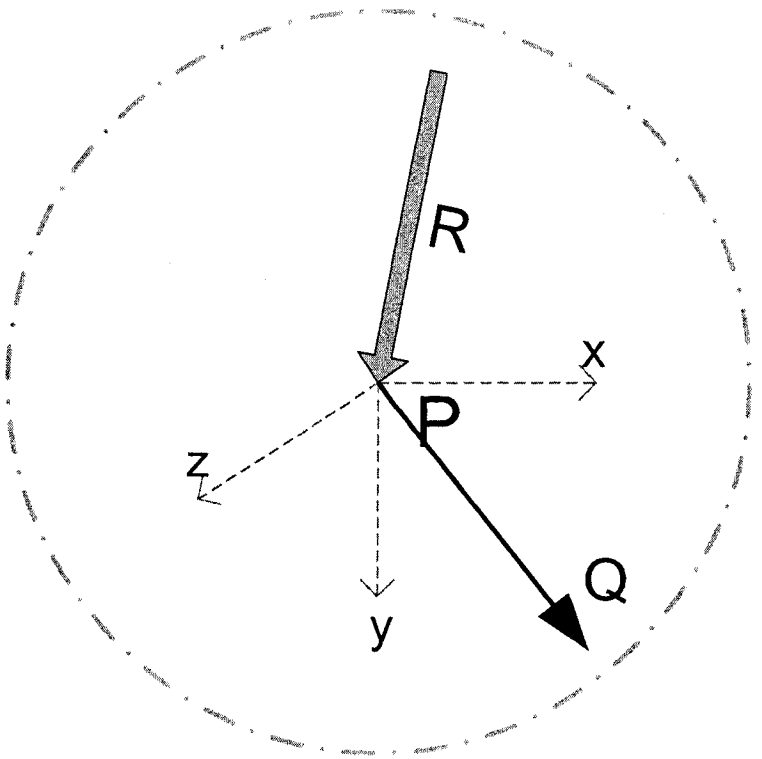

Figure 1: The Kelvin Problem

Let "$U_{ij}(P,Q)$" be the "d_j" component of the displacement at the point "Q" as a consequence of a concentrated force "R" exerted in direction "d_i" at point "P". Then according to the solution of the Kelvin problem, for "i = x, y, z" and "j = x, y, z" (with x, y and z being the standard notation for the Cartesian coordinate directions) we have:

$$U_{ij}(P,Q) = \frac{1}{16\pi G(1-\nu)R}\left[(3-4\nu)\delta_{ij} + \left\{\frac{d_i^{PQ} d_j^{PQ}}{R^2}\right\}\right]$$

Where,

Kronecker Delta :

$$\delta_{ij} = \begin{cases} 0 & i \neq j; \\ 1 & i = j; \end{cases}$$

$$d_i^{PQ} = d_i(Q) - d_i(P)$$

Displacement between "P" and "Q": $R = \| Q - P \|_2 = \sqrt{(d_x^{PQ})^2 + (d_y^{PQ})^2 + (d_z^{PQ})^2}$

G = Shear modulus of the object's material

and ν = Poisson's ratio.

A more detailed description as well as derivation of the Kelvin problem can be found in [38][39].

3.1.2 Numerical implementation of the displacement in Kelvin problem

Under the analytical solution of the Kelvin problem, a continuous volume definition is assumed. Thus, the process of discretizing the volume in order to be implemented for a point cloud (as in the case of 3D point based models) has to be undertaken. Also, resultant displacement with regards to a single force shall be the sum of the three components of forces i.e. The resultant displacement in direct "i" shall be the linear sum of all the displacement components in direction "i" at point "Q" due to the three components of the force at point "P".

Thus the displacement at any point "Q" as a consequence of an arbitrary directional force at point "P" can be defined as the sum of displacements due to the individual components of the force in the "x", "y" and "z" directions. Hence, if "$u_l(Q)$" is the "d_l" component of the displacement at point "Q", then

$$u_l(Q) = \sum_{k=x,y,z} U_{kl}(P,Q) \qquad l = x,y,z \text{ and } Q \in \Omega \qquad \ldots 1$$

It is also possible that due to the interactive requirements it may be necessary for the resultant solutions to abide by certain boundary conditions. These conditions can be enforced on the analytical solutions of the Kelvin problem by the insertion of a weight coefficient. The weight coefficients scale the resultant displacements at point "Q" in order to fulfill certain boundary conditions. That is, in order to meet the given boundary conditions the displacements at point "Q" are derived as"

$$u_l(Q) = \sum_{k=x,y,z} \alpha_k U_{kl}(P,Q) \qquad l = x,y,z \text{ and } Q \in \Omega \qquad \ldots 2$$

Where "α_k" are the weights with respect to each component of the force being applied. In order to formulate the equation (2) in a matrix form thereby forming a linear system of equations, equation (2) is written as:

$$\{u\} = [U]\{\alpha\} \qquad \ldots 3$$

With "u" being the values of corresponding displacements at the 3D data points, "U" being the corresponding analytical Kelvin solutions, and "α" being the weights of the respective displacement components.

Supposing that in a 3D point based model there are "m" 3D points which have a boundary condition defined for them, then the residual value "R" is defined as:

$$R = \sum_{l=x,y,z} \sum_{j=1}^{m} [u_l(Q_j) - u_l^0(Q_j)]^2 \qquad \ldots 4$$

Where "$u_i^0(Q_j)$" are the respective displacement boundary conditions defined at the corresponding 3D data points "Q_j". Minimizing the residual value "R" with respect to the weights "α_k" can determine the values of the given weights. This least squares minimization allows the process to meet the boundary constraints by finding coefficients that reduce the difference between the desired results and the analytical solutions. This minimization can be done using:

$$\frac{\partial R}{\partial \alpha_k} = 0 \qquad \text{with } k = x, y, z$$

This can be interpreted as a linear system of equation in the form:

$$[U]^T[U]\{\alpha\} = [U]^T\{u^0\} \qquad \ldots 5$$

Where "$\{u^0\}$" are the corresponding boundary conditions. The solution of equation (5) provides the values for the weights "α_k" and using these along with equation (2) the resultant displacement at any point on or within the 3D object due to a single force can be determined.

3.2 Computation of deformation (Multiple force points)

To undertake the physically based deformation of a 3D point based model under the effect of multiple forces, the principle of superposition allows for the convenient simplification of the system under consideration by dividing the resultant force at a

certain point into the linear sum of the effects on the point due to the various forces acting on the point. That is, assuming that "n" forces, namely "k_1", "k_2", "k_3" and so on till "k_n" are acting on a certain object, and the effects at a point (i.e. displacement or surface force) due to a force "f" is defined as "F(f)", then:

$$F(k_1 + k_2 + k_3 + \ldots + k_n) = F(k_1) + F(k_2) + F(k_3) + \ldots + F(k_n)$$

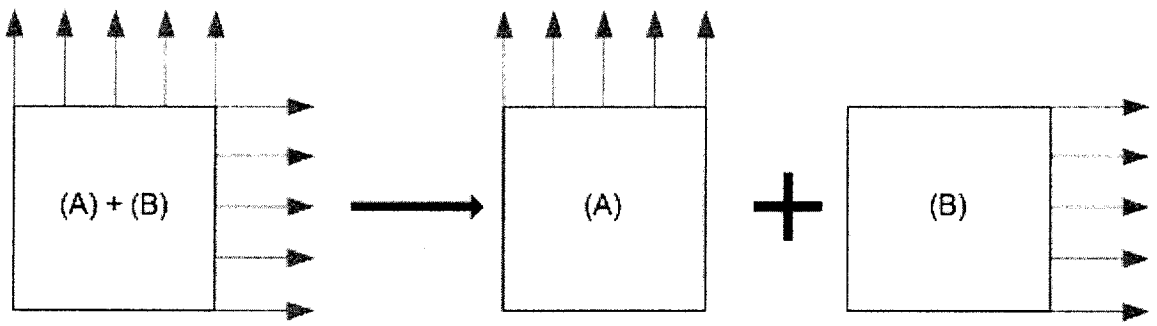

Figure 2: Principle of superposition

Now, let us say that "n" forces "P_1, P_2, P_3,..., P_n" are acting on the 3D point based model and at a point "Q" which belongs to the model, we know that the displacement due to a single force "P" is defined as (from equation (1)):

$$u_l(Q) = \sum_{k=x,y,z} U_{kl}(P,Q) \qquad l = x,y,z \text{ and } Q \epsilon \Omega$$

Then using the principle of superposition (figure 2.), we can find the resultant displacement in a direction "l" due to the "n" forces acting on the point "Q" as:

$$u_l(Q) = \sum_{i=1}^{n} \sum_{k=x,y,z} U_{kl}(P_i,Q) \qquad l = x,y,z \text{ and } Q \epsilon \Omega \qquad \ldots 6$$

As in section 3.1.2 we now assign weights "α_{ik}" to satisfy boundary conditions as defined. Thus we have,

$$u_l(Q) = \sum_{i=1}^{n} \sum_{k=x,y,z} \alpha_{ik} U_{kl}(P_i, Q) \quad l = x, y, z \text{ and } Q \in \Omega \qquad \ldots 7$$

Where "α_{ik}" is the weight belonging to component "k" of force "P_i".

Equation (7) can be written in a matrix notation as equation (3) and the weights "α_{ik}" to meet with boundary conditions "$\{u^0\}$" can now be calculated using equation (5).

3.3 Computation of surface force

The usage of haptic devices for the interactive physically based deformation of a 3D point based model requires a mechanism for providing a force feedback to the user. This force feedback shall allow the user to "feel" the object being deformed and as the process is physically based, it is imperative that the force felt by the user is directly proportional to the amount of deformation that the object has undertaken.

To this end, the computation of surface force is essential in order to provide for the complete immersion into the environment. The surface force is defined as the force which a point within the infinite sphere of the Kelvin problem applies to its neighboring regions. That is, under the stress of a force being applied at a certain point within the infinite sphere, the force that is propagated by a point after its displacement is the surface force.

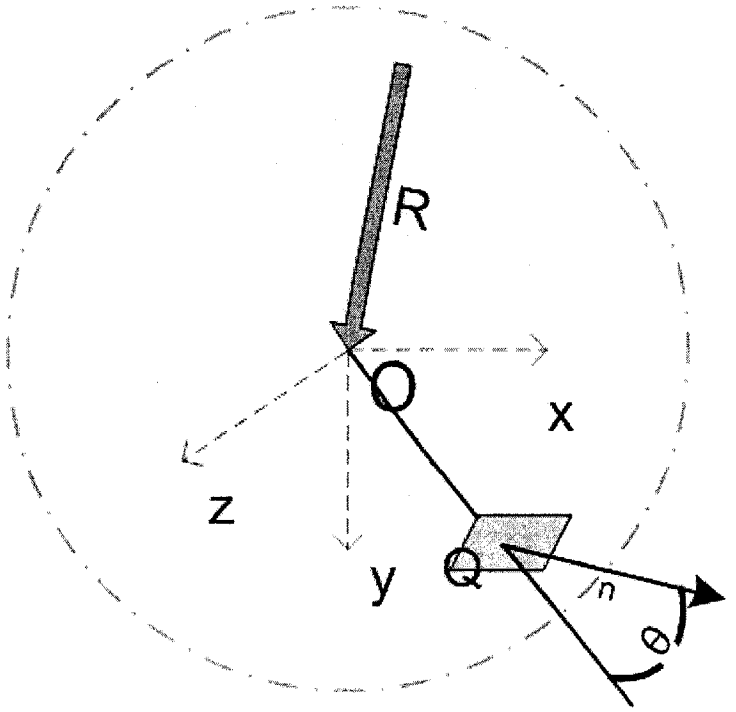

Figure 3: Computation of Surface force

Figure 3 shows a sphere object embedded within an infinite solid and being acted upon by a concentrated force "R" at the origin "O" in a random direction. Assuming that "Q" is a chosen point with normalized surface normal "$\hat{n}$" lying within this embedded sphere and the angle between the surface normal "$\hat{n}$" and the force location "O" to point "Q" vector i.e. "$(\vec{Q}-\vec{O})$" is "θ". The solution to the Kelvin problem for a surface force shall then provide us with the surface force at surface of point "Q" as a consequence of the force "R" at applied point "O" in a certain direction.

Let "$T_{ij}(P,Q,n)$" be the "d_j" component of the surface force at the point "Q", whose surface normal is "n", as a consequence of a single concentrated force "R" exerted in direction "d_i" at point "P". Then according to the solution for surface force of the Kelvin

problem, for "i = x, y, z" and "j = x, y, z" (with x, y and z being the standard notation for the Cartesian coordinate directions) we have:

$$T_{ij}(P,Q,\hat{n}) = \frac{\kappa}{R^2}\left[\left\{(1-2\nu)\delta_{ij} + 3\frac{d_i^{PQ}d_j^{PQ}}{R^2}\right\}\cos\theta - (1-2\nu)\left\{\frac{d_i^{PQ}n_j}{R} - \frac{d_j^{PQ}n_i}{R}\right\}\right] \quad \ldots 8$$

Where:

Kronecker Delta : $\delta_{ij} = \begin{cases} 0 & i \neq j; \\ 1 & i = j; \end{cases}$

$$d_i^{OQ} = d_i(Q) - d_i(P)$$

Displacement between "P" and "Q": $R = \| Q - P \|_2 = \sqrt{(d_x^{PQ})^2 + (d_y^{PQ})^2 + (d_z^{PQ})^2}$

$$\kappa = \frac{-1}{8\pi(1-\nu)} = \text{Constant}$$

$$\cos\theta = \frac{d_x^{PQ}}{R}n_x + \frac{d_y^{PQ}}{R}n_y + \frac{d_z^{PQ}}{R}n_z$$

and $\nu = \text{Poisson's ratio.}$

A more detailed description as well as derivation of the Kelvin problem can be found in [38][39].

3.3.1 Numerical implementation of the surface force in Kelvin problem

The process of numerically implementing the surface computation for the surface force in the Kelvin problem is analogous to the technique involved to implement the displacement

in the Kelvin problem. Also, resultant surface with regards to a single force shall be the sum of the three components of forces i.e. The resultant surface force in direct "i" shall be the linear sum of all the surface force components in direction "i" at point "Q" due to the three components of the force at point "P".

Thus the surface force at any point "Q" as a consequence of an arbitrary directional force at point "P" can be defined as the sum of surface forces due to the individual components of the force in the "x", "y" and "z" directions. Hence, if "$t_l(Q,\hat{n})$" is the "d_l" component of the surface force at point "Q" with normal "$\hat{n}$", then

$$t_l(Q,\hat{n}) = \sum_{k=x,y,z} T_{kl}(P,Q,\hat{n}) \qquad l = x,y,z \text{ and } Q \epsilon \Omega \qquad \ldots 9$$

It is also possible that due to the interactive requirements it may be necessary for the resultant solutions to abide by certain boundary conditions regarding surface forces. These conditions can be enforced on the analytical solutions of the Kelvin problem by the insertion of a weight coefficient. The weight coefficients scale the resultant surface force (and/or displacements) at point "Q" in order to fulfill the boundary conditions. That is, in order to meet the given boundary conditions the surface force at point "Q" is derived as"

$$t_l(Q,\hat{n}) = \sum_{k=x,y,z} \alpha_k T_{kl}(P,Q,\hat{n}) \quad l = x,y,z \text{ and } Q \epsilon \Omega \qquad \ldots 10$$

Where "α_k" are the weights with respect to each component of the force being applied. In order to formulate the equation (2) in a matrix form thereby forming a linear system of equations, equation (2) is written as:

$$\{t\} = [T]\{\alpha\} \qquad \ldots 11$$

With "t" being the values of corresponding surface forces at the 3D data points, "T" being the corresponding analytical Kelvin solutions, and "α" being the weights of the respective surface force components.

In the case of multiple force acting on the object, similar to equation (7), for "n" forces, the net resultant of the surface force can be computed as:

$$t_l(Q,\hat{n}) = \sum_{i=1}^{n} \sum_{k=x,y,z} \alpha_{ik} T_{kl}(P_i, Q, \hat{n}) \qquad l = x, y, z \text{ and } Q \in \Omega \quad \ldots 12$$

Where "α_{ik}" is the weight belonging to component "k" of force "P_i".

3.4 Computation of displacement and surface force with boundary conditions

Supposing that in a 3D point based model there are "mD" 3D points which have a displacement boundary conditions defined for them, and "mF" 3D points which have a surface force boundary condition defined, then the residual value "R" in this case is defined as:

$$R = \sum_{l=x,y,z} \sum_{j=1}^{mD} [u_l(Q_j) - u_l^0(Q_j)]^2 + \beta \sum_{l=x,y,z} \sum_{j=1}^{mF} [t_l(Q_j, \hat{n}) - t_l^0(Q_j)]^2 \quad \ldots 13$$

With "$t_l^0(Q_j)$" being the respective surface force boundary condition defined at the corresponding 3D data point "Q_j". "β" is defined as "ξ/E" with "ξ" being a positive constant and "E" being the Young's modulus. "β" is used to normalize the contributions

of forces and displacements in equation (13). Also, equations (7) and (12) can be represented together in a matrix form as:

$$\begin{Bmatrix} u \\ t \end{Bmatrix} = \begin{bmatrix} U \\ T \end{bmatrix} \{\alpha\} \qquad \ldots 14$$

Minimizing the residual value "R" with respect to the weights "α_k" can determine the values of the given weights. This least squares minimization allows the process to meet the boundary constraints by finding coefficients that reduce the difference between the desired results and the analytical solutions. This minimization can be done using:

$$\frac{\partial R}{\partial \alpha_{ik}} = 0 \qquad \text{with } k = x, y, z \text{ and i} = 1,2,3,\ldots,n$$

Now this can be interpreted as a linear system of equation in the form:

$$\begin{bmatrix} U \\ \beta.T \end{bmatrix}^T \begin{bmatrix} U \\ \beta.T \end{bmatrix} \{\alpha\} = \begin{bmatrix} U \\ \beta.T \end{bmatrix}^T \begin{Bmatrix} u^0 \\ \beta.t^0 \end{Bmatrix} \qquad \ldots 15$$

Where "$\{u^0\}$" and "$\{t^0\}$" are the corresponding boundary conditions. The solution of equation (15) provides the values for the weights "α_{ik}" and using these along with equation (14) the resultant displacement and surface forces for a set of 3D points can be determined.

Chapter 4

Adaptation and Interaction Design for Electronic Glove

In the previous chapter, we showed how the solutions to the Kelvin problem of elasticity can be adapted in order to achieve numerical solutions of displacements regarding individual points contained in a 3D point based model. In this chapter we continue the adaptation and the design of the solutions given in the previous chapter in order to enable the system to accept inputs from a motion capture device such as an electronic glove or a haptic interface.

The adaptation mechanism can be broken down into independently applicable processes which can then be suitably modified and implemented according to the motion capture devices being used. The methods defined in this chapter are conceptually imperative for the adaptation but provide enough freedom to be implemented within the framework of any motion capture system. The requirements for the following methods and their constraints and limitations have also been discussed along with the actual method implementation; this provides the system coherence and helps develop the adaptation naturally and logically. Furthermore, the interactivity and dependencies between the

various steps in the implementation and the point based 3D model along with the deformation mechanism has also been discussed where appropriate.

4.1 Adapting the motion capture input

In order to develop a physically based deformation system which takes data as input from an electronic glove, it is necessary to standardize the input data so as to optimize the techniques performance and reduce any noise errors which may arise due to the inherent electro-mechanical nature of the input device being used. Also, the high rates of data input which is provided by the electronic glove lends itself to redundant information with regards to minute position changes which might not be voluntary. For example, the data input rate for the electronic glove by Measurand Inc. has a data input rate exceeding 80 Hertz, this rate is most likely to be far superior than the required input rate which is essentially a real-time 3D editing rate of approximately 10-30 Hertz, and due to such high input rates, involuntary motions might also be perceived as motion. To this end, it is necessary to discretize the data in a manner which reduces the probability of accepting involuntary motions as inputs as well as reduces the total amount of data being input to a frequency which consists of minimally redundant information.

To solve the issues mentioned in this section, the input data is discretized. The discretization is undertaken on the basis of a minimum displacement offset for the point under consideration i.e. the point is perceived to move if and only if the displacement between the previous location of the point and the current location is greater than a

minimum offset "Δs". This solution to the issues of data input solves both the problems conveniently by avoiding the involuntary motions being perceived as actual motion by setting the value of "Δs" greater than the average noise in the equipment; and the issue of reducing the redundancies of data by ensuring that if and only if there is certain motion shall an input arrive at the system, thus the input rates exceeding 80 Hertz are rendered meaningless (while in case of quick motion, the complete capabilities of the equipment can be used). The size of "Δs" can be dependent on various parameters such as motion capture device quality (more noise implies a larger value for "Δs"), computational capabilities of the system (computational capability can be inversely proportional to the value of "Δs") etc. Once the adaptation of the input has been undertaken, the next step involves the registration of the motion capture data input with the 3D point based model in order to proceed with the deformation of the point based model. This registration and deformation are discussed in the following section.

4.2 Registration of motion capture data and point based model

Primarily, the process of registering the 3D point based model with the data input from a motion capture device can be defined as two independent processes, namely;

1) Placing both the 3D point based model and the motion capture point probe in the same 3D vector space.

2) Once the point probe of motion capture and the 3D point based model are in the same 3D vector space, detecting collisions between the pseudo-surface of the 3D point based model and the point probe as the point probe is moved using the data input from the motion capture device. Collision is said to occur when the point probe crosses the boundary of the object, which in turn triggers the mechanism of deformation in order to achieve an update which keeps the point probe on the "outside" of the object after the deformation.

The first of these is achieved by the applying various matrix transformations on the data input in order to match it with the scale and location of the 3D point based model. For example, if the motion capture device provides the inputs in terms of joint angles, then considering joint lengths and angles along with a stationary (or pre-located) pivot, the final positions of the finger tips can be converted into a 3D vector location, similarly if the point based model is large, the input can be scaled in order to achieve motions that can traverse the whole model in the range of the motion capture device. Other techniques such as special gestures can be used in cases where reorientation or repositioning is required with respect to the 3D point based model. Once both the 3D point based model and the motion capture driven point probe are placed in the same 3D vector space, the process of collision detection continued by deformation procedures can be undertaken.

4.2.1 Collision detection with point data

For the purpose of physically deforming a 3D point based model interactively via the input from an electronic glove, it is imperative to ensure that the motion capture driven

point probe(s) has interfered with the pseudo-surface represented by the 3D point based model. In circumstances where a volumetric point cloud is to be deformed, this surface can be represented by the out most set of the volumetric data, and any interference with this outermost set can be considered as interference with the point cloud itself. In the system being discussed it is assumed that all the interactivity which is undertaken from the motion capture comes solely from the motion of the finger tips of electronic glove or the haptic device. These finger tips can be modeled as point probes in the 3D vector space, and their motion being driven by the motion capture device input can be used to study the passage of the point probe through the pseudo surface represented by the 3D point based model.

The lack of an explicit surface representation being present warrants an improvisation in order to detect the passage of the point probe through the surface. While the essential concept of detecting the collision is similar, the presence of normal information on the individual points of the 3D point based model is treated with a slight difference while detecting collision. However, if no normal information is contained in the 3D point based model (such as models acquired from a 3D scanning device) then another variant of the collision detection test is undertaken. The two variants of the collision detection process are studied in the following two sections.

4.2.1.1 Collision with points containing normal information

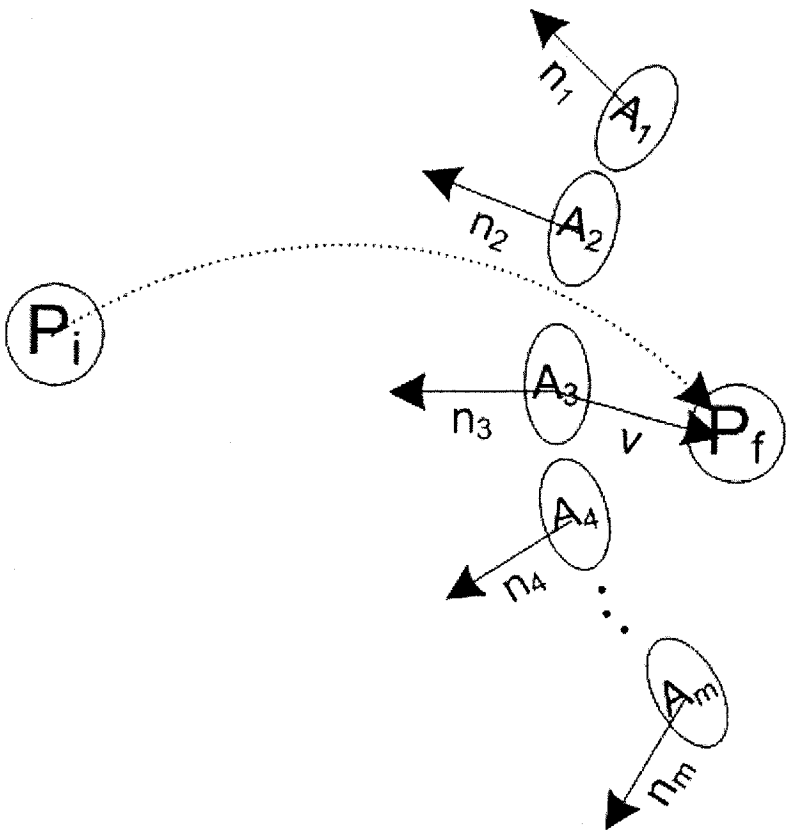

Figure 4: Collision detection with points containing normal information

In order to achieve collision detection of a motion capture driven point probe with respect to a 3D point based model which contains the normal information at each point, let us assume that the point probe "P" is moved from an initial position "P_i" to a final position "P_f" using the motion capture device (figure 4.). The 3D point models consists of "m" points "A_p" each containing a corresponding normal "n_p"; where "p" is the point number.

In order to detect if a collision with the pseudo-surface represented by the set of points "A_p" has taken place, first the nearest point "A_s" to "P_f" is searched, this is done by finding:

$$A_s = \min(\| P_f - A_p \|_2) \qquad p = 1, 2, 3, \ldots, m\text{-}1, m \qquad \ldots 1$$

Now to check if a collision has taken place, assuming that "P_i" was outside the pseudo-surface of the 3D point based model, the angle "θ" between the normal "n_s" at point "A_s" and the vector "v" is taken. Where "v" is defined as:

$$\vec{v} \equiv P_f - A_s \qquad \ldots 2$$

The angle "θ" can then be taken in the form of a cosine as follows:

$$\cos\theta = \frac{< n_s, \vec{v} >}{\| n_s \|_2 \| \vec{v} \|_2} \qquad \ldots 3$$

Where $< n_s, \vec{v} >$ is the dot product defined as:

$$< n_s, \vec{v} >= n_s^x \times \vec{v}^x + n_s^y \times \vec{v}^y + n_s^z \times \vec{v}^z \qquad \ldots 4$$

It can then be ascertained that the pseudo-surface for the 3D point based model has been crossed by the point probe at "P_f" if the value of "$\cos\theta$" is negative i.e. the angle "θ" is greater than 90^o.

4.2.1.2 Collision with points without normal information

Collision of the motion capture driven point probe with a 3D point based model which does not contain any normal information at the corresponding points is required in cases where the 3D point based model has been acquired using a 3D scanning device. In such cases the "approximate" location of the scanner or a reference point which is certain to lie outside the boundary of the 3D point based model, can be used to setup the vectors that are required in order to check for collisions between the 3D points based model and the motion capture driven point probe.

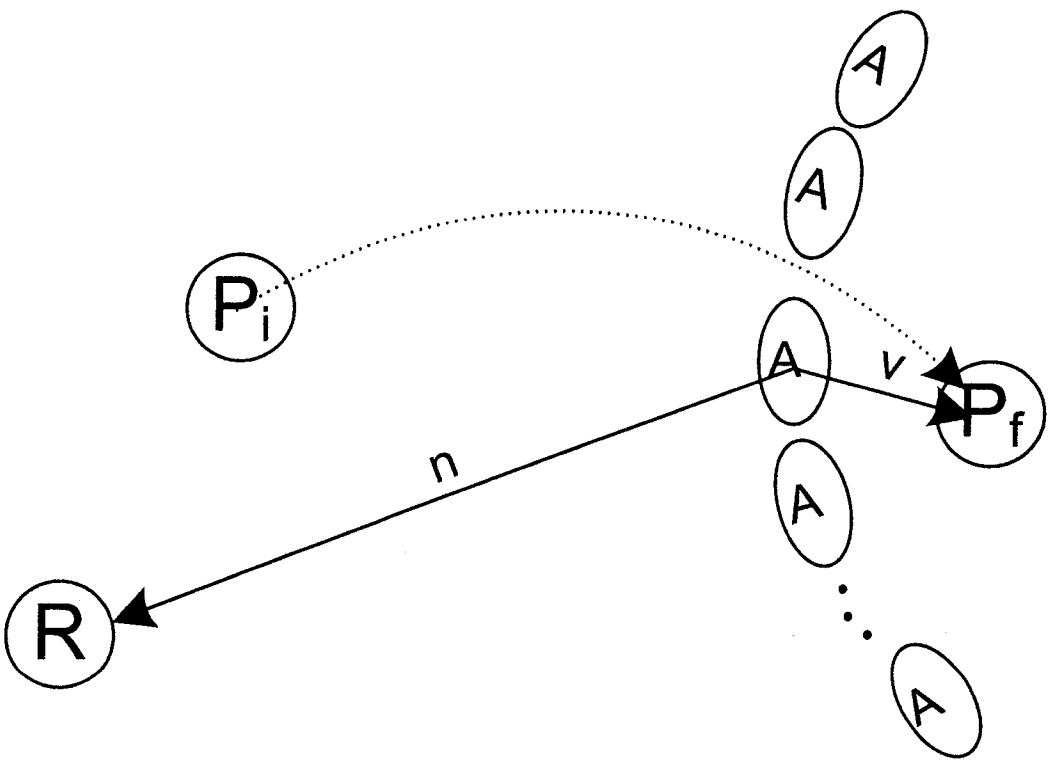

Figure 5: Collision detection with points without normal information

Let us assume that the motion capture drive point probe is moved from an initial position "P_i" to final position "P_f" with respect to a 3D point based model. After having found the point "A_s" which is the closest to the point "P_f" the following vectors are setup:

$$\vec{v} \equiv P_f - A_s$$

$$\vec{n} \equiv R - A_s \qquad \ldots 5$$

Where "R" is a predefined reference point which lies outside the 3D point based model. The cosine of the angle "θ" between the two vectors "$\vec{v}$" and "$\vec{n}$" is then calculated as follows:

$$\cos\theta = \frac{<\vec{n},\vec{v}>}{\|\vec{n}\|_2 \|\vec{v}\|_2} \qquad \ldots 6$$

Using the value obtained from equation (6) we can compute the angle between the two vectors and if the value of "cosθ" is negative, then the angle "θ" is greater than 90^o and thus a collision has occurred.

Once the occurrence of a collision has been ascertained, the process of setting up force vectors is undertaken, and following the force vectors the deformation is performed. In the next section, the technique used towards the setting up of force vectors along with the process of deformation is discussed.

4.3 Deformation

After the occurrence of a collision between the point probe(s) and 3D point based model, the process of computing the deformation can be broadly divided into three main tasks which are:

1) The dynamic setup of force vectors in order to simulate the force being applied by the use of the motion capture device onto the 3D point based model.
2) The dynamic setup of boundary conditions for a subset of the 3D point based model for assuring certain responses towards application of force.
3) Using the force vectors and the boundary conditions to compute the deformation of the 3D point based model as a consequence of the forces being applied on the model.

The primary requirement of these procedures is to adapt the data produced by the motion capture device into a stable mathematical formulation so that suitable parametric input values can be extracted from the motion capture data. It is also necessary that these adaptations are not computationally expensive and that a major portion of the computational capabilities are directed towards the computation of the deformation rather than towards the handling or modification of peripherals such as data adaptation and organization. Also, due to the numerical nature of the computations being performed, it is necessary to ensure stability in the system so as to avoid exceptional cases where the computation might lead to errors or singularities (ensuring stability is discussed later).

Further in this section a discussion about the procedures and consideration towards their implementation has been undertaken.

4.3.1 Setup of force vectors

The setup of the force vectors is required in order to obtain information regarding the origin of the force as well as information related to the direction and magnitude of the force. Both these information sets are required in order to be able to use the deformation technique provided and this information is needed to be extracted from the motion capture device's input data dynamically in order to realize an interactive system.

As the deformation technique being discussed is physically based, and the target system is intended to be intuitive i.e. model as closely as possible real world interactions, the setting up of the vectors need to be undertaken such that real world scenarios fit the setup to at least a certain degree. In the following two sections we present procedure used towards the dynamic setup of the force vectors and extraction of their origin, magnitude and directions relative to the 3D point based model.

4.3.1.1 Force vectors for inwards force

The origin of the force is organized such that it is one of the discrete point probe positions; this is done in order to ensure that the point probe stays outside the 3D point based model without exception. The reason and implementation of this shall be presented

subsequently. Along with the origin, the direction and magnitude of the force vector are computed by calculating the amount of potential penetration (not actual, as the surface is always deformed such that the point probe is always on the outside i.e. the deformation is performed such that the point probe cannot cross the boundary of the 3D point based model). Thus if the point probes final position is inside the 3D point based model by a small amount then the deformation force has a lesser magnitude as compared to the magnitude under the situation of greater penetration.

Let us assume that the motion capture driven point probe moves from the point "P_1" to "P_2" moving on to "P_3". Furthermore, a collision occurs at point "P_3" and the nearest point to it is a point "A_3" belonging to the 3D point based model (figure 6.)

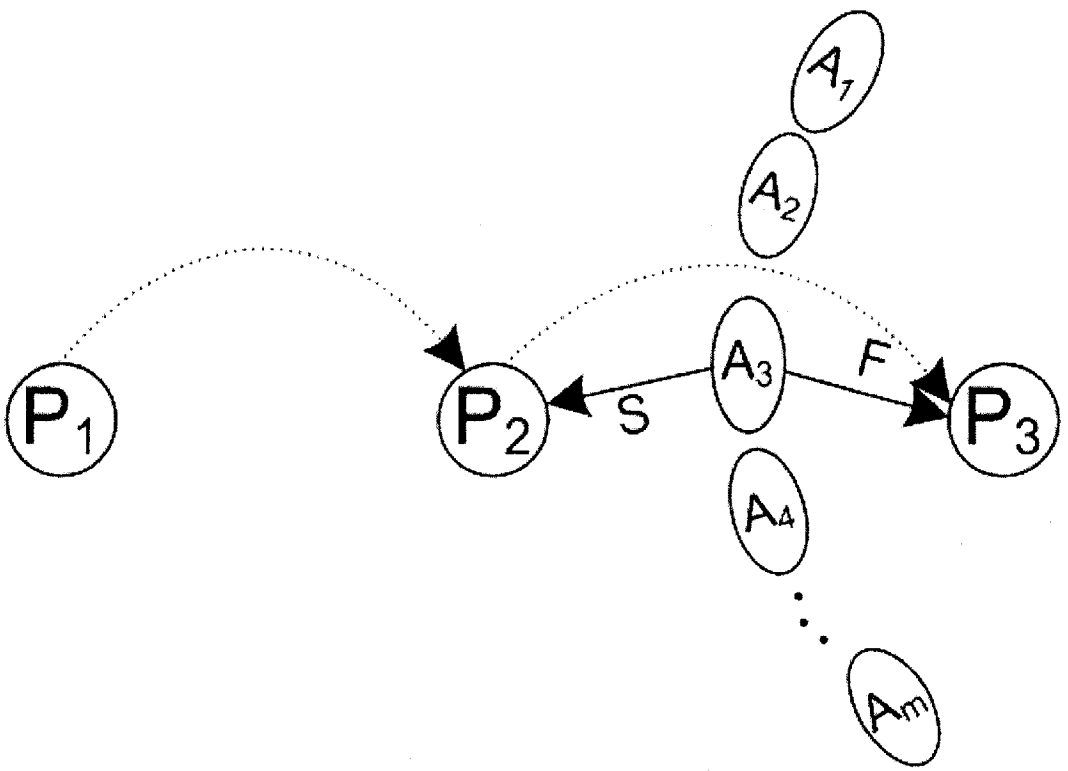

Figure 6: Setup of force vector

Then the force vector is computed as:

$$\vec{F} = P_3 - A_3 \qquad \ldots 7$$

The magnitude of this force vector "$\vec{F}$" is the magnitude of the force being applied, this solves the requirement for a force which is directly proportional to the amount of potential penetration that takes places inside the 3D point based model.

The origin of this force is set to be the position of the point probe just before the collision occurred; this is done due to the fact that it is imperative that the force be applied to the system from outside the 3D point based model. Thus the origin of the force "$\vec{F}$" is presumed to be at point "P_2". However, as the analytical solution to the Kelvin problem is singular at the force point, it is necessary to avoid this singularity (thus the need for keeping the point probe on the outside) i.e.

$$U(P,P) \Rightarrow R = 0 \Rightarrow U(P,P) = \infty \qquad \ldots 8$$

Due to the numerical nature of the computation, a very small value of "R" can also lead to a computational explosion, thus a minimum force origin offset of "Δd" is set. The displacement of the point probe prior to the collision with respect to "A_3" is computed, which give "S" and if "S" is less than the minimum offset "Δd", then the position of the point probe prior to the current force origin point is considered as the origin for the force. That is, in figure 6. :

$$\| P_2 - A_3 \|_2 \quad = \quad \| S \|_2 \begin{cases} \geq \Delta d \Rightarrow Origin(P_2) \\ < \Delta d \Rightarrow Origin(P_1) \end{cases} \qquad \ldots 9$$

Thus after finding the origin using "S" and the direction and magnitude of the force vector using "$\vec{F}$", the complete setup regarding the force vector can be undertaken.

4.3.1.2 Force vectors for outward force

The setup of force vectors with regards to an outward force is primarily analogous to the inwards force vector setup. The major difference being that the point probe is now constrained in such a manner so as to never be able to leave the confines of the 3D point based model.

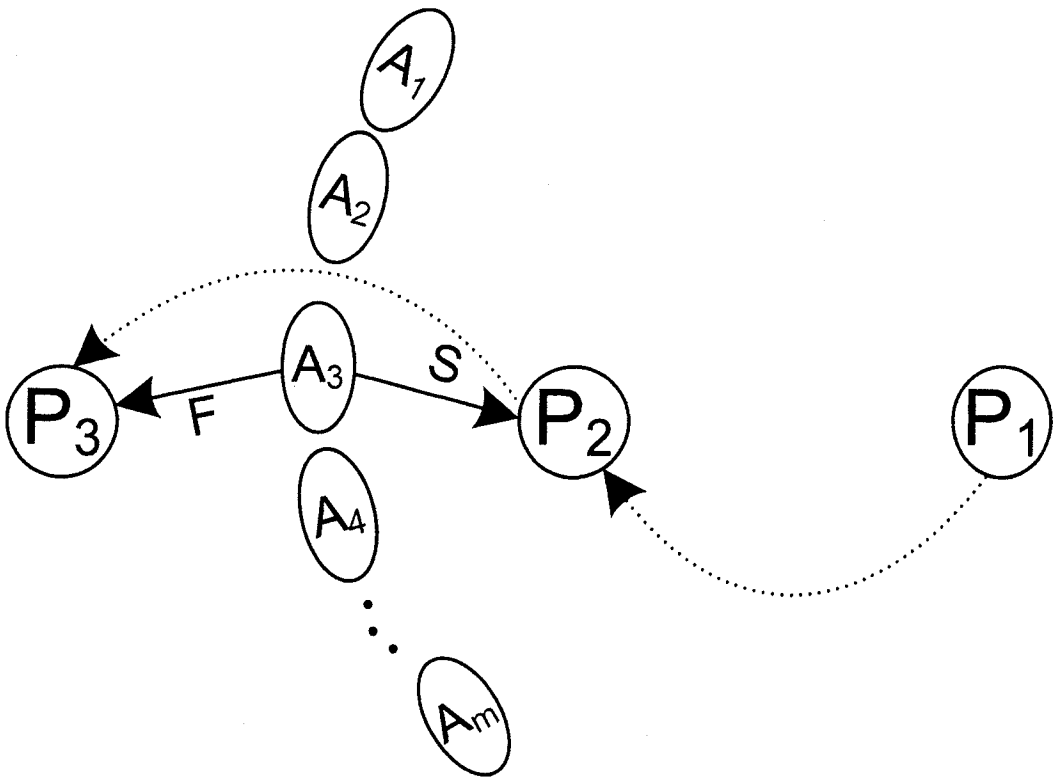

Figure 7: Setup of force vector for outwards force

The rest of the setup process is essentially mirrored to the process concerning an inwards force, as shown in figure 7. Yet, this is to be emphasised that where ever the point probe begins is initial location, it can not from there onwards, cross the pseudo-surface of the 3D point based model because the boundary conditions are implemented in such a way so as to ascertain that after the deformation has been computed due to a collision/penetration of the surface, the point probe is again outside the object boundary. This setup of boundary conditions ensure both stability against singularities and prevents the point probe from entering the deformed surface.

4.3.2 Setup of boundary conditions

In order to specify boundary conditions intuitively, specifications of conditions is undertaken such that it ensures that the point probe does not reside inside the 3D point based model's pseudo-surface after the deformation has taken place (alternatively for an outwards force, the point probe should not be outside the 3D point based model after the application of the deformation).

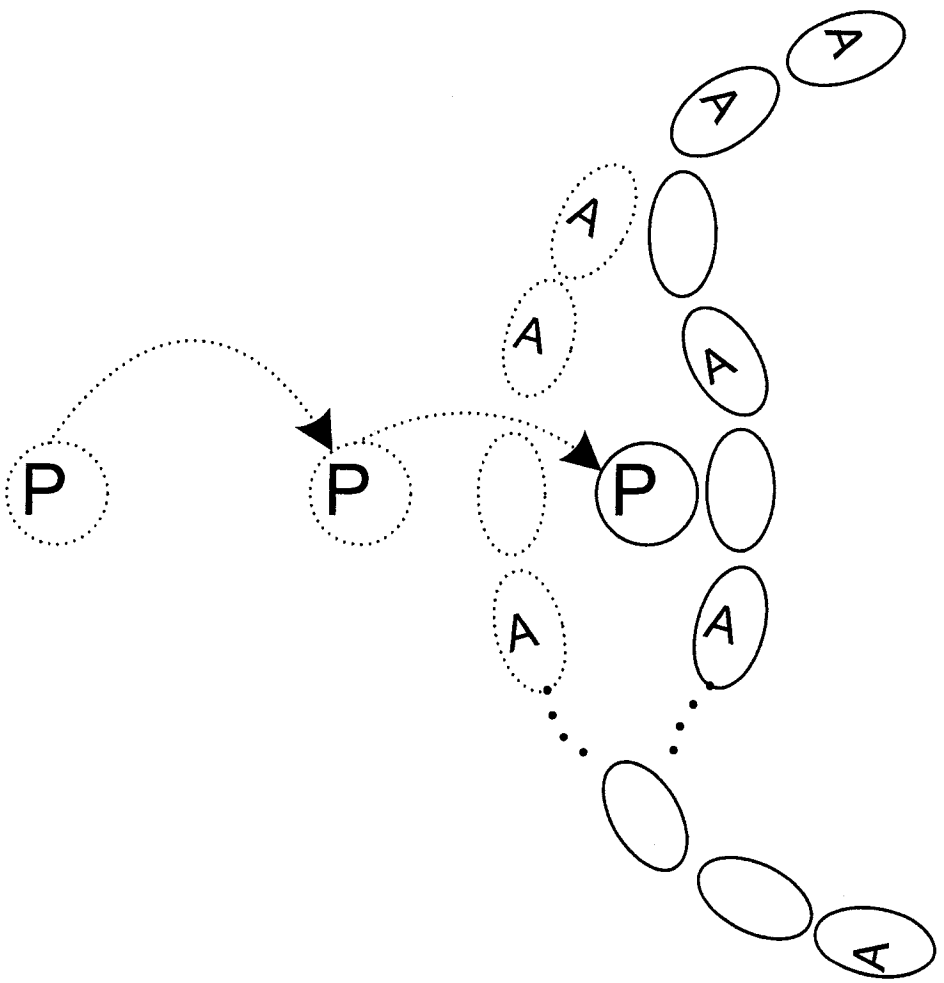

Figure 8: Setup of boundary conditions

This is achieved by defining the displacement boundary conditions for the points closest to the point of intersection (that is, the point "A_s" and the points lying in it's ε-neighborhood with ε being small enough so as to consider all the points within the ε-radius of "A_s" to have a similar position after deformation) and defining their boundary values such that the displacement after the deformation at each of these points is equal to the amount needed for the point probe at "P_3" to be outside (conversely inside for outwards forces) the 3D point based model (figure 8.). This rule can be conveniently defined as:

$$\forall A : [A \in A_s(\varepsilon)] \Rightarrow A^0 : [A^0 = P_3 - A] \qquad \ldots 10$$

Where "$A_s(\varepsilon)$" is the ε-neighborhood of "A_s" and "A^0" are the defined boundary conditions that move the corresponding points to the final position of the point probe, thereby ensuring that no singularities occur in solutions and the point probe does not cross the pseudo-surface of the 3D point based model after deformation is computed.

4.3.2.1 Setup of boundary conditions for multiple force points

In a scenario where multiple forces are being applied on the object simultaneously, it is important still to ensure that no singular solutions arise, towards this end another displacement constraint is enforced such that if the a collision occurs at a certain force point or points, then if a point probe is outside the object but within a certain distance

"Δr" from its nearest data point, then the nearest data point and the points in its ε-neighborhood are constrained such that after deformation the point probe remains outside the object.

In case where the point probe is farther than "Δr" and an extreme deformation is undertaken, and following the deformation, the point probe is found to be inside the pseudo-surface of the 3d point based model, then the point probe is moved to the nearest data point such that it resides outside the 3d point based data model. This ensures that under no circumstances does the point probe – 3D model deformation computation results in a singular solution and enhances the stability of the system. Also, the displacement "Δr" can be modified so as to allow the user the specific interactivity required for the deformation, for example, if "Δr" is large, then even if a point probe is outside the object by some distance, it will act as if it is constraining the motion of the object.

While multiple collisions simultaneously are treated as a linear sum of the solutions, and the techniques involved in one collision are superposed for the various instances.

4.3.3 Calculation of deformation and model updating

Once the initial setup of force vectors and boundary conditions has been undertaken with the following details:

Force Vector = F

Boundary Conditions = $\{u^0\}$

Following this the matrix of the corresponding analytical solutions to the Kelvin problem with respect to each of the individual points in the 3D point based data model is constructed, i.e. the matrix "[U]" is constructed using the force vector "F" on the data model.

To compute the values of the weights assigned to the force in order to comply with the requirements of the boundary constraints the system of linear equation is solved as follows:

$$[U]\{\alpha\} = \{u^0\} \qquad \ldots 11$$

Since "[U]" is most probably not a square matrix, it can be converted into the generic form of representing a linear system of equations as:

$$[U]^T[U]\{\alpha\} = [U]^T\{u^0\} \qquad \ldots 12$$

This can be written as:

$$[M]\{\alpha\} = \{m^0\} \qquad \ldots 13$$

This can then be solved by:

$$[M]^{-1}[M]\{\alpha\} = [M]^{-1}\{m^0\} \quad \ldots 15$$

$$\Rightarrow \{\alpha\} = [M]^{-1}\{m^0\} \quad \ldots 16$$

Using the values found for "$\{\alpha\}$" in the equation (16), we can calculate the displacement at each point in the 3D point based model with:

$$u_l(Q) = \sum_{k=x,y,z} F_k \alpha_k U_{kl}(P,Q) \quad l = x,y,z \text{ and } Q \epsilon \Omega \quad \ldots 17$$

Where

$u_l(Q)$: Displacement at point "Q" in direction "l"

F_k : Component of applied force in direction "k"

α_k : The "k" component of the weights computed for force "F"

$U_{kl}(P,Q)$: Analytical solution for displacement in direction "l" at point "Q" due to a unit force at "P" in direction "k".

4.3.3.1 Calculation of deformation for multiple force points

In order to compute the deformation due to multiple forces which have the following defined from the previous sections:

Force Vectors = F_i ("i"th force vector)

Boundary Conditions = $\{u^0\}$

Analogous to the previous section, the net displacement at the point "Q" due to the forces "F_i" can be computed as:

$$u_l(Q) = \sum_{i=1}^{n} \sum_{k=x,y,z} F_{ik} \alpha_{ik} U_{kl}(P_i, Q) \qquad l = x, y, z \text{ and } Q \in \Omega \qquad \ldots 18$$

Where

F_{ik} : The "k" component of the force vector "F_i".

α_{ik} : The "k" component of weight for the "i"th force vector.

4.4 Speed-up heuristics

As can be seen from the previous sections, which define the process of registering the motion capture device input with the 3D point based model followed by the computation of deformation, the complete procedure is computationally intensive. Therefore under situations dealing with a 3D point based model consisting of an extremely large set of points on a relatively slow computer (the experiments were undertaken on a 2.8GHz Quad core processor with 4GB of memory), it is possible that the processing capabilities of the system might not be able to provide the user with interactive response rates when dealing with the deformation of this large data set.

To this end two methods are being suggested which assist in the achievement of greater interactivity with regards to frame rates. The two techniques that follow are presented keeping in mind two scenarios with two requirements which a user of such a system might be unwilling to compromise on.

The first of the two techniques, namely, Reduced model set, is targeted towards the user's requirement which states that it is imperative for the user to be able to visualize the global effects of the deformation being applied while the local and extremely detailed modification are not of utmost concern under this scenario. The first technique can be used in the initial stages of a virtual sculpting process where the user is starting to build a model and is working on highly generic and global deformations which require him to visualize the overall presentation of the model while the exact intricate detail is not of any concern yet.

The second technique, namely, Localization of deformation, is primarily concerned with an user's need to be able to modify highly local regions of the 3D point based model without having to concern with the global effects that the deformation process might be having, this is needed in cases where a highly intricate modeling work is needed to be in a certain region of the 3D point based model while the other regions remain unchanged. In the following two sections we discuss in detail the two speed-up heuristics and present their implementation procedures and issues in detail.

4.4.1 Reduced model set

The speed-up heuristic presented in this section stresses solely on a user whose main requirement is the interactive visualization of global deformation due to the application of various forces on a large 3D point based model. This technique limits itself to the visualization of only global changes and local or intricate details about the deformation

are not of primary importance in this method for enhancing the interactivity in terms of frame rates.

To achieve interactive frame rates, the subset of the large 3D point based model is taken, thereby reducing the number of points for which deformation computation needs to be undertaken. However, it is not advisable to take the subset concentrated at a certain region of the 3D point based model because it shall then not be possible to visualize the effects of the deformation being undertaken in the regions where the 3D point density is very sparse.

It is recommended that the 3D point based model which is taken as a subset of the originally large 3D point based model should be a model with regular point density i.e. the concentration of points should be avoided and a regular distribution is more appropriate. This allows the user to visualize an overall global deformation and interact with the model intuitively. This session of interaction can be saved as a motion capture input and can then be applied off-line to the original large 3D point based model to achieve the exact deformations which had been noticed with the sub set model. However, in this scenario, the locally intricate details cannot be visualized and thus any predication about their behavior is out of context.

4.4.2 Localization of deformation

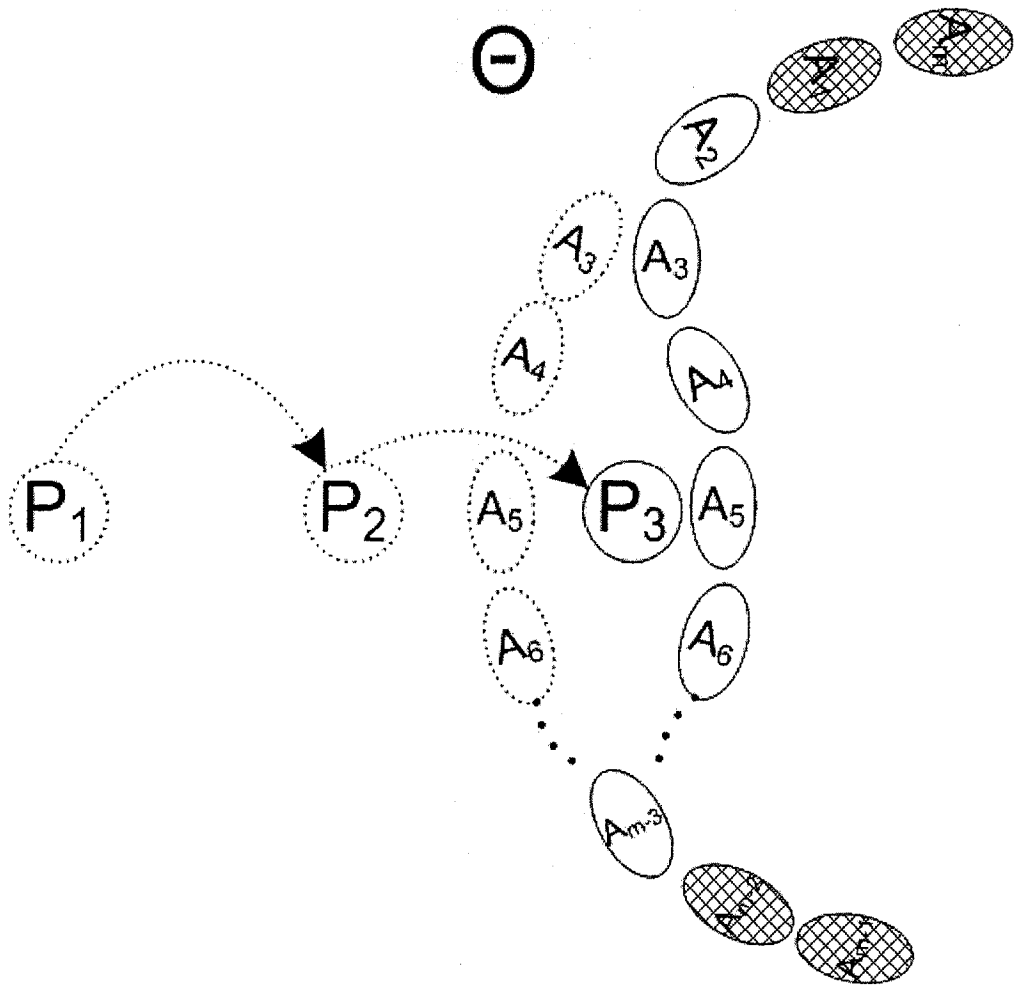

Figure 9: Localization of deformation

The speed-up heuristic discussed in this section is mainly targeted towards a more details deformation requirement by the user rather than global changes, thus in this case, it is important to be able to visually update the deformation of the high resolution 3D point based model interactively, and therefore the reduced point set technique does not suffice for this requirement. However, due to the need for high resolution only at a certain place in the 3D point based model it can safely be assumed that the user does not need to make

any global changes as a result of these highly intricate local deformations. Thus, it is possible to only update and change the locally high resolution model subset of the region that needs to be modified while all the while keeping the rest of the model as static. This can be done by the use of boundary constraints to ensure no "tearing" of the surface takes place.

Assume that the region "Θ" in figure 9 is the region that is required to be deformed while being able to visually interact with the high resolution definition of the 3D point based model. Then it is possible to set the boundary constraints of all the points lying within a certain ε-neighborhood of the boundary of the region "Θ" such that they are non-moving after the deformation. That is, in figure 9, the displacement boundary condition for the points A_1, A_m and A_{m-1}, A_{m-2} is set to zero. This ensures that these points do not move after the deformation has occurred.

Following the setup of boundary constraints for the "Θ" region, the computation for deformation is only undertaken for the points lying within the "Θ" region and any points lying out side "Θ" are not deformed. Thus the computation required is directly proportional now to the size of the region "Θ" and the overall size of the 3D point based model is not of any concern. This also leads to a localization of the deformation as only local effects can be modeled in this manner. It is also to be noted that large amounts of deformation in the localized region might lead to the "tearing" of the 3D point model's pseudo surface and might need surface reconstruction or other similar processes to be applied in that case.

4.5 Considerations/Discussion

One of the points that need to be described in passing is the offline computation mechanism which needs to be implemented in order to achieve deformation of high resolution 3D point based models. As it is an essential requirement for the reduced model set technique it is imperative that the motion capture data be saved during interaction and later use the saved data itself to apply it off-line to the high resolution model.

However, to optimize the performance in the offline mode of the system, it is possible to neglect any collision detections that might have taken place by keeping a track of the collisions and force vectors that occurred during the interactive session and those points and forces can then be applied without any computational overheads to the original high resolution point based model. Therefore in the offline mode a high degree of computation can be reduced by this technique and can thereby reduce the time required to implement a motion capture deformation sequence on a high resolution 3D point based model. The decrease in time is mainly due to the lack of collision detection with the model as we have ensure during the saving of the motion capture data that the only points that are saved are the collision points and the force vectors which can then be directly applied to the high resolution model.

Chapter 5

Implementation and Results

In this chapter we present a few illustrative examples of interactive deformation using the implementation of the technique described in the previous chapters. An analysis of various deformation mechanism and their results and limitations are discussed along with the setup parameters and considerations that had to be made in order to achieve intuitive as well and physically realistic results. The examples presented here for the deformation/editing of 3D point based models using an electronic motion capture device input have been broadly classified into six major categories of deforming/editing, which are as follows:

1) Applying an inwards force on the model i.e. pushing in the points on the model in order to edit features of a 3D point based model.
2) Applying a force which is outwards with respect to the 3D point based model, in this case the result is that of pinching or pulling out points from the model.
3) Applying multiple forces on the object simultaneously.
4) Applying an extreme deformation which results in the tearing of the pseudo-surface of the 3D point based model.
5) First speed-up heuristic: Reduced model set - in order to achieve global deformation with regards to large data models.

6) Second speed-up heuristic: Localized deformation – in order to achieve local deformations with minimal global effects and allowing the user to visualize the high resolution model interactively.

Generally the implementation of the deformation process was done on a two dual core processor system of 2.19 GHz and 2 GB of memory. The code was converted to the executable format using the Matlab 2007b compiler with the executable requiring the appropriate environment and data inputs from the motion capture device. For a motion capture device the electronic glove from Measurand Inc. (ShapeGlove) was used (figure 10.) This glove provides inputs in the form of the Cartesian coordinate positions of the joints of the hand i.e. nineteen tracking point data is received for the joints on the hand. Out of these nineteen points only the finger tip locations along with their orientation and positioning were considered as it was assumed that the deforming/editing process shall only be undertaken by the "touch" of the finger tips rather than the remaining parts of the user's hands.

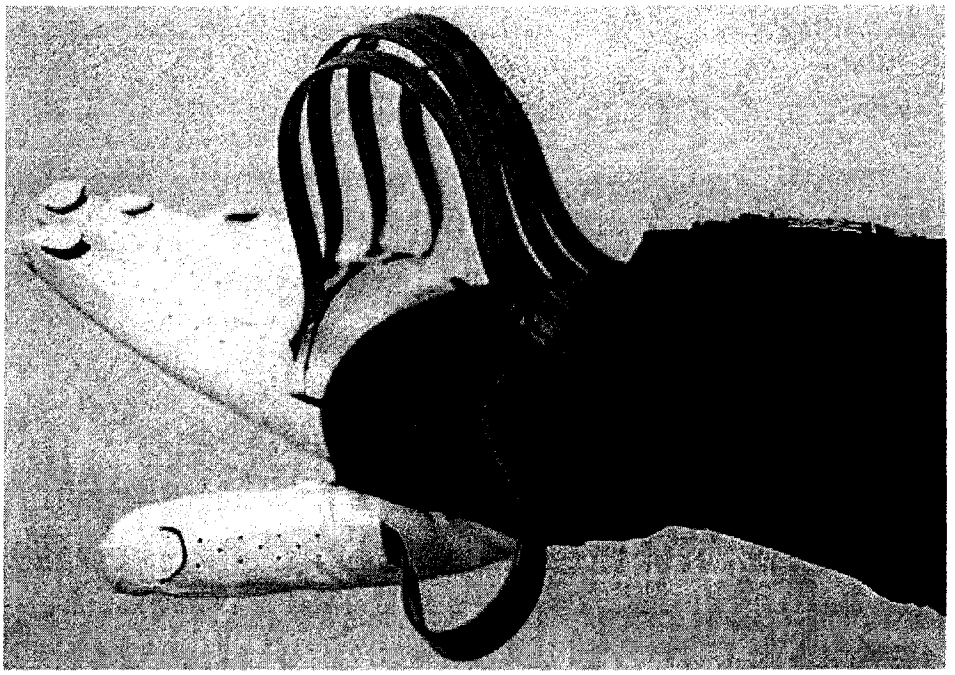

Figure 10: An electronic glove from Measurand Inc.

5.1 Inwards force

For the application of an inwards force on a 3D point based model it was necessary to setup up initial parameters in order to achieve physically based deformation with respect to the data input that were being received from the motion capture device. The implementation of the inwards force process was done using the following parameters:

Physical Parameters :

Shear Modulus(G)	3×10^{-5}
Poisson's Ratio(v)	5×10^{-1}
Young's Modulus(E)	1×10^{-2}

The parameters used to discretize the data inputs as well as to avoid singularities were as follows (in the scale of the model coordinates):

Offset Parameters :

$\Delta s \quad 1 \times 10^{-1}$

$\Delta d \quad 1 \times 10^{-2}$

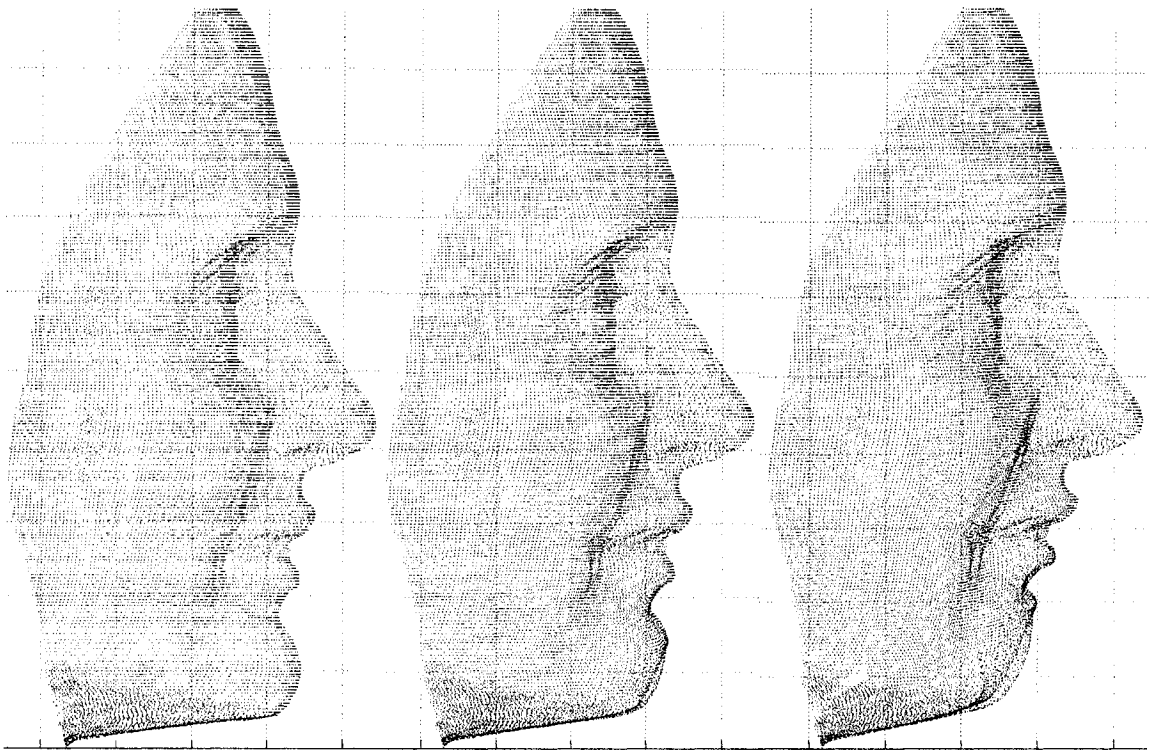

Figure 11: Editing the chin of a point based face model

Figure 11 shows the adjustment of the chin of the 3D point based model of a Face which consists of 40,800 points along with their color (RGB) and normal information. The deformation is undertaken in 60 collision computation frames and has been undertaken offline due to the large size of the data. The global effects were studied using the reduced Model set heuristic by using a 10% in point density model for interactive deformation

and then the motion capture data was applied to the original high resolution model. The computation time for the deformation shown in figure 11 ranged to about 1/24 frames per second for the high resolution model and 5 frames per second for reduced model set system.

Figure 12: Deforming the Nose of a point based face model

Figure 12 above presents the process of pushing in the nose of the 3D point based model of a face. The model consists of 40,800 points along with color information and normal information. The nose deformation was undertaken in 42 computation frames offline and the reduced model set with 10% density was computed at a frame rate of 5 frames per second for interactivity. The high resolution deformation produced a frame rate of 1/24 frames per second.

Figure 13 below presents the deformation of the Igea model which consists of 134,345 points with each point contain color information (RGB) and normal information. The back of the model is pushed in using the motion capture input as reference data and the deformation is continuous and smooth with the intuitive response of elasticity for the model. The deformation takes 85 collisions to process and the motion capture glove data is provided to the high resolution model offline. The 85 frames are computed at the rate of 1/175 frames per second due to the extremely large data set for the Igea model.

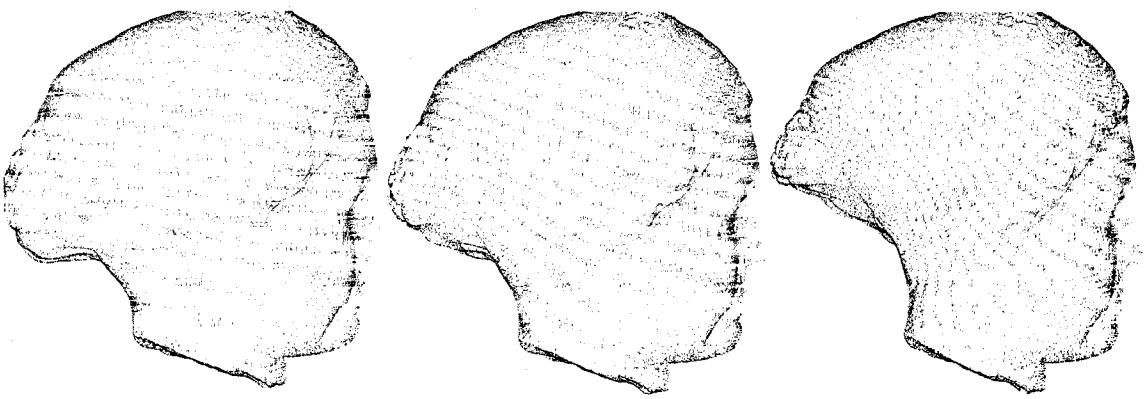

Figure 13: Editing the Igea point based model

Figure 14 shows an overlay of the initial and final states of the Igea model for the back deformation. It also shows the comparison in the global effects of the deformation undertaken. These global effects are necessary for the physically based behavior of the deformation process as they produce the effect of partial volume conservation during the deformation process of the 3D point based model.

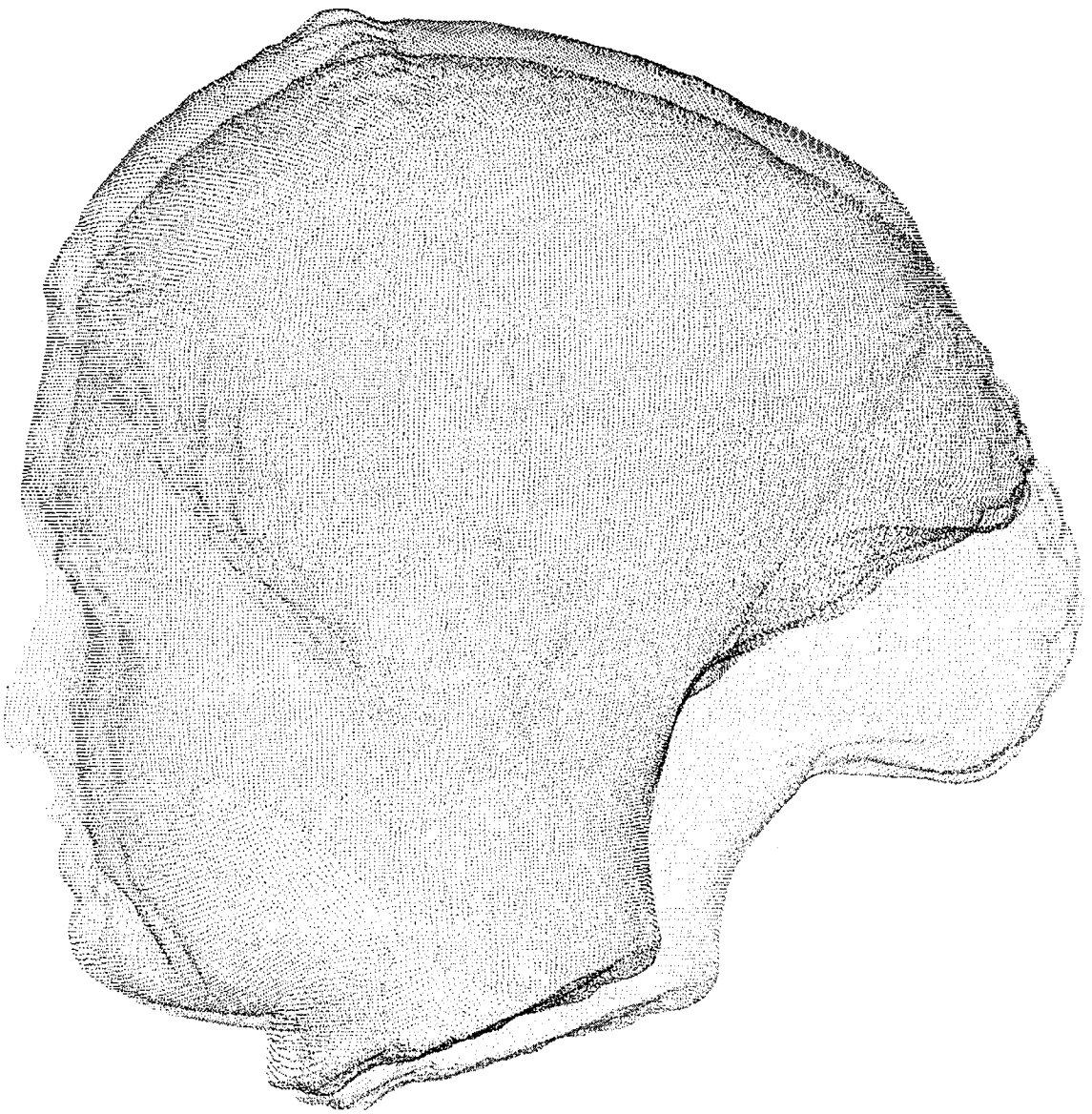

Figure 14: Comparision of Initial and Final positions of the Ideal model

5.2 Outwards force

For the application of an outwards force on a 3D point based model it was necessary to setup up initial parameters in order to achieve physically based deformation with respect to the data input that were being received from the motion capture device. The implementation of the inwards force process was done using the following parameters:

Physical Parameters :

Shear Modulus(G)	3×10^{-5}
Poisson's Ration(v)	5×10^{-1}
Young's Modulus(E)	1×10^{-2}

The parameters used to discretize the data inputs as well as to avoid singularities were as follows:

Offset Parameters :

Δs	1×10^{-1}
Δd	1×10^{-2}

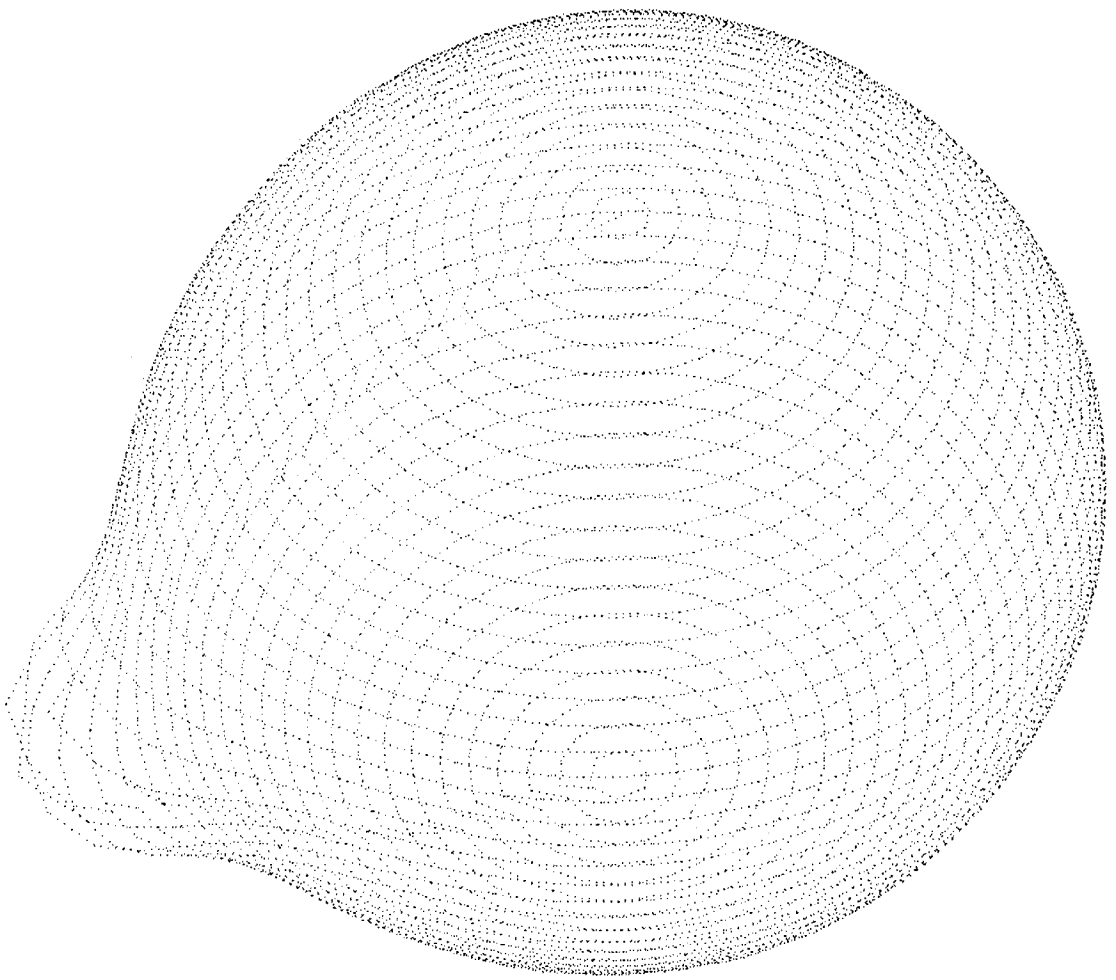

Figure 15: Illustrative protrusion of a sphere 3D point based model

Figure 15 shows an illustrative example of the pulling out or the pseudo-surface of the point based sphere. As can be seen the protrusion is smooth and behaves in a physically based manner to allow for intuitive interaction in terms of protrusions. The sphere model contains 3203 points and was interactively deformed at a rate of 18 collision frames per second.

Figure 16: Pulling out the nose in 3D point based model of a face.

Figure 16 presents the extrusion of the nose of a point based model of a face which was undertaken offline due to the large size of the model. The face model consists of 40,800 points and each point consists of color (RGB) as well as normal information. The deformation is undertaken in 92 collision computation frames. The global effects were studied interactively using the reduced Model set heuristic by using a 10% in point density model for interactive deformation and then the motion capture data was applied to the original high resolution model. The computation time for the deformation shown in Figure 16 ranged to about 1/24 frames per second for the high resolution model and 5 frames per second for reduced model set system.

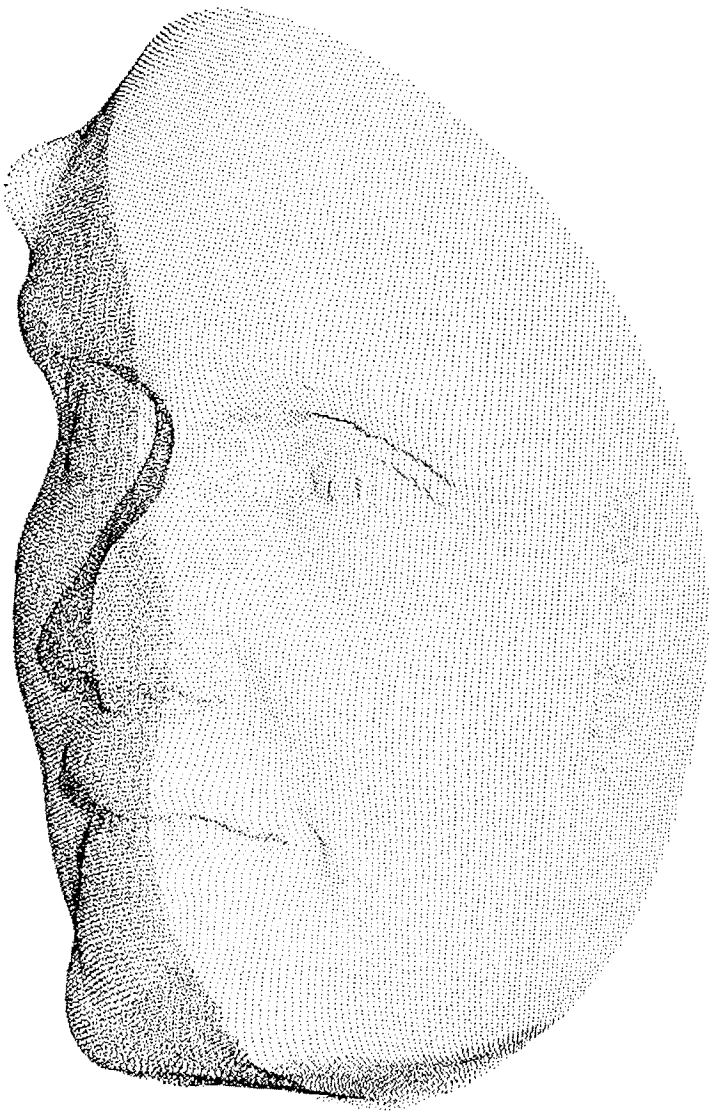

Figure 17: Protruding the forehead of the face model

Figure 17 above presents the process of pulling out a section of the forehead of the 3D point based model of a face. The model consists of 40,800 points along with color information and normal information. The nose deformation was undertaken in 47 computation frames offline and the reduced model set with 10% density was computed at a frame rate of 5 frames per second for interactivity. The high resolution deformation produced a frame rate of 1/24 frames per second.

5.3 Multiple force points

For the application of a multiple inwards forces on a 3D point based model it was necessary to setup up initial parameters with regards to the multiple force points which are derived from the motion capture device. This is undertaken to achieve physically based deformation with respect to the data input that were being received from the motion capture device. The implementation of the inwards force process was done using the following parameters:

Physical Parameters :

Shear Modulus(G)	2.5×10^{-5}
Poisson's Ration(v)	5×10^{-1}
Young's Modulus(E)	1×10^{-2}

The parameters used to discretize the data inputs as well as to avoid singularities were as follows:

Offset Parameters :

Δs	1×10^{-1}
Δd	1×10^{-2}
Δr	1×10^{-1}

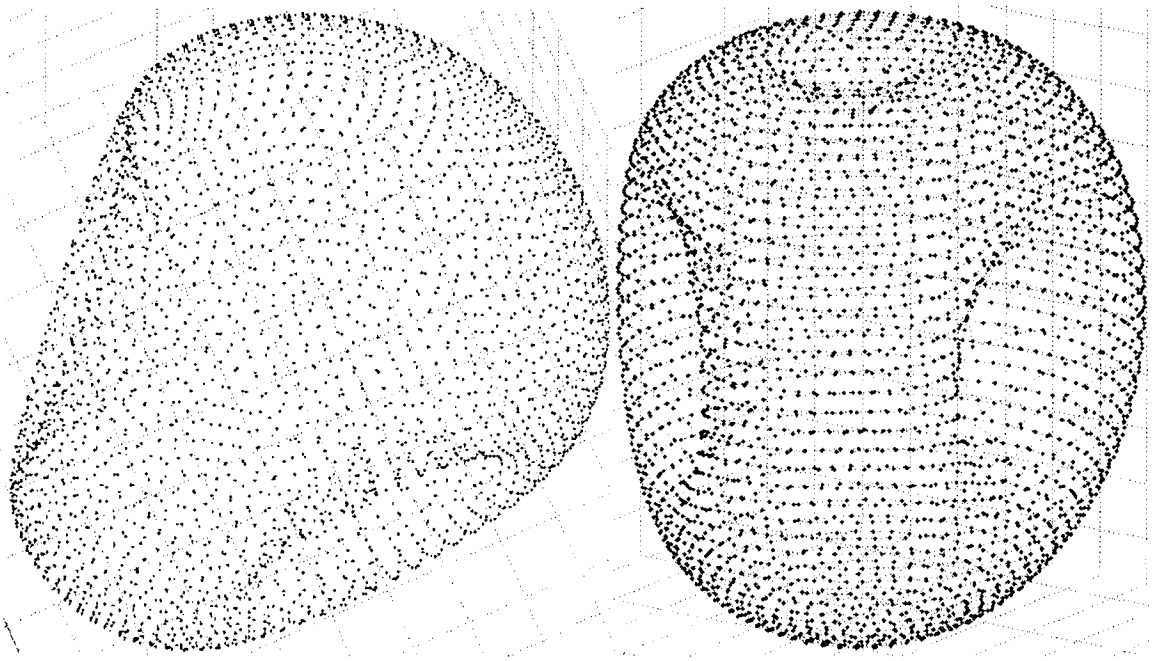

Figure 18: Deformation of a sphere using two and three force points simultaneously

Figure 18 above shows an illustrative example of using multiple forces simultaneously on a model of a sphere which consists of 3203 points with the color (RGB) and normal information. The two simultaneous forces model in the Figure 8 (left) was interactively modified at a rate of 16 frames per second with 70 collision frames. The three simultaneous forces model in Figure 8 (right) was interactively modified at a rate of 15 frames per second. Figure 8 shows the effect of pressing the object from two or three sides using simultaneous forces and elongation in the direction normal to the forces is clearly visible in the figure.

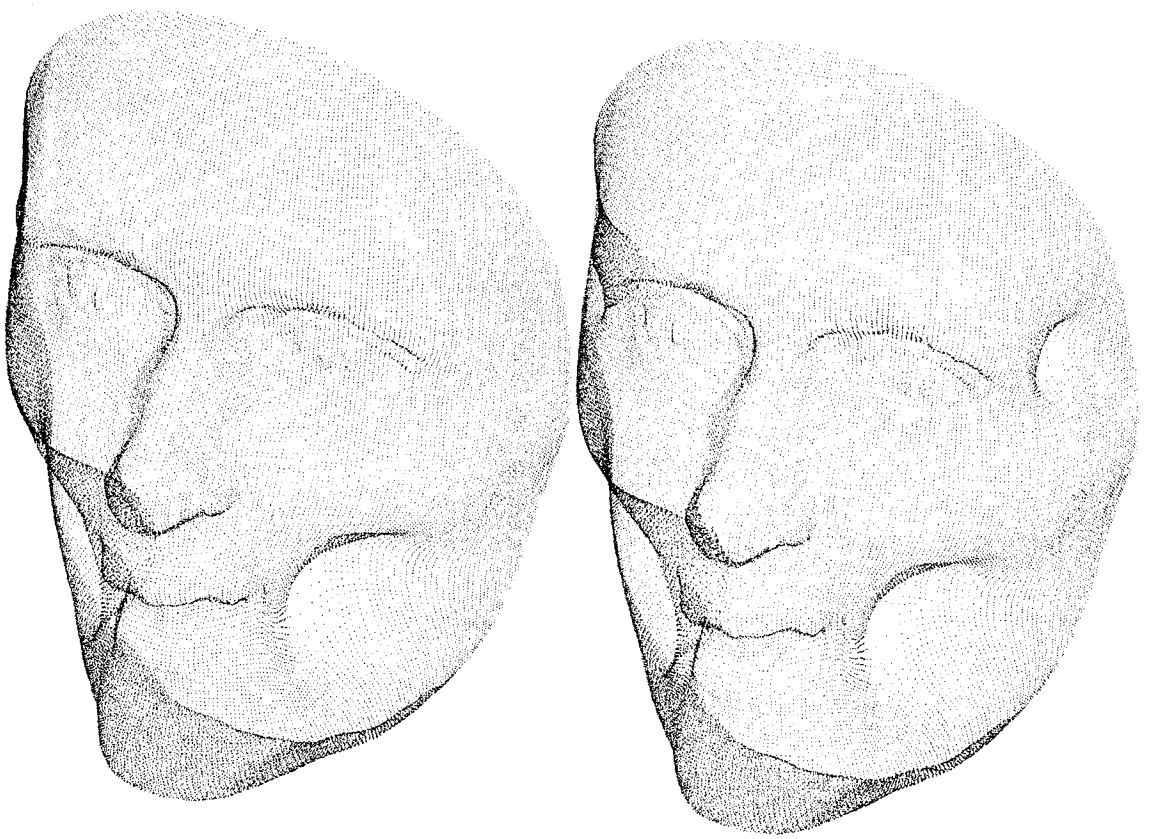

Figure 19: Editing of the face model using multiple force points

Figure 19 above shows the simultaneous deformation of the face 3D point based model. The model consists of 40,800 points along with color (RGB) and normal information. The simultaneous deformation was undertaken offline for the high resolution model at a rate of 1/26 frames per second for 36 collision frames. The reduced model set with 10% point density was deformed interactively with at a rate of 4 frames per second. The forehead deformations in figure 9 (right) took 8 collision frames to produce the effect.

5.4 Extreme deformation

With the deformation process being studied earlier, it is possible to undertake extreme deformations while interacting with the 3D point based model. These extreme deformation result in the "tearing" of the pseudo-surface of the 3D point based model. While the results are still physically based an intuitively true, the lack of point density renders possibilities of "holes" in the model.

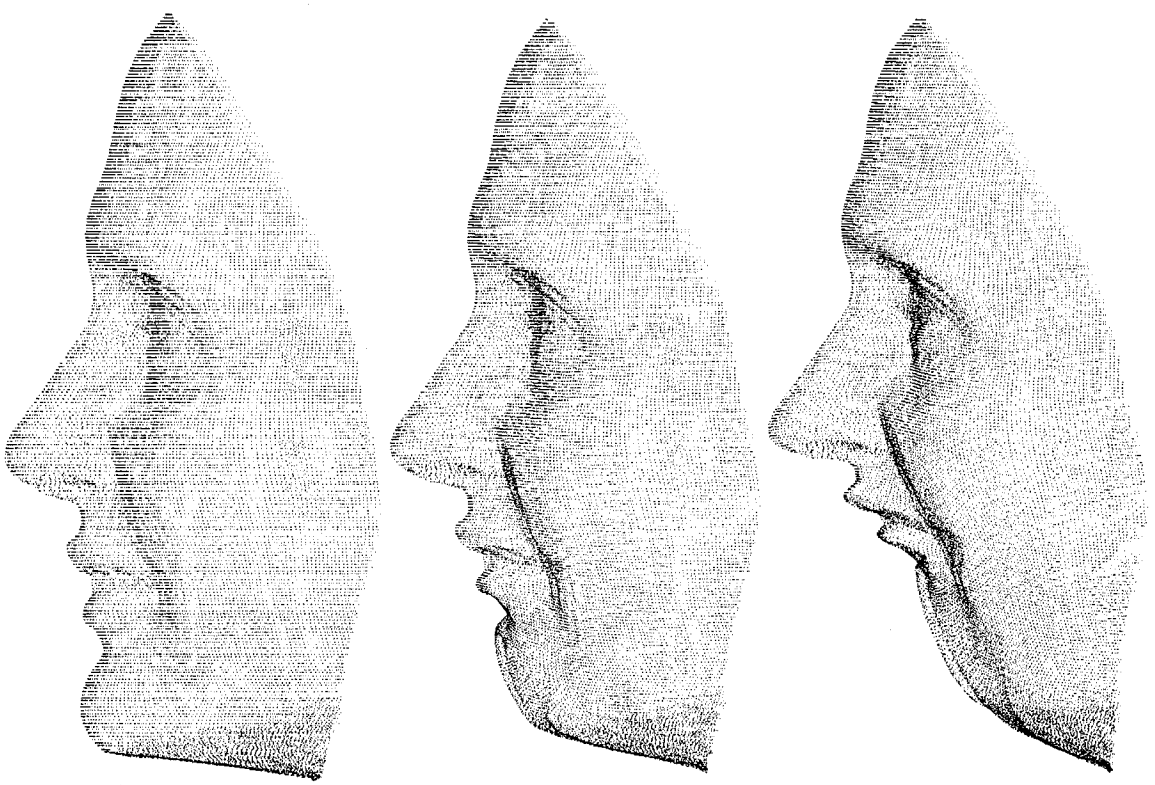

Figure 20: An example of extreme deformation of the face model

Figure 20 above shows the effect of an extreme deformation of the chin in the 3D point based model of a face. The collision frame count is 190 and the deformation leads to the "tearing" of the chin portion with the distance between individual points is enlarged

sufficiently so as to seem that "holes" are present in the model. The 170 frames were computed offline at a rate of 1/24 frames per second with the model consisting of 40,800 points.

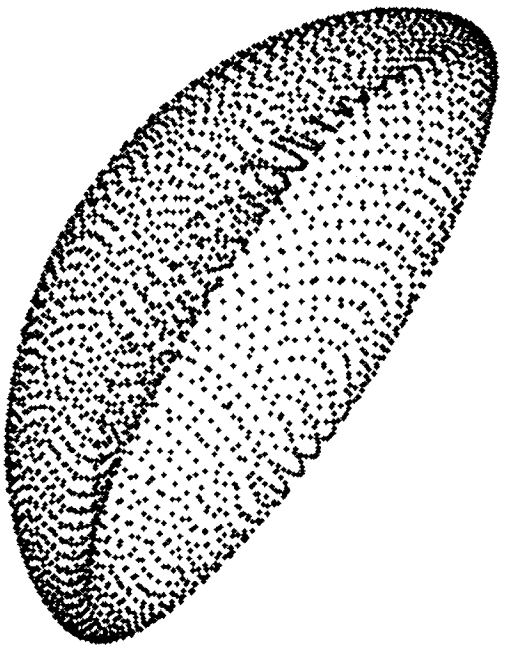

Figure 21: An extremely deformed sphere

Figure 21 shows the physically based results of even an extremely large deformation. The deformation is undertaken with 340 collision frames interactively at a rate of 18 frames per second. The sphere consisted of 3203 points and the model even after such an extreme deformation results in an intuitively reasonable solution. The sphere after such a deformation consists in reduced point density close to the point of application of the force and as such a "tearing" of the pseudo-surface of the 3D point based model occurs under such scenarios.

5.5 Reduced model set samples

In order to achieve interactivity with respect to globally deforming really high resolution 3D point based models it is necessary to visually process a reduced model set for the high resolution model by taking a subset of the data with reduced density. This provides the user with an illustration of the overall effects and the motion capture device inputs can then be mapped offline to then high resolution model.

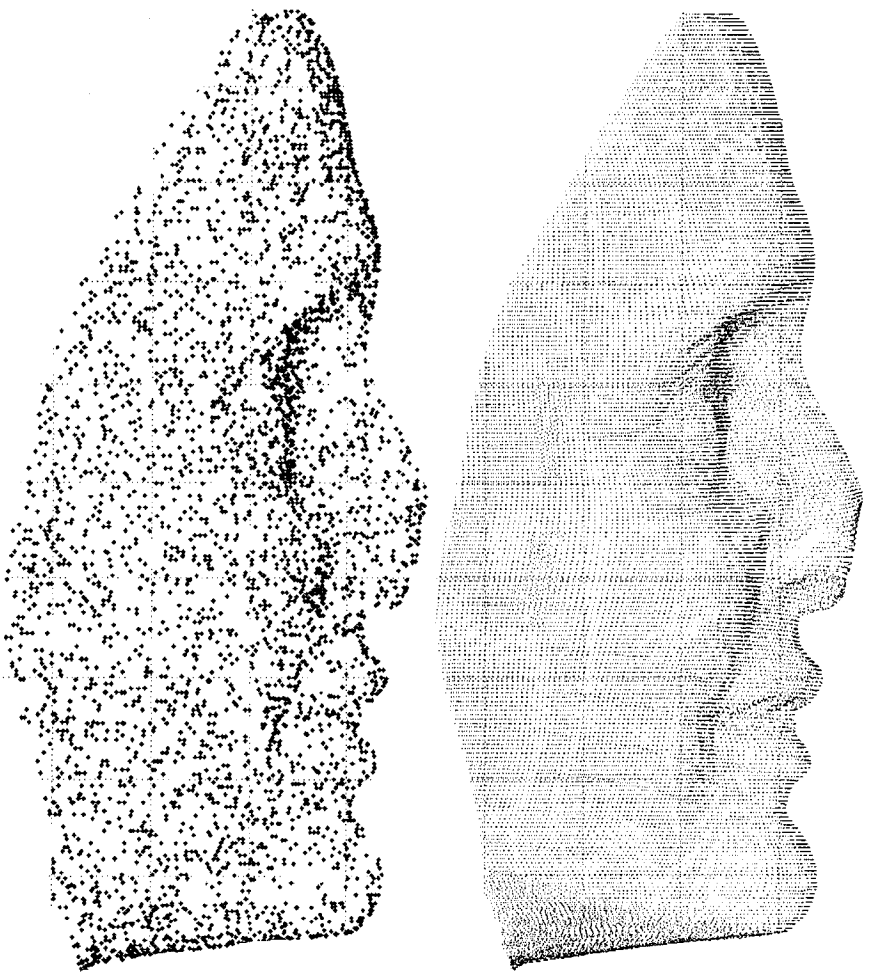

Figure 22: Reduced model set (left) for the high resolution face model (right)

Figure 22 shows the reduced model set (left) with 10% density of the high resolution model (right) with 40,800 points. The reduced model set thus consists of 4,080 points and can be interactively edited at a rate of 5-7 collision frames per second. The motion capture data was then applied offline to the high resolution model to achieve the same deformation of the nose for the model. The offline process cannot be undertaken at interactive frame rates and in the case of the high resolution face model the rate is about 1/25 collision frames per second which is only suitable for offline processing. This helps in both visualizing global deformation and also editing in an interactive way a high resolution 3D point based model.

Also, an analysis between the two deformation solutions, i.e. the reduced model set and the complete model shows that the percentage of mean difference between the two models for the set of the common 3D data points between the two models for the same motion capture data input, varies for different models between 0.08% to 0.32%. This mean difference variation is due the fact that in certain cases the collision detection is not triggered at the same motion capture data points for the two 3D point based models, and hence a difference of a deformation frame (in case of high frequency inputs, the difference would be more than a frame, but with high frequency), the overall change between more than one frames is still within the rage of 0.32% an in better circumstances, the percentage of mean difference is only 0.08%.

5.6 Localized deformation

The localized deformation speed-up heuristic can be used when there is a need for visual interaction with a high resolution 3D point based model for detailed and highly specific deformation or editing. Thus it is assumed that the user does not need any global changes to take place in the model and only location specific changes are required. This can be undertaken by defining boundary conditions for points lying at the border of the region outside which not deformation is needed and then editing can be performed.

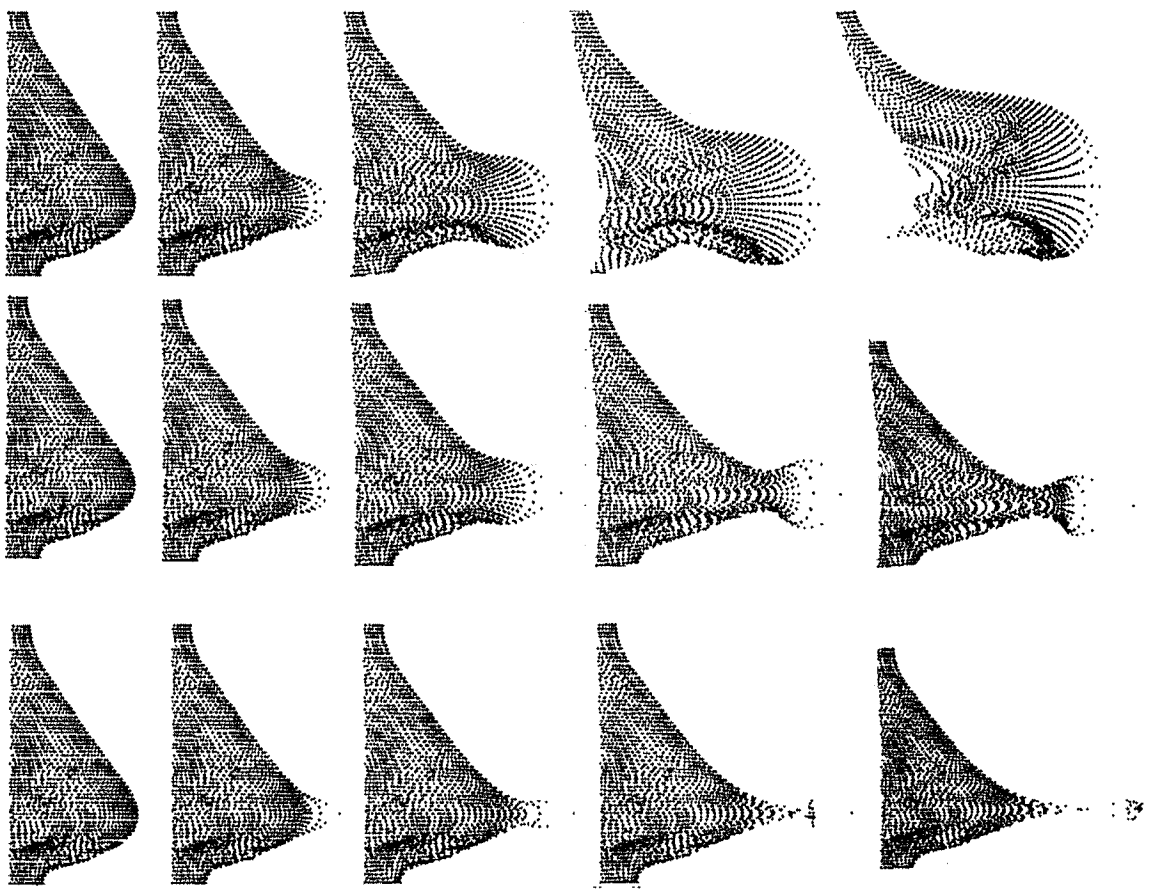

Figure 23: Illustration of deformation under different boundary constraints

Figure 23 illustrates the localization of deformation by setting various boundary constraints at the points lying at the base of the nose in the face 3D point based model. Assuming that the user only wants to make special and intricate changes to the nose of the model, the user can define the boundary conditions and edit the nose while the points lying at the junction of the editable and non-editable region are kept constant thus no tearing shall occur.

As in Figure 23, the top illustration is the progressive deformation of the nose without defining any boundary constraints. As the nose only contains 2804 points (the face model contains 40,800 points in total) the nose can be deformed interactively using the motion capture driven point probe at a rate of 19 collision frames per second with 96 collision frames. It is also apparent that the junction of the nose and the rest of the face shall have a "hole" present when this model is replaced back into the face model because the left most point which connect the face to the nose have been moved along with the rest of the deformation. Thus this process is not valuable but the progressive deformation in the middle of Figure 23 displays the deformation using the same input data and in this case the tearing is reduced as the points on the left most region remain almost static and do not move. In this deformation only 3 points lying on the base of the nose were given boundary conditions such that they remain motion-less. The middle progressive deformation was undertaken interactively at a rate of 18 collision frames per second with 96 collision frames.

In the bottom progressive deformation in Figure 23 it is clear that the points lying on the base of the model are completely static and no tearing takes place in the model at the junction due to this deformation. In this case 10 points lying that the junction were given boundary conditions thus are enforcing a more stringent criterion for meeting boundary conditions and deformation. The bottom localized progressive deformation of the nose was also undertaken interactively at a rate of 15 frames per second with 96 collision frames.

It is, however, to be noted that this localization results in reduced tolerance to high degrees of deformation and results in pinching of the point model at a lower extent of deformation. As in Figure 23 the top deformation has no "pinching" taking place due to the deformation in 96 frames, but the middle set with some boundary constraints and minimal localization results in "pinching" at about frame 69 while in the bottom localized deformation, the editable extent for deformation is reduced such that the pinching takes place at frame 46. For highly local and specific deformation, a high boundary constraint can be very conveniently used, but extreme deformations under this scenario can result in pinching" and thus it is only reasonable for precise changes to the model.

Figure Number	Number of Points	Number of Fingers	Number of Collision frames	Frame rate (per second)
11	40800	1	60	1/24
12	40800	1	42	1/24
13	134345	1	85	1/175
15	3203	1	50	18
16	40800	1	92	1/24
17	40800	1	47	1/24
18	3203	2	70	16
18	3203	3	72	15
19	40800	2	36	1/26
19	40800	2	8	1/26
20	40800	1	190	1/24
21	3203	1	340	18
22	4080	1	42	10
23	2803	1	96	19

Table 1: Table of results

5.6 Discussion

The results of the implementation have shown the high degree of applicability and intuitive nature of solutions that the underlying models/techniques are capable of providing. The results show a clear presence of physics based behavior and are highly stable (due to the offsets and thresholds applied) under sufficiently stressful but non-extreme editing conditions. While the interactivity rates in terms of dealing with large data models is a concern, the two speed-up heuristics presented help alleviate the issue by providing the user with real time editing capabilities in terms of both global and local change requirements.

The technique has also been shown to be able to achieve both inwards and outwards editing which is a highly desirable feature for any 3D modeling technique. It can be seen that both the forms of editing are highly intuitive in their behavior to forces and act as they can be expected to without any non-acceptable "digital" artifacts appearing unless the editing forces are extreme. Thus, this technique and system can be a very useful and handy tool which can be built upon to provide any 3D artist with virtual sculpting capabilities in both motion capture and haptics based environments.

Chapter 6

Conclusion and Future Work

With the introduction of ever increasing faster processing capabilities in terms of personal computers as well as industrial machines, it is only a matter of time when the art and entertainment providers begin requiring a virtual world where they can work and design their art without the hassles and non-editability of the present and more "classical" materials/tools. However, over the last few years higher degrees of interactivity has been targeted in terms of creating an environment for the artists who do not have a technical background to help them wade through the various complex layouts of some of the generic user interfaces. While there are various extremely intuitive forms of interactive painting and sketching digital tools available for artists in the form of pens/tablets, this is unfortunately not the case in terms of 3D modeling techniques.

In this thesis we have attempted to put forth a technique with the idea to consider this bottle neck being reduced/eliminated. We have presented a highly adaptable technique with extremely intuitive interaction handles which requires minimum handling from the user and would leave only the "classical" art editing being required by the artist as he/she uses his/her hands using the glove to sculpt/edit a 3D point based model. The technique adapted and implemented is physically based and thus the results seem to map intuitively to real behavior of various objects, say rubber, plastic, clay, etc.

Also keeping in mind the need for being able to "touch" and "feel" the object which needs to be interacted with, a model is presented in which via surface force feed back a 3D digital object can be edited by an user in a manner which would allow him/her to be able to manually feel his interactions with the object being manipulated.

In the deformation process, our method only computes displacement of existing points. In the case of exaggerated deformation, it may be possible that the sampling density of the point-based surface becomes very non-uniform. This may cause a problem in subsequent processing, as many point-based rendering techniques mandate near uniform sampling densities. If uniform point sampling density is required, we may have to adapt what most other approaches do, namely, carry out a re-sampling operation for introducing new points into the model.

While the examples given here only consider surface deformation of the point sampled surface data sets, we believe this technique can also be effectively used to deform/modify volumetric point sampled data. Hence it may be worth investigating its application in the medical domain, where such volumetric scanned data are used.

We believe that the ultimate goal of a design and interaction tool is to be able to give the user the best of both the real world/"classical" art forms as well as the data handling capabilities and visualization of the digital world. Thus, it was imperative to keep in mind that minimal (if any) compromises were being made (in terms of reducing the advantages

of "classical" and/or "digital" art) while modeling this technique in order to obtain the desired results.

The presented technique has also been applied in order to generate simple yet highly intuitive edits of complex 3D point based model sets. The stability of the technique has been ensured by the use of various thresh-holds and off-sets in order to always allow the user to work without any possible glitches in terms of behavior of the technique being implemented in the system. The reason for considering point based data to model was essentially due to the fact that its raw nature would allow the technique being developed to be able to be applied at any and all data sets which are 3 dimensional and/or volumetric in nature and thus the applicability of the research is increased.

We also studied the results of the technique and found that the technique can provide the user with good feed back results and in the future it would be appropriate to study further optimizations that can be done in order to achieve even higher rates of interaction. While with the present constraints we have presented two independent techniques which allow the user to achieve real time interactivity with high density point models and the two techniques in tandem cover almost all the design requirements which the user might need in order to achieve interactive performance. The first technique allows the user to interactively make global level changes on the data set interactively, while the second technique allows the user to make local highly detailed changes. Thus, as in the case of "classical" sculpting, the user can start off with global changes interactively and then proceed to make more refined changes in real time using the two speed-up techniques.

In the future we expect to enhance this technique to be able to be used with a head mounted display, thus allowing a complete immersive experience for the user of the system. Also, various gesture controls to orient and organize the 3D model in virtual space would also be a task which needs to be investigated. While this technique is highly stable and interactive a study need to be made into other enhancements and the possibility of using tools of "classical" sculpting in virtual worlds. Furthermore, we would also like to study the possibility of being able to add together multiple 3D point based models to create one single model, such as when an artist adds pieces of clay in various regions of a sculpture to add features and details in that specific region. It is believed that this thesis is only a small step towards the achievement of total human immersion into a digital artistic form where the freedom of the real world is only enhanced (and not in any way restricted) by the presence of a computing machine as an underlying platform to achieve creative content.

Bibliography

[1] **CSURI, C., HACKATHORN, R., PARENT, R., CARLSON, W. AND HOWARD, M.**, "Towards an Interactive High Visual Complexity Animation System," *Computer Graphics,* Vol. 13, No. 2, August, 1979, pp. 289-298.

[2] **BLINN, JAMES F.**, "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces," *Computer Graphics,* Vol. 16, No. 3, July, 1982, pp. 21-29.

[3] **REEVES, WILLIAM T.**, "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects," *Computer Graphics,* Vol. 17, No. 3, July, 1983, pp. 359-376.

[4] **SMITH, ALVY RAY**, "Plants, Fractals and Formal Languages," *Computer Graphics,* Vol. 18, No. 3, July, 1984, pp. 1-10.

[5] **LEVOY, M. and WHITTED, T.**, The use of points as a display primitive. TR 85-022, University of North Carolina at Chapel Hill, 1985.

[6] **GROSSMAN, J.P. and DALLY, W.J.** Point sample rendering. *Eurographics Rendering Workshop 1998*, pages 181–192, June 1998.

[7] **H. PFISTER, M. ZWICKER, J. VAN BAAR, AND M. GROSS.** Surfels: Surface elements as rendering primitives. *Proceedings of SIGGRAPH 2000*, pages 335–342, July 2000.

[8] **ZWICKER, M., PAULY, M., KNOLL, O.**, AND GROSS, M. 2002. Pointshop3d: An interactive system for point-based surface editing. SIGGRAPH, 322–329.

[9] **RUSINKIEWICZ, S., AND LEVOY, M. 2000.** Qsplat: A multi-resolution point rendering system for large meshes. SIGGRAPH, 343–352.

[10] **MARC ALEXA , JOHANNES BEHR , DANIEL COHEN-OR , SHACHAR FLEISHMAN , DAVID LEVIN , CLAUDIO T. SILVA**, Point set surfaces, Proceedings of the conference on Visualization '01, October 21-26, 2001, San Diego, California

[11] **MARC ALEXA , JOHANNES BEHR , DANIEL COHEN-OR , SHACHAR FLEISHMAN , DAVID LEVIN , CLAUDIO T. SILVA**, Computing and Rendering Point Set Surfaces, IEEE Transactions on Visualization and Computer Graphics, v.9 n.1, p.3-15, January 2003

[12] **ANDREW NEALEN, MATTHIAS MÜLLER, RICHARD KEISER, EDDY BOXERMAN AND MARK CARLSON**, Physically Based Deformable Models in Computer Graphics, Eurographics STAR Report 2005.

[13] **TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.**: Elastically deformable models. In Computer Graphics Proceedings (July 1987), Annual Conference Series, ACM SIGGRAPH 87, pp. 205–214.

[14] **BARAFF D., WITKIN A.**, Large steps in cloth simulation. In Proceedings of SIGGRAPH 1998 (1998), pp. 43–54.

[15] **DESBRUN M., SCHRÖDER P., BARR A. H.**, Interactive animation of structured deformable objects. In Graphics Interface '99 (1999).

[16] **CHEN D. T., ZELTZER D.**, Pump it up: computer animation of a biomechanically based model of muscle using the finite element method. In *Proceedings of SIGGRAPH*, 1992; 89-98.

[17] **KOCH R.M., GROSS M.H., CARLS F.L., BUREN D.F., FANKHAUSER G., PARISH Y.I.H.**, Simulating facial surgery using finite element methods. In *Proceedings of SIGGRAPH*, 1996; 421-428.

[18] **JAMES D. L., PAI D. K**, ArtDefo, accurate real time deformable objects. In Computer Graphics Proceedings (Aug. 1999), Annual Conference Series, ACM SIGGRAPH 99, pp. 65–72.

[19] **S. COQUILLART**, Extended free-form deformation: A sculpturing tool for 3d geometric modeling. Computer Graphics, 24(4):187–196, 1990.

[20] **T. W. SEDERBERG AND S. R. PARRY**, Free-form deformation of solid geometric models. Computer Graphics, 20(4):151–160, 1986.

[21] **RICARDO S. AVILA, LISA M. SOBIERAJSKI**, "A Haptic Interaction Method for Volume Visualization," *vis*, p. 197, Seventh IEEE Visualization 1996 (VIS'96), 1996

[22] **SARAH F. FRISKEN , RONALD N. PERRY , ALYN P. ROCKWOOD , THOUIS R. JONES**, Adaptively sampled distance fields: a general representation of shape for computer graphics, Proceedings of the 27th annual conference on Computer graphics and interactive techniques, p.249-254, July 2000

[23] **T. V. THOMPSON II, D. E. JOHNSON, AND E. COHEN**. Direct haptic rendering of sculptured models. In Symposium on Interactive 3D Graphics, pp 167–176, 1997.

[24] **FRANK DACHILLE, HONG QIN, ARIE KAUFMAN, JIHAD EL-SANA**, Haptic Sculpting of Dynamic Surfaces, In Symposium on Interactive 3D Graphics, pp 103-109, 1999.

[25] **ZHAN GAO AND IAN GIBSON**, Haptic B-spline Surface Sculpting with a Shaped Tool of Implicit Surface, Computer Aided Design & Applications, 2(1-4), pp 263-272, 2005.

[26] **H. QIN AND D. TERZOPOULOS**, "D-Nurbs: A Physics-Based Framework for Geometric Design," *IEEE Trans. Visualization and Computer Graphics,* vol. 2, no. 1, pp. 85-96, Mar. 1996.

[27] **A. OPALACH AND M. CANI-GASCUEL**. Local deformations for animation of implicit surfaces. In W. Straßer, editor, 13th Spring Conference on Computer Graphics, pages 85–92, 1997.

[28] **ALON RAVIV , GERSHON ELBER**, Interactive Direct Rendering of Trivariate B-Spline Scalar Functions, IEEE Transactions on Visualization and Computer Graphics, v.7 n.2, p.109-119, April 2001

[29] **ASKES H.**: Everything you always wanted to know about the Element-Free Galerkin method, and more. Tech. rep., TU Delft nr. 03.21.1.31.29, 1997.

[30] **M. MÜLLER , R. KEISER , A. NEALEN , M. PAULY , M. GROSS , M. ALEXA**, Point based animation of elastic, plastic and melting objects, Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, August 27-29, 2004, Grenoble, France

[31] **LIU, GL.**, Mesh Free Methods, *Moving Beyond the Finite Element Method.* CRC Press LLC; 2003.

[32] **CHANG, J., ZHANG, J. J.** 2004. Mesh-free deformations. In Computer Animation, Virtual Worlds, 211.218.

[33] **R. SZELISKI AND D. TONNESEN**. Surface modeling with oriented particle systems. SIGGRAPH, 1992.

[34] **A. WITKIN AND P. S. HECKBERT.** Using particles to sample and control implicit surfaces. SIGGRAPH, 1994.

[35] **M. PAULY, L. KOBBELT, AND M. GROSS.** Multiresolution modeling of point-sampled geometry. TH Zurich Technical Report, 2002.

[36] **M. PAULY, R. KEISER, L. KOBBELT, AND M. GROSS.** Shape modeling with point-sampled geometry. SIGGRAPH, 2003.

[37] **GUO XIAOHU, JING HUA, AND HONG QIN**, Touch-based Haptics for Interactive Editing on Point Set Surfaces, IEEE CGA, 24(6), pp31-39, 2004.

[38] **SADD M. H. ELASTICITY**: *Theory, Application and Numerics*. Elsevier Academic Press, 2005.

[39] **SAADA A. S.** Elasticity Theory and Applications (2^{nd} edn.). Krieger Publishing Company, 1993.