DOI: 10.1111/j.1467-8659.2007.01102.x

COMPUTER GRAPHICS *forum*
*Volume 26 (2007), number 4 pp. 824–852*

# Manipulating, Deforming and Animating Sampled Object Representations

M. Chen[3]†, C. Correa[2], S. Islam[3], M. W. Jones[3], P.-Y. Shen[1], D. Silver[2], S. J. Walton[3] and P. J. Willis[1]

[1]Department of Computer Science, University of **Bath**, UK
[2]Department of Electrical and Computer Engineering, **Rutgers**, The State University of New Jersey, USA
[3]Department of Computer Science, University of Wales **Swansea**, UK

**Abstract**

*A sampled object representation (SOR) defines a graphical model using data obtained from a sampling process, which takes a collection of samples at discrete positions in space in order to capture certain geometrical and physical properties of one or more objects of interest. Examples of SORs include images, videos, volume datasets and point datasets. Unlike many commonly used data representations in computer graphics, SORs lack in geometrical, topological and semantic information, which is much needed for controlling deformation and animation. Hence it poses a significant scientific and technical challenge to develop deformation and animation methods that operate upon SORs. Such methods can enable computer graphics and computer animation to benefit enormously from the advances of digital imaging technology.*

*In this state of the art report, we survey a wide range of techniques that have been developed for manipulating, deforming and animating SORs. We consider a collection of elementary operations for manipulating SORs, which can serve as building blocks of deformation and animation techniques. We examine a collection of techniques that are designed to transform the geometry shape of deformable objects in sampled representations and pay particular attention to their deployment in surgical simulation. We review a collection of techniques for animating digital characters in SORs, focusing on recent developments in volume animation.*

**Keywords:** sampled object representations, volume datasets, point clouds, images, manipulation, deformation, animation, volume visualization, surgical simulation

**ACM CCS:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.3.6 [Computer Graphics]: Methodology and Techniques - Graphics data structures and data types, Interaction techniques; I.3.7 [Computer Graphics]: 3D Graphics and Realism - Animation

## 1. Introduction

With technical advances and cost reduction, digital imaging technology is rapidly becoming one of the most effective ways of collecting data and information. In computer graphics, many of the most recent developments have gravitated towards the handling of sampled data directly in graphics pipelines. For instance, *volume rendering* [Lev88, Wes90] enables direct rendering of three-dimensional (3D) volume data captured by 3D scanning devices such as computed tomography scans; *image-based rendering* [Che95, LH96] enables direct rendering of graphical models and scenes that are specified with a set of photographic images; *point-based rendering* [PZvBG00, ZPBG01] enables direct rendering of a large collection of sampling points representing a surface object (e.g., a sampled dataset acquired using laser-scanning).

However, these developments are yet to have a significant impact upon computer animation, which encompasses a wide-range of graphics techniques and is often considered the ‘crown jewels’ of computer graphics. It is not difficult to

---

†Contact Author: Min Chen, Department of Computer Science, University of Wales Swansea, Singleton Park, Swansea SA2 8PP, United Kingdom. Tel: +44 1792 295663. Fax: +44 1792 295708. Email: m.chen@swansea.ac.uk

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

Submitted 27 September 2005
Revised 16 November 2006
Accepted 10 January 2007

see that computer animation could benefit enormously from the advances of digital imaging technology, should appropriate techniques be integrated into animation pipelines for handling sampled data directly. This would enable animators to gain easy access to a huge collection of complex models defined in the forms of images, points and volumes. In the following discussions, we collectively refer to image-, point- and volume-based representations as *sampled object representations* (SORs).

In this survey, we consider a wide-range of techniques that have been developed for manipulating, deforming and animating SORs. This particular focus on SORs enables us to review the previous work in the field from a non-traditional perspective, and to cover some areas which have not been examined in other major surveys on deformation and animation techniques. In addition to computer animation as a benchmarking application, the survey can serve as a state-of-the-art report on a class of techniques that are essential to applications such as surgical simulation, medical illustration, computer graphics special effects and virtual reality.

In computer graphics, techniques for *manipulation*, *deformation* and *animation* are inter-related, and sometimes terms are used interchangeably. In this survey, in the context of *manipulation*, we consider a collection of elementary operations for processing SORs. All these operations can serve as building blocks of deformation and animation techniques, and some also involve minor deformation to realize a desired manipulation. In the context of *deformation*, we consider a collection of techniques that are designed to transform the geometrical shape of a deformable SOR. While minor geometrical changes are often a ‘side effect’ of a manipulation operation, a deformation operation is intended to precipitate geometrical changes. In the context of *animation*, we consider a collection of techniques for animating digital characters in SORs focusing on modelling movements of articulated figures in SORs.

The survey is organized as follows: in Section 2, we give an overview of the general scope of deformation and animation techniques in computer graphics. In particular, we briefly describe the major advances in surface-based techniques, which are not the focus of this report, but can serve as benchmarks for techniques based on SORs. In Section 3, we provide a formal definition of the class of SORs, and discuss their geometrical and graphical attributes. In Section 4, we examine a collection of elementary operations on SORs, highlight the fact that such operations will cause minor and often unintended, geometrical changes, and prepare for the further discussion about their use in deformation and animation. In Section 5, we consider various deformation techniques that have been, or can be, used to realize intended deformation. In particular, after a brief overview of empirical deformable models, physically-based deformable models and direct deformation rendering, we examine the deployment of these techniques in surgical simulation, an application area where SORs can have a major role. In Section 6, we focus on the main components in the animation pipeline for modelling the movements of digital characters in SORs. Finally, we offer our observations and concluding remarks in Section 7.

## 2. General Scopes of Deformation and Animation

In computer graphics and its applications, deformation modelling and computer animation are two closely related fields. While the literatures on deformation and animation are dominated by surface-based modelling and rendering techniques, it is certainly sensible and meaningful to consider the deformation and animation of SORs in the backdrop of these techniques. In this section, we give a general overview of the scopes of deformation and animation, their relationships with other fields in computer graphics and their applications. This is followed by an outline of major technical advances in deformation modeling, and in computer animation.

### 2.1 Overview

The temporal behaviour of a graphical object may come in a variety of forms, such as changes of positioning attributes, geometrical shape, color and illumination properties, and many other behavioural parameters. Among those, the change of geometrical shape, that is, *deformation*, is the most well studied in the literature. A range of representation schemes for deformable objects were developed, and several physically-based and empirical computational models were formulated. Applications of deformation techniques include graphical modelling, computer animation, scientific visualization, haptic interaction, surgical simulation, medical imaging, path planning and computer vision.

In the technical scope of computer graphics, the term *computer animation* is primarily referred to the modelling, controlling and rendering of temporal behaviour of graphical objects. It addresses a wide-range of technical issues, such as motion dynamics, kinetic control, collision detection, actor modelling, animation control and so on [WW92a]. Deformation of graphical objects, including articulated and soft objects, is an integral part of computer animation techniques.

In a broader term, *computer animation* usually refers to the process of creating temporal sequences of computer generated images and digital visual effects. This process involves not only computer graphics techniques, but also an entire production pipeline including story development, visual development, character design, motion capture, camera tracking, texture painting, image processing, image retouching, image composition, colour grading and so on [Ker04]. In the entertainment industry, the use of the term also extends to computer-assisted animation, such as computer generated in-betweens in key-frame animation.

Figure 1 depicts the overall scope of deformation modeling and computer animation, and highlights the main

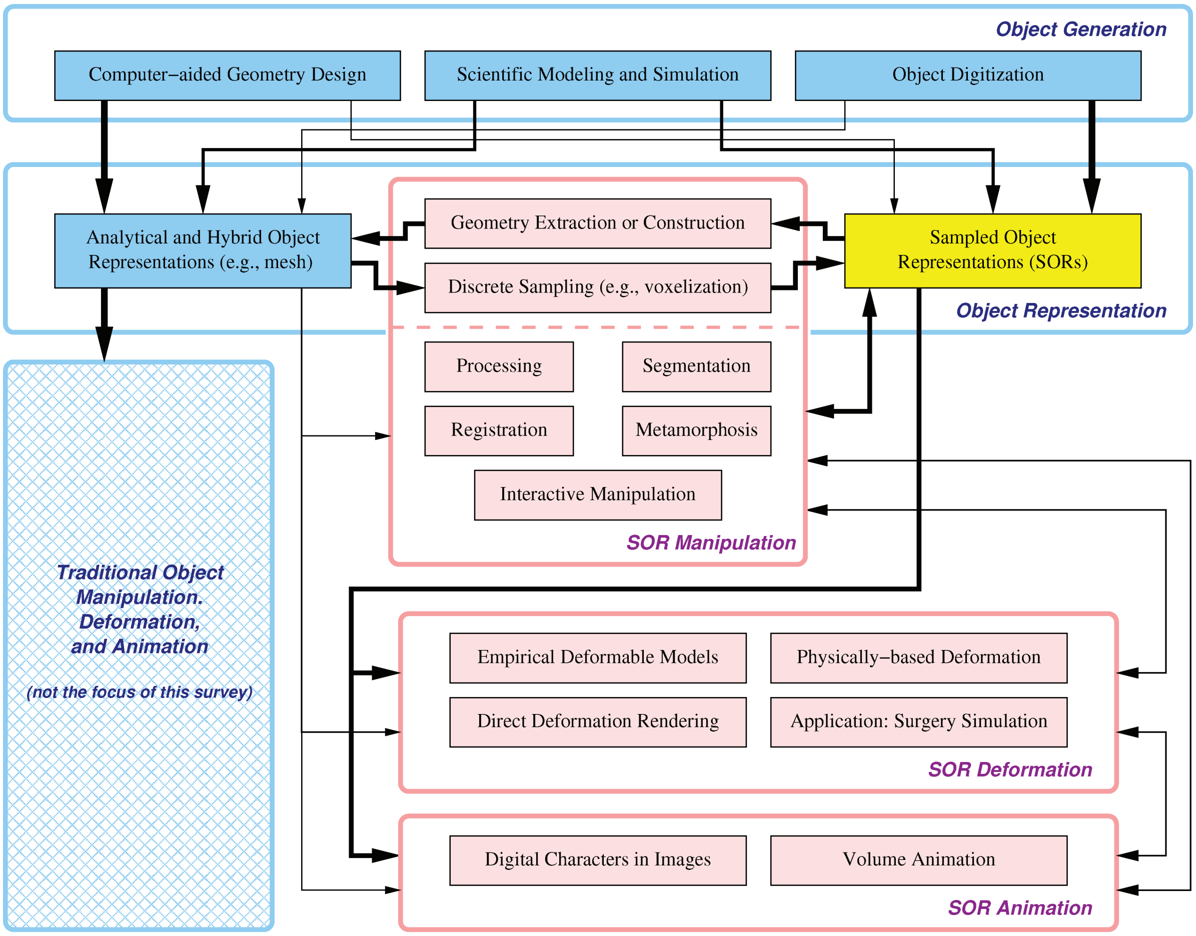

**Figure 1:** *The main scope of this survey and related topics.*

technical areas of manipulating, deforming and animating SORs, which will be discussed in detail in Sections 4, 5 and 6, respectively. Traditional deformation and animation is not the focus of this survey, but will be outlined in 2.2 and 2.3 as the technical backdrop of this survey.

A *sampled object representation* (SOR) defines a graphical model using data obtained from a sampling process, which takes a collection of samples at discrete positions in space in order to capture certain geometrical and physical properties of one or more objects of interest.

At the other end of the spectrum, many object representations are specified in an analytical manner, for instance, using a mathematical function to define the shape of an object, or using a user interface to define a set of interconnected geometric entities. Here, such a representation is referred to as an *analytical object representation* (AOR).

Colloquially, an object modelled in a SOR is often referred to as a 'discrete object', whilst one in an AOR as a 'continuous object'. This can be rather misleading. Though an object in a SOR may contain disjoint components, each component is continuous from the perspectives of both modelling and rendering (see the detailed definition in Section 3). Hence, in terms of continuity, there is no fundamental distinction between objects specified in a SOR and an AOR.

The top two boxes in Figure 1 do not correspond exactly to the taxonomy and dataflow of volume visualization proposed by [KCY93] because the latter placed its focus on one particular SOR, that is, regular volume grid or volume buffer. This survey considers a much broader taxonomy in terms of the types of representations as well as the methods for handling these representations.

There is a rich collection of data manipulation techniques for processing and altering SORs, perhaps far more than those for AORs. Most of these techniques may introduce, often unintentionally, minor geometrical changes in SORs. Some are designed to extract analytical information (e.g., connectivity) from a SOR. The information is then used to construct a corresponding AOR, or to create a *hybrid* *object representation* (HOR) by combining the original SOR with the generated analytical components. Such an AOR or HOR is commonly used to assist in deforming and animating the SOR. We will examine these techniques in Section 4.

### 2.2 Major advances in deformation

For deformation, readers are encouraged to refer to several extensive surveys, including surveys of deformation techniques in the context of the following subject areas.

- computer graphics and animation [GM97, NMK*06],
- animation of articulated figures [LW98],
- facial modelling and animation [NN98],
- medical image analysis [MT96],
- surgical simulation [LTCK03],
- computer vision [CF01].

The models and techniques for deformation fall into the following categories:

- *Functional and Procedural Models*—Models in this category are normally defined mathematically and are not accompanied by a discrete geometrical representation, such as control points. The control of deformation is hence manipulated by changing global parameters (e.g., the radius of a sphere) or local parameters in a constructive model (e.g., parameters of a primitive function in implicit surfaces). Some commonly used models in this category are algebraic implicit surfaces, superquadrics, generalized cylinders (or swept volumes) and global and local deformation of solids.
- *Parametric Models*—Models in this category are defined mathematically, and facilitate a continuous distortion across the surface of a deformed object, though they are usually accompanied by a discrete geometric representation which provides the principle control of deformation. Some commonly used models in this category include a variety of parametric bicubic surface models, free form deformation, active contour models (snakes) and active surface models.
- *Polygonal Meshes*—Models in this category are defined by a collection of inter-connected polygons with shared vertices and edges. Because these representations, especially triangular meshes, facilitate fast rendering, they are most widely-used in computer graphics and animation. Whilst it is possible to deform such a model by directly manipulating its geometric elements such as vertices or faces, direct manipulation becomes impractical for accurate control of deformation when there are more than hundreds of geometric elements in a model. Hence, it is common to associate a deformable polygonal mesh with a high-level deformable model that is more controllable. For example, as in free-form deformation, a polygon mesh may be bounded by a parametric volume, whose deformation governs that of the polygonal mesh.
- *Physically-based Deformable Models*—Models in this category facilitate the computation of motion and deformation based on specific physical laws. Numerical simulation is an indispensable tool in implementing most of these models. Commonly used physical models and computation methods include Lagrangian dynamics, mass-spring models, finite element methods, finite difference, continuum models and particle systems.

### 2.3 Major advances in animation

For a general coverage of computer animation, readers may refer to several books on this subject, including [FvDFH96, WW92a, Vin92, O'R95, MTT96, Par01] and [Poc01, Ker04]. In addition, the technical literature is rich in papers in this area. Readers are also encouraged to consult various computer graphics journals, and proceedings of some major conferences which include:

- SIGGRAPH Annual Conferences (1974-present);
- Eurographics Annual Conferences (1980-present);
- Eurographics Workshops on Animation and Simulation (1990-2001) and SIGGRAPH/Eurographics Symposiums on Computer Animation (2002-present);
- International Conferences on Computer Animation (1988-2002) and on Computer Animation and Social Agents (2003-present);
- Workshops on Lifelike Computer Characters (1994-1996, 1998);
- Workshops on Virtual Humans (1996-1998);
- and many one-time conferences and workshops (http://www.cs.ubc.ca/∼van/ani.html).

Motion and deformation are the two most fundamental temporal features in an animation. It is also necessary for an animation system to address the interaction between different objects. The techniques for motion and deformation control include the followings.

- *Articulated Models*—Articulated models attracted attention from the early days of 3D animation, and they facilitate the kinematical control at joints of articulated parts. A summary of various techniques can be found in [BBZ91].
- *Procedural Control*—General procedural control is based on scripts controlling actors or on interactive motion planning. In both cases the input is interpreted to generate the required movement of the characters and the operation of the camera, so that a rendering system can produce the frames. Procedural techniques may involve other methods listed here, for example, by implementing physical laws.

**Table 1:** *Example data capture modalities, and their typical characteristics and representation schemes.*

| Example Sampling Modality (*physical property*) | Data Dimension | Number of Channels | Representation Scheme |
|---|---|---|---|
| Black-white photography (*light reflection*) | 2 | 1 | 2D regular grid |
| Color photography (*light reflection*) | 2 | 3 | 2D regular grid |
| Raw laser scans (*distance to a plane*) | 2.5 | 1 | 2D regular grid |
| Circular full-body scans (*distance to an axis*) | 2.5 | 1 | 2D curvilinear grid |
| Computed tomography (*X-ray attenuation*) | 3 | 1 | 3D regular grid |
| Magnetic resonance imaging (*relaxation of magnetized nuclei*) | 3 | 1 | 3D regular grid |
| Raw 3D Ultrasonography (*sonic reflection*) | 2.5 | 1 | unstructured 2D images |
| Processed 3D Ultrasonography (*sonic reflection*) | 3 | 1 | 3D regular grid |
| Electron microscopy (*electron diffraction*) | 3 | 1 | 3D regular grid |
| Spatial distance fields (*distance to a surface*) | 3 | 1 | 3D regular grid |
| Spatial vector fields (*e.g., velocity*) | 3 | 3 | 3D regular grid |
| 3D photographic imaging (*light reflection*) | 3 | 3 | 3D regular grid |
| Movies and videos (*time-varying light reflection*) | 3 | 3 | 3D regular grid |
| Particle simulation results (*space-time position, etc.*) | 4 | 1 | time-series, 3D point set |
| Motion capture data (*space-time position*) | 4 | 1 | time-series, 3D point set |
| Seismic measurements (*space-time density, temperature, etc.*) | 4 | n | time-series, 2D point set |

- *Constraint-based Control*—This approach is common in robot systems. As such, there are obvious parallels with, and applications in, computer animation.
- *Physically-based Control*—Physically-based simulation has a long history, with commercial applications in flight simulators being an early example. Human and animal movements are also amenable to physical simulation and of course these same techniques can be used to give human-like movements to non-human objects.
- *Stochastic Control*—Particle systems and their rendering to produce fireworks, fire, water sprays and other fuzzy objects are the examples of stochastic control.
- *Behavioural Control*—This area has traditionally concerned itself with gross control, for instance, for modelling flocks, herds and schools. It has more recently been used to control subtle effects in human animation, for example, in the way emotion affects how a character walks.

## 3. Sampled Object Representation (SOR)

The general notion of a SOR is a set of samples $V = \{(p_i, v_i) \mid i = 1, 2, \ldots, n\}$, where $v_i$ is a value of a specific data type (e.g., Boolean, scalar, vector, tensor or constructive), which represents some property (e.g., luminance intensity or sampling confidence) at each sampling location $p_i$ in $k$-D Euclidean space $\mathbb{E}^k$. Typically these samples are associated with a spatial domain $\mathbb{D}^k (\subseteq \mathbb{E}^k)$, which is normally continuous or consists of several disjoint sub-domains. An object specified by a SOR is thus a function $F(p)$ that defines the value at every $p \in \mathbb{D}^k$. $F(p)$, which is commonly referred to as an *interpolation* or *blending* function, typically derives a value $v$ at an arbitrary point $p \in \mathbb{D}^k$ from a number of known point-value pairs in $V$. The continuity of the object within $\mathbb{D}^k$ (or each of its continuous sub-domains) is thus determined by that of $F$.

*Digitization* is the primary technology for acquiring SORs of real-life objects and phenomena. This technology, which is based on measuring various physical properties, is available in a wide-range of modalities as listed in Table 1. In most of these modalities, a sampling process may involve the processing of multi-channel or multi-dimensional signals, including convolution and deconvolution, quantization and signal space conversion. Here, we consider only the resulting digital representations of such a sampling process.

In some modalities, sampling positions are defined by a regular grid in the object space. For example, computed tomography (CT) scanning normally utilizes a 3D anisotropic grid, where the sampling interval in the $z$-direction differs from that in the $x$ and $y$ directions. In many other modalities, sampling positions are defined by a regular grid in the image space. A primary example of such modalities is photography, where sampling results are recorded on a 2D isotropic grid though individual samples may not correlate uniformly to signal sources in the object space.

SORs can also be obtained by sampling AORs. For example, a parametric surface model can be approximated by an unstructured point dataset using a randomized discretization process or by a volume dataset using a voxelization process. In many science and engineering disciplines, such as finite element analysis and computational fluid dynamics, a SOR (e.g., a point set) is usually accompanied by analytical components (e.g., a face list), resulting in an HOR.

SORs commonly exhibit a subset of the following characteristics, which collectively signify the differences between

SORs and other schemes for representing graphical objects and scenes.

- *Limited geometrical information*—Most SORs do not contain any explicit geometrical description of the objects represented, while some contain partial geometric information (e.g., in a point set). It is common to translate sampled physical information (e.g., X-ray attenuation) to geometrical information (e.g., an isosurface of a tumour). In addition, SORs are particularly suited for modelling amorphous objects, such as fire, dust and smoke, for which a precise geometrical description is difficult to obtain.
- *Limited topological information*—The only topological information available in a SOR is the spatial or temporal order in which samples were captured. Such information does not imply a definite topological relationship between any two data points in the object space, although it is often used to derive, analytically or statistically, more meaningful topological information, such as the possible connectivity between two sampling points in the context of 3D model acquisition and the association of a set of voxels to the same object in the context of segmentation.
- *Little semantic information*—Although a SOR, such as a photographic image and a computed tomography scan, may capture a collection of objects in a scene, it does not normally contain any semantic information, about the objects of interest, such as object identification and object hierarchy.
- *Multiple data channels*—Many SORs capture data from a complex signal source (e.g., reflectance) or multiple signal sources (e.g., a combination of density, sonic, temperature and imagery logging in seismic measurements).
- *Multi-valued data channels*—Many SORs contain data sampled in an integer or floating-point real domain. In some situations, this facilitates a high level of accuracy (e.g., the texture of a piece of textile in an image), but in others, this brings about a degree of uncertainty (e.g., the boundary of a piece of textile in an image).

The development of techniques for manipulating, deforming and animating SORs can be built upon theoretic advances in areas of signals and sampling [PM96], point-set [Mor90], discrete topology [CK95] and level-set [Tsi95, Set96], as well as technological advances in areas of deformable object modelling, computer animation, scientific visualization, volume graphics, point-based graphics, image-based modelling and rendering, image processing, computer vision and medical imaging. Meanwhile, such development will also have a profound scientific impact in these areas and the field of computer graphics in general. It will deliver a collection of usable and effective techniques and tools to a wide range of applications in science, engineering, medicine and industries including manufacturing, media and entertainment.

## 4. Manipulating Sampled Object Representations

In this survey, we consider *manipulation* as the application of elementary operations that alter the sampled properties of SORs, or convert a SOR to an AOR or HOR. There is not an agreeable term for encapsulating these operations. The term 'manipulation' captures the essence of 'data manipulation' as this is what is concerned here. Although one may use the term 'transformation', it does not capture the same scope due to its traditional use in graphics for geometric transformation.

As discussed above, most SORs contain limited geometrical, topological and semantic information, and the traditional notion of 'geometry' is usually constructed from sampled physical information. Hence, any operation of a SOR that leads to changes of sampled physical properties may result in alterations to geometrical, topological and semantic attributes derived from the SOR. Such alterations are in effect 'minor deformations'. For instance, varying intensity values of some voxels in a CT dataset may lead to a different isosurface in surface extraction, and modifying confidence level of some points in a laser scan may result in different topological connectivity of these points in surface reconstruction.

In this section, we examine a range of methods for altering and converting the raw physical properties of SORs. We first consider three sets of algorithms, namely *surface extraction* from volume data, *surface reconstruction* from point data, and *skeletonization*, all of which are used for constructing 'geometry', in a traditional sense, from SORs. We then describe a collection of basic operators for *altering* data elements in a single 2D or 3D imagery dataset, and discuss more complex SOR manipulation in the context of image and volume morphing. This is followed by a brief review of *segmentation* techniques and *registration* techniques where surface-based deformation algorithms are often deployed. Finally we mention a few *interactive systems* for manipulating SORs. As many of these topics, such as surface extraction and segmentation, cover a large domain of the literature, we only give a brief overview of each topic in order to outline the overall scope of SOR manipulation, and prepare for the further discussions on SOR deformation in Section 5 and animation in Section 6.

### 4.1 Extracting geometry from SORs

There are 2D algorithms for extracting contours and skeleton from images, and for constructing contours from points. Because they are relatively trivial in comparison with 3D algorithms, they do not attract the same level of attention in the literature. Here, we concentrate on algorithms for 3D geometry extraction. We also assume that the sampled values in SORs do not suffer from sampling errors or regional variance, which will be considered in 4.3.

### 4.1.1 *Surface extraction from volume data*

One common approach for handling 3D volume datasets is to approximate an interested object in the volume by a polygonal mesh that can then be rendered using a surface-based graphics system. Such approximation is usually in the form of an *iso-surface* (also called a level surface), which is the set of all points in a scalar field with a specific scalar value $\tau$ (i.e., iso-value). The most well-known method for extracting an iso-surface from a regular volume dataset is the *marching cubes* algorithm [LC87]. A more complicated algorithmic problem is extracting an *interval volume* and approximating the extraction, for example, by a tetrahedral mesh [NS97].

What complicates surface extraction algorithms is the fact that many basic cases are ambiguous. A similar but much simpler ambiguity problem also exists in a class of 2D contouring algorithms that extract contour lines from 2D SORs, such as an image. [NH91] provided a computational solution, called asymptotic decider, to resolve the 2D and 3D ambiguity problems.

Some methods were proposed to accelerate the process of marching cubes by reducing the search space of an iso-surface. A family of indexing structures were used for iso-surfacing, including active list [GH90], octree [Wv92], MIN-MAX cell index [Jon95, CMPS96], extrema graph [IK95], span space [LSJ96] and interval tree [CMM*97]. Another approach is to track an iso-surface from a known seed point or seed cube [HL78, BPS96]. More recently, algorithms were developed specifically for surface extraction from very large volume datasets [CS97, BS03b].

The number of triangles generated by the marching cubes algorithm can be excessively large, often leading to inefficiency in storing and rendering the extracted iso-surface. A noticeable amount of effort has been made to reduce the number of triangles or to replace triangles with other geometric primitives. There are two categories of algorithms:

- *During marching cubes*—Examples of such algorithms include producing surfaces adaptively [MS93], extracting points instead [CLL*88], replacing triangles with polygonal volume primitives [YP92].
- *After marching cubes*—Examples of such algorithms include removing vertices followed by local re-triangulation [SZL92], dispersion of new vertices on top of the original mesh, followed by global re-triangulation [Tur92] and edge manipulation [HDD*93]. A large collection of further development in this area can be found in the context of both multi-resolution surface modelling (e.g., [Hop96, Gar99]) and surface reconstruction from point data (see 4.1.2).

Recent advances in surface extraction include the reconstruction of a dual iso-surface in the form of quad patches [Nie04], high-dimensional isosurfacing [BWC00], feature-sensitive isosurfacing [VKKM03] and topology-controlled isosurfacing [vKvOB*97, GP00, TFT04, CSv04],

### 4.1.2 *Surface reconstruction from point data*

Many data acquisition techniques (e.g., laser range scanning [LPC*00] and alpha matte acquisition [MBR*00]) generate output in the form of an arbitrary set of points in space Where the properties of these points cannot be discerned directly, they must be inferred algorithmically. While there is a class of algorithms for direct rendering of point datasets (e.g., surfels [PZvBG00] and QSplat [LPC*00]), there have also been a collection of algorithms for reconstructing continuous surfaces from point datasets. Because of noise and imperfections introduced in the acquisition stage, such algorithms must list noise tolerance as a priority. Most algorithms available can be classified as:

- *Primitive list*—A polygonal mesh is constructed by adding topological connectivities (i.e., analytical components) to points. For example, Turk and Levoy [TL94] used triangulation to fit a triangular mesh to the point data and performed weighted averaging in overlapped areas. Amenta *et al.* [ABK98] used a Voronoi-based approach to reconstruct a triangular mesh. Bernardini *et al.* [BMR*99] developed a method for reconstructing a triangular mesh by connecting neighboring points with a ball pivoted around a seed point.
- *Functional and parametric surface*—A functional parametric surface is found to approximate the surface. For example, Hoppe *et al.* [HDD*92] devised a reconstruction method based on determining the zero set of an estimated distance function. Distance fields were also used by Curless and Levoy [CL96] and Wheeler *et al.* [WSI98] for reconstructing an implicit representation from point datasets. Lei *et al.* [LBC96] fitted high degree implicit polynomials to point data. Krishnamurthy and Levoy [KL96] proposed to fit smooth surfaces to the reconstructed polygonal meshes. Pratt [Pra87] developed a least-square fitting algorithm for defining an algebraic surface over a point dataset. Lee [Lee00], Alexa *et al.* [ABCO*01], Mederos *et al.* [MVdF03] Amenta and Kil [AK04] used a moving least squares method to fit a continuous surface to a set of points. Carr *et al.* [CBC01] used radial basis functions for their reconstruction.

### 4.1.3 *Skeletonization*

A skeleton is a useful shape abstraction that captures the essential topology of an object in both two and three dimensions. It is used extensively in commercial computer animation packages, and is therefore of interest for volume manipulation and animation. It refers to a thinned version of the original object but still retains the shape properties of the original object. In 2D, the skeleton is also referred to as the medial-axis. In 3D, the term skeleton has been used for both a medial-surface and a more line-like representation. In [Blu67], a grass-fire analogy to the skeleton is given, that is, the skeleton consists of the points where different fire fronts intersect. If a fire was simultaneously started on the perimeter of the grass, the fire would proceed to burn towards the interior of the object. When two fire fronts meet each other the fire will be quenched. In 2D, the fire will quench along a curve. In 3D, the two fire fronts will meet along a surface or a curve.

Recently, there has been interest in extracting a line-like 1D skeletal-representation from a 3D object. The line-like skeleton is also referred to as a curve-skeleton [SNS02], inverse-kinematic bone skeleton (IK-skeleton) [Dis04] or centerline skeleton. In this survey we refer to it as a curve-skeleton. Fine curve-skeletons are useful for many different geometric tasks, such as virtual colonoscopy and virtual endoscopy [HHCL01, PC87], 3D object registration [AJWB01, AB02, PFY*99], computer animation (both polygonal and volume animation) [Blo02, Dis04, GS99, GS01, TK03, WP02, LWM*03], matching [SKK02, SSGD03], surface reconstruction [Ley03], vessel tracking [AB02] and curved planar reformation [KFW*02]. While there is no precise definition for a curve-skeleton, there are numerous desirable properties of both the skeleton and the skeleton computation process. These properties depend upon the application that the curve-skeleton is being used for, and include thinness, centeredness, joint separation, reliability, etc. Because there can be many different curve-skeletons for a particular object, it is important that skeletonization algorithms have the capability of extracting multi-scale (hierarchical) sets of skeletons. A more comprehensive survey can be found in [CSM05, CSM07].

The use skeleton as a means specify manipulation or deformation of a volumetric model in volume animation will be further discussed in Section 6.3.

## 4.2 Processing and morphing

There is a huge collection of techniques for processing and manipulating images (e.g., [Cas96, GW01, SP86]). Many techniques do not involve dimensionally-dependent data structures and can easily be extrapolated to three or higher dimensional problems. Hence, most techniques reviewed in this section are applicable to both 2D and 3D SORs.

### 4.2.1 *Processing SORs*

The processing of SORs can serve to enhance the visual interpretation of the data, carry out structural alterations to objects contained within or convert the representation into an alternative for better transmission or storage.

- *Grey level transformations*—Basic grey level transformations operating on a value by value basis are employed to enhance SORs, for instance, to increase contrast during visualization. Gamma correction is a common operation used to non-linearly brighten or darken in order to enhance contrast on non-linear display devices.
- *Statistical processing*—Statistical properties of SORs (e.g., histograms), can be computed and used to enhance SORs (e.g., equalization), *High dynamic range* (HDR) methods were applied to images using histogram equalization [DM97] and their use for displaying volume data on HDR displays was recently studied [GTH05].
- *Spatial filtering*—Filtering can be used to smooth or sharpen features in SORs, and is also used for determining the derivative of a discrete image or volume function. Low-pass filters average neighbouring values to remove noise and to blur SORs, high-pass filters implement spatial differentiation to highlight edges and discontinuities. A number of high-pass filters are used to calculate derivatives (e.g. Sobel, Gaussian, Zucker-Hummel [ZH81], central differences, intermediate differences), along with adaptive schemes [THB*90]. Filtering has been used to denoise volume data to enhance visual effects, such as refraction [RC06], and to create anti-aliased voxelizations of objects [ŠK00]. Filtering can also take place in the frequency domain of a SOR [Mal93].
- *Arithmetic and logical operations*—Arithmetic and logic operations can be applied to multiple SORs to facilitate combinational operations, such as masking and blending. For example, arithmetic and logical operations were used to provide the SOR equivalent of *constructive solid geometry* (CSG), namely *constructive volume geometry* (CVG) [CT00]. Change detection operations, which are built upon mainly arithmetic and logic operations, were used to construct video volumes for the purposes of video visualization [DC03].
- *Morphological operations—Erosion*, *dilation*, *opening* and *closing* are commonly used morphological operations on 2D and 3D SORs [GW01]. *Erosion* removes external parts of an object (depending upon the structuring element), while *dilation* adds parts to the boundary of the object. *Opening* enlarges cracks and cavities, while *closing* closes up cavities and smooths spikes. Morphological operations can be applied to binary data as well as multi-valued data. *Distance field SORs* can be used to facilitate fast morphological operations [Jon01].
- *Distance transforms* [JBŠ06]—A distance field $D$ is a representation where at each point $p$ within the field, $D(p)$ represents the distance from $p$ to the closest point on any object within the domain. Given a SOR, *distance transforms* are the process of applying templates to each pixel/voxel in the SOR in order to propagate distances around the SOR. Methods for distance transforms include *Chamfer* [SNS02], *Vector* [SJ01] and *Fast Marching Methods* [Tsi95, Set96]. Distance field SORs can be used in volume morphing (see 4.2.2), and for voxelizing AORs (e.g., implicit surfaces) and HORs (e.g. triangular mesh objects) [Jon96, JS01].

There are other SOR processing methods that may also introduce geometrical changes to SORs, though intentionally these methods aims to minimize such changes. Such methods including Fourier transform (e.g., [Mal93]), wavelet transform (e.g., [SS96]), watermarking (e.g., [WGKH01] for volume and [CWPG04] for point datasets) and compression (e.g., [Jon04]).

### 4.2.2 *Metamorphosis of SORs*

Given a starting SOR $D_a$ and a finishing SOR $D_b$, *metamorphosis* (or commonly referred to as *morphing*) is a process that generates a sequence of in-between SORs, $D_1, D_2, \ldots, D_n$, which represent a smooth transformation from $D_a$ to $D_b$. For SORs defined upon regular grids, such as images and volume datasets, this transformation from one SOR to another is usually under the influence of two control structures. Let $C_a$ and $C_b$ be the control structures associated with $D_a$ and $D_b$, respectively. For each SOR $D_i$, an in-between control dataset $C_i$ is first obtained as an interpolation of $C_a$ and $C_b$. $C_i$ is then used, in conjunction with $C_a$ and $C_b$ respectively, to deform $D_a$ and $D_b$, resulting in two distorted SORs $D_{ia}$ and $D_{ib}$. The in-between volume $D_i$ is then obtained as the interpolation of $D_{ia}$ and $D_{ib}$.

According to the use of control structures, approaches to metamorphosis of SORs can be classified into the following three categories:

- *Cross Dissolving*—Methods in this category require no control datasets. The simplest cross-dissolving method is a linear interpolation between the two SORs with the same grid organization in the spatial domain. In volume morphing, when the emphasis is given to a particular iso-surface, distance fields are first constructed for the starting and finishing volumes, and the cross-dissolving process is then applied to the distance fields [PT92, CLS98, BW01]. To enhance the smoothness of the in-between SORs, the Fourier transform has been used to schedule the interpolation in the frequency domain by favoring high-frequency components that are defined by the threshold of an interested iso-surface [Hug92]. Wavelet transform has also been employed in volume morphing in a multi-resolution manner [HWK94].

  The methods in this category are generally easy to use and require little human interference. The smoothness of the in-between SORs is well achieved with methods based on wavelets and distance fields. However, methods in this category have difficulties in specifying complex morphing involving geometric transformations such as rotation.
- *Mesh Warping*—Mesh warping methods utilize control structures to define spatial subdivisions as well as coordinate mappings. With images, a mesh with triangular or quadrilateral elements is usually used as a planar subdivision defined over the images [SP86]. In 3D, a mesh with tetrahedral elements or hexahedral elements (with quadrilateral faces) is used as a volume subdivision over volume datasets [CJT95].

  In mesh warping, the distortion is constrained by individual elements, and it is therefore relatively easier to achieve a desired transformation without causing 'ghost shadows' [BN92] provided there is no 'fold-over' structure. However, in many cases, these methods require a control mesh consisting of a very large number of elements. The manipulation of 3D subdivisions through a user interface also seems to be somewhat problematic.
- *Field Morphing*—In field morphing, control structures are used to specify the corresponding features of in the SORs concerned. Although it requires user input of the control structures, feature-based deformation has demonstrated its flexibility and controllability. A variety of geometric shapes, such as points, lines, boxes and discs, have been used to specify features and coordinate mappings [BN92, CJT95, LGL95, CJT96, FSRR00]. Due to the needs of coordinate mappings, supplementary vectors are required for some 3D shapes, resulting in the difficulty in defining and manipulating features without a sophisticated user interface.

## 4.3 Segmenting SORs

Segmentation is an extensive subject, due to both the demand for segmentation (particularly in image and video processing, and medical analysis) and the complexity of segmentation in general. Many segmentation tools are available in the forms of commercial software and freeware. Segmentation for 2D SORs (e.g., images) has been widely researched and documented [Cas96, GW01, Yoo04, SWL05]. While great efforts have been made to devise techniques for segmenting 3D SORs (e.g., volume datasets), unfortunately, most 2D methods do not lend themselves naturally to 3D. Instead, volume datasets are commonly segmented slice by slice. In this section, we focus on the methodologies adopted by different techniques without making explicit dimensional distinction, and give references to 3D work wherever possible.

Segmentation is an effective means for adding semantic information into a SOR, and such information can be useful to volume deformation (e.g., in feature specification [CCC06b]) and volume animation (e.g., in determining the physical properties of a voxel). Meanwhile, deformable models, such as snakes, are also deployed as a tool in some segmentation techniques. Segmentation techniques for both images and volumes typically fall into one of the following categories.

### 4.3.1 *Stochastic methods*

*Thresholding* is the most basic segmentation technique, where sampled points are classed strictly according to their values. The technique fails with low-contrast volumes, and introduces aliasing artifacts [LK03].

Related to thresholding is the *watershed transform* [VS91], which can be extended to 3D easily. An improved method by [HP03] extends the immersion-based transform by building a hierarchical tree structure from the resulting basins, allowing for quick identification of individual objects without the necessity of merging the basins entirely.

Algorithms can be designed to choose thresholds automatically from histogram analysis as in [KEK03]. When the thresholds have a transition zone in between, the classification relies on the standard deviation of the 26 neighbours of the voxel concerned. The algorithm also relies on morphological closure and region-growing techniques, with some user interaction required for guidance where, for example, there is a very fine gap between two objects.

Region-growing techniques attempt to group similar voxels into regions of increasing size. [LJT01] created an unseeded version of the algorithm where seeds are automatically defined, removing the requirement for user interaction.

### 4.3.2 *Biologically-inspired methods*

A recent trend in segmentation has been the use of *artificial neural networks* [RA00]. The technique facilitates learning from past examples of segmented datasets. [AF97] used a two-stage system comprising of self-organizing components analysis networks and self-organizing feature maps for segmenting a CT dataset. Self-organizing feature maps attempt to represent the 3D data in only two dimensions, grouping together similar objects with no user interaction. They are also trained automatically.

Locally excitatory globally inhibitory oscillator network (LEGION) systems attempt to mimic the manner in which the brain analyses features from visual cortex oscillations. This framework was derived from theoretical work and recent experimental evidence. The idea is that the oscillations in the visual cortex can be implemented in software, attempting to reverse-engineer the brain's framework for detecting and identifying objects in a scene. Because of the complexity of such a system, simplified algorithms are developed to work with large datasets [SWY99].

### 4.3.3 *Data mining methods*

*Clustering* techniques attempt to group voxels together that display similar predefined characteristics, according to a measurement of similarity in a SOR. An $N$-dimensional vector is built from each voxel based on the properties of that voxel, the set of all $N$-dimensional vectors is then fed into a clustering algorithm. One of the most popular clustering methods is $K$-means clustering, which attempts to form $n$ disjoint, nonempty subsets by grouping together 'similar' voxels. Closely related is fuzzy clustering, which utilizes fuzzy *if-then* rules to determine object membership.

Popular algorithms from graph theory can be used in conjunction with clustering. Edges are built between vertices (voxels) that display similar properties [WL93], and are then removed from the graph where the vertices touching the edge fail to satisfy the similarity measure. The result is a graph that consists of $n$ unconnected subgraphs, which correspond to the segmented regions.

*Markov Random Field* is a statistical model that can be used alongside clustering to achieve automatic segmentation. Such a field stochastically defines local properties of the dataset in a completely generalized manner by modeling spatial interaction between voxels [RGR97]. Such algorithms unfortunately are computationally expensive and are heavily influenced by the controlling parameters.

### 4.3.4 *Knowledge-based segmentation*

Knowledge-based approaches utilize additional input to assist with the segmentation process, usually using a pre-generated atlas for the known dataset, as adopted in [SvL*03]. This is combined with a watershed transform to create partitions of similar areas of the dataset [SLK*03]. The atlas is a pre-segmented dataset that is spatially aligned with the target dataset (referred to as registration [MV98] or atlas-warping). A disadvantage of this method is that multiple datasets of the same object are required for atlas generation, and there is no guarantee that the atlas will match the target dataset in a satisfactory manner.

Atlas information can also be factored into previously stochastic methods, such as snakes or statistically-based segmentation [FRZ*05] for enhanced precision. Grau *et al.* give an improved watershed transform that uses prior information to improve accuracy [GMA*04]. In general, atlas-guided approaches excel with medical data since they assist with not only the segmentation stage, but also with the correct labeling and identification of the resulting regions.

### 4.3.5 *Energy-minimizing deformable models*

Deformable models offer a flexible and accurate approach to segmentation. Such models generally handle image noise extremely well, and also allow for subvoxel accuracy. Energy-minimizing snakes [KWT87, MT95] remain the most popular deformable model for image and volume segmentation. This is essentially a parametric contour model that actively (i.e. with user interaction) attempts to minimize its energy such that it forms a smooth and tight boundary around an object to be segmented. With the snake model, segmenting a volume dataset involves a slice-by-slice segmentation process, with a 3D segment built from the resulting 2D segments. [MBL*91] extended the energy-minimization model into 3D by constructing a polygonal 'balloon' inside the object that grows to conform to the surface of the object of interest. [TK95] used a reaction-diffuse-based modification of the bubble model which improves the behavior near sharp edges.

### 4.3.6 *Interaction-intensive approaches*

Human interaction is often required in medical segmentation to fine-tune any segmentation efforts made by the system. This approach is used with great effect in the PAVLOV system [KK99]. A parallel CPU system powers the rendering of the segmented dataset to allow for real-time updates. The user is invited to segment the dataset using thresholding, and erosion and dilation morphological operations.

[SHN03] used a GPU-based implementation of the seed-fill algorithm to segment areas of interest. Combined with hardware-accelerated rendering, the user is able to segment the dataset and view results in real-time.

## 4.4 Registering SORs

Registration is a process that involves three main algorithmic steps: (i) identifying features in the SORs to be registered, (ii) establishing correspondence between features and (iii) computing the appropriate transformation to be applied to the SORs in order to derive spatially consistent SORs. The transformation may be global (i.e., the same transformation applied over the whole SOR) or local (i.e., different transformations applied to regions of the SOR). The transformation may be rigid (i.e., only translations and rotations), affine (including translations, rotations, dilations and shears), or projective or elastic (e.g., lungs expanding, heart beating, joint deforming).

Registration techniques now span a wide range of applications, including remote sensing (e.g., multispectral classification), medical imaging (e.g., combining SORs from different modalities), cartography (e.g., map updating), computer vision (e.g., target localization). Comprehensive surveys of image registration methods can be found in [Bro92, ZF03]. [MV98] presented a full classification of the stages involved in medical registration. [MCS*02] offered a survey and review of various techniques for rigid and elastic (cardiac) registration, which present the additional problem of registering 4D SORs (3D spatial and time).

In the following two subsections, we consider the first two steps in 4.4.1 and the third step in 4.4.2. Again we do not make explicit dimensional distinction, and give references to 3D work wherever possible.

### 4.4.1 *Feature extraction and correspondence*

The first step is to identify features in each SOR to be registered, and the second is to establish a correspondence between those features. The very simplest techniques (and perhaps most accurate), use markers introduced to the SOR. In comparing the various medical registration methods, researchers refer to the use of (fiducial) markers as the 'gold standard' [PWL*98], where known markers are physically introduced to the scanned objects to help feature correspondence. Such markers are chosen so that they can be identified accurately and automatically thus removing errors and costly interaction. Their accuracy ensures that such methods may be used as a benchmark for alternative approaches.

Where markers cannot be introduced (existing data or introducing the markers will obscure or confuse the data), then *landmarks* already in the data have to be identified for correspondence. In this case, techniques such as cross correlation (in the spatial or frequency domain), may be used to identify the landmarks. Penney *et al.* [PWL*98] reviewed several of these techniques including cross correlation. In some cases, some user interaction may be required to 'correct' misjudged correspondences [Jon01], or if large enough numbers of correspondences are found, those that exceed the mean by a large amount may be rejected. Other approaches have used object normals within the 3D scan data and 2D X-ray image data to create correspondences [TLSP03], or have used B-splines to define non-rigid transformations and then varying the control points whilst applying correlation [RSH*99].

### 4.4.2 *Transform determination*

Once a feature correspondence has been created, an optimization process must be followed in order to determine the transformation. Quite often a simpler rigid body transformation is used to calculate the gross alignment between two SORs (calculated globally), followed by an elastic transform to create a finer match between corresponding points [ZA96, BF03].

In many cases (e.g. [PSRP00]) the user will give an initial indication of correspondence and alignment of fiducial marker(s) (sometimes called the *known point method*), and then an iterative process will adjust parameters within a range of translations and rotations around the initial estimate until the best solution for the global rigid transformation is found. Often hierarchical approaches are employed to simplify computation during the exploration of the parameter search space [Bor88, RSH*99, BF03].

*Optical flow* methods [BFB94] create a vector field indicating the correspondence between two SORs (i.e., the vector field indicates the direction and velocity that each voxel should move with in order to reach its corresponding point). Barron *et al.* [BFB94] give a good review of the various approaches for calculating the flow field.

*Surface-based* or *segmentation-based* methods rely on a segmentation of the surface contained within the data (usually into contour sets). A distance transform is used on one of the contour sets, and then the distances covered by the other contour are used to calculate the root-mean-square distance between the two contour sets [Bor88]. This is minimized by adjusting the translation and rotation between the contour sets (rigid transformation). Hierarchical methods may be used to increase convergence.

In *voxel similarity*, *mutual information* or *relative entropy* methods [SHH96], a 2D histogram is created where each point (*i*, *j*) has the number of voxels that occur with value *i* at a position in the first volume and value *j* at the same position in the second. Maximizing the values in the histogram is equivalent to registering the two volumes. Studholme *et al.* [SHH96] performed this optimization by using a multiresolution method to iterate through various translational and rotational values. Fei *et al.* [BF03] optimized feature correspondence based on the volumetric information around the features, and modelled the local transform using thin-plate splines. Rueckert *et al.* [RSH*99] used a similar method to find the global transform, and modelled the local transform using B-splines.

In general, the best approach (in terms of speed and accuracy) is to determine the global rigid transform hierarchically using optimization. For the local elastic transform, a parametric model (such as, B-splines or thin-plate splines) is used, and then optimized until the approximating transform is found.

### 4.5 Interactive manipulation

A small number of software systems have been developed for interactive modelling of volume objects using procedural tools and operators [WK95, GH91, AS96]. In particular, the metaphor of sculpting and painting has been employed to manipulate volume objects, including both solids (e.g., marble and wood) and soft objects (e.g., clay or wax-like sculptures). For example, a sawing tool may be used to remove a large piece of material from an object, while a heat gun may be used to melt away soft materials on the object.

The use of sculpting metaphor for surface-based geometric modelling has been studied extensively (e.g., [Coq90, Nay90, SPE*90, SP86]). Galyean and Hughes [GH91] first introduced this concept to interactive volume modelling. A number of tools, which are also discretized, were developed, including ‘toothpaste’ for adding voxels, ‘heat gun’ for melting away voxels, ‘sandpaper’ for smoothing an object by wearing away the ridges and filling the valleys. [WK95] extended this approach to include ‘carving’ and ‘sawing’ tools. Particular attention was paid to prevent the aliasing caused by the sculpting process. [AS96] used a force feedback articulated arm to command volume sculpting tools, including ‘paint’, ‘melt’ ‘construct’, ‘burn’, ‘squirt’, ‘stamp’ and ‘airbrush’. [Bær98] proposed an octree-based approach to accelerate volume sculpting.

Recently, [RE00] proposed a hierarchical approach based on the scalar tensor product uniform trivariate B-spline function. The sculpted object is evaluated as zero set of the sum of the collection of the trivariate functions defined over a 3D working space, resulting in multi resolution control capabilities. The continuity of the sculpted object was governed by the continuity of the trivariates. A collection of B-spline patches with arbitrary position, orientation and size was used to represent the scalar filed. To sculpt the volume, user selects the patch in which object is defined using a tool to modify the scalar coefficients. More recently, [BC02] showed that using level sets is beneficial for real-time manipulation of object.

## 5. Deforming Sampled Object Representations

In this survey, the term *deformation* refers to intended change of geometric shape of an object under the control of some external influence such as a force. To facilitate the computation of geometric changes, a *deformable model* normally has two primary components, a data representation and an algorithm based on a physical or mathematical concept. Applications of deformation techniques include computer animation, object modelling, computer-aided illustration, surgical simulation and scientific visualization.

While deformation of SORs is our main focus here, we also consider some techniques designed for AORs and HORs but potentially applicable to SORs, in addition to those operating directly on SORs. The reason for including the former category is to draw our attention to the need for technical developments in the latter category, especially in areas of physically-based deformation. Tables 2 and 3 list some of the previous developments in this area (including those for AORs and HORs), and their main technical characteristics. After a summary of empirical deformable models (Table 2), we briefly examine several major physically-based models (Table 3). We then consider techniques for rendering deformation directly. Finally, we give an overview of deformation techniques in the context of a particular application, namely surgical simulation, where SOR deformation could potentially play a major role.

As shown in Table 3, physically-based models are usually defined on mesh-based data representations. In some applications, such as surgical simulation, it has been common to add analytical information (e.g., connectivity) to sampled data, prior to the application of physically-based models. However, recent research has demonstrated that meshless physically-based models can also be used for deformation, and they are particularly effective in coping with fracturing, liquid or granular effects. In order to give a complete perspective of SOR deformation, we have included many mesh-based deformation models used in a context that is strongly relevant to SORs.

### 5.1 Empirical deformable models

*Empirical deformable models* are non-physically-based deformable models, which are designed to imitate physical behaviours of deformable objects with little or very limited physics in their computation algorithms.

Since SORs contain limited geometrical and topological information, one common approach is to associate a SOR

**Table 2:** *Summary of a collection of empirical deformable models.*

| Reference | Data Representation | Computational Model | Application Context |
|---|---|---|---|
| [Bar84] | Generic solid models | Global/local deformation | |
| [Bar86] | Parametric/implicit surfaces | Space warp | |
| [CBS96] | Implicit surface | Swept objects | |
| [Coq90] | Polygonal mesh | Free-form deformation | |
| [CR94] | Polygonal mesh | Free-form deformation | |
| [CSW*03] | Volume | Spatial transfer function | Volume visualization/animation |
| [CCC06a] | Volume | Displacement map | Illustrative deformation/visualization |
| [FLW93] | Superquadrics | Parametric model | |
| [Gib97] | Volume grid | Chainmail | |
| [GS00] | Volume | Skeleton subspace deformation | Volume animation |
| [HHK92] | Rectangular mesh | b-spline-based FFD | |
| [IDSC04] | Volume | Spatial transfer function | Splitting and explosion |
| [KY95] | Volume | Ray deflectors | Volume visualization |
| [LKHW03] | Polygonal mesh | Level set | |
| [LW94] | Curve, surface, solid | NURBS-based FFD | |
| [MTB03] | Volume | Procedural models | Volume visualization |
| [NC99] | Rectangular mesh | NURBS-based FFD | |
| [PKKG03] | Point-based | Free-form deformation | |
| [RSSSG01] | Volume | Gradient deformation | Hardware-assisted rendering |
| [SP86] | Generic solid | Free-form deformation | |
| [SP91] | Implicit surface | Deformation map | |
| [SS04] | Texture | Skeleton subspace deformation | Volume animation |
| [TM91] | Superquadrics | Parametric model | |
| [WMW86] | Implicit surface | Procedural model | |
| [WGG99] | Implicit surface | Procedural model | |
| [WS01] | Texture | Free-form deformation | |
| [WW92b] | b-spline surface | Parametric model | |

with a geometry-based control structure, which deforms upon the input of a deformation specification and then transfers its geometric changes to the SOR concerned.

Several traditional functional and parametric models mentioned in 2.2 have been applied to SOR deformation, which include:

- applying *global and local deformation* [Bar84, Bar86] to volume datasets through ray deflectors [KY95], spatial transfer functions [CSW*03] and more recently, displacement maps [CCC06a];
- applying free-form deformation [SP86] to volume datasets through volume bounding boxes [CSW*03], and controlling skeleton-based volume deformation using parametric curves [WJ06];
- employing a collection of pre-defined procedural deformation specifications to segmented volume datasets in interactive data exploration [MTB03];
- transforming a volume dataset to an *implicit model*, which is then used to facilitate parametric control for deforming the volume dataset [HQ04];
- applying splitting operations to volume datasets and hypertexture in a combinational manner using spatial transfer functions [IDSC04, ISC07], and applying force to move segments of a volume object apart [BG06].
- applying feature-sensitive operations involving cuts, peels and dissections to volume datasets in interactive illustrative deformation [CCC06b].

On the other hand, SORs are sometimes superimposed upon surface or solid models to assist in deformation computation. For example, level sets were employed for deforming surface-based objects [Whi04, LKHW03], and *deformable distance fields* were used to estimate penetration depth for elastic bodies [FL01].

An interesting development of empirical deformable models is the *chain-mail algorithm* [Gib97], which utilizes the grid topology in a volume dataset to propagate displacement as 'messages'.

In recent years, empirical deformable models have been applied to point-based representations. These include free-form deformation [PKKG03], and haptic-texturing [HBS99].

### 5.2 Physically-based models

Although empirical models can be implemented in real-time for very large datasets, accurate deformations cannot always

**Table 3:** *Summary of a collection of physically-based deformable models.*

| Reference | Data Representation | Computational Model | Application Context |
|---|---|---|---|
| [BAZT04] | Dynamic models | Finite element | Shape recognition in 2D images |
| [BC96] | Tetrahedral mesh | Finite element | Surgical simulation |
| [BS03a] | Spherical mesh | Radial element method | Surgical simulation |
| [BMW01] | Triangular mesh | Mass spring | Soft tissue simulation |
| [CB01] | Long element mesh | Finite element | Surgical simulation |
| [CD99] | Tetrahedral mesh | Finite element | Real time surgical simulation |
| [CZK98b] | Triangular mesh | Mass-spring, finite element | |
| [DCA99] | Tetrahedral mesh | Finite element and tensor mass | Surgical simulation |
| [DDCB01] | Hierarchical tetrahedral mesh | 'Explicit' finite element | Real-time deformation |
| [Jon03] | Volume | Finite differences | Ice melting |
| [KCM00] | Polyhedral mesh, NURBS | Mass spring, finite element | Surgical simulation |
| [KMH*04] | Point-based | Finite element (linear elasticity) | Collision handling |
| [KWT87] | 2D curves on images | Energy minimizing snakes | |
| [MDM*02] | Tetrahedral mesh | Finite element | Real-time deformation |
| [MKN*04] | Point-based | Finite element (linear elasticity) | |
| [MSVS03] | Tetrahedral mesh | Mass spring | Surgical simulation |
| [MTG04] | Hexahedral mesh | Finite element | |
| [NT98] | Triangular mesh | Mass spring | |
| [PDA03] | Tetrahedral mesh | Finite element | Liver laparoscopic, haptic |
| [PLDA00] | Tetrahedral mesh | Finite element | Laparoscopic surgical simulation |
| [PKA*05] | Point-based | Element-free Galerkin method | Fracture simulation |
| [PW89] | Generic solid | Finite element | |
| [RRTP99] | Unstructured surface mesh | Mass spring | Surgical simulation |
| [SBH*00] | Rectangular mesh | Finite element | Laparoscopic surgical simulation |
| [TBHF03] | Triangular mesh | Finite volume | Skeletal muscle simulation |
| [TPBF87] | Curve, surface, solid | Elasticity | |
| [WT04] | Triangular/tetrahedral mesh | Finite element | Soft tissue simulation |
| [WW90] | Generic solid | Low degree-of-freedom | |
| [XHW*05] | Rectangular mesh | Finite element | Heart modelling |
| [ZB05] | Point-based and distance field | Particle-in-cell method | Granular objects and liquids |

be realized, especially in emulating physical responses to an input force. For this reason, there have been many physically-based models proposed for deformation.

As shown in Table 3, almost all physically-based models are associated with a mesh data representation, typically with triangular or rectangular elements for surfaces and tetrahedral or hexahedral elements for solids or volumes. In most applications involving SORs, such data representations can be extracted or reconstructed from SORs using the techniques discussed in Section 4. They can be coupled with SORs and act as the control structure for SOR deformation.

Typical physically-based models include *continuum mechanics*, *mass-spring systems*, *particle systems*, *smoothed particle hydrodynamics* and *fluid dynamics*. In these models, a deformed object is essentially a function of the forces acting on the material properties of the original object. Deformation is computed by finding solution to the equilibrium state of energy functionals. *Finite difference*, *finite element* and *finite volume* methods are commonly used to obtain approximate solutions of mesh-based partial differential equations found in the Lagrangian formulation of motion in continuum mechanics.

For mesh-centered physically-based deformable models, readers are especially encouraged to consult several important surveys, including those mentioned in 2.2, and in particular and the recent Eurographics State-of-the-Art Report by [NMK*06]. To maintain a certain degree of self-containment of this survey, we briefly describe several commonly used approaches to physically-based deformation.

- *Mass-spring Models*—In these models, an object is approximated as a finite mesh of points. The mechanics of deformation is defined as coupled ordinary differential equations, which specifies equilibrium at the mesh points. Vertices are adopted as nodes in a mass-spring model, which are connected via springs to their neighbors. An initial condition can be assigned to each vertex and the internal force acting on a vertex is calculated based upon its local neighbors. This force is then used to calculate vertex motion using Newton's law of motion.

- *Finite Element Methods* (FEM)—Unlike mass-spring models, where the equilibrium equation is discretized and solved at finite mass points, the FEM system is discretized by representing the desired function within each element (e.g., line, triangular, quadrilateral, tetrahedral and hexahedral elements), as a finite sum of element-specific interpolation functions. FEM is used extensively in computer graphics for deformation (e.g., [PW89, BC96]). In computational science, FEM is normally used in conjunction with a non-linear elasticity model, while it is common in computer graphics to employ a reduced linear model as discussed below.
- *Low Degree-of-freedom Models*—For many applications, such as surgical simulations and interactive modelling, real-time solutions are necessary. This class of models are designed to reduce the computational costs of above-mentioned physically-based models. For example, one may use a system of equations that are linearly independent [SPE*90], have a restricted class of deformation functions [WW90], use iterative solutions for the first-order differential equation of deformation [BBC*94], preprocessing non-linearity in high-order differential models [CD99] and linear elasticity theory [MG04, KMH*04].

Point-based data representations are becoming a popular alternative to mesh-based representations in computer graphics. As SORs, they lack in the topological connectivity necessary for the application of most physically-based deformation models. One can either superimpose a mesh structure over a point set to create an HOR, or define neighbour-hood using an approximation, such as moving least squares [MKN*04, KMH*04, MHTG05]. The latter is referred to as *meshless* or *mesh-free* deformation.

### 5.3 Rendering deformation

Traditionally, deformation is performed at the modelling stage, which results in an explicit deformed object to be forwarded to the rendering stage. It is often desirable to couple the modelling and rendering of deformation together to facilitate interactive deformation or reduce the needs for generating explicit deformed objects at each time step. This approach is particularly effective when deformation rendering is accelerated by using modern graphics cards. The major developments in this area include:

- Kurzion and Yagel [KY95, YK97] introduced the concept *ray-deflector* of for deforming volume datasets during the rendering stage by deforming viewing rays using linear or non-linear *deflectors*. In addition to traditional geometric transformation, there are deflectors designed to split and see through external surfaces of a volume object.
- Chen *et al.* [CSW*03] developed the concept of *spatial transfer functions* which enable deformation defined as volume objects in a volume scene graph. Deformation, which can be specified in a hierarchical manner, is realized during the rendering of the scene graph. The concept was deployed for volume visualization, free-form deformation, image-swept volumes, volume and hypertexture splitting, and volume animation [CSW*03, IDSC04, ISC07].
- Correa *et al.* [CCC06a] employed 3D displacement maps as a means for complex illustrative deformation. They generalized the traditional notion of displacement maps to accommodate discontinuous, non-orthogonal and large deformations, and provided a real-time implementation using 3D texture mapping hardware. In addition, they further introduced two feature-aligned methods [CCC06b], which enables users to specify deformation accurately and interactively.
- Westermann and Salama [WS01] utilized 3D texture mapping hardware to achieve interactive deformation of volume datasets. Coupled with backward distortion of 3D texture coordinates, deformation is realized using deformed textures during rendering. Rezk-Salama *et al.* [RSSSG01] used GPU, coupled with edge and face constraints, to facilitate the adaptive subdivision of piecewise patches of a volume object. Georgii and Westermann [GW06] employed a multi-grid approach to achieve real-time simulation and rendering of deformation on a desktop computer.
- Singh and Silver [SSC03] used a proxy geometry to specify manipulation about joints in the volume. Deformation about the joints is achieved with the mid-plane geometry. Bounding boxes are used which can then be moved and rendered with the texture mapping hardware (see also 6.2).
- Müller *et al.* published a series of work on hardware-assisted deformation [MDM*02, MST*04, MG04, MTG04]. They integrated both a plasticity and a fracture model into the pseudo-linear computation of elastic forces. A multi-resolution approach was adopted to simulate object deformation in real time. In their recent work [MKN*04, MHTG05], a meshless approach was employed to deform point-based representations in conjunction with moving least squares approximation.

### 5.4 Deformable models in surgical simulation

In surgery simulation and training, deformable models are required for collision detection, rendering and haptics simulation. When the user interacts through a virtual tool, forces applied to the model produce a deformation, described as a set of displacements of the underlying geometry, and they generate internal forces and vibrations which are fed back to the user as haptic stimuli.

Although non-physical models, such as 3D Chain-Mail [Gib97] and free form deformation [SP86], are computationally inexpensive, physically-based models are the dominant paradigm because of their accuracy. These include mass-spring models [CZK98a, MSVS03] and FEM. Of these two, the latter is more common, because it is more accurate and can accommodate different material properties through a small number of parameters. Furthermore, the focus of most surgical simulation systems is a simulation on localized regions, which FEM can handle properly (i.e., no need to simulate large displacements).

FEM, however, is computationally expensive for real-time simulation, since it requires solving large partial differential equations (PDEs). Techniques for achieving real-time finite element simulation can be classified into two categories: those that simplify the mathematics, and those that speed-up the solution algorithms. In the former category we find approaches that simplify the modeling of elastic tissue using linear models [ZCK98, BC96]. Linear elasticity is often preferred because it reduces the problem to a linear equation that can be solved quickly by pre-computing the inverse of the stiffness matrix. However, linear elasticity only is accurate for small deformations. Large deformations, such as global rotations, usually result in an unrealistic volume growth of the model. For this reason, different techniques have been proposed to handle large deformations, such as warping of the stiffness matrix [MDM*02] and quasi non-linear deformation [CD99]. Zhuang and Canny propose real-time deformation using non-linear elasticity [ZC01]. In surgery simulation, the problem is often described as a dynamic problem. Alternatively, the problem can be reduced to a static problem, which ignores body forces, inertia and energy dissipation. BroNielsen proposed this simplification for surgery simulation for obtaining real-time response [BC96]. However, loss of dynamics may affect the realism of the simulation and static systems are mostly used in surgery planning, where the desired solution is the equilibrium state of the deformable model after being subjected to forces, with no interest in the intermediate states.

The second category for real-time deformation includes techniques for speeding up the solution of the resulting equations. BroNielsen and Cotin proposed a technique based on condensation [BC96], which reduces the size of the PDE to be solved by ignoring the internal finite elements in the computation. They also proposed explicit integration over implicit integration for its reduced computation time and memory requirements. However, explicit integration leads to instability for large time steps. Another possibility for speed-up is the use of multi-resolution techniques, as suggested by Debunne *et al.* [DDCB01] and Wu [WDGT01]. Wu and Tendick propose a multi-grid integration scheme [WT04] to solve non-linear deformations in real-time. Real-time deformation has also been possible with increased computation capabilities, such as parallel processing and specialized hardware [SBH*00, FTBS01] and implementation of matrix solvers on GPUs [BFGS03].

A challenge in surgical simulation that prevents extensive use of pre-computed quantities in FEM is real-time cutting. Pre-computation of the stiffness matrix and applied forces forbids topology changes in the mesh, required for simulating cutting. Cotin *et al.* [CD99] propose a hybrid approach for real-time cutting that uses a static model in regions that do not require topology changes and a dynamic model for a limited region where cutting and tearing is needed.

Other approaches to surgery simulation include tensor-mass models [DCA99, PDA03], long elements methods (LEM) [CB01], later extended to radial elements method (REM) [BS03a], and finite volume methods (FVM) [TBHF03]. Although they differ from FEM, these methods follow the same idea: the sampling of the continuum into elements and the integration of forces and displacements based on the material properties of those elements. Further details of these approaches can be found in [LTCK03].

## 6. Animating Sampled Object Representations

Computer animation refers to the simulation of motion and deformation of objects or figures. While simple rigid-body movements can be achieved by rotation or translations, more complicated animation that involves the movements of articulated figures requires more sophisticated representation schemes and control mechanisms. Object representations that are commonly used in animation usually facilitate at least $G^0$ geometric continuity in object descriptions, as well as controls of motion and deformation via parameters or control-points. These include contour-based figures in 2D cartoon animation, and 3D articulated animation. For instance, in 3D computer animation, digital characters are commonly represented by articulated figures which 'are hierarchical structures composed of a set of rigid links that are connected at rotary joints' [HB03].

However, when a digital character is captured in a SOR, the much desirable geometrical, topological and semantic information is not available for animating the digital character. There are so far only a few pieces of research that have been reported in the literatures. In this section, we first examine techniques for animating 2D digital characters in images. This is followed by discussions on the animation (and movement) of 3D volumetric objects.

### 6.1 Animating digital characters in images

In the studio production of digital composition, image quality plays a critical role. There are really two aspects here: animating digital characters which are images of fixed resolution and the resolution of the composite images which are the final product.

The demand for high-resolution raises the memory loading. Froumentin *et al.* [FLW00] developed a 2.5D rendering

(a)snake model of swan (b) arbitrary warping

(c) Digital composition: left: original image;
right: snake-boundary representation.

**Figure 2:** *2D SORs can be deformed and animated using a 2.5D rendering and composition system.*

and composition system for animating image-based digital character. Basically, their method is based on image compositing using either 2D geometric shapes or raster images as input primitives. The resolution of the final image is virtually unlimited.

Traditional interpolation methods will blur the sharpness of edges and degrade the quality of texture features, especially the tiny texture features under warping and morphing operations. That is, visual flaws can become noticeable when part of an image that represents a digital character is under deformation. Thus, they develop a continuous model for such image-based characters, which maintains the sharpness of edges and the smoothness of flat areas. As demonstrated by Froumentin *et al.* in [FLW00], snakes can be used to define edges independently of the pixel resolution. Their snakes in fact pass 'through' pixels and precisely where is determined by the image data itself. As photographic images seldom have areas of constant color (unlike cartoons), they segment the image on texture. It is these areas of constant texture which are then bounded by closed snake curves.

As shown in Figure 2, this representation of the goose maintains the edge information well and thus the composition quality is maintained, even when the required output is at much higher resolution than the raw image data (right).

The previous method requires some user-intervention to identify the areas of constant texture. In an alternative approach, Su and Willis [SW04] have investigated using image interpolation techniques to represent an image at any resolution, independent of the original sampling rate. They again note that edges are important visually and so these should be retained when resampling an image. Each group of four pixels is tested, on the assumption that an edge passes between them. If so, one pixel value will be an outlier and the edge can be presumed to pass between it and its companions. Rather than examine the edge further, they use a single extra bit to indicate which diagonal is closest to the direction of this edge. As this is a very local test, they then run over these bits and complement any bit which is not in the majority of those around it, in effect extending the edge determination to an area of four by four pixels. These diagonals, in conjunction with the four surrounding pixels, create a triangulation of the entire image. Simple bilinear interpolation within any triangle gives the value at any required point. To resample an image, the new sampling grid is logically placed over the triangulated image and the point values calculated. As this is equivalent to Gouraud shading, this can be done in real-time on proprietary graphics cards but is in any case simple enough to be fast in any implementation. The samples can of course be rotated if the image required is a rotated one; or be non-uniformly placed, if a warped image is wanted. The resulting images are visually as good as more complex methods.

The above image interpolation method was also used in the Quasi-3D animation system of Qi and Willis [QW03], which builds on the mentioned work from Labrosse, Froumentin in their earlier work with Willis. Traditional cartoon animation uses a stack of painted cels, with the layer ordering determining visibility. Each sack is photographed to make one frame of animation, then one or more elements are changed and a new frame photographed. Qi and Willis extended this approach for computer use. In their system, every cell is a digital image. They retain the layering but permit the cels to be anywhere in 3D. This includes intersecting cels. Each cel can be animated, independently lit from front and behind, and moved around in 3D. As a result of this freedom of layout, the user has more scope to construct the animated world. Conventional pixel images are supported, when the system has some parallel with imposters in virtual reality. However, they can also use the interpolated images of Su and Willis, meaning that the rendering quality is automatically adjusted to that of the final image. Moreover, the ability to light each cel from front and back extends the visual effects possible. Finally, they use the beta channel color model [Odd91], which permits multi-layer transparency, color blending and front and back lighting, of a realism not possible with alpha channel.

In-between frames are calculated semi-automatically in traditional animation productions. However, in motion capture systems [BLCD02], undesirable visual artefacts can be introduced by in-between frames.

The main idea in a motion capture cartoon system is to parameterize cartoon motion with a combination of affine-transformations and key-weight vectors. Thus, the solution to the problem of inconsistent in-between frames is to construct the extended linear space for every combination of hand-picked key-shapes and then generate mean-shapes in between key-shapes, by solving the affine matrix and weight coefficients.

### 6.2 Block-based volume animation

The work in volume animation is focused on repositioning an acquired volumetric model into a new pose, for example, making the visible man dataset sit or walk. Most of the techniques below do not model deformations, that is, only the voxels are moving and the muscles are not deforming. Clearly, for large movements, breakage could occur at joints. Furthermore, since now a volume is available, the issue of how and which internal structures of the volume move about a particular joint is a difficult problem. Below is a summary of the methods which utilized block-like bounds to group parts of the volume together for manipulation:

- Wu and Prakash [WP00] first proposed a block-based approach for controlling the motion of the visible man dataset. The volumetric representation is first dissected into blocks of voxels, each representing a major segment of the digital character. For each movement, the deformation of the block structure is computed using the finite element method. Each deformed block is then re-voxelized using 3D texture mapping. The combination of these re-voxelized blocks represents the motion of the digital character at each time step.
- Chen, Silver and others [CSW*03] employed the concepts of spatial transfer function and constructive volume geometry to achieve the block-based animation without the need for re-voxelization at each time step.
- Singh and Silver [SSC03, SS04] also decomposed a volume into blocks in order to animate a digital characters in volume datasets. The blocks were used in conjunction to a skeleton. This enabled real-time manipulations along the IK-skeleton since each block could be transformed and rendered using the texture memory. The blocks were stretched about the joints to prevent breaking.
- Islam *et al.* [IDSC04] further developed this approach by incorporating volume splitting into the deformation, facilitating an animation series with motion and explosion They have also studied the scalability of animation modeling and rendering in terms of the number of blocks used.

### 6.3 Skeleton-based volume animation

In conventional 3D computer animation, digital characters are animated using a ‘skeleton’, which is a stick figure representation of the object. The animator first creates a skeleton of the model (called IK skeleton) and then binds the polygons to the skeleton. The skeleton can then be manipulated or animated to cause the corresponding movement in the model using key framing, inverse kinematics or motion capture. This process is adopted by many commercial animation packages.

Gagvani and Silver [GS99, GS00, GS01] developed a methodology for animating volumetric models similar to the process used for surface polygonal models. This process includes three steps:

1. skeletonization and attachment of volume to skeleton,
2. modification or manipulation of the skeleton and
3. reconstruction of the volume and rendering.

In this process, a new volume is created for each manipulation of the skeleton, similar to a key-frame process. The first step is to skeletonize a volumetric representation of an object to be animated and to choose bones/joints to form the IK skeleton. The IK skeleton can be computed using a skeletonization algorithm (described in 4.1.3). In [GS00] a thinning procedure is used to first thin the volume based upon the distance field and then the animator chooses joints and bones. While the joints and bones (IK skeleton) are used for animation, the ‘thinned volume’ is used to reconstruct the volume in the final step. Because a volume is available, the IK-skeleton can either be ‘centered’ within the volume as in traditional computer graphics, or can lie along the actual skeleton of the volume if one is available (as in the case of a human, or an animal).

The IK-skeleton is then manipulated. This can be done in any animation package such as Character Studio [Dis02], or Maya [Ali02] by applying standard animation techniques such as key-framing, inverse kinematics or even motion capture. Transformations are applied to the IK-skeleton, which in turn specifies the transformation on the ‘thinned volume’. Each of the voxels in the thin volume has an associated distance field, which specifies a solid sphere of texture centered at that voxel. The volumetric object is reconstructed by scan-filling the solid spheres about the transformed joint while inversely mapping into the original volume for correct sample values. The deformation about the joints is an empirical solution based upon kinematics. The spheres act as uncoupled volume particles. When a sufficient number of spheres are placed about the joints, breakage at joints will not occur and a smooth deformation can be achieved [GS01].

An example of the visible man jumping rope is shown in Figure 3. Each frame is created by mapping motion-capture data (a jump-rope sequence) to an IK-skeleton of the visible man and reconstructing a volume around each key frame. Thus each frame is a new 3D volume in the new pose. Once the volume is created, it can be rendered with standard volume rendering algorithms. Because a new volume is created, it is available to be used in any application that takes a 3D volume as input.

## 7. Conclusions

In this state-of-the-art report, we have examined a technical challenge for deforming and animating SORs. We have outlined the overall scope for this challenge and examined a family of methods and techniques that have been developed for manipulating, deforming and animating SORs. In

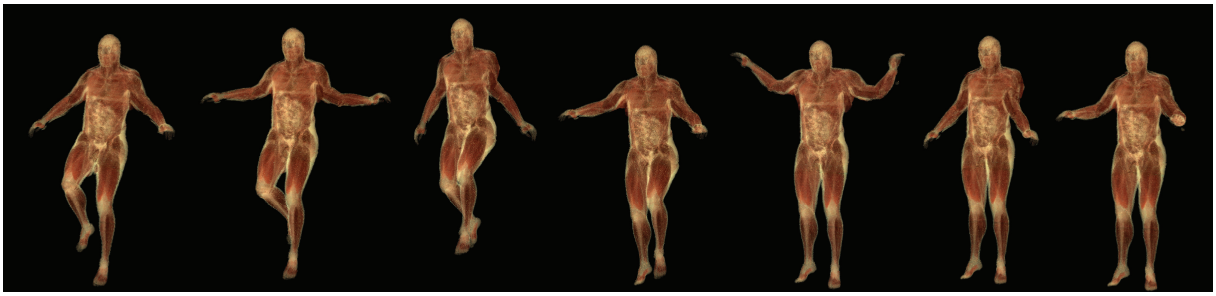

**Figure 3:** *Animating the Visible Man dataset with complex movements using a skeleton-based approach [GS 01].*

**Table 4:** *Summary of the volumes of published work on techniques for manipulating, deforming and animating SORs, AORs and HORs.*

| Techniques | AORs and HORs | SORs |
|---|---|---|
| Manipulation | Relatively small | Very large |
| Deformation | | |
| (*Empirical*) | Reasonably large | Comparable |
| (*Physical*) | Relatively large | Very small |
| Animation | Very large | Very small |

comparison with AORs and HORs, we have observed a significant imbalance between techniques developed for these two types of data representations, as summarized in Table 4.

In general, techniques for manipulating SORs have reached a relatively mature status, with many well-studied technical problems and solutions. Techniques for deforming SORs still rely heavily on those originally developed for surface and solid object representations. However, recent advances in meshless deformation techniques, as well as in close coupling of deformation and rendering of SORs, have opened an exciting new frontier. In terms of animating SORs, although there are several major breakthroughs, the overall effort made in this area is so far limited.

Considering the rapid advances in imaging technology as well as SOR rendering technology, it is highly desirable to take the research and development in aspects of deforming and animating SORs to a new level.

## Acknowledgments

This work is conducted under the framework of the *van* project, which is funded by an EPSRC programme with grants GR/S44198 and GR/S44204, which were awarded to Swansea and Bath respectively, and for which Professor Silver (Rutgers) is a guest partner and adviser. This survey is also based in part upon work supported by the National Science Foundation under award number 011876 for D. Silver. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

[AB02] AYLWARD S. R., BULLITT E.: Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE Transactions on Medical Imaging 21*, 2 (2002), 61–75.

[ABCO∗01] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Point set surfaces. In *Proc. IEEE Visualization* (2001), pp. 21–28.

[ABK98] AMENTA N., BERN M., KAMVYSSELIS M.: A new Voronoi-based surface reconstruction algorithm. In *Computer Graphics (Proc. SIGGRAPH 98)* (1998), pp. 415–421.

[AF97] AHMED M. N., FARAG A. A.: Two-stage neural network for volume segmentation of medical images. *Pattern Recognition Letters 18*, 11–13 (1997), 1143–1151.

[AJWB01] AYLWARD S. R., JOMIER J., WEEKS S., BULLITT E.: Registration and analysis of vascular images. *International Journal of Computer Vision 55*, 2-3 (2001), 123–138.

[AK04] AMENTA N., KIL Y. J.: Defining point-set surfaces. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003) 22*, 3 (2004), 264–270.

[Ali02] ALIAS WAVEFRONT: Maya: A 3d animation and visual effects tool, 2002. http://www.aliaswavefront.com/en/products/maya/index.shtml.

[AS96] AVILA R. S., SOBIERAJSK L. M.: A haptic interaction method for volume visualization. In *Proc. IEEE Symposium on Volume Visualization* (1996), pp. 197–204.

[Bær98] BÆRENTZEN A.: Octree-based volume sculpting. In *Proc. IEEE Visualisation Late Breaking Hot Topics* (1998).

[Bar84] BARR A.: Global and local deformation of solid primitives. *Computer Graphics (Proc. SIGGRAPH 84) 18*, 3 (1984), 21–30.

[Bar86] BARR A.: Ray tracing deformed surfaces. *Computer Graphics (Proc. SIGGRAPH 86) 20*, 4 (1986), 287–296.

[BAZT04] BERGNER S., AL-ZUBI S., TONNIES K.: Deformable structural models. In *Proc. International Conference on Image Processing* (2004), vol. 3, pp. 1875–1878.

[BBC∗94] BARRETT R., BERRY M., CHAN T. F., DEMMEL J., DONATO J., DONGARRA J., EIJKHOUT V., POZO R., ROMINE C., VAN DER VORST H.: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd ed. SAIM, 1994.

[BBZ91] BADLER N. I., BARSKY B. A., ZELTZER D. (Eds.): *Making Them Move: Mechanics, Control and Animation of Articulated Figures*. Morgan Kaufmann, 1991.

[BC96] BRO-NIELSEN M., COTIN S.: Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum 15*, 3 (1996), 57–66.

[BC02] BæRENTZEN J. A., CHRISTENSEN N. J.: Interactive modelling of shapes using the level-set method. *International Journal on Shape Modeling 8*, 2 (2002), 79–97.

[BF03] B. FEIA C. KEMPERA D. L. W.: A comparative study of warping and rigid body registration for the prostate and pelvic MR volumes. *Computerized Medical Imaging and Graphics 27*, 4 (2003), 267–281.

[BFB94] BARRON J. L., FLEET D. J., BEAUCHEMIN S. S.: Performance of optical flow techniques. *International Journal of Computer Vision 12*, 1 (1994), 43–77.

[BFGS03] BOLZ J., FARMER I., GRINSPUN E., SCHRÖDER P.: Sparse matrix solvers on the GPU: conjugate gradients and multigrid. *ACM Transactions on Graphics 22*, 3 (2003), 917–924.

[BG06] BRUCKNER S., GRÖLLER M. E.: Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1077–1084.

[BLCD02] BREGLER C., LOEB L., CHUANG E., DESHPANDE H.: Turing to the masters: motion capturing cartoons. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (2002), 399–407.

[Blo02] BLOOMENTHAL J.: Medial based vertex deformation. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), pp. 147–151.

[Blu67] BLUM H.: A transformation for extraction new descriptors of shape. In *Models for the Perception of Speech and Visual Form*,WATHEN-DUNN W., (Ed.). MIT Press, 1967, pp. 362–380.

[BMR∗99] BERNARDINI F., MITTLEMAN J., RUSHMEIER H., SILVA C., TAUBIN G.: The ball-pivoting algorithm for surface. *IEEE Transactions on Visualization and Computer Graphics 5*, 4 (1999), 349–359.

[BMW01] BRUYNS C., MONTGOMERY K., WILDERMUTH S.: A virtual environment for simulated rat dissection: A case study of visualization for astronaut training. In *Proc. IEEE Visualization* (2001), pp. 21–26.

[BN92] BEIER T., NEELY S.: Feature-based image metamorphosis. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 35–42.

[Bor88] BORGEFORS G.: Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 10*, 6 (1988), 849–865.

[BPS96] BAJAJ C. L., PASCUCCI V., SCHIKORE D. R.: Fast isocontouring for improved interactivity. In *Proc. IEEE Symposium on Volume Visualization* (1996), pp. 39–46.

[Bro92] BROWN L. G.: A survey of image registration techniques. *ACM Computing Surveys 24*, 4 (1992), 326–376.

[BS03a] BALANIUK R., SALISBURY JR. J. K.: Soft-tissue simulation using the radial elements method. In *Proc. International Symposium on Surgery Simulation and Soft Tissue Modeling* (2003), pp. 48–58.

[BS03b] BORDOLOI U. D., SHEN H.-W.: Space efficient fast isosurface extraction for large datasets. In *Proc. IEEE Visualization* (2003), pp. 201–208.

[BW01] BREEN D. E., WHITAKER R. T.: A level-set approach for the metamorphosis of solid models. *IEEE Transactions on Visualization and Computer Graphics 7*, 2 (2001), 173–192.

[BWC00] BHANIRAMKA P., WENGER R., CRAWFIS R.: Isosurfacing in higher dimensions. In *Proc. IEEE Visualization* (2000), pp. 267–274, 566.

[Cas96] CASTLEMAN K. R.: *Digital Image Processing*. Prentice-Hall, 1996.

[CB01] COSTA I. F., BALANIUK R.: LEM – an approach for real time physically based soft tissue simulation. In *Proc. IEEE International Conference on Robotics and Automation* (2001), pp. 2337–2343.

[CBC01] CARR J. C., BETSON R. K., CHERRIE J. B.: Reconstruction and representation of 3D objects with radial basis functions. In *Computer Graphics (Proc. SIGGRAPH 2001)* (2001), pp. 67–76.

[CBS96] CRESPIN B., BLANC C., SCHLICK C.: Implicit sweep objects. *Computer Graphics Forum 13*, 3 (1996), 165–174.

[CCC06a] C. CORREA D. S., CHEN M.: Discontinuous displacement mapping for volume graphics. In *Proc. Volume Graphics* (2006), pp. 9–16.

[CCC06b] C. CORREA D. S., CHEN M.: Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1069–1076.

[CD99] COTIN S., DELINGETTE H.: Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics 5*, 1 (1999), 62–71.

[CF01] CAMPBELL R. J., FLYNN P. J.: A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding 81*, 2 (2001), 166–210.

[Che95] CHEN S. E.: QuickTime VR – an image-based approach to virtual environment navigation. In *Computer Graphics (Proc. SIGGRAPH 95)*. Addison-Wesley, 1995, pp. 29–38.

[CJT95] CHEN M., JONES M. W., TOWNSEND P.: Methods for volume metamorphosis. In *Image Processing for Broadcast and Video Production*,PAKER Y., WILBUR S., (Eds.). Springer-Verlag, 1995.

[CJT96] CHEN M., JONES M. W., TOWNSEND P.: Volume distortion and morphing using disk fields. *Computers and Graphics 20*, 4 (1996), 567–575.

[CK95] COHEN D., KAUFMAN A.: Fundamentals of surface voxelization. *CVGIP Graphics Models and Image Processing 57*, 6 (1995), 453–461.

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *Computer Graphics (Proc. SIGGRAPH 96)* (1996), Addison-Wesley, pp. 303–312.

[CLL∗88] CLINE H. E., LORENSEN W. E., LUDKE S., CRAWFORD C. R., TEETER B. C.: Two algorithms for the three-dimensional reconstruction of tomograms. *Medical Physics 15*, 3 (1988), 320–327.

[CLS98] COHEN-OR D., LEVIN D., SOLOMIVICI A.: Three-dimensional distance field metamorphosis. *ACM Transactions on Graphics 17*, 2 (1998), 116–141.

[CMM∗97] CIGNONI P., MARINO P., MONTANI E., PUPPO E., SCOPIGNO R.: Speeding up isosurface extraction using interval trees. *IEEE Transactions on Visualization and Computer Graphics 3*, 2 (1997), 158–170.

[CMPS96] CIGNONI P., MONTANI C., PUPPO E., SCOPIGNO R.: Optimal isosurface extraction from irregular volume data. In *Proc. IEEE Symposium on Volume Visualization* (1996), pp. 31–38.

[Coq90] COQUILLART S.: Extended free-form deformation: Asculpting tool for 3D geometric modeling. *Computer Graphics (Proc. SIGGRAPH 90) 24*, 4 (1990), 187–196.

[CR94] CHANG Y., ROCKWOOD A. P.: A generalized de casteljau approach to 3D free-form deformation. In *Computer Graphics (Proc. SIGGRAPH 94)*. ACM Press and Addison-Wesley, 1994, pp. 257–260.

[CS97] CHIANG Y.-J., SILVA C. T.: I/O optimal isosurface extraction. In *Proc. IEEE Visualization* (1997), pp. 293–300.

[CSM05] CORNEA N., SILVER D., MIN P.: Curve skeleton applications. In *Proc. IEEE Visualization* (2005), pp. 95–102.

[CSM07] CORNEA N. D., SILVER D., MIN P.: Curve-skeleton properties, applications and algorithms. *IEEE Transactions on Visualization and Computer Graphics* (2007). to appear.

[CSv04] CARR H., SNOEYINK J., VAN DE PANNE M.: Simplifying flexible isosurfaces using local geometric measures. In *Proc. IEEE Visualization* (2004), pp. 497–504.

[CSW∗03] CHEN M., SILVER D., WINTER A. S., SINGH V., CORNEA N.: Spatial transfer functions – a unified approach to specifying deformation in volume modeling and animation. In *Proc. Volume Graphics* (2003), pp. 35–44.

[CT00] CHEN M., TUCKER J.: Constructive volume geometry. *Computer Graphics Forum 19*, 4 (2000), 281–293.

[CWPG04] COTTING D., WEYRICH T., PAULY M., GROSS M.: Robust watermarking of point-sampled geometry. In *Proc. Shape Modeling International* (2004), pp. 233–242.

[CZK98a] CHEN Y., ZHU Q., KAUFMAN A.: Physically-based animation of volumetric objects. In *Proc. IEEE Computer Animation* (1998), pp. 154–160.

[CZK98b] CHEN Y., ZHU Q., KAUFMAN A.: Real-time biomechanically-based muscle volume deformation using FEM. In *Proc. IEEE Computer Animation* (1998), pp. 154–160.

[DC03] DANIEL G., CHEN M.: Video visualization. In *Proc. IEEE Visualization* (2003), pp. 409–416.

[DCA99] DELINGETTE H., COTIN S., AYACHE N.: A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. In *Proc. IEEE Computer Animation* (1999), pp. 70–81.

[DDCB01] DEBUNNE G., DESBRUN M., CANI M.-P., BARR A. H.: Dynamic real-time deformations using space & time adaptive sampling. In *Computer Graphics (Proc. SIGGRAPH 2001)* (2001), pp. 31–36.

[Dis02] DISCREET: Character studio animation toolkit, 2002. http://www.discreet.com/products/cs/.

[Dis04] DISCREET: 3D Studio Max, (3DSMax), 2004. http://www.discreet.com/.

[DM97] DEBEVEC P. E., MALIK J.: Recovering high dynamic range radiance maps from photographs. In *Computer Graphics (Proc. SIGGRAPH 97)* (1997), ACM Press, pp. 369–378.

[FL01] FISHER S., LIN M.: Fast penetration depth estimation for elastic bodies using deformed distance fields. In *Proc. IEEE Intelligent Robots and Systems* (2001), vol. 1, pp. 330–336.

[FLW93] FERRIE F. P., LAGARDE J., WHAITE P.: Darboux frames, snakes, and super-quadrics: Geometry from the bottom up. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 8 (1993), 771–783.

[FLW00] FROUMENTIN M., LABROSSE F., WILLIS P.: A vector-based representation for image warping. *Computer Graphics Forum 19*, 3 (2000), C385–C394, C428.

[FRZ∗05] FREEDMAN D., RADKE R. J., ZHANG T., JEONG Y., LOVELOCK D. M., CHEN G. T. Y.: Model-based segmentation of medical imagery by matching distributions. *IEEE Transactions on Medical Imaging 24*, 3 (2005), 281–292.

[FSRR00] FANG S., SRINIVASAN R., RAGHAVAN R., RICHTSMEIER J.: Volume morphing and rendering – an integrated approach. *Computer Aided Geometric Design 17*, 1 (2000), 59–81.

[FTBS01] FRANK A. O., TWOMBLY I. A., BARTH T. J., SMITH J. D.: Finite element methods for real-time haptic feedback of soft-tissue models in virtual reality simulators. In *Proc. Virtual Reality* (2001), pp. 257–263.

[FvDFH96] FOLEY J., VAN DAM A., FEINER S., HUGHES J.: *Computer Graphics: Principles and Practice*, 2nd ed. Addison Wesley, 1996.

[Gar99] GARLAND M.: Multiresolution modeling: survey and future opportunities. In *Eurographics State of the Art Report* (1999), pp. 111–131.

[GH90] GILES M., HAIMES R.: Advanced interactive visualization for cfd. *Computing Systems in Engineering 1*, 1 (1990), 51–62.

[GH91] GALYEAN T. A., HUGHES J. F.: Sculpting: an interactive volumetric modeling technique. *Computer Graphics (Proc. SIGGRAPH 91) 25*, 4 (1991), 267–274.

[Gib97] GIBSON S.: 3D chainmail: a fast algorithm for deforming volumetric objects. In *Proc. Symposium on Interactive 3D Graphics* (1997), pp. 149–154.

[GM97] GIBSON S., MIRTICH: *A survey of deformable modeling in computer graphics*. Tech. Rep. TR97-19, MERL Technical Report, 1997.

[GMA∗04] GRAU V., MEWES A. U. J., ALCANIZ M., KIKINIS R., WARFIELD S. K.: Improved watershed transform for medical image segmentation using prior information. *IEEE Transactions on Medical Imaging 23*, 4 (2004), 447–458.

[GP00] GERSTNER T., PAJAROLA R.: Topology preserving and controlled topology simplifying multiresolution isosurface extraction. In *Proc. IEEE Visualization* (2000), pp. 259–266, 565.

[GS99] GAGVANI N., SILVER D.: Parameter controlled volume thinning. *Graphical Models and Image Processing 61*, 3 (1999), 149–164.

[GS00] GAGVANI N., SILVER D.: Animating the visible human dataset. In *Proc. the 3rd Visible Human Project Conference* (2000). http://www.nlm.nih.gov/research/visible/vhpconf2000/.

[GS01] GAGVANI N., SILVER D.: Animating volumetric models. *Graphical Models 63*, 6 (2001), 443–458.

[GTH05] GHOSH A., TRENTACOSTE M., HEIDRICH W.: Volume rendering for high dynamic range displays. In *Proc. Volume Graphics* (2005), pp. 91–98, 231.

[GW01] GONZALEZ R., WOODS R. E.: *Digital Image Processing*, 2nd ed. Prentice-Hall, 2001.

[GW06] GEORGII J., WESTERMANNA R.: A multigrid framework for real-time simulation of deformable bodies. *Computers and Graphics 30*, 3 (2006), 408–415.

[HB03] HEARN D., BAKER M. P.: *Computer Graphics with OpenGL*, 3rd ed. Prentice-Hall, 2003.

[HBS99] HO C.-H., BASDOGAN C., SRINIVASAN M. A.: Efficient point-based rendering techniques for haptic display of virtual objects. *Presence 8*, 5 (1999), 477–491.

[HDD∗92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. *Computer Graphics (Proc. SIGGRAPH 86) 20*, 4 (1992), 295–302.

[HDD∗93] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Mesh optimization. In *Computer Graphics (Proc. SIGGRAPH 93)* (1993), pp. 19–26.

[HHCL01] HE T., HONG L., CHEN D., LIANG Z.: Reliable path for virtual endoscopy: Ensuring complete examination of human organs. *IEEE Transactions on Visualization and Computer Graphics 7*, 4 (2001), 333–342.

[HHK92] HSU W. M., HUGHES J. F., KAUFMAN H.: Direct manipulation of free form deformation. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 177–184.

[HL78] HERMAN G. T., LIU H. K.: Dynamic boundary surface detection. *Computer Graphics and Image Processing 7* (1978), 130–138.

[Hop96] HOPPE H.: Progressive meshes. In *Computer Graphics (Proc. SIGGRAPH 97)* (1996), pp. 99–108.

[HP03] HAHN H. K., PEITGEN H.: IWT-interactive watershed transform: a hierarchical method for efficient interactive and automated segmentation of multidimensional gray-scale images. In *Proc. Medical Imaging* (2003), vol. SPIE 5032, pp. 643–653.

[HQ04] HUA J., QIN H.: Haptics-based dynamic implicit solid modeling. *IEEE Transactions on Visualization and Computer Graphics 10*, 5 (2004), 574–586.

[Hug92] HUGHES J. F.: Scheduled fourier volume morphing. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 43–46.

[HWK94] HE T., WANG S., KAUFMAN A.: Wavelet-based volume morphing. In *Proc. IEEE Visualization* (1994), pp. 85–91.

[IDSC04] ISLAM S., DIPANKAR S., SILVER D., CHEN M.: Temporal and spatial splitting of scalar fields in volume graphics. In *Proc. IEEE Symposium on Volume Visualization and Graphics* (2004), pp. 87–94.

[IK95] ITOH T., KOYAMADA K.: Automatic isosurface propagation using an extrema graph and sorted boundary cell lists. *IEEE Transactions on Visualization and Computer Graphics 1*, 4 (1995), 319–327.

[ISC07] ISLAM S., SILVER D., CHEN M.: Volume splitting and its applications. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007). to appear.

[JBŠ06] JONES M. W., BÆRENTZEN A., ŠRÁMEK M.: Discrete 3D distance fields: Techniques and applications. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (2006), 581–599.

[Jon95] JONES M. W.: *The Visualisation of Regular Three Dimensional Data*. PhD thesis, University of Wales, 1995.

[Jon96] JONES M. W.: The production of volume data from triangular meshes using voxelisation. *Computer Graphics Forum 15*, 5 (1996), 311–318.

[Jon01] JONES M. W.: Facial reconstruction using volumetric data. In *Proc. Vision, Modeling, and Visualisation* (2001), pp. 135–142.

[Jon03] JONES M. W.: Melting objects. *The Journal of WSCG 11*, 2 (2003), 247–254.

[Jon04] JONES M.: Distance field compression. *Journal of WSCG 12*, 2 (2004), 247–254.

[JS01] JONES M. W., SATHERLEY R.: Shape representation using space filled sub-voxel distance fields. In *Proc. Shape Modelling and Applications* (2001), pp. 316–325.

[KCM00] KUHNAPFEL U., CAKMAK H. K., MASS H.: Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computers and Graphics 24*, 5 (2000), 671–682.

[KCY93] KAUFMAN A., COHEN D., YAGEL R.: Volume graphics. *IEEE Computer 26*, 7 (1993), 51–64.

[KEK03] KANG Y., ENGELKE K., KALENDER W. A.: A new accurate and precise 3-D segmentation method for skeletal structures in volumetric CT data. *IEEE Transactions on Medical Imaging 22*, 5 (2003), 586–598.

[Ker04] KERLOW I. V.: *The Art of 3D Computer Animation and Effects*. John Wiley & Sons, Inc, 2004.

[KFW*02] KANITSAR A., FLEISCHMANN D., WEGENKITTL R., FELKEL P., GRÖLLER M. E.: CPR: Curved planar reformation. In *Proc. IEEE Visualization* (2002), pp. 37–44.

[KK99] KREEGER K., KAUFMAN A.: Interactive volume segmentation with the PAVLOV architecture. In *Proc. IEEE Symposium on Parallel Visualization and Graphics* (1999), pp. 61–68.

[KL96] KRISHNAMURTHY V., LEVOY M.: Fitting smooth surfaces to dense polygon meshes. In *Computer*

*Graphics (Proc. SIGGRAPH 96)* (1996), pp. 313–324.

[KMH*04] KEISER R., MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Contact handling for deformable point-based objects. In *Proc. Vision, Modeling, Visualization* (2004), pp. 315–322.

[KWT87] KASS M., WITKIN A., TERZOPOULOS D.: Snakes: Active contour models. *International Journal of Computer Vision 1*, 4 (1987), 321–331.

[KY95] KURZION Y., YAGEL R.: Space deformation using ray deflectors. In *Proc. 6th Eurographics Workshop on Rendering* (1995), pp. 21–32.

[LBC96] LEI Z., BLANE M. M., COOPER D. B.: 3L fitting of higher degree implicit polynomials. In *Proc. 3rd IEEE Workshop on Applications of Computer Vision* (1996), pp. 148–153.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics (Proc. SIGGRAPH 87) 21*, 4 (1987), 163–169.

[Lee00] LEE I.-K.: Curve reconstruction from unorganized points. *Computer Aided Geometric Design 17*, 2 (2000), 161–177.

[Lev88] LEVOY M.: Display of surfaces from volume data. *IEEE Computer Graphics and Applications 8*, 3 (1988), 29–37.

[Ley03] LEYMARIE F.: *3D Shape Representation via the Shock Scaffold*. PhD thesis, Brown University, 2003.

[LGL95] LERIOS A., GARFINKLE C., LEVOY M.: Feature-based volume metamorphosis. In *Computer Graphics (Proc. SIGGRAPH 95)* (1995), pp. 449–456.

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Computer Graphics (Proc. SIGGRAPH 96)* (1996), pp. 31–42.

[LJT01] LIN Z., JIN J., TALBOT H.: Unseeded region growing for 3D image segmentation. In *Proc. CRPITS '00: Selected Papers from the Pan-Sydney Workshop on Visualisation* (2001), pp. 31–37.

[LK03] LAKARE S., KAUFMAN A.: Anti-aliased volume extraction. In *Proc. Eurographics/IEEE TCVG Symposium on Data Visualisation* (2003), pp. 113–122.

[LKHW03] LEFOHN A. E., KNISS J. M., HANSEN C. D., WHITAKER R. T.: Interactive deformation and visualization of level set surfaces using graphics hardware. In *Proc. IEEE Visualization* (2003), pp. 75–82.

[LPC*00] LEVOY M., PULLI K., CURLESS B., RUSINKIEWICZ S., KOLLER D., PEREIRA L., GINZTON M., ANDERSON S., DAVIS J., GINSBERG J., SHADE J., FULK D.: The digital Michelangelo project: 3D scanning of large statues. In *Computer Graphics (Proc. SIGGRAPH 2000)* (2000), pp. 131–144.

[LSJ96] LIVNAT Y., SHEN H. W., JOHNSON C. R.: A near optimal isosurface extraction algorithm using the span space. *IEEE Transactions on Visualization and Computer Graphics 2*, 1 (1996), 73–84.

[LTCK03] LIU A., TENDICK F., CLEARY K., KAUFMANN C.: A survey of surgical simulation: applications, technology and education. *Presence 12*, 6 (2003), 599–614.

[LW94] LAMOUSIN H. J., WGGENSPACK W. N.: Nurbs-based free-form deformations. *IEEE Computer Graphics and Application 14*, 6 (1994), 59–65.

[LW98] LEMOS R., WYVILL B.: *A survey of layered construction techniques using deformable models in the animation of articulated figures*. Tech. Rep. TR1998-637-28, University of Calgary, 1998.

[LWM*03] LIU P., WU F., MA W., LIANG R., OUHYOUNG M.: Automatic animation skeleton construction using repulsive force field. In *Proc. Pacific Graphics* (2003), pp. 409–413.

[Mal93] MALZBENDER T.: Fourier volume rendering. *ACM Transactions on Graphics 12*, 3 (1993), 233–250.

[MBL*91] MILLER J. V., BREEN D. E., LORENSEN W. E., O'BARA R. M., WOZNY M. J.: Geometrically deformed models: a method for extracting closed geometric models from volume data. *Computer Graphics (Proc. SIGGRAPH 91) 25*, 4 (1991), 217–226.

[MBR*00] MATUSIK W., BUEHLER C., RASKAR R., GORTLER S. J., MCMILLAN L.: Image-based visual hulls. In *Computer Graphics (Proc. SIGGRAPH 2000)* (2000), pp. 343–352.

[MCS*02] MAKELA T., CLARYSSE P., SIPILA O., PAUNA N., PHAM Q. C., KATILA T., MAGNIN I. E.: A review of cardiac image registration methods. *IEEE Transactions on Medical Imaging 21*, 9 (2002), 1011–1021.

[MDM*02] MÜLLER M., DORSEY J., MCMILLAN L., JAGNOW R., CUTLER B.: Stable real-time deformations. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), pp. 49–54.

[MG04] MÜLLER M., GROSS M.: Interactive virtual materials. In *Proc. Graphics Interface* (2004), pp. 239–246.

[MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005) 24*, 3 (2005), 471–478.

[MKN*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., CROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Proc. ACM SIGGRAPH Symposium on Computer Animation* (2004), pp. 141–151.

[Mor90] MORGAN II J. C.: *Point Set Theory*. Marcel Dekker, 1990.

[MS93] MUELLER H., STARK M.: Adaptive generation of surfaces in volume data. *The Visual Computer 9* (1993), 182–199.

[MST*04] MÜLLER M., SCHIRM S., TESCHNER M., HEIDELBERGER B., GROSS M.: Interaction of fluids with deformable solids. *Journal of Computer Animation and Virtual Worlds 15*, 3-4 (2004), 159–171.

[MSVS03] MOLLEMANS W., SCHUTYSER F., VAN CLEYNENBREUGEL J., SUETENS P.: Tetrahedral mass spring model for fast soft tissue deformation. In *Proc. International Symposium on Surgery Simulation and Soft Tissue Modeling* (2003), pp. 145–154.

[MT95] MCINERNEY T., TERZOPOULOS D.: Topologically adaptable snakes. In *Proc. 5th International Conference on Computer Vision* (1995), pp. 840–845.

[MT96] MCINERNEY T., TERZOPOULOS D.: Deformable models in medical image analysis: A survey. *Medical Image Analysis 1*, 2 (1996), 91–108.

[MTB03] MCGUFfiN M. J., TANCAU L., BALAKRISHNAN R.: Using deformations for browsing volumetric data. In *Proc. IEEE Visualization* (2003), pp. 401–408.

[MTG04] MÜLLER M., TESCHNER M., GROSS M.: Physically-based simulation of objects represented by surface meshes. In *Proc. Computer Graphics International* (2004), pp. 26–33.

[MTT96] MAGNENAT-THALMANN N., THALMANN D. (Eds.): *Interactive Computer Animation*. Prentice Hall, 1996.

[MV98] MAINTZ J. B. A., VIERGEVER M. A.: A survey of medical image registration. *Medical Image Analysis 2*, 1 (1998), 1–36.

[MVdF03] MEDEROS B., VELHO L., DE FIGUEIREDO L. H.: Moving least squares multiresolution surface approximation. In *Proc. SIBGRAPI* (2003), pp. 19–26.

[Nay90] NAYLOR B.: SCULPT: an interactive solid modeling tool. In *Proc. Graphics interface* (1990), pp. 138–148.

[NC99] NOBLE R. A., CLAPWORTHY G. J.: Direct mannipulation of surfaces using NURBS-based free-form deformations. In *Proc. IEEE Information Visualisation* (London, UK, July 1999), pp. 238–243.

[NH91] NIELSON G., HAMANN B.: The asymptotic decider: resolving the ambiguity in marching cubes. In *Proc. IEEE Visualization* (1991), pp. 83–90.

[Nie04] NIELSON G. M.: Dual marching cubes. In *Proc. IEEE Visualization* (2004), pp. 489–496.

[NMK*06] NEALEN A., MÜLLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically based deformable models in computer graphics. *Computer Graphics Forum 25*, 4 (2006), 809–836.

[NN98] NOH J., NEWMANN U.: *A Survey of Facial Modeling and Animation Techniques*. Tech. Rep. TR1998-99-705, Integrated Media Systems Center, University of Southern California, 1998.

[NS97] NIELSON G., SUNG J.: Interval volume tetrahedralization. In *Proc. IEEE Visualization* (1997), pp. 221–228.

[NT98] NEDEL L. P., THALMANN D.: Real-time muscle deformation using mass-spring system. In *Proc. Computer Graphics International* (1998), pp. 156–165.

[O'R95] O'ROURKE M.: *Principles of Three-Dimensional Computer Animation*. W. W. Norton and Company, 1995.

[Par01] PARENT R.: *Computer Animation: Algorithms and Techniques*. Morgan-Kaufmann, 2001.

[PC87] PERRY A. E., CHONG M. S.: A description of eddying motions and flow patterns using critical-point concepts. *Annual Review of Fluid Mechanics 19* (1987), 125–155.

[PDA03] PICINBONO G., DELINGETTE H., AYACHE N.: Non-linear anisotropic elasticity for real-time surgery simulation. *Graphical Models 65*, 5 (2003), 305–321.

[PFY*99] PIZER S. M., FRITSCH D. S., YUSHKEVICH P. A., JOHNSON V. E., CHANEY E. L.: Segmentation, registration, and measurement of shape variation via image object shape. *IEEE Transactions on Medical Imaging 18*, 10 (1999), 851–865.

[PKA*05] PAULY M., KEISER R., ADAMS B., DUTRÉ; P., GROSS M., GUIBAS L. J.: Meshless animation of fracturing solids. *ACM Transactions on Graphics 24*, 3 (2005), 957–964.

[PKKG03] PAULY M., KEISER R., KOBBELT L., GROSS M.: Shape modeling with point-sampled geometry. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003) 22*, 3 (2003), 641–650.

[PLDA00] PICINBONO G., LOMBARDO J.-C., DELINGETTE H., AYACHE N.: Anisotropic elasticity and force extrapolation to improve realism of surgery simulation. In *Proc. IEEE Robotics and Automation* (2000), pp. 596–602.

[PM96] PROAKIS J. G., MANOLAKIS D. G.: *Digital Signal Processing*, 3rd ed. Prentice-Hall, 1996.

[Poc01] POCOCK L.: *The Computer Animator's Technical Handbook*. Morgan Kaufmann, 2001.

[Pra87] PRATT V.: Direct least-squares fitting of algebraic surfaces. *Computer Graphics (Proc. SIGGRAPH 87) 21*, 4 (1987), 145–152.

[PSRP00] PORTER B. C., STRANG J. G., RUBENS D. J., PARKER K. J.: 3D volume registration: experiments on tissue phantom and liver. In *Proc. IEEE Ultrasonics Symposium* (2000), vol. 2, pp. 1581–1584.

[PT92] PAYNE B. A., TOGA A. W.: Distance field manipulation of surface models. *IEEE Computer Graphics and Applications 12*, 1 (1992), 65–71.

[PW89] PENTLAND A., WILLIAMS J.: Good vibrations: modal dynamics for graphics and animation. *Computer Graphics (Proc. SIGGRAPH 89) 23*, 3 (1989), 215–222.

[PWL*98] PENNEY G. P., WEESE J., LITTLE J. A., DESMEDT P., HILL D. L. G., HAWKES D. J.: A comparison of similarity measures for use in 2-D-3-D medical image registration. *IEEE Transactions on Medical Imaging 17*, 4 (1998), 586–595.

[PZvBG00] PfiSTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: surface elements as rendering primitives. In *Computer Graphics (Proc. SIGGRAPH 2000)* (2000), pp. 335–342.

[QW03] QI M., WILLIS P.: Quasi-3D cel-based animation. In *Proc. Vision, Video and Graphics* (2003), pp. 111–116.

[RA00] REYES-ALDASORO C. C., ALGORRI M. E.: A combined algorithm for image segmentation using neural networks and 3D surface reconstruction using dynamic meshes. *Revista Mexicana de Ingeníeria Biomédica 21*, 3 (2000), 73–81.

[RC06] RODGMAN D., CHEN M.: Refraction in volume graphics. *Graphical Models 68*, 5-6 (2006), 432–450.

[RE00] RAVIV A., ELBER G.: Three-dimensional freeform sculpting via zero sets of scalar trivariate functions. *Computer Aided Design 32*, 8-9 (2000), 513–526.

[RGR97] RAJAPAKSE J. C., GIEDD J. N., RAPOPORT J. L.: Statistical approach to segmentation of single-channel cerebral MR images. *IEEE Transactions on Medical Imaging 16*, 2 (1997), 176–186.

[RRTP99] RADETZKY A., RNBERGER A., TEISTLER M., PRETSCHNER D.: Elastodynamic shape modeling in virtual medicine. In *Proc. Shape Modeling and Application* (1999), pp. 172–178.

[RSH*99] RUECKERT D., SONODA L. I., HAYES C., HILL D. L. G., LEACH M. O., HAWKES D. J.: Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging 18*, 8 (1999), 712–721.

[RSSSG01] REZK-SALAMA C., SCHEUERING M., SOZA G., GREINER G.: Fast volumetric deformation on general purpose hardware. In *Proc. SIGGRAPH/Eurographics Graphics Hardware Workshop* (2001), pp. 17–24.

[SBH*00] SZEKELY G., BRECHBUHLER C., HUTTER R., RHOMBERG A., IRONMONGER N., SCHMID P.: Modelling of soft tissue deformation for laparoscopic surgery simulation. *Medical Image Analysis 4*, 1 (2000), 57–66.

[Set96] SETHIAN J. A.: A fast marching level set method for monotonically advancing fronts. *Proc. National Academy of Sciences of the USA - Paper Edition 93*, 4 (1996), 1591–1595.

[SHH96] STUDHOLME C., HILL D. L. G., HAWKES D. J.: Automated 3-D registration of MR and CT images of the head. *Medical Image Analysis 1*, 2 (1996), 163–175.

[SHN03] SHERBONDY A., HOUSTON M., NAPEL S.: Fast volume segmentation with simultaneous visualization using programmable graphics hardware. In *Proc. IEEE Visualization* (2003), pp. 171–176.

[SJ01] SATHERLEY R., JONES M. W.: Vector-city vector distance transform. *Computer Vision and Image Understanding 82*, 3 (2001), 238–254.

[ŠK00] ŠRÁMEK M., KAUFMAN A.: vxt: a C++ class library for object voxelization. In *Volume Graphics*, CHEN M., KAUFMAN A. E., YAGEL R., (Eds.). Springer Verlag, 2000, pp. 119–134.

[SKK02] SEBASTIAN T., KLEIN P., KIMIA B.: Shock-based indexing into large shape databases. In *Proc. European Conference on Computer Vision* (2002), vol. 3, pp. 731–746.

[SLK*03] STRAKA M., LA CRUZ A., KÖCHL A., FLEISCHMANN M. D., GRÖLLER E.: 3D watershed transform combined with a probabilistic atlas for medical image segmentation. In *Proc. Medical Informatics and Technologies* (2003), pp. 1–8.

[SNS02] SVENSSON S., NYSTROM I., SANNITI DI BAJA G.: Curve skeletonization of surface-like objects in 3D images guided by voxel classification. *Pattern Recognition Letters 23*, 12 (2002), 1419–1426.

[SP86] SEDERBERG T. W., PARRY S. R.: Free-form deformation of solid geometric models. *Computer Graphics (Proc. SIGGRAPH 86) 20*, 4 (1986), 151–160.

[SP91] SCLAROFF S., PENTLAND A.: Generalized implicit functions for computer graphics. *Computer Graphics (Proc. SIGGRAPH 91) 25*, 4 (1991), 247–250.

[SPE*90] SCHAROFF S. E., PENTLAND A., ESSA I., FRIEDMANN M., HOROWITZ B.: The ThingWorld modeling system: virtual sculpting by modal forces. *Computer Graphics (Proc. SIGGRAPH 90) 24*, 4 (1990), 143–144.

[SS96] SCHRODER P., SWELDENS W. (Eds.): *Wavelets in Computer Graphics*. ACM SIGGRAPH '96 Course Notes, 1996.

[SS04] SINGH V., SILVER D.: Interactive volume manipulation with selective rendering for improved visualization. In *Proc. IEEE Symposium on Volume Visualization and Graphics* (2004), pp. 95–102.

[SSC03] SINGH V., SILVER D., CORNEA N.: Real-time volume manipulation. In *Proc. Volume Graphics* (2003), pp. 45–51.

[SSGD03] SUNDAR H., SILVER D., GAGVANI N., DICKINSON S.: Skeleton based shape matching and retrieval. In *Proc. Shape Modeling and Applications* (2003), pp. 130–138.

[SvL*03] STRAKA M., LA CRUZ A., GRÖLLER E., KÖCHL A., FLEISCHMANN D.: Bone segmentation in CT-angiography data using a probabilistic atlas. In *Proc. Vision, Modeling and Visualization* (2003), pp. 505–512.

[SW04] SU D., WILLIS P.: Image interpolation by pixel level data-dependent triangulation. *Computer Graphics Forum 23*, 2 (2004), 189–201.

[SWL05] SURI J. S., WILSON D., LAXIMINARAYAN S. (Eds.): *Handbook of Biomedical Image Analysis*. Plenum, 2005.

[SWY99] SHAREEF N., WANG D. L., YAGEL R.: Segmentation of medical images using LEGION. *IEEE Transactions on Medical Imaging 18*, 1 (1999).

[SZL92] SCHROEDER W. J., ZARGE J. A., LORENSEN W. E.: Decimation of triangle meshes. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 65–70.

[TBHF03] TERAN J., BLEMKER S., HING V., FEDKIW R.: Finite volume methods for the simulation of skeletal muscle. In *Proc. Eurographics/ACM SIGGRAPH Symposium on Computer Animation* (2003), pp. 68–74.

[TFT04] TAKAHASHI S., FUJISHIRO I., TAKESHIMA Y.: Topological volume skeletonization and its application to transfer function design. *Graphical Models 66*, 1 (2004), 24–49.

[THB*90] TIEDE U., HÖHNE K. H., BOMANS M., POMMERT A., RIEMER M., WIEBECKE G.: Investigation of medical 3D-rendering algorithms. *IEEE Computer Graphics and Applications 10*, 2 (1990), 41–53.

[TK95] TEK H., KIMIA B. B.: Image segmentation by reaction-diffusion bubbles. In *Proc. 5th International Conference on Computer Vision* (1995), pp. 156–162.

[TK03] TAL A., KATZ S.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003) 22*, 3 (2003), 954–961.

[TL94] TURK G., LEVOY M.: Zippered polygon meshes from range images. In *Computer Graphics (Proc. SIGGRAPH 94)* (1994), pp. 311–318.

[TLSP03] TOMAZEVIC D., LIKAR B., SLIVNIK T., PERNUS F.: 3-D/2-D registration of CT and MR to X-ray images. *IEEE Transactions on Medical Imaging 22*, 11 (2003), 1407–1416.

[TM91] TERZOPOULOS D., METAXAS D.: Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 7 (1991), 703–714.

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *Computer Graphics (Proc. SIGGRAPH 87) 21*, 4 (1987), 205–214.

[Tsi95] TSITSIKLIS J. N.: Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control 40*, 9 (1995), 1528–1538.

[Tur92] TURK G.: Retiling polygonal surfaces. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 55–64.

[Vin92] VINCE J.: *3-D Computer Animation*. Addison-Wesley, 1992.

[VKKM03] VARADHAN G., KRISHNAN S., KIM Y. J., MANOCHA D.: Feature sensitive subdivision and isosurface reconstruction. In *Proc. IEEE Visualization* (2003), pp. 99–106.

[vKvOB*97] van Kreveld M., van Oostrum R., Bajaj C. L., Pascucci V., Schikore D. R.: Contour trees and small seed sets for isosurface traversal. In *Proc. ACM Symposium on Computational Geometry* (1997), pp. 212–220.

[VS91] Vincent L., Soille P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 6 (1991), 583–598.

[WDGT01] Wu X., Downes M. S., Goktekin T., Tendick F.: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. *Computer Graphics Forum 20*, 3 (2001), 349–358.

[Wes90] Westover L.: Footprint evaluation for volume rendering. *Computer Graphics (Proc. SIGGRAPH 90) 24*, 4 (1990), 367–376.

[WGG99] Wyvill B., Guy A., Galin E.: Extending the CSG tree. warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphics Forum 18*, 2 (1999), 149–158.

[WGKH01] Wu Y., Guan X., Kankanhalli M. S., Huang Z.: Robust invisible watermarking of volume data using the 3D DCT. In *Proc. Computer Graphics International* (2001), pp. 359–362.

[Whi04] Whitaker R.: Modeling deformable surfaces with level sets. *IEEE Computer Graphics and Applications 24*, 5 (2004), 6–9.

[WJ06] Walton S. J., Jones M. W.: Volume wires: a framework for experical non-linear deformation of volumetric datasets. *Journal of WSCG 14*, 1-3 (2006), 81–88.

[WK95] Wang S., Kaufman A.: Volume sculpting. In *Proc. Symposium on Interactive 3D Graphics* (1995), pp. 151–156.

[WL93] Wu Z., Leahy R.: An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 11 (1993), 1101–1113.

[WMW86] Wyvill B., McPheeters C., Wyvill G.: Animating soft objects. *The Visual Computer 2*, 4 (1986), 235–242.

[WP00] Wu Z., Prakash E. C.: Visual human animation. In *Volume Graphics*,Chen M., Kaufman A. E., Yagel R., (Eds.). Springer, 2000, pp. 243–252.

[WP02] Wade L., Parent R.: Automated generation of control skeletons for use in animation. *The Visual Computer 18*, 2 (2002), 97–110.

[WS01] Westermann R., Salama C.: Real-time volume deformations. *Computer Graphics Forum 20*, 3 (2001), 443–451.

[WSI98] Wheeler M. D., Sato Y., Ikeuchi K.: Consensus surfaces for modeling 3D objects from multiple range images. In *Proc. IEEE Computer Vision* (1998), pp. 917–924.

[WT04] Wu X., Tendick F.: Multigrid integration for interactive deformable body simulation. In *Proc. International Symposium on Medical Simulation* (2004), vol. LNCS 3078, pp. 92–104.

[Wv92] Wilhelms J., van Gelder A.: Octrees for faster isosurface generation. *ACM Transactions on Graphics 11*, 3 (1992), 201–227.

[WW90] Witkin A., Welch W.: Fast animation and control of nonrigid structures. *Computer Graphics (Proc. SIGGRAPH 90) 24*, 4 (1990), 243–252.

[WW92a] Watt A., Watt M.: *Advanced Animation and Rendering Techniques: Theory and Practice*. ACM Press, 1992.

[WW92b] Welch W., Witkin A.: Variational surface modeling. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 157–166.

[XHW*05] Xia L., Huo M., Wei Q., Liu F., Crozier S.: Analysis of cardiac ventricular wall motion based on a three-dimensional electromechanical biventricular model. *Physics in Medicine and Biology 50*, 8 (2005), 1901–1917.

[YK97] Yagel R., Kurzion Y.: Interactive space deformation with hardware-assisted rendering. *IEEE Computer Graphics and Applications 17*, 5 (1997), 66–77.

[Yoo04] Yoo T.: *Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis*. A K Peters, 2004.

[YP92] Yun H., Park K. H.: Surface modelling method by polygonal primitives for visualizing three-dimensional volume data. *The Visual Computer 8*, 4 (1992), 246–259.

[ZA96] Zuk T. D., Atkins M. S.: A comparison of manual and automatic methods for registering scans of the head. *IEEE Transactions on Medical Imaging 15*, 5 (1996), 732–744.

[ZB05] Zhu Y., Bridson R.: Animating sand as a fluid. *ACM Transactions on Graphics 24*, 3 (2005), 965–972.

[ZC01] ZHUANG Y., CANNY J. F.: Real-time global deformations. In *Algorithmic and Computational Robotics*,DONALD B. R., LYNCH K. M., RUS D., (Eds.). A K Peters, 2001, pp. 97–107.

[ZCK98] ZHU Q.-H., CHEN Y., KAUFMAN A. E.: Real-time biomechanically-based muscle volume deformation using FEM. *Computer Graphics Forum 17*, 3 (1998), 275–284.

[ZF03] ZITOVA B., FLUSSER J.: Image registration methods: A survey. *Image and Vision Computing 21*, 11 (2003), 977–1000.

[ZH81] ZUCKER S. W., HUMMEL R. A.: A three-dimensional edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence 3*, 3 (1981), 324–331.

[ZPBG01] ZWICKER M., PfiSTER H., BAAR J., GROSS M.: EWA volume splatting. In *Proc. IEEE Visualization* (2001), pp. 29–36.