Deforming and Animating Discretely 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 discretely sampled object representation (DSOR) defines a graphical model using data obtained by 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 DSORs include images, videos, volume datasets and point datasets. Unlike many commonly used data representations in computer graphics, DSORs 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 DSORs. 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 DSORs. We consider a collection of elementary operations for manipulating DSORs, 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 DSORs and pay particular attention to their deployment in surgical simulation. We review a collection of techniques for animating digital characters in DSORs, focusing on recent developments in volume animation.

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

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 and in graphics pipelines. For instance, *volume rendering* [Lev88, Wes90] enables direct rendering of 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 see that computer animation could benefit enormously from the advances of digital imaging technology, should appropriate techniques be integrated in animation pipelines for

† 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

handling digitally sampled data directly. This would enable animators to gain easy access to a vast amount 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 *discretely sampled object representations (DSORs)*.

In this state of the art report, we survey a wide-range of techniques that have been developed for manipulating, deforming and animating DSORs. In computer graphics, techniques for *manipulation*, *deformation* and *animation* are closely 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 DSORs. All these operations can serve as building blocks of deformation and animation techniques, and some also involve 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 DSOR. 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 DSORs focusing their capability of modeling movements of articulated figures in DSORs.

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 DSORs. In Section 3, we provide a formal definition of the class of DSORs, and discuss their geometrical and graphical attributes. In Section 4, we examine a collection of elementary operations on DSORs, 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 DSORs can have a major role. In Section 6, we focus on the main components in the animation pipeline for modeling the movements of digital characters in DSORs. 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 modeling and computer animation are two closely inter-related fields. While the literatures on deformation and animation are dominated by surface-based modeling and rendering techniques, it is certainly sensible and meaningful to consider the deformation and animation of discretely sampled object representations (DSORs) 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 behavior 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 behavioral 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 modeling, 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 modeling, controlling and rendering of temporal behavior of graphical objects. It addresses a wide range of technical issues such as motion dynamics, kinetic control, collision detection, actor modeling, 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, color grading and so on [Ker04]. In the entertainment industry, the use of the term also extends to computer-assisted animation such as computer generated inbetweening in key-frame animation.

Figure 1 depicts the overall scope of deformation modeling and computer animation, and highlights the main technical areas of manipulating, deforming and animating DSORs, 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. Continuous object representations, which can be extracted from DSORs, are often used to assist in deforming and animating DSORs, and most techniques for manipulating DSORs may introduce, often unintentionally, minor geometrical changes in DSORs. We will briefly examine these techniques in Section 4.

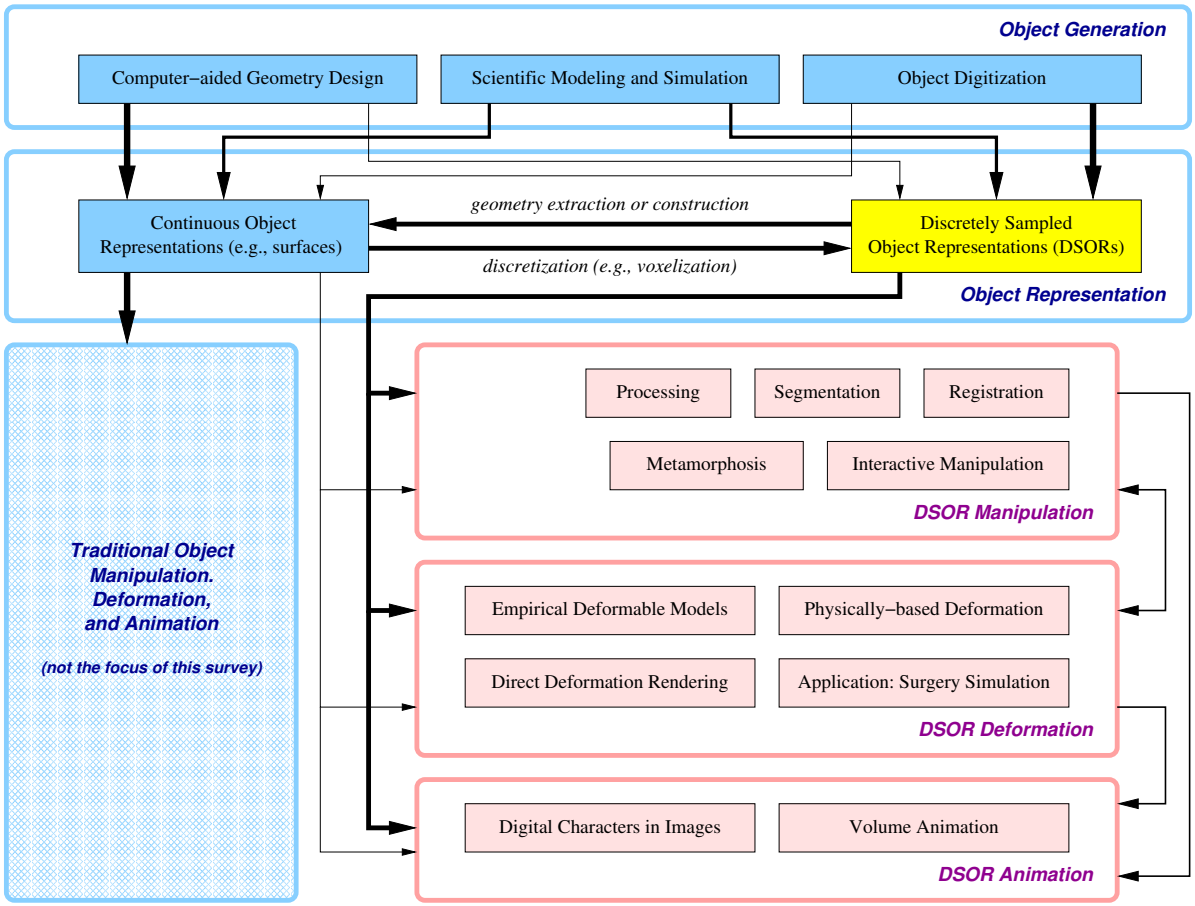


Figure 1: *The main scope of this survey and related topics.*

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*05],
- animation of articulated figures [LW98],
- facial modeling 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 a set of 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 [Bli82]). Some commonly used models in this category are algebraic implicit surfaces [Bli82, Kle89, SP91, Ped95, WGG99], superquadrics [TM91], generalized cylinders (or swept volumes) [DBJ97, Coq87, Mil88, Sny92, CBS96] and global and local deformation of solids [Bar84].
- *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 [KWT87, GP89, LW94, CR94], free form deformation [SP86, HHK92, CJ91] active contour models (snakes) [KWT87] and active surface models [TPBF87]
- *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 [SP86], 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 [TQ94], mass-spring models [FE77], finite element methods [ZT89], finite difference [LL93], continuum models [TPBF87] and particle systems [LTSE92, RHK*00].

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. A summary of various techniques can be found in [BBZ91].
- *Procedural Control* — General procedural control is based on scripts controlling actors [Rey82] or on interactive motion planning [JW89]. 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.
- *Constraint-based Control* — This approach is common in robot systems. As such, there are obvious parallels with computer animation. Among many publications, are an early PhD thesis [Dui86], efficient interface work [ZC95] and applications to facial animation [Rut99].
- *Physically-based Control* — Physically-based simulation has a very 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 [LvF00]. Again, there are many papers in this general area.
- *Stochastic Control* — Particle systems and their rendering to produce fireworks, fire, water sprays and other fuzzy objects are the definitive examples of stochastic control [Ree83].
- *Behavioral Control* — This area has traditionally concerned itself with gross control, with Reynolds work on modeling flocks, herds and schools [Rey87] as the key paper. However it has more recently been used to control quite subtle effects in human animation, for example in the way emotion effects the way a character walks [UT91, DW97].

3. Discretely Sampled Object Representations

A discretely sampled object representation (DSOR) defines a graphical model using data obtained by 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.

Digitization is the primary technology for acquiring DSORs 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. In this report, 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

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

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

grid though individual samples may not correlate uniformly to signal sources in the object space.

DSORs can also be obtained by sampling continuous object representations. For example, a continuous surface representation 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, DSORs are commonly used to approximate continuous spatial and temporal data representations derived from theoretic studies, scientific modeling and computer simulation.

DSORs commonly exhibit a subset of the following characteristics, which collectively signify the differences between DSORs and other schemes for representing graphical objects and scenes.

- *Limited geometrical information* — Most DSORs 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 tumor). In addition, DSORs are particularly suited for modeling 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 DSOR 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 DSOR, 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 DSORs 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 DSORs 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 DSORs 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 modeling, computer animation, scientific visualization, volume graphics, point-based graphics, image-based modeling 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 Discretely Sampled Object Representations

As discussed above, most DSORs contain limited geometrical, topological and semantic information, and the traditional notion of 'geometry' is usually constructed from sampled physical information. Hence, any manipulation of a DSOR that leads to changes of sampled physical properties may result in alterations to geometrical, topological and semantic attributes derived from the DSOR. 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 manipulating the raw physical properties of DSORs. 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 DSORs. We then describe a collection of basic operators for *manipulating* data elements in a single 2D or 3D imagery dataset, and discuss more complex DSOR 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 DSORs. 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 DSOR manipulation, and prepare for the further discussions on DSOR deformation in Section 5 and animation in Section 6.

4.1. Extracting Geometry from DSORs

4.1.1. Surface Extraction from Volume Data

One common approach for handling 3D volume datasets is to approximate a volume by a continuous 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 τ (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 DSORs such as an image. [NH91] provided a computational solution, called asymptotic decide, 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 isosurfacing, 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, BS03].

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 modeling (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 isosurface 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 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 polynormals 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.

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

4.2.1. Processing DSORs

The processing of DSORs 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. Most techniques reviewed in this section are applied to both 2D [GW01] and 3D DSORs.

- *Grey level transformations* — Basic grey level transformations operating on a value by value basis are employed to enhance DSORs, 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 DSORs (e.g., histograms), can be computed and used to enhance DSORs (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 DSORs, and is also used for determining the derivative of a discrete image or volume function. Low-pass filters average neighboring values to remove noise and to blur DSORs, 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 [RC04], and to create anti-aliased voxelizations of objects [ŠK00]. Filtering can also take place in the frequency domain of a DSOR [Mal93].

- *Arithmetic and logical operations* — Arithmetic and logic operations can be applied to multiple DSORs to facilitate combinational operations such as masking and blending. For example, arithmetic and logical operations were used to provide the DSOR equivalent of 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 DSORs [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 DSORs* can be used to facilitate fast morphological operations [Jon01].
- *Distance transforms* — 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 DSOR, *distance transforms* are the process of applying templates to each pixel/voxel in the DSOR in order to propagate distances around the DSOR. Methods for distance transforms include *Chamfer* [SNS02], *Vector* [SJ01] and *Fast Marching Methods* [Tsi95, Set96]. Distance field DSORs can be used in volume morphing (see 4.2.2), and for voxelizing continuous object representations (e.g. triangular mesh objects) [Jon96, JS01].

There are other DSOR processing methods that may also introduce geometrical changes to DSORs, 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 DSORs

Given a starting DSOR D_a and a finishing DSOR D_b, *metamorphosis* (or commonly referred to as *morphing*) is a process that generates a sequence of inbetween DSORs, $D_1, D_2, \ldots, D_n$, which represent a smooth transformation from D_a to D_b. For DSORs defined upon regular grids, such as images and volume datasets, this transformation from one DSOR 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 DSOR D_i, an inbetween 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 DSORs D_{ia} and D_{ib}. The inbetween 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 DSORs 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 DSORs with the same grid organization in the spatial domain. In volume morphing, when the emphasis is given to a particular isosurface, 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 inbetween DSORs, 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 isosurface [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 inbetween DSORs 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 DSORs 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 DSORs

Segmentation is an extensive subject, due to both the demand for segmentation (particularly in medical analysis) and the complexity of segmentation in general. Segmentation for 2D DSORs (e.g., images) has been widely researched and documented. While great efforts have been made to devise techniques for segmenting 3D DSORs (e.g., volume datasets), unfortunately, most 2D methods do not lend themselves naturally to 3D. Instead, volume datasets are commonly segmented slice by slice.

Segmentation is an effective means for adding semantic information into a DSOR, and such information can be useful to volume deformation 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 can typically fall into one of the following categories.

4.3.1. Stochastic Methods

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

Related to thresholding is the watershed transform [VS91], and has the advantage that it can be extended to 3D quite 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], where Gaussian curves for both thresholds are automatically found. If the thresholds have a transition zone inbetween, the classification relies on the standard deviation of the 26 neighbors of the voxel. The remainder of the algorithm 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, but the quality of result largely relies on the choice of a seed point to begin. Lin *et al.* [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 strength of using such a technique is that it can learn from past examples of segmented datasets. Ahmed and Farag [AF97] use 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 three-dimensional data in only two dimensions, grouping together similar objects with no user interaction. They are also trained automatically.

LEGION (locally excitatory globally inhibitory oscillator network) 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 can be defined as a measurement of similarity in image regions. Clustering techniques attempt to group voxels together that display similar predefined characteristics. 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 G 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 an 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. Deformable Models

Deformable models offer a flexible, accurate approach to segmentation. such models generally handle image noise extremely well, and also allow for sub-voxel accuracy. Energy-minimizing snakes [KWT87, MT95] remain the most popular deformable model for image and volume segmentation. Snakes are parametric contour models that actively (i.e. with user interaction) attempt to minimize their energy as:

$$E_{snake} = \int_0^1 E_{int}(s) + E_{ext}(s) + E_{img}(s)$$

where E_{int}, E_{ext} and E_{img} represent the internal (due to bending), external (user-defined constraints) and image forces (edge detection), respectively.

Segmentation of volume datasets with the snake model involves a slice-by-slice segmentation, with a 3D surface built from the resulting contours from each slice. Ballerini and Bocchi [BB03] extend the idea slightly by using multiple snakes that communicate to segment 2D radiogram images of the hand. Miller *et al.* [MBL*91] extended the energy-minimization model into the third dimension by constructing a polygonal 'balloon' inside the object that grows to conform to the surface of the object of interest. Interestingly, the object classification is a simple threshold operation. Tek and Kimia [TK95] use 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.

Sherbondy *et al.* [SHN03] use a GPU-based implementation of the seed-fill algorithm to segment areas of interest. A significant speedup over that of SSE2-accelerated CPU code was achieved. Combined with hardware-accelerated rendering, the user is able to segment the dataset and view results in real-time.

4.3.7. Generalized Algorithms

The availability of high-performance computers has accelerated the use of volume visualization, which in turn is pulling the research focus of image segmentation into volume segmentation. The requirement of accurate and automated segmentation techniques for use particularly with medical data often prompts researchers to devise new algorithms and hybrid techniques which are optimized specifically for a subset of the human anatomy. Such algorithms tend to excel at the task they are given, often automatically.

Generalized segmentation algorithms require a large amount of user interaction to provide a start point within the dataset, or to guide the algorithm as it attempts to locate areas of interest. Such algorithms are usually stochastic in nature, attempting to locate areas of interest according to the data found during the run. It is currently unclear whether a completely general volume segmentation algorithm or unified set of segmentation methods can be devised to work proficiently on a variety of datasets.

4.4. Registering DSORs

Registration normally requires a three step approach. In the first step features are extracted from the DSORs. The second step carries out a feature correspondence in order to derive the spatial transformation between the DSORs, and the third step requires the calculation of the deformation to take place between the DSORs.

The transformation may be global (i.e., the same transformation applies over the whole DSOR) or local (i.e., different transformations apply to subsets of the DSOR). The transformation may be rigid (only translations and rotations are involved), affine (translations, rotations, dilations and shears), projective or elastic (e.g., lungs expanding, heart beating, joint deforming).

The process of registration for projective transformations (and the special cases of affine and rigid transformations) involves determining the 4×4 matrix for the transformation. This may need to be determined at spatial subsets in the case of local transformations. Elastic registration usually determines the transformation as a function based upon location. In both cases discontinuities may arise at boundaries.

Brown [Bro92] published a comprehensive survey of image registration methods. Further registration methods from the period 1992 to 2003 are surveyed by Zitova and Flusser [ZF03]. Image registration techniques now span a wide range of research activities and applications, including: applications in remote sensing (multispectral classification, environmental monitoring, creating super-resolution

images, integrating information into geographic information systems), applications in medicine (combining computer tomography and MRI data to obtain more complete information about the patient, monitoring tumour growth, treatment verification, comparison of the patient's data with anatomical atlases), applications in cartography (map updating), and applications in computer vision (target localization, automatic quality control), to name a few [ZF03, MV98].

Maintz and Viergever [MV98] presented a full classification of the stages involved in medical registration in their excellent survey, where they identify nine criteria for the classification. Makela *et al.* [MCS*02] offered a survey and review of various rigid and elastic (cardiac) registration techniques, and commented on the performance, accuracy and suitability of each technique. Cardiac registration can present the additional problem of calculating the registration in 4D (3D spatial and time). Zuk and Atkins [ZA96] provided a comparison of several techniques for rigid transformations.

4.4.1. Feature Extraction and Correspondence

The first stage is to identify features in each DSOR to be registered, and secondly to establish a correspondence between those features. The very simplest techniques (and perhaps most accurate), use markers introduced to the DSOR. 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 calculated the gross alignment between two DSORs (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 DSORs (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 calculated 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* or *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] use a multi-resolution approach and iterate through various translational and rotational values to maximize the function. Fei *et al.* [BF03] also use a feature correspondence which is refined using optimization of mutual information in volumes of interest around the features. The local transform is then modeled using a thin-plate spline transformation which is optimized iteratively. Reuckert *et al.* also use a similar method to find the global transform, and then use B-splines to model the local transform, and optimize those using a smoothness penalty and multiresolution approach.

In general, it seems the best approach (in terms of speed and accuracy) is to determine the global rigid transform hierarchically using an optimization of choice. For the local elastic transform, some 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 modeling 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 modeling has been studied extensively (e.g., [Coq90, Nay90, SPE*90, SP86]). Galyean and Hughes [GH91] first introduced this concept to interactive volume modeling. A number of tools were developed, which are also discretized and which include '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'. [Bar98] 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 the a tool to modify the scalar coefficients.

5. Deforming Discretely 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 modeling, computer-aided illustration, surgical simulation, and scientific visualization.

In this section, we focus on those techniques that are of a strong relevance to discretely sampled object representations (DSORs). these include those techniques that operate directly on DSORs, as well as those involve geometric representations extracted from DSORs. Table 2 lists a collection of previous developments in this area, and their main technical characteristics. After a summary of non-physically-based deformable models, we briefly examine several major physically-based models that have been deployed for deforming DSORs. We then consider techniques for rendering deformation directly. Finally we give an overview of DSOR deformation techniques in the context of a particular application, namely surgical simulation, where DSOR deformation could potentially play a major role.

5.1. Empirical Deformable Models

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

Since DSORs contain limited geometrical and topological information, one common approach is to associate a DSOR with a geometry-based control structure, which deforms upon the input of a deformation specification and then transfers its geometric changes to the DSOR concerned.

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

- applying *global and local deformation* [Bar84, Bar86] to volume datasets through ray deflectors [KY95] or spatial transfer functions [CSW*03];
- applying free-form deformation [SP86] to volume datasets through volume bounding boxes [CSW*03];
- 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].

On the other hand, DSORs 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].

Reference	Data Representation	Computational model	Application Context
[Bar84]	generic solid models	empirical (global/local deformation)	
[Bar86]	parametric/implicit surfaces	empirical (space warp)	
[BAZT04]	dynamic models	finite element	shape recognition in 2D images
[BC96]	tetrahedral mesh	finite element	surgical simulation
[BJ03]	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
[CBS96]	implicit surface	empirical (swept objects)	
[Coq90]	polygonal mesh	empirical (free-from deformation)	
[CR94]	polygonal mesh	empirical (free-from deformation)	
[CSW*03]	volume	empirical (spatial transfer function)	volume visualization/animation
[CZK98a]	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
[FLW93]	superquadrics	empirical (superquadrics)	
[Gib97]	volume grid	empirical (chainmail)	
[HHK92]	rectangular mesh	empirical (b-spline-based FFD)	
[IDSC04]	volume	empirical (spatial transfer function)	splitting and explosion
[Jon03]	volume	finite differences	ice melting
[KCM00]	polyhedral mesh, NURBS	mass spring, finite element	surgical simulation
[KMH*04]	point-based	linear elasticity	collision handling
[KWT87]	2D curves on images	energy minimizing snakes	
[KY95]	volume	empirical (ray deflectors)	volume visualization
[LKHW03]	polygonal mesh	empirical (level set)	
[LW94]	curve, surface, solid	empirical (NURBS-based FFD)	
[MDM*02]	tetrahedral mesh	finite element, elasticity	real-time deformation
[MKN*04]	point-based	moving least squares	
[MSCS03]	tetrahedral mesh	mass spring	surgical simulation
[MTB03]	volume	empirical (procedural models)	volume visualization
[MTG04]	Hexahedral mesh	finite element	
[NC99]	rectangular mesh	empirical (NURBS-based FFD)	
[NT98]	triangular mesh	mass spring, elasticity	
[PDA03]	tetrahedral mesh	finite element, non-linear elasticity	liver laparoscopic, haptic
[PKKG03]	point-based	empirical (free-form deformation)	
[PLDA00]	tetrahedral mesh	finite element, elasticity	laparoscopic surgical simulation
[PPG04]	point-based	moving least squares	contact handling
[PW89]	generic solid	finite element	
[RRTP99]	unstructured surface mesh	mass spring	surgical simulation
[RSSSG01]	volume	empirical (gradient deformation)	hardware-assisted rendering
[SBH*00]	rectangular mesh	finite element	laparoscopic surgical simulation
[SP86]	generic solid	empirical (free-from deformation)	
[SP91]	implicit surface	empirical (deformation map)	
[TBHF03]	triangular mesh	finite volume	skeletal muscle simulation
[TM91]	superquadrics	empirical (superquadrics)	
[TPBF87]	curve, surface, solid	elasticity	
[WMW86]	implicit surface	empirical (implicit surface)	
[WGG99]	implicit surface	empirical (implicit surface)	
[WT04]	triangular/tetrahedral mesh	finite element	soft tissue simulation
[WW90]	generic solid	low degree-of-freedom	
[WW92b]	b-spline surface	empirical (parametric model)	
[XHW*05]	rectangular mesh	finite element	heart modeling

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

5.2. Physically-based Models

As shown in Table 2, 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 DSORs, such data representations can be extracted or reconstructed from DSORs using the techniques discussed in Section 4. They can be coupled with DSORs and act as the control structure for DSOR deformation. For mesh-centered physically-based deformable models, readers are especially encouraged to consult several important surveys, including .... and STAR4. To maintain a certain degree of self-containment of this survey, we briefly describe several commonly used physically-based deformable models.

- *Mass-spring Models* — In mass-spring models, an object is approximated as a finite mesh of points. The mechanics of deformation is defined as a second-order differential equation, 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 vertices 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, as a finite sum of element-specific interpolation functions. FEM is used extensively in computer graphics for deformation (e.g., [PW89, BC96]).
- *Continuum Model* — A continuum model of a deformable object is a function of the forces acting on the material properties of the object. Since it is not always possible to find a closed-form analytic solution of such a continuum model, a number of numerical methods are used to approximate the object deformation [TPBF87, TW88].
- *Low Degree-of-freedom Models* — This class of models are designed to reduce the computational costs of abovementioned 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], add physical behavior to the traditional geometric modeling primitives as parametric surface patches [WW92b], 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 [MKN*04, KMH*04].

5.3. Rendering Deformation

Traditionally, deformation is performed at the modeling stage, which results in an explicit deformed object to be forwarded to the rendering stage. It is often desirable to couple the modeling 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] 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-from deformation, image-swept volumes, volume and hypertexture splitting, and volume animation [CSW*03, IDSC04].
- 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 general purpose rendering hardware, coupled with edge and face constraints, to facilitate the adaptive subdivision of piecewise patches of a volume object.
- Singh and Silver [SSC03] used a proxy geometry to specify manipulation about joints in the volume. 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 pseudolinear computation of elastic forces. A multiresolution approach was adopted to simulate object deformation in real time.

5.4. Deformable Models in Surgical Simulation

The role of deformable models in surgery simulation and training is diverse, since they 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 ChainMail [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 [CZK98b, MSCS03] and finite element methods (FEM). Of these two, the latter is the most 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 [ZC00]. 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 multiresolution techniques, as suggested by Debunne *et al.* [DDCB01] and Wu [WDGT01]. Wu and Tendick propose a multigrid 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 precomputed 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) [BJ03], 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. Surveys can be found in [LTCK03].

6. Animating Discretely 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 DSOR, 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

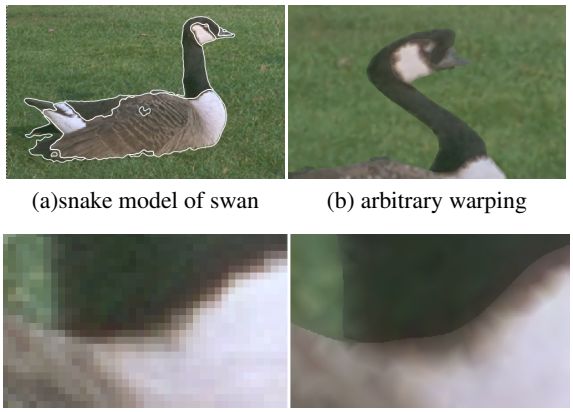
In the studio production of digital composition, image quality plays a critical role. There are really two aspects here: animating digital characters which are images of fixed resolution and the resolution of the composite images which are the final product.

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



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

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

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

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.

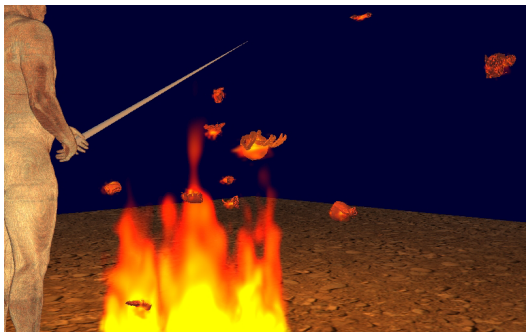
The above image interpolation method was also used in the Quasi-3D animation system of Qi and Willis [QW03], which builds on the mentioned work from Labrosse, Froumentin in their earlier work with Willis. Traditional cartoon animation uses a stack of painted cels, with the layer ordering determining 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 have 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.

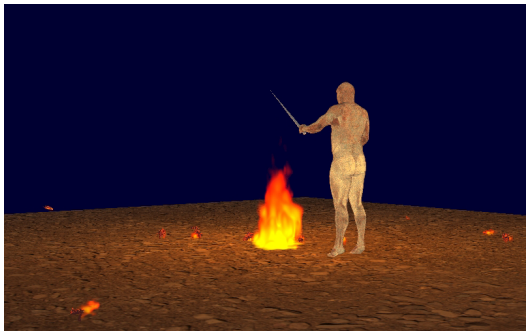
The main idea in a motion capture cartoon system is to parameterize cartoon motion with a combination of affine-transformations and key-weight vectors. Thus the solution to the problem of inconsistent in-between frames is to construct the extended linear space for every combination of hand-picked key-shapes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

The Visible Man was about to barbecue a lobster.

The lobster struggled to get free, but ended in the fire.



The lobster exploded.



The Visible Man waved his 'magic wand', hoping ...

Figure 3: *Animating the motion and splitting of a lobster using a block-based approach in conjunction with motion of the Visible Man. Both the lobster and the Visible Man are defined using discrete volume representations and the fire is defined using a procedural volume representation [IDSC04].*

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, i.e., 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 allowed for 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 (Figure 3). 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 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 the visible man, or an animal).

The IK-skeleton is then manipulated. This can be done in any animation package such as Character Studio [ani02a], or Maya [ani02b] 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. Because there is overlap in the spheres, coverage is maintained for small angles about the joints. (Please refer to [GS01] for more details.) An example of the visible man jumping rope is shown in Figure 4. 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 discretely sampled object representations (DSORs). 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 DSORs. In general, techniques for manipulating DSORs have reached a relatively mature status, with many well studied technical problems and solutions. Techniques for deforming DSORs are still replied heavily on those originally developed for surface and solid object representations. However, recent advances in close coupling of deformation and rendering of DSORs have opened an exciting new frontier. In terms of animating DSORs, 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 DSOR rendering technology, it is highly desirable to take the research and development in aspects of manipulating, deforming and animating DSORs 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/S46574, 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* (2002). 119

[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. 119

[ABK98] AMENTA N., BERN M., KAMVYSSELIS M.: A new voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH 1998* (1998), ACM Press, pp. 415–421. 119

[AF97] AHMED M. N., FARAG A. A.: Two-stage neural network for volume segmentation of medical images. *Pattern Recognition Letters 18* (1997), 1143–1151. 121

[AJWB01] AYLWARD S. R., JOMIER J., WEEKS S., BULLITT E.: Registration and analysis of vascular images. *International Journal of Computer Vision* (2001). 119

[AK04] AMENTA N., KIL Y. J.: Defining point-set surfaces. *ACM Trans. Graph. 23*, 3 (2004), 264–270. 119

[ani02a] Character studio animation toolkit, 2002. http://www.discreet.com/products/cs/. 130

[ani02b] Maya: A 3d animation and visual effects tool, 2002. http://www.aliaswavefront.com/en/products/maya/index.shtml. 130

[AS96] AVILA R. S., SOBIERAJSK L. M.: A haptic interaction method for volume visualization. In *Proc. IEEE Symposium on Volume Visualization* (San Francisco, CA, October 1996), pp. 197–204. 124

Figure 4: *Animating the Visible Man dataset with complex movements using a skeleton-based approach [GS01].*

[Bar84] BARR A.: Global and local deformation of solid primitives. *Computer Graphics (Proc. SIGGRAPH 84) 18*, 3 (July 1984), 21–30. 115, 124, 125

[Bar86] BARR A.: Ray tracing deformed surfaces. *Computer Graphics (Proc. SIGGRAPH 86) 20*, 4 (July 1986), 287–296. 124, 125

[Bar98] BARENTZEN A.: Octree-based volume sculpting. In *Proc. IEEE Visualisation Late Breaking Hot Topics* (1998). 124

[BAZT04] BERGNER S., AL-ZUBI S., TONNIES K.: Deformable structural models. In *Proc. International Conference on Image Processing* (Singapore, 2004), pp. 1875–1878. Volume 3. 125

[BB03] BALLERINI L., BOCCHI L.: Bone segmentation using multiple communicating snakes. In *Proc. Medical Imaging 2003* (2003), vol. 5032, pp. 1621–1628. 122

[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, Philadelphia, PA, 1994. 126

[BBZ91] BADLER N. I., BARSKY B. A., ZELTZER D. (Eds.): *Making Them Move : Mechanics, Control and Animation of Articulated Figures*. Morgan Kaufmann, 1991. 116

[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. 125, 126, 127

[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* (2003), 267Ű281. 123

[BFB94] BARRON J. L., FLEET D. J., BEAUCHEMIN S. S.: Performance of optical flow techniques. *Int. J. Comput. Vision 12*, 1 (1994), 43–77. 123

[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. 127

[BJ03] BALANIUK R., JR. J. K. S.: Soft-tissue simulation using the radial elements method. In *Proc. International Symposium on Surgery Simulation and Soft Tissue Modeling* (2003), pp. 48–58. 125, 127

[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 (July 2002), 399–407. 128

[Bli82] BLINN J. F.: A generalization of algebraic surface drawing. *ACM Transactions on Graphics 1*, 3 (July 1982), 235–256. 115

[Blo02] BLOOMENTHAL J.: Medial based vertex deformation. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Antonio, Texas, 2002), pp. 147–151. 119

[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, Cambridge, MA, 1967, pp. 362–380. 119

[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 (October/December 1999), 349–359. 119

[BMW01] BRUYNS C., MONTGOMERY K., WILDERMUTH S.: A virtual environment for simulated rat dissection: A case study of visualization for astronaut training. In *IEEE Visualization 2001* (San Diego, California, Oct. 2001), pp. 21–26. 125

[BN92] BEIER T., NEELY S.: Feature-based image metamorphosis. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 35–42. 120

[Bor88] BORGEFORS G.: Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 10*, 6 (Nov 1988), 849–865. 123

[BPS96] BAJAJ C. L., PASCUCCI V., SCHIKORE D. R.:

Fast isocontouring for improved interactivity. In *Proc. IEEE 1996 Symposium on Volume Visualization* (October 1996), pp. 39–46. 118

[Bro92] BROWN L. G.: A survey of image registration techniques. *ACM Computing Surveys 24* (1992), 326–376. 122

[BS03] BORDOLOI U. D., SHEN H.-W.: Space efficient fast isosurface extraction for large datasets. In *Proc. IEEE Visualization* (2003), pp. 201–208. 118

[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. 120

[BWC00] BHANIRAMKA P., WENGER R., CRAWFIS R.: Isosurfacing in higher dimensions. In *Proc. IEEE Visualization* (2000), pp. 267–274,566. 118

[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. 125, 127

[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. 119

[CBS96] CRESPIN B., BLANC C., SCHLICK C.: Implicit sweep objects. *Computer Graphics Forum 13*, 3 (1996), 165–174. 115, 125

[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. 125, 126, 127

[CF01] CAMPBELL R. J., FLYNN P. J.: A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding 81* (2001), 166–210. 115

[Che95] CHEN S. E.: QuickTime VR – an image-based approach to virtual environment navigation. In *Computer Graphics (Proc. SIGGRAPH 95)*. Addison-Wesley, Los Angeles, California, 1995, pp. 29–38. 113

[CJ91] COQUILLART S., JANCÈNE P.: Animated free-form deformation: an interactive animation technique. *Computer Graphics (Proc. SIGGRAPH 91) 25*, 4 (1991), 23–26. 115

[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. 120

[CJT96] CHEN M., JONES M. W., TOWNSEND P.: Volume distortion and morphing using disk fields. *Computers and Graphics 20*, 4 (1996), 567–575. 120

[CK95] COHEN D., KAUFMAN A.: Fundamentals of surface voxelization. *CVGIP Graphics Models and Image Processing 57*, 6 (1995), 453–461. 117

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *Computer Graphics (Proc. SIGGRAPH 96)*. Addison-Wesley, New Orleans, Louisiana, 1996, pp. 303–312. 119

[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. 118

[CLS98] COHEN-OR D., LEVIN D., SOLOMIVICI A.: Three-dimensional distance field metamorphosis. *ACM Transactions on Graphics 17*, 2 (1998), 116–141. 120

[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. 118

[CMPS96] CIGNONI P., MONTANI C., PUPPO E., SCOPIGNO R.: Optimal isosurface extraction from irregular volume data. In *Proc. IEEE 1996 Symposium on Volume Visualization* (October 1996), pp. 31–38. 118

[Coq87] COQUILLART S.: A control-point-based sweeping technique. *IEEE Computer Graphics and Applications 17*, 11 (November 1987), 36–45. 115

[Coq90] COQUILLART S.: Extended free-form deformation: Asculpting tool for 3D geometric modeling. *Computer Graphics (Proc. SIGGRAPH 90) 24*, 4 (August 1990), 187–196. 124, 125

[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, Orlando, Florida, July 1994, pp. 257–260. 115, 125

[CS97] CHIANG Y.-J., SILVA C. T.: I/O optimal isosurface extraction. In *Proc. IEEE Visualization '97* (1997), pp. 293–300. 118

[CSv04] CARR H., SNOEYINK J., VAN DE PANNE M.: Simplifying flexible isosurfaces using local geometric measures. In *Proc. IEEE Visualization* (2004), pp. 497–504. 118

[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* (Tokyo, Japan, 2003), pp. 35–44. 124, 125, 126, 129

[CT00] CHEN M., TUCKER J.: Constructive volume geometry. *Computer Graphics Forum 19*, 4 (2000), 281–293. 120

[CWPG04] COTTING D., WEYRICH T., PAULY M., GROSS M.: Robust watermarking of point-sampled geometry. In *SMI '04: Proceedings of the Shape Modeling International 2004 (SMI'04)* (Washington, DC,

USA, 2004), IEEE Computer Society, pp. 233–242. 120

[CZK98a] CHEN Y., ZHU Q., KAUFMAN A.: Physically-based animation of volumetric objects. In *Proc. IEEE Computer Animation* (1998), pp. 154–160. 125

[CZK98b] CHEN Y., ZHU Q., KAUFMAN A.: Physically-based animation of volumetric objects. In *Proc. IEEE Computer Animation'98* (1998), pp. 154–160. 127

[DBJ97] D. BLACKMORE M. C. LEU L. P. W., JIANG H.: Swept volumes: a retrospective and prospective view. *Neural, Parallel and Scientific Computations 5* (1997), 81–102. 115

[DC03] DANIEL G., CHEN M.: Video visualization. In *Proc. IEEE Visualization* (2003), pp. 409–416. 120

[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. the Computer Animation* (Washington, DC, USA, 1999), p. 70. 125, 127

[DDCB01] DEBUNNE G., DESBRUN M., CANI M.-P., BARR A. H.: Dynamic real-time deformations using space & time adaptive sampling. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), pp. 31–36. 125, 127

[Dis04] DISCREET: 3D Studio Max, (3DSMax), 2004. http://www.discreet.com/. 119

[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. 119

[Dui86] DUISBERG R. A.: *Constraint-Based Animation: Temporal Constraints in the Animus System*. PhD thesis, Dept. of Computer Science, Univerity of Washington, 1986. Also Tektronix Laboratories Technical Report CR-86-37. 116

[DW97] DENSLEY D. J., WILLIS P. J.: Emotional posturing: A method towards achieving emotional figure animation. In *Proc. IEEE Computer Animation* (1997), pp. 8–14. 116

[FE77] FISCHLER M., ELSCHLAGER R.: The representation and matching of pictorial structures. In *CMetImAly77* (1977), pp. 31–56. 116

[FL01] FISHER S., LIN M.: Fast penetration depth estimation for elastic bodies using deformed distance fields. In *IROS2001* (2001). 124

[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. 125

[FLW00] FROUMENTIN M., LABROSSE F., WILLIS P.: A vector-based representation for image warping. *Computer Graphics Forum 19*, 3 (2000), C385–C394, C428. 127, 128

[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. *to appear in IEEE Transactions on Medical Imaging* (2005). 122

[FSRR00] FANG S., SRINIVASAN R., RAGHAVAN R., RICHTSMEIER J.: Volume morphing and rendering - an integrated approach. *Computer Aided Geometric Design 17*, 1 (January 2000), 59–81. 120

[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–xxx. 127

[FvDFH96] FOLEY J., VAN DAM A., FEINER S., HUGHES J.: *Computer Graphics: Principles and Practice*, 2nd ed. Addison Wesley, 1996. ISBN 0201848406. 116

[Gar99] GARLAND M.: Multiresolution modeling: survey and future opportunities. In *State of the Art Report* (1999), pp. 111–131. 118

[GH90] GILES M., HAIMES R.: Advanced interactive visualization for cfd. *Computing Systems in Engineering 1*, 1 (1990), 51–62. 118

[GH91] GALYEAN T. A., HUGHES J. F.: Sculpting: an interactive volumetric modeling technique. *Computer Graphics (Proc. SIGGRAPH 91) 25*, 4 (1991), 267–274. 124

[Gib97] GIBSON S.: 3D chainmail: a fast algorithm for deforming volumetric objects. In *Proc. 1997 Symposium on Interactive 3D Graphics* (April 1997), pp. 149–154. 124, 125, 126

[GM97] GIBSON S., MIRTICH: *A survey of deformable modeling in computer graphics*. Tech. Rep. TR97-19, MERL Technical Report, 1997. 115

[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. In *IEEE Transactions on Medical Imaging* (2004), vol. 23, pp. 447–458. 122

[GP89] GRIESSMAIR J., PURGATHOFER W.: Deformation of solids with trivariate b-splines. In *Proceedings of Eurographics* (1989). 115

[GP00] GERSTNER T., PAJAROLA R.: Topology preserving and controlled topology simplifying multiresolution isosurface extraction. In *Proc. IEEE Visualization* (2000), pp. 259–266,565. 118

[GS99] GAGVANI N., SILVER D.: Parameter controlled volume thinning. *Graphical Models and Image Processing 61*, 3 (1999), 149–164. 119, 129

[GS00] GAGVANI N., SILVER D.: Animating the visible

human dataset. In *Proc. the Visible Human Project Conference* (2000). 129, 130

[GS01] GAGVANI N., SILVER D.: Animating volumetric models. *Graphical Models 63*, 6 (2001), 443–458. 119, 129, 130, 131

[GTH05] GHOSH A., TRENTACOSTE M., HEIDRICH W.: Volume rendering for high dynamic range displays. In *Proc. Volume Graphics* (2005), p. to appear. 119

[GW01] GONZALEZ R., WOODS R. E.: *Digital Image Processing*, 2nd ed. Prentice-Hall, 2001. 119, 120

[HB03] HEARN D., BAKER M. P.: *Computer Graphics with OpenGL*, 3rd ed. Prentice-Hall, 2003. 127

[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. 124

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *Proceedings of ACM SIGGRAPH'92* (August 1992), Computer Graphics Proceedings, Annual Conference Series, ACM, pp. 295–302. 119

[HDD*93] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Mesh optimization. In *Computer Graphics (Proc. SIGGRAPH 93)*. ACM Press and Addison-Wesley, Anaheim, California, 1993, pp. 19–26. 118

[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. 119

[HHK92] HSU W. M., HUGHES J. F., KAUFMAN H.: Direct manipulation of free form deformation. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (July 1992), 177–184. 115, 125

[HL78] HERMAN G. T., LIU H. K.: Dynamic boundary surface detection. *Computer Graphics and Image Processing 7* (1978), 130–138. 118

[Hop96] HOPPE H.: Progressive meshes. In *Computer Graphics (Proc. SIGGRAPH 97)*. ACM Press, 1996, pp. 99–108. 118

[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* (2003), vol. 5032, pp. 643–653. 121

[HQ04] HUA J., QIN H.: Haptics-based dynamic implicit solid modeling. *IEEE Transactions on Visualization and Computer Graphics 10*, 5 (2004), 574–586. 124

[Hug92] HUGHES J. F.: Scheduled fourier volume morphing. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (July 1992), 43–46. 120

[HWK94] HE T., WANG S., KAUFMAN A.: Wavelet-based volume morphing. In *Proc. Visualization 1994* (Los Alamitos, CA, 1994), IEEE Comuter Society Press, pp. 85–91. 120

[IDSC04] ISLAM S., DIPANKAR S., SILVER D., CHEN M.: Temporal and spatial splitting of scalar fields in volume graphics. In *Proc. IEEE VolVis2004* (Austin, Texas, October 2004), IEEE, pp. 87–94. 124, 125, 126, 129

[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. 118

[Jon95] JONES M. W.: *The Visualisation of Regular Three Dimensional Data*. PhD thesis, University of Wales, 1995. 118

[Jon96] JONES M. W.: The production of volume data from triangular meshes using voxelisation. *Computer Graphics Forum 15*, 5 (1996), 311–318. 120

[Jon01] JONES M. W.: Facial reconstruction using volumetric data. In *Proc. Vision, Modeling, and Visualisation*. IOS Press, 2001, pp. 135–142. 120, 123

[Jon03] JONES M. W.: Melting objects. *The Journal of WSCG 11*, 2 (2003), 247–254. 125

[Jon04] JONES M.: Distance field compression. *WSCG 12*, 2 (2004), 199–204. 120

[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. 120

[JW89] JOHN N. W., WILLIS P. J.: The Controller animation system. *Computer Graphics Forum 8* (1989), 133–138. 116

[KCM00] KUHNAPFEL U., CAKMAK H. K., MASS H.: Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computer & Graphics 24* (2000), 671–682. 125

[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. 121

[Ker04] KERLOW I. V.: *The art of 3D computer animation and effects*. John Wiley & Sons, Inc, 2004. 114, 116

[KFW*02] KANITSAR A., FLEISCHMANN D., WEGENKITTL R., FELKEL P., GRÖLLER M. E.: CPR: Curved planar reformation. In *Proc. IEEE Visualization 2002* (2002), pp. 37–44. 119

[KK99] KREEGER K., KAUFMAN A.: Interactive volume segmentation with the PAVLOV architecture. In *Proc. IEEE Symposium on Parallel Visualization and Graphics* (1999), ACM Press, pp. 61–68. 122

[KL96] KRISHNAMURTHY V., LEVOY M.: Fitting smooth surfaces to dense polygon meshes. In *Computer Graphics (Proc. SIGGRAPH 96)*. Addison-Wesley, New Orleans, Louisiana, 1996, pp. 313–324. 119

[Kle89] KLECK J.: *Modeling Using Implicit Surfaces*. Master's thesis, University of California at Santa Cruz, Santa Cruz, CA, 1989. 115

[KMH*04] KEISER R., MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Contact handling for deformable point-based objects. In *Proc. Vision, Modeling, Visualization (VMV)* (November 2004), pp. 315–322. 125, 126

[KWT87] KASS M., WITKIN A., TERZOPOULOS D.: Snakes: Active contour models. *International Journal of Computer Vision 1*, 4 (1987), 321–331. 115, 122, 125

[KY95] KURZION Y., YAGEL R.: Space deformation using ray deflectors. In *Proc. 6th Eurographics Workshop on Rendering* (1995), pp. 21–32. 124, 125, 126

[LBC96] LEI Z., BLANE M. M., COOPER D. B.: 3l fitting of higher degree implicit polynomials. In *WACV '96: Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96)* (Washington, DC, USA, 1996), IEEE Computer Society, p. 148. 119

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics (Proc. SIGGRAPH 87) 21*, 4 (July 1987), 163–169. 118

[Lee00] LEE I.-K.: Curve reconstruction from unorganized points. *Computer Aided Geometric Design 17*, 2 (2000), 161–177. 119

[Lev88] LEVOY M.: Display of surfaces from volume data. *IEEE Computer Graphics and Applications 8*, 3 (May 1988), 29–37. 113

[Ley03] LEYMARIE F.: *3D Shape Representation via the Shock Scaffold*. PhD thesis, Brown University, 2003. 119

[LGL95] LERIOS A., GARFINKLE C., LEVOY M.: Feature-based volume metamorphosis. In *Computer Graphics (Proc. SIGGRAPH 95)*. Addison-Wesley, Los Angeles, California, 1995, pp. 449–456. 120

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Computer Graphics (Proc. SIGGRAPH 96)*. Addison-Wesley, New Orleans, Louisiana, 1996, pp. 31–42. 113

[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), Australian Computer Society, Inc., pp. 31–37. 121

[LK03] LAKARE S., KAUFMAN A.: Anti-aliased volume extraction. In *Proc. Eurographics/IEEE TCVG Symposium on Data Visualisation* (Grenoble, France, 2003), Eurographics, pp. 113–122. 121

[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* (2003), IEEE, pp. 75–82. 124, 125

[LL93] LEVY H., LESSMAN F.: *Finite Difference Equations*. Dover Publications, 1993. 116

[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 *Proc. SIGGRAPH 2000* (2000), ACM Press, pp. 131–144. 118

[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). 118

[LTCK03] LIU A., TENDICK F., CLEARY K., KAUFMANN C.: A survey of surgical simulation: applications, technology and education. *Presence 12*, 6 (December 2003), 599–614. 115, 127

[LTSE92] LOKE T., TAN D., SEAH H., ER M.: Rendering fireworks displays. *IEEE Computer Graphics and Application 12*, 3 (May 1992), 33–43. 116

[LvF00] LASZLO J., VAN DE PANNE M., FIUME E.: Interactive control for physically-based animation. In *Computer Graphics (Proc. SIGGRAPH 2000)*. ACM Press, 2000, pp. 201–209. 116

[LW94] LAMOUSIN H. J., WGGENSPACK W. N.: Nurbs-based free-form deformations. In *Proc. of IEEE Computer Graphics and Application* (1994), IEEE, pp. 59–65. 115, 125

[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. 115

[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* (October 2003). 119

[Mal93] MALZBENDER T.: Fourier volume rendering. *ACM Transactions on Graphics 12*, 3 (1993), 233–250. 119, 120

[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 (July 1991), 217–226. 122

[MBR*00] MATUSIK W., BUEHLER C., RASKAR R., GORTLER S. J., MCMILLAN L.: Image-based visual hulls. In *Proc. SIGGRAPH 2000* (2000), ACM Press, pp. 343–352. 118

[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. 123

[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* (San Antonio, Texas, 2002), pp. 49–54. 125, 126, 127

[MG04] MÜLLER M., GROSS M.: Interactive virtual materials. In *Proc. Graphics Interface (GI 2004)* (May 2004), pp. 239–246. 126

[Mil88] MILLER G.: The motion dynamics of snakes and worms. *Computer Graphics (Proc. SIGGRAPH 88) 22*, 4 (1988), 169–178. 115

[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. 125, 126

[Mor90] MORGAN II J. C.: *Point Set Theory*. Marcel Dekker, 1990. 117

[MS93] MUELLER H., STARK M.: Adaptive generation of surfaces in volume data. *The Visual Computer 9* (1993), 182–199. 118

[MSCS03] MOLLEMANS W., SCHUTYSER F., CLEYNENBREUGEL J. V., 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. 125, 127

[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. 126

[MT95] MCINERNEY T., TERZOPOULOS D.: Topologically adaptable snakes. In *Proc. 5th International Conference on Computer Vision* (1995), IEEE Computer Society, p. 840. 122

[MT96] MCINERNEY T., TERZOPOULOS D.: Deformable models in medical image analysis: A survey. *Medical Image Analysis 1*, 2 (1996), 91–108. 115

[MTB03] MCGUFFIN M. J., TANCAU L., BALAKRISHNAN R.: Using deformations for browsing volumetric data. In *Proc. IEEE Visualization 2003* (2003), Seattle, WA, pp. 401–408. 124, 125

[MTG04] MÜLLER M., TESCHNER M., GROSS M.: Physically-based simulation of objects represented by surface meshes. In *Proc. Computer Graphics International (CGI)* (June 2004), pp. 26–33. 125, 126

[MTT96] MAGNENAT-THALMANN N., THALMANN D. (Eds.): *Interactive Computer Animation*. Prentice Hall, 1996. 116

[MV98] MAINTZ J. B. A., VIERGEVER M. A.: A survey of medical image registration. *Medical Image Analysis 2*, 1 (1998), 1–36. 121, 123

[MVdF03] MEDEROS B., VELHO L., DE FIGUEIREDO L. H.: Moving least squares multiresolution surface approximation. In *Proc. SIBGRAPI* (October 2003), IEEE Press, pp. 19–26. 119

[Nay90] NAYLOR B.: Sculpt: an interactive solid modeling tool. In *Proceedings on Graphics interface '90* (1990), pp. 138–148. 124

[NC99] NOBLE R. A., CLAPWORTHY G. J.: Direct manipulation of surfaces using nurbs-based free-form deformations. In *Proc. of International Conference on Information Visualisation* (London, UK, July 1999), IEEE, pp. 238–243. 125

[NH91] NIELSON G., HAMANN B.: The asymptotic decider: resolving the ambiguity in marching cubes. In *Proc. IEEE Visualization* (Octorber 1991), IEEE Computer Society, pp. 83–90. 118

[Nie04] NIELSON G. M.: Dual marching cubes. In *Proc. IEEE Visualization* (2004), pp. 489–496. 118

[NMK*05] NEALEN A., MÜLLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically based deformable models in computer graphics. In *Eurographics State of the Art Report* (2005). 115

[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. 115

[NS97] NIELSON G., SUNG J.: Interval volume tetrahedralization. In *Proc. IEEE Visualization'97* (October 1997), pp. 221–228. 118

[NT98] NEDEL L. P., THALMANN D.: Real-time muscle deformation using mass-spring system. In *Proc. Computer Graphics International* (June 1998), pp. 156–165. 125

[O'R95] O'ROURKE M.: *Principles of Three-Dimensional Computer Animation*. W. W. Norton and Company, 1995. 116

[Par01] PARENT R.: *Computer Animation: Algorithms and Techniques*. Morgan-Kaufmann, 2001. 116

[PC87] PERRY A. E., CHONG M. S.: A description of eddying motions and flow patterns using critical-point concepts. *Ann. Review of Fluid Mechanics 19* (1987), 125–155. 119

[PDA03] PICINBONO G., DELINGETTE H., AYACHE N.: Non-linear anisotropic elasticity for real-time surgery simulation. *Graphical Models 65*, 5 (2003), 305–321. 125, 127

[Ped95] PEDERSEN H.: Displacement mapping using

flow fields. In *Computer Graphics (Proc. SIGGRAPH 95)*. Addison-Wesley, Los Angeles, California, 1995, pp. 291–300. 115

[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. 119

[PKKG03] PAULY M., KEISER R., KOBBELT L., GROSS M.: Shape modeling with point-sampled geometry. In *SIGGRAPH 2003* (2003). 124, 125

[PLDA00] PICINBONO G., LOMBARDO J., DELINGETTE H., AYACHE N.: Anisotropic elasticity and force extrapolation to improve realism of surgery simulation. In *In Proc. IEEE Intl. Conf. Robotics & Automation* (April 2000). 125

[PM96] PROAKIS J. G., MANOLAKIS D. G.: *Digital Signal Processing*, third ed. Prentice-Hall, 1996. 117

[Poc01] POCOCK L.: *The Computer Animator's Technical Handbook*. Morgan Kaufmann, 2001. 116

[PPG04] PAULY M., PAI D. K., GUIBAS L. J.: Quasi-rigid objects in contact. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation* (2004). 125

[Pra87] PRATT V.: Direct least-squares fitting of algebraic surfaces. *Computer Graphics (Proc. SIGGRAPH 87) 21*, 4 (1987), 145–152. 119

[PSRP00] PORTER B. C., STRANG J. G., RUBENS D. J., PARKER K. J.: 3D volume registration: experiments on tissue phantom and liver. *IEEE Ultrasonics Symposium 2* (2000), 1581–1584. 123

[PT92] PAYNE B. A., TOGA A. W.: Distance field manipulation of surface models. *IEEE Computer Graphics and Applications 12*, 1 (1992), 65–71. 120

[PW89] PENTLAND A., WILLIAMS J.: Good vibrations: modal dynamics for graphics and animation. *Computer Graphics (Proc. SIGGRAPH 89) 23*, 3 (July 1989), 215–222. 125, 126

[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* (August 1998), 586–595. 123

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: surface elements as rendering primitives. In *Computer Graphics (Proc. SIGGRAPH 2000)*. ACM Press, 2000, pp. 335–342. 113, 118

[QW03] QI M., WILLIS P.: Quasi-3D cel-based animation. In *Proc. Vision, Video and Graphics 2003* (Bath, July 2003), Eurographics, pp. 111–116. 128

[RA00] REYES-ALDASORO C. C., ALGORRI M. E.: A combined algorithm for image segmentation using neural networks and 3D surface reconstruction using dynamic meshes. In *Rev Mex Ing Biomed* (2000), vol. 21, pp. 73–81. 121

[RC04] RODGMAN D., CHEN M.: Volume denoising for visualizing refraction. In *Scientific Visualization, Dagstuhl Scientific Visualization 2003*, Bonneau G. P., Nielson G.,, Ertl T., (Eds.). Springer, 2004, p. to appear. 119

[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. 124

[Ree83] REEVES W. T.: Particle systems – a technique for modeling a class of fuzzy objects. *Computer Graphics (Proc. SIGGRAPH 83) 17*, 3 (1983), 359–376. 116

[Rey82] REYNOLDS C. W.: Computer animation with scripts and actors. *Computer Graphics (Proc. SIGGRAPH 82) 16*, 3 (1982), 289–296. 116

[Rey87] REYNOLDS C. W.: Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics (Proc. SIGGRAPH 87) 21*, 4 (1987), 25–34. 116

[RGR97] RAJAPAKSE J. C., GIEDD J. N., RAPOPORT J. L.: Statistical approach to segmentation of single-channel cerebral MR images. In *IEEE Transactions on Medical Imaging* (1997), vol. 16, pp. 176–186. 121

[RHK*00] RUSÀK Z., HORVÀTH I., KUCZOGI G., VERGEEST J. S. M., JANSSON J.: First results of the development of particle system modelling for shape conceptualisation. In *Proc. International Design Conference 2000* (Dubrovnik, May 2000), pp. 339–346. 116

[RRTP99] RADETZKY A., RNBERGER A., TEISTLER M., PRETSCHNER D.: Elastodynamic shape modeling in virtual medicine. In *Int. Conf. on Shape Modeling and Application, 1999* (Piscataway, NJ, 1999), IEEE Computer Society, pp. 172–178. 125

[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. 123

[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* (2001), pp. 17–24. 125, 126

[Rut99] RUTTKAY Z.: *Constraint-based facial animation*. Tech. rep., CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, The Netherlands, 1999. 116

[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. 125, 127

[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. 117, 120

[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 (June 1996), 163–175. 123

[SHN03] SHERBONDY A., HOUSTON M., NAPEL S.: Fast volume segmentation with simultaneous visualization using programmable graphics hardware. In *Proc. IEEE Visualization 2003* (2003), pp. 171–176. 122

[SJ01] SATHERLEY R., JONES M. W.: Vector-city vector distance transform. *Computer Vision and Image Understanding 82*, 3 (2001), 238–254. 120

[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, London, 2000, pp. 119–134. 119

[SKK02] SEBASTIAN T., KLEIN P., KIMIA B.: Shock-based indexing into large shape databases. *ECCV 3* (2002), 731–746. 119

[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. MIT 2003* (2003), pp. 1–8. 121

[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* (2002), 1419–1426. 119, 120

[Sny92] SNYDER J. M.: *Generative Modeling for Computer Graphics and CAD*. Academic Press, Boston, 1992. 115

[SP86] SEDERBERG T. W., PARRY S. R.: Free-form deformation of solid geometric models. *Computer Graphics (Proc. SIGGRAPH 86) 20*, 4 (1986), 151–160. 115, 116, 120, 124, 125, 126

[SP91] SCLAROFF S., PENTLAND A.: Generalized implicit functions for computer graphics. *Computer Graphics 25*, 1 (1991), 247–250. 115, 125

[SPE*90] SCHAROFF S. E., PENTLAND A., ESSA I., FRIEDMANN M., HOROWITZ B.: The thingworld modeling system: virtual sculpting by modal forces. *SIGGRAPH Computer Graphics 24*, 2 (1990), 143–144. 124, 126

[SS96] SCHRODER P., SWELDENS W. (Eds.): *Wavelets in Computer Graphics*. ACM SIGGRAPH '96 Course Notes, 1996. 120

[SS04] SINGH V., SILVER D.: Interactive volume manipulation with selective rendering for improved visualization. In *IEEE Symposium on Volume Visualization and Graphics* (October 2004), pp. 95–102. 129

[SSC03] SINGH V., SILVER D., CORNEA N.: Real-time volume manipulation. In *Proc. Eurographics/IEEE TCVG Workshop on Volume Graphics 2003* (Tokyo, Japan, 2003), pp. 45–51. 126, 129

[SSGD03] SUNDAR H., SILVER D., GAGVANI N., DICKINSON S.: Skeleton based shape matching and retrieval. In *Proc. Shape Modeling and Applications Conference (SMI 2003)* (May 2003), pp. 130–138. 119

[SvL*03] STRAKA M., ŠRÁMEK, LA CRUZ A., GRÖLLER E., KÖCHL A., FLEISCHMANN D.: Bone segmentation in CT-angiography aata using a probabilistic atlas. In *Proc. VMV 2003* (2003). 121

[SW04] SU D., WILLIS P.: Image interpolation by pixel level data-dependent triangulation. *Computer Graphics Forum 23*, 2 (2004), 189–201. 128

[SWY99] SHAREEF N., WANG D. L., YAGEL R.: Segmentation of medical images using LEGION. *IEEE Transactions on Medical Imaging 18*, 1 (1999). 121

[SZL92] SCHROEDER W. J., ZARGE J. A., LORENSEN W. E.: Decimation of triangle meshes. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 65–70. 118

[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. 125, 127

[TFT04] TAKAHASHI S., FUJISHIRO I., TAKESHIMA Y.: Topological volume skeletonization and its application to transfer function design. *Graphical Models 66*, 1 (2004), 24–49. 118

[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. 119

[TK95] TEK H., KIMIA B. B.: Image segmentation by reaction-diffusion bubbles. In *Proc. ICCV95* (1995), pp. 156–162. 122

[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. 119

[TL94] TURK G., LEVOY M.: Zippered polygon meshes from range images. In *Computer Graphics (Proc. SIGGRAPH 94)* (Orlando, Florida, 1994), ACM Press and Addison-Wesley, pp. 311–318. 119

[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. 123

[TM91] TERZOPOULOS D., METAXAS D.: Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Transactions on Pat-*

tern Analysis and Machine Intelligence 13, 7 (1991), 703–714. 115, 125

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *Computer Graphics (Proc. SIGGRAPH 87) 21*, 4 (1987), 205–214. 115, 116, 125, 126

[TQ94] TERZOPOULOS D., QIN H.: Dynamic nurbs with geometric constraints for interactive sculpting. *ACM Transactions on Graphics 13*, 2 (1994), 103–136. 116

[Tsi95] TSITSIKLIS J. N.: Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control 40*, 9 (September 1995), 1528–1538. 117, 120

[Tur92] TURK G.: Retiling polygonal surfaces. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 55–64. 118

[TW88] TERZOPOULOS D., WITKIN A.: Physically based models with rigid and deformable components. *IEEE Computer Graphics and Applications 8*, 6 (November 1988), 41–51. 126

[UT91] UNUMA M., TAKEUCHI R.: Generation of human motion with emotion. In *Proc. Computer Animation* (1991), pp. 77–88. 116

[Vin92] VINCE J.: *3-D Computer Animation*. Addison-Wesley, 1992. 116

[VKKM03] VARADHAN G., KRISHNAN S., KIM Y. J., MANOCHA D.: Feature sensitive subdivision and isosurface reconstruction. In *Proc. IEEE Visualization* (2003), pp. 99–106. 118

[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. 118

[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. 121

[WDGT01] WU X., DOWNES M. S., GOKTEKIN T., TENDICK F.: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. *Comput. Graph. Forum 20*, 3 (2001). 127

[Wes90] WESTOVER L.: Footprint evaluation for volume rendering. *Computer Graphics (Proc. SIGGRAPH 90) 24*, 4 (August 1990), 367–376. ISBN 0-201-50933-4. 113

[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. 115, 125

[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* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 359–362. 120

[Whi04] WHITAKER R.: Modeling deformable surfaces with level sets. *IEEE Computer Graphics and Applications 24*, 5 (Septmber-October 2004), 6–9. 124

[WK95] WANG S., KAUFMAN A.: Volume sculpting. In *Proc. Symposium on Interactive 3D Graphics* (Monterey, CA, April 1995), ACM Press, pp. 151–156. 124

[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. 121

[WMW86] WYVILL B., MCPHEETERS C., WYVILL G.: Animating soft objects. *The Visual Computer 2*, 4 (August 1986), 235–242. 125

[WP00] WU Z., PRAKASH E. C.: Visual human animation. In *Volume Graphics*, Chen M., Kaufman A. E.,, Yagel R., (Eds.). Springer, London, 2000, pp. 243–252. 129

[WP02] WADE L., PARENT R.: Automated generation of control skeletons for use in animation. *The Visual Computer 18*, 2 (March 2002), 97–110. 119

[WS01] WESTERMANN R., SALAMA C.: Real-time volume deformations. *Computer Graphics Forum 20*, 3 (2001). 126

[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. 119

[WT04] WU X., TENDICK F.: Multigrid integration for interactive deformable body simulation. In *Proc. ISMS* (2004), pp. 92–104. 125, 127

[Wv92] WILHELMS J., VAN GELDER A.: Octrees for faster isosurface generation. *ACM Transactions on Graphics 11*, 3 (July 1992), 201–227. ISSN 0730-0301. 118

[WW90] WITKIN A., WELCH W.: Fast animation and control of nonrigid structures. *Computer Graphics (Proc. SIGGRAPH 90) 24*, 4 (1990), 243–252. 125, 126

[WW92a] WATT A., WATT M.: *Advanced Animation and Rendering Techniques: Theory and Practice*. ACM Press, 1992. 114, 116

[WW92b] WITKIN A., WELCH W.: A. witkin and w. welch. *Computer Graphics (Proc. SIGGRAPH 92) 26*, 2 (1992), 157–166. 125, 126

[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* (2005), 1901–1917. 125

[YP92] YUN H., PARK K. H.: Surface modelling method by polygonal primitives for visualizing three-dimensional volume data. *The Visual Computer 8* (1992), 246–259. 118

[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. 123

[ZC95] ZICHENG L., COHEN M. F.: An efficient symbolic interface to constraint based animation systems. In *Proc. 6th Eurographics Workshop on Computer Animation and Simulation* (1995), pp. 210–235. 116

[ZC00] ZHUANG Y., CANNY J. F.: Real-time global deformations. In *Proc. 4th International Workshop on Algorithmic Foundations of Robotics* (2000). 127

[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. 127

[ZF03] ZITOVA B., FLUSSER J.: Image registration methods: A survey. *Image and Vision Computing 21* (2003), 977–1000. 122, 123

[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. 119

[ZPBG01] ZWICKER M., PFISTER H., BAAR J., GROSS M.: EWA volume splatting. In *Proc. IEEE Visualization 2001* (San Diego, CA, 2001), IEEE/TCVG, pp. 29–36. 113

[ZT89] ZIENKIEWICZ O. C., TAYLOR R. L. (Eds.): *The Finite Element Method*. McGraw-Hill, 1989. 116