Multiscale Spatiotemporal Visualisation

STREP # FP7-248032

Report on Best Practice

D4.1

Work package 4: Best Practice for MSV Software Development

Nigel McFarlane (BED)
Xiangyin Ma (BED)
Gordon Clapworthy (BED)
Nik Bessis (BED)
Debora Testi (B3C)

12/01/2012

SEVENTH FRAMEWORK PROGRAMME

eHealth

Document Information

IST Project Number	FP7-248032	**Acronym**	MSV
Full title	Multiscale Spatiotemporal visualisation: development of an open-source software library for the interactive visualisation of multiscale biomedical data		
Project URL	http://www.msv-project.eu		
Document URL	http://www.biomedtown.org/biomed_town/MSV/associates/reviewers/		
EU Project officer	Ivo Locatelli		

Deliverable Number	D4.1	**Title**	Best Practice for MSV Software Development
Work package Number	4	**Title**	Best Practice

Date of delivery	**Planned**	31-12-2011	**Actual**	12-01-2012
Status	Version v3		final ☒ Draft ☐	
Nature	Prototype ☐ Report ☒ Demonstrator ☐ Other ☐			
Dissemination Level	Public ☒ Consortium ☐			

Authors (Partner)	Nigel McFarlane (BED), Xiangyin Ma (BED), Gordon Clapworthy (BED), Nik Bessis (BED), Debora Testi (B3C), with input from all partners			
Responsible Author	Gordon Clapworthy		**Email**	gordon.clapworthy@beds.ac.uk
	Partner	BED	**Phone**	+44 1582 743440

Abstract (for dissemination)	The aim of the MSV project is to develop an open source library for the visualisation and interaction with multiscale biomedical data. This document summarises visualisation techniques in general, and multiscale visualisation techniques in particular, in order to identify a coherent approach to multiscale visualisation that can be adapted to whatever data the user is confronted with. It presents possible features of the data and other factors that the user should consider when designing multiscale visualisation to be used in a specific application.	
Keywords	Multiscale visualisation	Best practice

Version Log

Issue Date	Rev No.	Author	Change
10/11/2011	V1	Nigel McFarlane, Xiangyin Ma	Initial version circulated to partners
29/11/2011	V2	Nigel McFarlane, Xiangyin Ma, Gordon Clapworthy, Nik Bessis	V1 corrected, comments from partners incorporated
12/01/2012	V3	Nigel McFarlane, Xiangyin Ma, Gordon Clapworthy, Nik Bessis, Debora Testi	Minor changes to V2, wrapped in official format for submission

Project Consortium Information

Disclaimer: This document is property of the MSV Consortium. There is no warranty for the accuracy or completeness of the information, text, graphics, links or other items contained within this material. This document represents the common view of the consortium and does not necessarily reflect the view of the individual partners.

List of Abbreviations

3D	Three-dimensional
AMR	Adaptive Mesh Refinement
API	Application Programming Interface
AUK	University of Auckland, NZ
B3C	BioComputing Competence Centre - SCS srl, Italy
BED	University of Bedfordshire, UK
BVH	Bounding Volume Hierarchy
CFD	Computational Fluid Dynamics
CPR	Curved Planar Reformation
CT	Computed Tomography
DTI	Diffusion Tensor Imaging
DVR	Direct Volume Rendering
FEM	Finite Element Method
FPS	Frames per second
GPU	Graphic Processing Units
GUI	Graphic User Interface
HCI	Human-computer interaction
KIT	Kitware Inc., USA
LIC	Line Integral Convolution
LoD	Level of Detail
MPR	Multi-Planar Reconstruction
MRI	Magnetic Resonance Imaging
MSV	Multiscale Visualisation and Interaction project
MSVE	Multiscale Virtual Environment
OBB	Oriented Bounding Box
PSC	Power Scaled Coordinates
TAC	Time Activity Curve
UPF	Universitat Pompeu Fraba, Spain
US	Ultrasound
VPH	Virtual Physiological Human
VTK	Visualization Toolkit

Contents

1 Introduction

WP4, *Best Practice*, aims to form a bridge between WP3, *Exemplary Problems* and WP5, *Shared Implementation*. Following the identification of the exemplary problems, the objective of WP4 is to identify and, by the means of exploration and discussion, to support and formalise the development process of the shared implementation toolkit for MSV that will take place in WP5. It will help assist the final implementation by providing best practice methods and suggest the use of applicable state-of-the-art technologies for adoption during WP5.

The term "multiscale visualisation" is generally understood to mean that an image or scene contains detail over a range of scales that exceeds the resolution of the display or the human eye. The term encompasses a wide variety of types of data and the techniques required to visualise it; for example, the multiscale may be spatial or temporal; data may be a single image, a multi-object scene, or perhaps a high-resolution graph; and techniques may be required to deal with occlusions, ill-conditioning, levels-of-detail and scene navigation.

This deliverable will survey the different forms that multiscale data may take, the common problems that need to be addressed and the techniques that may be applied to deal with them. It will propose a classification of multiscale data and visualisation styles by which the challenges posed by a particular problem can be matched to corresponding techniques. Techniques appropriate for pre-analysing the multiscale structure of a scene will be surveyed and applied to example data. Finally, for the purposes of the MSV project, recommendations will be made concerning the kinds of technique that might be included in a multiscale visualisation library.

2 Literature survey - overview

Domains such as medical visualisation, architecture and urban design, geospatial scanning, astrophysics, biochemistry and abstract data analysis are regularly producing massive datasets containing features that are many orders of magnitude apart in scale. The challenges posed by such datasets can be divided into two categories: those of interactively navigating the content and those of structuring and serialising data which is too large for the typical 1GB graphics pipeline. This work is largely concerned with the former; we will include a brief discussion of the size issue later, though a comprehensive review of this field would be beyond the scope of this report.

This survey will add to the preliminary reviews of multiscale applications and techniques that have already been reported in the MSV White Paper (D2.1) and Problem Assessment Exercise (D3.1). This overview section gives a review of activity in multiscale visualisation from the point of view of four research fields: cartography, astrophysics, biomedicine and information visualisation (InfoVis). The techniques arising from this overview will be discussed in detail in later sections.

2.1 Cartography and GIS

The most well-known example of a multiscale visualisation is Google Earth (Garfinkel, 2007; Google Earth, 2011), in which the dataset consists of the whole planet, from the entire globe down to street level. The main visualisation in Google Earth is a 2D cartographic view based on a zooming and Level of Detail (LoD) approach: the more the user zooms at a higher resolution the more detail is rendered. The user interaction has been proven to be very effective and user-friendly and it might be taken into consideration in the navigation of biomedical datasets. Google Earth also includes a 3D visualisation at street level, with a click-and-fly interaction, in which the mouse is used to move through the scene. The transition from 2D map to street level is discrete, requiring a dialog interaction to insert the user in the desired location.

Another example from the geosciences is the software package CoViz4D (CoViz 4D, 2011) which is designed for visualising changes in oil reservoirs with time and includes topographic and seismic visualisations with a temporal dimension. Navigation at different spatial scales is based on a zoom-in/zoom-out interaction.

The challenges of navigation in a multiscale cartographic environment have been addressed in detail in McCrae *et al*. (2009). A navigation tool called "cubemap" projects a range map of the nearest objects onto the orthogonal faces of a cube and uses this to automatically sense the scale of the immediate surroundings for the purposes of visualisation and flying speed. The automatic sensing of scale allows the user to make a seamless transition from a 2D cartographic view to 3D street level and to move in and out of buildings. Ill-conditioning of the coordinates of small features was found to be a serious problem when the origin was fixed at the centre of the Earth; this was avoided by rendering all objects relative to a mobile origin. All interactions, such as flying and hovering were controlled via the mouse. Figure 1 shows a cubemap navigation from the global scale to the inside of a jug. The figure 1 sequence also shows how the vertical changes smoothly from the North Pole to the true vertical as the view approaches ground level.

Figure 1. Navigation between scales in 3D scenes. Here, the user navigates from thousands of kilometres above the Earth's surface to come to rest inside a jug on a table only centimetres in diameter. Note the smooth change in the vertical direction as the view approaches ground level. (McCrae et al., 2009)

Further analysis of navigation in multiscale environments was reported by McCrae *et al*. (2010). It is difficult for users to orient themselves (i.e. understand their pose and location) in cluttered multiscale scenes. A large variety of navigation widgets such as radars and maps were proposed to assist with user orientation. Navigation can be entirely freeform, or guided to specific targets, which might be pre-defined by the creators of the scene, or computed by some form of space partitioning algorithm such as a Voronoi tessellation.

All multiscale visualisations must consider what to do about the Level of Detail (LoD) which is to be rendered at a particular scale. Does the data need to be resampled, redrawn or relabelled at different scales? Should data which is too small to see be marked by a placeholder, redrawn to suit the scale, or simply allowed to vanish? In its simplest form, LoD might simply consist of modifying the resolution of an image to avoid processing voxels which are too small to be seen. In cartography, it is of paramount importance, since properties such as connectivity must be conserved when the scale changes, and a large body of algorithms exists describing how features should be added, deleted and redrawn at different scales. The principles of LoD in cartography are comprehensively described in Li (2007).

A further issue in multiscale cartography is the display of Geographical Information Systems (GIS) data, where the visualisation may consist of many different layers, which can originate from different sources and even be copyrighted to different owners (Sayar *et al*., 2009). The layering paradigm is significant because the composite data contains different semantic components, each of which might have a characteristic scale, so a change in one can cause an implicit change in the other. Medical data is not yet as far down this road as GIS, but the integration of levels with different scales and semantics, from genetics to cells and organs, is currently being driven by programmes such as VPH (2011), IUPS Physiome (2011) and NSR Physiome (2011) and this coupling of scale with semantics is something which should be borne in mind.

2.2 Astrophysics

Applications such as astrophysics, where a single scene could in principle contain visible objects from the subatomic scale to the intergalactic scale, provide a good proving ground for multiscale problems. From the scale of galactic super-clusters down to the quarks in atomic nuclei, we have approximate sizes ranging from 10^{25} meters down to 10^{-15} m. Figure 2 shows a poster of the structure of the known universe. The illustration uses a call-out technique to magnify sub-volumes and link them to their position in the whole.

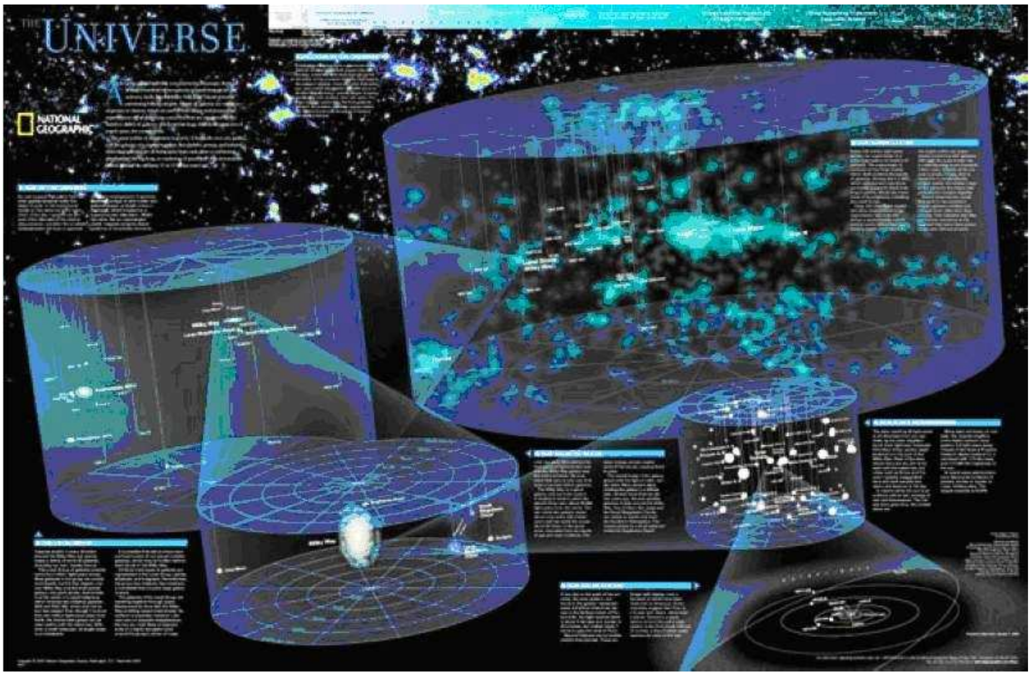

Figure 2 Artist's rendering of the known universe from our solar system to the most distant super clusters (McCrae et al., 2010, National Geographic)

It is important to find a way to allow many orders of magnitude to be represented without arithmetic precision errors. Hanson *et al*. (2000) were able to visualise 40 orders of magnitude with log scale homogeneous coordinates, in which the fourth homogeneous coordinate was interpreted as an order-of-magnitude exponent. All foreground objects were rendered at or close to unit scale and background objects too distant to present any motion parallax were rendered as environment textures.

Various multiscale visualisation platforms have now been developed and are being used in astrophysics. Li *et al*. (2006) developed an astrophysical navigation system called a scalable WIM (world in miniature) using power-scaled coordinates and log scales to manage the distances and magnifying call-outs and various in-view widgets for navigation. The SCISS Uniview architecture (Uniview, 2011) is a real-time capable visualisation platform, aimed at planetarium displays, based on nested scene graphs containing data of arbitrary scale into a joint ScaleGraph representation. The AMUSE project (Amuse Project, 2011) is a Python based framework for the multiscale simulations of dense stellar systems. This includes some parallel computing codes to manage large simulations. AMUSE is free to download and aims to attract an active community of developers. FESTIVAL (Auchere *et al*., 2008) is a visualisation tool developed for viewing satellite images of the solar corona and tracking coronal mass ejections as they grow by orders of magnitude in the heliosphere. The system integrates images from several different instruments into a composite image and includes stereo visualisation.

2.3 Biology and biomedicine

In biology and biomedicine there has been an exponential increase in the size and complexity of datasets, creating challenges in usability, visual analytics and standardisation (O'Donoghue *et al*., 2010a). Biological data visualisation often deals with a broad range of scales, from whole organisms to macromolecules, where even genes and proteins are themselves multiscale structures. Akkiraju *et al*. (1996) used an immersive virtual environment called a CAVE to explore protein structure in multiscale. Some organs such as the lungs have an intricate multiscale structure and are the subject of much modelling research, e.g. Wiechert *et al*. (2011). To be useful, the graphical representations need to be adjustable, so as to display the level of detail appropriate to a particular scale. For example, the standard visualisations of protein structure vary from ball-and-stick representations of individual atoms, to ribbons and sheets representing particular structures, to space-filling models of the whole protein (O'Donoghue *et al*., 2010b; Petsko & Ringe, 2009). In genomics it is a considerable challenge to move seamlessly between scales, from nucleotide to chromosome (Nielsen *et al*., 2010). A novel multiscale interface has been implemented in the VISTA Synteny Viewer, a tool for comparing genomes.

A web-based interface for navigating human anatomy, called the Body Browser, was created by Google in 2010, and is currently awaiting relaunch under new ownership as Zygote Body (Zygote Body, 2011). The tool allowed organs to be explored using a navigation widget and different levels of transparency. A similar 3D anatomy browser can be found at the Biodigital Human (2011).

Systems and pathologies in the human body are often hierarchical, with cumulative events on the molecular and cellular scales propagating upwards to the tissue and organ levels. The ability to use a single integrated tool for the visualisation of multiscale simulation data is important for understanding the effects that events at one scale have on events in the other. Figure 3 shows a montage of various multiscale systems in the human body, taken from O'Donoghue *et al*. (2010a).

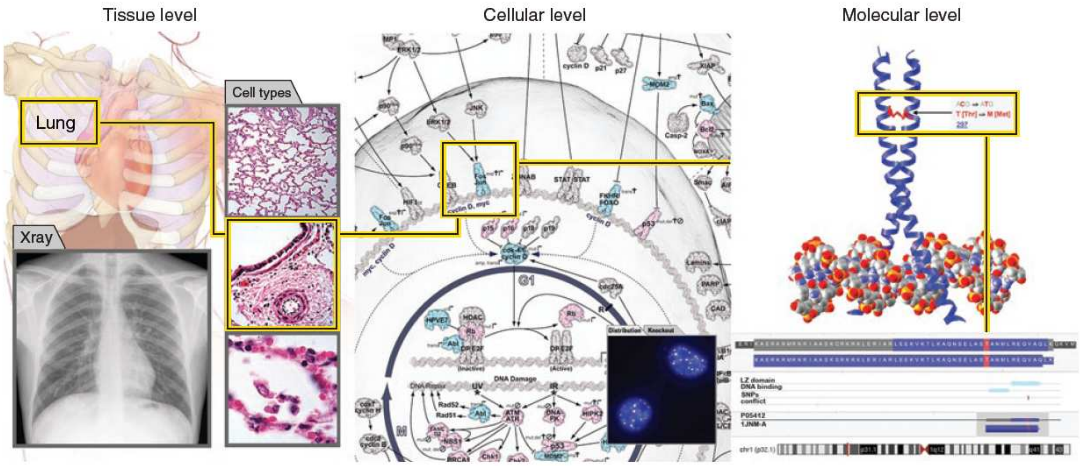

Figure 3. Possible integrated visualisation environment, in which biologists may be able to seamlessly move between data from tissue, cellular and molecular scale (O'Donoghue et al., 2010a).

An example of a multiscale system is the blood flow in the brain. Insley *et al*. (2011) developed an integrated visualisation for simulations of brain aneurysms. Two main scales were important: the macro level of the blood flow through the vessels; and the micro level of the arterial wall, where platelet deposition occurs. The visualisation included simultaneous views at different scales, but did not include any in-view navigation between scales. The two-scale model of blood flow, consisting of the local fluid dynamics or cellular processes as one component and the macro circulation as the other, is commonly used.

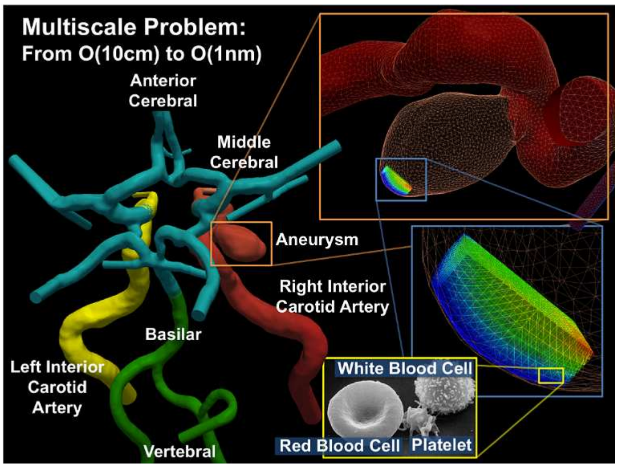

Figure 4. Blood flow in the brain is a multiscale problem. Shown on the left is the macro domain where the spectral element equations are solved; different colours correspond to different computational patches. Shown in the inset (right), located next to the arterial wall of the aneurysm, is the micro domain where dissipative particle dynamics is applied (Insley et al., 2011).

An important imaging method for the brain is diffusion tensor MRI (DTMRI) which measures the diffusion of water, revealing the structure of the nerve fibres in the brain. Associated with this is the visualisation technique of tractography, which tracks the fibres, showing how the brain is connected. A platform called Saturn (Cardenes *et al*., 2010), developed for analysis of DTMRI images, includes many multiresolution features for clustering the fibres – an important LoD technique known as bundling, which is encountered when groups of fibres or connections are viewed at different scales.

A review of medical imaging methods (Walter *et al*., 2010) indicated various challenges that may be encountered when visualising spatial scales from cells to organisms. The integration of data that originates from multiple simulation tools requires the solution of the recurring problem of their having different file formats and different reference systems. Even the DICOM format is not entirely standardised and it is often not used because of its complexity. Non-rigid registration of images from different devices was also identified as a problem. The need for methods for organising, sharing and searching biological images with associated metadata was also identified in Auer *et al*. (2007).

A multiscale view for medical data was developed by McFarlane *et al*. (2008) and Viceconti *et al*. (2011). This was a click-and-zoom view showing nested images of a femur from 400mm to 500 microns. Issues of ill-conditioning were evident at the smallest scale. Viceconti *et al*. (2011) included examples of both spatial and temporal multiscale, showing the electrical activity of a muscle at different timescales.

Despite numerous calls for multiscale visualisation in the biomedical field (Brook & Waters, 2007) there has so far been little demand from clinical applications for the kind of interactive multiscale views that are now common in cartography and astrophysics. Auer *et al*. (2007) questioned whether it was worthwhile to collect multiscale data without having the means to visualise it adequately. Why is there no equivalent of Google Earth for biomedical images, although the orders of magnitude involved would be similar? It may be useful to consider for a moment some fundamental differences between cartography and biomedical imaging which may explain this disparity. These are summarised in Table 1.

The first major difference is the availability of datasets. Whilst cartographic datasets are widely available, medical datasets are restricted by issues of ethics and privacy, and their publication takes significant effort in terms of anonymisation, description and quality control. In addition to the effort involved, research groups regard their datasets as hard-won and valuable assets for their own research, and are understandably reluctant to part with them.

Consider also the ease of capturing images. In cartography and astrophysics, the effort and cost per image is very small and different scales can be captured by the same satellite or telescope by changing the magnification. Composites of millions of images can easily be collected, registered and stitched together because the subjects are rigid and relatively static. In contrast, each medical image requires time and expert attention. Multiple images can be exhausting or harmful to a patient, so clinical imaging tends to be kept to a minimum. Registration is very difficult because the subjects are non-rigid. Thus, high resolution whole body datasets such as the Visible Human (Visible Human Project, 2011) are rare and cannot be created from living patients.

The approach to scale in these fields is also different. In cartography, it is natural to think of scale changing continuously, across many orders of magnitude, with little change in semantics. In biomedical imaging, it is more common to think of scales as discrete. Different scales have different semantic meaning - for example, organs, tissues and cells - and they tend to be modelled and imaged separately from each other, with intermediate scales ignored. There might be wide gaps between the scales of interest in a given application.

It should also be noted that there is only one Earth, but there are millions of medical patients, none of whose medical data is of clinical value to anyone else, so a Visible Human is not as universally useful as a Google Earth. The value of complex medical datasets tends to be in research, particularly in the modelling of complex systems such as the genome, the metabolome and the brain and it is in such areas that multiscale visualisation is most required.

	Cartography and Astrophysics	**Biomedicine**
Availability	Widely available.	Restricted by issues of ethics and privacy. Publishing requires anonymisation, metadata and quality control. Research groups reluctant to release valuable data.
Effort per image	Negligible.	Large.
Composite images	Whole Earth or large swathes of the Universe easy to capture and stitch.	Multiple images may be exhausting or harmful to patient. Whole body rare and not possible on living subjects.
Instrumentation	Different scales captured by the same instrument (satellite or telescope)	Different scales captured by different instruments and technologies.
Registration	Rigid	Non-rigid
Number of subjects	Only one Earth and one Universe to map.	Millions of different patients.
Gaps between scales	Many orders of magnitude with no gaps.	Images or models may be far apart in scale, depending on clinical need.
Semantic continuity between scales	Continuous scale. Many orders of magnitude with no semantic change.	Discrete scales. Scales are strongly labelled and have distinct semantics.

Table 1. Contrasting the relative difficulty of producing multiscale datasets in cartography and astrophysics with that in biomedical imaging.

2.4 Information visualisation and genomics

Information visualisation, or InfoVis, is the art and science of visualising all types of data, mainly that which is not spatially located and not visualised by traditional 3D graphics. It is a sister field to what is often called scientific visualisation, though there is some overlap between the two. The variety of data visualised and the number of techniques devised to present it is enormous.

As with 3D spatial data, InfoVis data can be very large and multiresolution. For example, the human genome contains almost 3 billion base pairs - requiring a range of visualisation scales greater than Google Earth. The VISTA multiscale viewer (Nielsen *et al*., 2010) has been mentioned in the previous section.

Multiscale methods have been applied since the mid-nineties to database exploration. The DataSplash system (Olston *et al*., 1998) supported panning, zooming and hyperlinking. Users were able to design their own multiscale interface to their database, presenting queries as charts, adding hyperlinks and specifying the zoom behaviour of targets. A system based on data cubes (Gray *et al*., 1996) was developed in Stolte *et al*. (2003), which generalised the concept of scale to abstract concepts such as time, location and product in order to perform multiresolution searches of databases. Each concept was associated with an ontological list of scale levels, such as {month, quarter, year} for time, {city, state, country} for location and {product, coffee, espresso} as the concept hierarchy for a particular product.

Zomit, a click-and-zoom interface with a rich set of navigation tools, was presented in Pook *et al*. (2000) and applied to both InfoVis, as a library navigation tool, and genomics.

3 Multiscale techniques

This section presents a detailed review of current multiscale techniques.

3.1 Out of core

In this section, we shall focus on describing approaches for visualising datasets larger than the main memory available. Communication between fast internal memory and slower external memory is a major bottleneck in many large-scale applications. This work has had significant impact given that, in recent years, there has been a rapid increase in the raw size of datasets. The field of out-of-core visualisation has a large body of work. Engel *et al. (*2004) present techniques used to render large data on the GPU, while Silva *et al. (*2002) provide a survey of external memory techniques. A review of massive data sets and their visualisation has also been produced by Joy (2009).

3.1.1 Chunking

While multidimensional arrays are usually stored in a file using linear storage, a common way of improving access to them is to reorganise the file using chunked storage. That happens when data is split in chunks (cubes or bricks) of equal size and the same dimensionality as the original volume and each individual chunk is stored contiguously in the file using linear storage. This increases the likelihood that data that are physically close in the n-dimensional volume will be stored at locations that are close together in the file.

Sarawagi & Stonebraker (1994) reorganised the files using chunked storage and maintained several copies of the data that would match each access pattern. While this method is very effective and widely used, it has the disadvantage that the data needs to be reorganised or copies of the data need to be made with different organisation. More & Choudhary (2000) reorganised their data according to the expected query type and the likelihood that data values will be accessed together. The Active Data Repository used chunking to reduce overall access costs and to achieve balanced parallel I/O (Chang *et al*., 2000).

Dynamic chunking views a dataset stored using linear storage as if it were chunked into blocks of configurable size and shape. As soon as an item in any block is accessed, the entire block is read. Dynamic chunking was described in the context of a slicer visualisation application (Lipsa *et al*., 2007) and it was shown that it provides some of the benefits of file chunking without having to reorganise or maintain multiple copies of the file. This work was extended to apply to an arbitrary direction slicer and to ray casting (Lipsa *et al*., 2009). They proposed an optimisation that could take advantage of knowledge about the iteration pattern and significantly improve performance.

Agrawal *et al*. (2010) proposed an efficient out-of-core method for the exploration of large volume data that is suitable for data-flow-oriented systems. It improves the performance of out-of-core exploration of large volume data based on the following two key concepts: firstly, to use knowledge of the storage order of the data on disk and of the access pattern required by the visualisation algorithm to optimise disk access; secondly, to create discrete multiresolution representations of the data – they use a low-resolution version for representing the entire dataset and a higher-resolution version only for visualising smaller regions of interest. This approach allows the user to roam through multi-gigabyte volume datasets in real time, with low memory requirements.

3.1.2 Prefetching

Prefetching has also been an active area of research. Common ways to mediate the effect of slow I/O are to use prefetching when the user is thinking what to do as in Doshi *et al*. (2003) or to use a separate thread to overlap rendering with data I/O, as in Gao *et al*. (2005). Brown *et al. (*2001) describe a hint based method that effectively accelerates paged virtual memory performance using an operating system that takes advantage of compiler generated hints and multiple disks. Rhodes *et al. (*2005) use information about the access pattern provided by an iterator object to calculate a cache block shape that reduces the number of reads from the file. The iteration-aware prefetching speeds up a point or block iteration along one of the principal axes. Chisnall *et al. (*2006) use data inferred from previous accesses to an out-of-core octree, to improve out-of-core rendering of a point dataset using discrete ray tracing.

The interactive out-of-core volume data exploration has requirements relating to importing, accessing, visualising and extracting a part of a very large volume dataset. Problem-specific approaches are not enough; visualisation systems aim to be able to visualise volume datasets of many kinds using a variety of volume rendering techniques (Preim & Bartz, 2007a) and most systems also provide the user with volume smoothing, segmentation operations, colour mapping functions, etc.

3.2 Click-and-zoom

The click-and-zoom interaction is familiar to users of Google Earth (Google Earth, 2011). Sub-scale data which is too small to be resolved on the display screen is marked by a placeholder token, and clicking the mouse on the token invokes a zoom to the target data. In this report we will define a *token* as any glyph, landmark or label which acts as a placeholder for data in the scene at that point. There have not been many attempts to extend the click-and-zoom interaction into 3D. One such example was described in Thurmond *et al*. (2005), where geological outcrops were visualised using VRML, the locations of sub-scale data were indicated by hyperlinked tokens and clicking on a token opened the hyperlinked target data. The ability to place hyperlinks in 3D scenes makes VRML a convenient platform for this type of interaction, though the programming language is not rich enough to create the complex variety of visualisations required in a field such as medical imaging. Hyperlinks are useful for displaying metadata associated with a feature; they offer a quick solution for implementing a click-and-zoom view, but are an unsubtle and disorientating method of moving around a 3D scene.

A 3D click-and-zoom view for medical data was described in McFarlane *et al*. (2008) and Viceconti *et al*. (2011) as part of the LHDL project (Living Human Digital Library, 2011). The data consisted of nested images of bone at different scales: a 400mm whole femur, a 20mm microCT and a 500 micron nanoCT. Objects in the scene were represented in different ways depending on their size on the screen: as tokens in the form of small coloured cubes when too small, or invisible when too large. Threshold screen sizes were defined for the transition between scale representations, with hysteresis to prevent flickering from one form to another. A large number of features were identified as essential to make this type of interaction work well. For example, in 3D the tokens were perceived as being within the scene and therefore had be seen to change their size in response to a zoom, but without growing too large or too small. This contrasts with 2D cartographic displays such as Google Earth, where the tokens are labels and pins on top of the map, but do not change their size with it. Zooms were deliberately performed slowly, beginning with a rotation towards the target, to show the user where they were going. It was important to distinguish objects with current "attention" from those peripheral to the view, in order to set the camera and other features correctly. The mouse interaction was assisted by a minimal set of GUI buttons, consisting of a camera reset (user got lost), a go-back (user clicked the wrong token) and a zoom-out (user got tired of dragging the mouse).

The major problems identified in McFarlane *et al*. (2008) were occlusions and floating-point precision.

- Occlusion is not a problem in an open, uncluttered visualisation such as a slice through a volume, where all the tokens can be clearly seen. However, it is difficult to generalise the interface to different types of data and different visualisations, because the tokens are easily hidden inside larger objects. Methods need to be developed to indicate the presence and location of such hidden sub-scale data. Possible solutions might take the form of "sprites" shining through the occluding surface, protruding flags, or annotating call-outs pointing to the target from around the edge of the viewport.
- Problems of floating-point precision were reported by McFarlane *et al*. (2008) in positioning a 500 micron image in a 400mm scene. This was caused by the magnitude of the floating point values in the micron image being close to a preset precision limit in version 4 of the Visualization Toolkit (VTK) (Kitware, 2003; Schroeder *et al*., 2002a). Ill-conditioning has been previously addressed by McCrae *et al*. (2009), in which a mobile origin was used and Hanson *et al*. (2000), in which log-scale coordinates were used, with all objects being rendered at or near unit scale. In medical datasets, the fixed-origin problem might be avoided by registering the positions of small objects with respect to local parents. Local groups could then be visualised in their own units at unit scale, with out-of-scale and ill-conditioned objects excluded from the rendering. Medical data is often pre-arranged into such a hierarchy and at least one visualisation platform (Viceconti *et al*., 2007) already supports a tree structure in which objects are positioned relative to their parent node.

Another example of click-and-zoom is that of Pook *et al*. (2000). This describes a zoomable user interface, or ZUI, called Zomit, which is mainly designed for information visualisation and genomics. The Zomit interface contains a rich set of navigational aids. The user selects the form of interaction from a pop-up pie menu. A context layer is a transparent overlay which shows the user where they are. A transparent copy of the view can be zoomed in and out, whilst holding the current view static. A history layer is another overlay which shows the user where they have been and allows them to retrace their path. A hierarchy tree is permanently displayed.

3.3 Click-and-fly

Click-and-fly is the interaction used in immersive scenes, in which a target is clicked and the user moves towards it. A typical click-and-fly interaction is seen at the street view scale of Google Earth (Google Earth, 2011). Fundamentally, the interaction is the same as the same as click-and-zoom, in that both involve camera motion initiated by a mouse click but the perception is different. In click-and-zoom the aim of the click is to magnify the target; in click-and-fly, the aim is cause the user to move through the scene. Click-and-fly interactions need not involve a change of scale.

3.4 Lensing

A lens is a magnified region of an image or scene, located in the scene at the point being magnified. Lenses are normally used to inspect image data and are a useful tool for analysing small-scale phenomena within an enlarged visualisation. Lenses provide magnification of detail which is in-place and therefore retains the position and context in the global view.

A major advantage of lenses is that they can be employed to show different information in the lens regions, such as a parameter or scalar other than that currently being displayed in the background image and they are thus very useful in multiparameter visualisations (Preim & Bartz, 2007b). The concept of the movable lens for exploring multiparameter data was introduced by Bier *et al. (*1993). The interaction style was termed “see through interfaces” and the lenses referred to as “magic lenses” to distinguish them from conventional magnifying lenses. The magic lens might also be semi-transparent, acting as a small moveable overlay.

Viega *et al. (*1996) extended the concept of the magic lens to 3D including both flat, planar lenses and volumetric lenses. Viewpoint-dependent distortion of 3D data, see Carpendale *et al*. (1997) for example, can highlight regions of interest by dedicating more space to them.

Relatively little work has been done on lenses in the domain of volume visualisation. LaMar *et al. (*2001) integrated a 3D magnification lens with a hardware-texture based volume renderer. Zooming was accomplished by modifying texture coordinates and the 2D “perspective correct textures” technique was extended to 3D in order to obtain the correct texture coordinates for the lens border. Wei *et al. (*2001) applied fisheye views to magnify particle track volume data using nonlinear magnification functions. Cohen & Brodlie (2004) magnified volume data by generating a new volume using inverse distortion functions; however, this method was slow and memory-intensive. Further research is clearly needed to design better lenses and find efficient implementations for volume data. Wang *et al. (*2005) proposed a focus + context framework that used various standard and advanced magnification lens rendering techniques to magnify the features of interest, while compressing the remaining volume regions without clipping them away completely. Some of these lenses could be interactively configured by the user to specify the desired magnification patterns, while others were feature adaptive. Taerum *et al. (*2006) demonstrated an illustration-inspired system for depicting contextual close-up views of selected regions of interest within volumes.

An interesting combination of lenses and call-outs was reported by Brosz *et al*. (2011), in which the global context and connectivity was provided by magnifying lenses, whilst an undistorted magnification was provided by an accompanying call-out.

3.5 Power Scaled Coordinates

Navigation through a virtual environment and the placement of objects in the environment utilise what are considered to be the standard six degrees of freedom: three orientation parameters, which are independent of scale and three position parameters, which must be adapted to our scaling requirements. In order to support local scales near unity, one additional parameter is introduced in Hanson *et al*. (2000), the power scale, which is effectively an index into an array of local, mipmap-like, unit-scale environments whose scales are related by exponentiation of the indices.

Fu & Hanson (2006) developed large-scale visualisation techniques based on Power Scaled Coordinates (PSC). These methods can transparently integrate visualisation of objects at vast scales that might otherwise be limited by depth buffer and matrix transformation precision. Spatial scaling methods based on PSC permits representation and rapid transformation of positions and vectors at any arbitrary scale.

Power scaled coordinates are composed of a four-tuple, (x, y, z, s) representing the 3D position (x, y, z) $*k^s$, where k is any positive exponent base, usually chosen as 10. Based on this construction, the PSC representation nicely decouples the directional term (x, y, z) and the exponential scale term s from a given 3D coordinate so that we can effectively represent positions of objects at different scales in a uniform fashion. By using power scaled coordinates, we can represent spatial data in scale-independent form $(x, y, z, s) = (\lambda x, \lambda y, \lambda z, s - \log_k \lambda)$.

The idea of PSC-based depth rescaling is as follows: given an object far beyond the far plane, if we shrink all its vertices along their lines of sight towards the eye point, its effective size will decrease and it will become viewable at the camera without explicitly extending the distance of the far plane. Since all vertices are distorted in the same manner along their lines of sight, we will not notice any difference at the camera location.

Li *et al. (*2006) proposed that mapping a user control to the exponential power term of distance would provide smooth adaptive speed control, enabling fluid travel between regions of interest that are very far apart.

3.6 Temporal zoom

Data may be multiscale in time as well as space, in that the data contains features of interest on a range of timescales. The human body, even if we exclude biochemistry, undergoes processes at timescales ranging from milliseconds to decades. The electrical activity of the human body as measured by electromyography (EMG), electrocardiography (ECG) and electroencephalography (EEG) is a rich source of temporal multiscale data. Viceconti *et al*. (2011) presented a temporal multiscale view of EMG data, showing features at timescales of 2s, 200ms and 20ms. The graph of electrical data was also coupled with an animated spatial view of the electrical activity in the muscle. The temporal zoom expands the time-axis of the graph to show the activity on that scale, while the corresponding spatial animation is slowed down in order to be visible to the user.

Because time is one-dimensional, the basic operation of selecting the time of interest and the timescale is relatively simple compared with the spatial case. However, unlike spatial data, the sampling frequency of time data can be very high, with the actual scales of interest somewhere in between the highest and lowest frequencies. Techniques such as the wavelet analysis of Woodring &

Shen (2009) have been developed to assist the user in locating the times and timescales of interest in the data. Another important difference is that time data is frequently periodic, in which case an option to select and track small periodic features over longer timescales would be a useful tool. This is used to time the capture of cardiac images and signals and is called *gating*. Further examples of time series visualisation are given in a later section.

3.7 GUI navigation

Jones *et al. (*2009) discussed three kinds of 3D geospatial GUI; these are also relevant to 3D biomedical visualisation:

- *Standard GUI*. The 3D model is rendered in a normal GUI window and the 3D visualisation exists entirely within a user interface based on windows, icons, menu bar, etc. This type of GUI provides the user with a porthole through which to view the virtual 3D world and hence is generally considered as semi immersive. Central to desktop-based GUIs is the depiction of the file system as a hierarchy of iconic folders and files. To work with a given dataset, the user must navigate to the correct file by traversing the file-system hierarchy.
- *Immersive 3D visualisation*. The main visual cue for the user to locate a dataset is the position of the data in 3D space, rather than its location in the file system. The model does not exist within a user interface, the model is the interface. To locate a given dataset, one navigates in 3D space to the appropriate position in the virtual model. The model also acts as the interface through which the user can access additional data that are not inherently spatial (e.g. metadata, photos, literature references, etc.), but which are tied to a given locality or object.
- *Hybrid GUI*. This combines a high degree of interactivity via the 3D model with interaction through standard desktop-based GUI widgets. In this way, there is a positive redundancy so that the user always has alternative methods to access data in the model.

Besides projecting the 3D data to the 2D screen, volumetric displays are a relatively new platform for displaying 3D imagery (Favalora, 2005; Grossman & Balakrishnan, 2006). Volumetric displays are unique in that they provide imagery in true 3D space; as a result, they can improve depth perception and shape recognition.

3.8 Call-outs

In visualisation and illustration, a call-out is an annotation that is associated with a point in an image and connected to the point by means of a pointer. This could be a text label connected to an image feature by a line or arrow, a flag, or even a speech balloon. In multiscale illustration, a magnifying call-out is an enlarged sub-region, expanded out of the parent image and magnified into a new image. The technique is most often seen in static illustrations, such as the poster shown in Figure 2. However, call-outs have also been used interactively; Tory & Swindells (2003) developed a system called ExoVis which used magnifying call-outs for interactive 3D visualisation. ExoVis allows a subvolume of the image to be pulled out of the image and magnified into a new image, with pointers connecting it to the corresponding region in the original. This is an excellent way of viewing sub-scale detail and global position at the same time, since the whole path through the scales is visible and located at each step. The ability to view two or more scales simultaneously is potentially powerful, especially if the data is time-varying. The disadvantage of this technique would appear to be that it is extravagant with screen space, since each change in scale requires a new image, but this is not an insurmountable problem, since two images could easily be displayed on a split screen without much loss of size and more could be accommodated given some management of the screen space.

An unusual and visually striking form of magnification was presented by Hsu *et al*. (2011), in which the multiscale detail was pulled seamlessly out of the target image with increasing magnification, in the manner of a continuous call-out, or a stretched-out lens. This was achieved using a non-linear ray-tracing technique. The results were remarkable in terms of illustrative rendering, occasionally resembling the non-linear projections of the artist M.C. Escher (Ernst & Brigham, 1986). However, in terms of scientific visualisation, there were some disadvantages in comparison to a standard sequence of discrete call-outs: it was not possible to display the same detail simultaneously on different scales; it did not show how much magnification had taken place; and it could cause confusion regarding the relative size of features. The non-linear rendering was also computationally very expensive, and required some user intervention to set up, so the technique in this form probably cannot be considered for real-time rendering.

3.9 Level of detail

As datasets used in interactive visualisation applications grow in size and complexity, techniques for adapting the detail of objects in the scene to ensure interactive frame rates gain increasing importance. Level of detail, or LoD for short, is a general approach to managing the trade-off between quality/complexity and performance. The fundamental concept behind LoD is simple: when rendering an object in the scene, use a less detailed representation for distant, small or unimportant objects. The basic idea behind representing the same model at multiple levels of detail is to use a hierarchical structure to represent multiple versions of the model, with the detail of the model increasing with depth in the hierarchy.

Building a multiresolution data hierarchy from a large amount of data allows us to visualise data at different scales and balance image quality and computation speed. The most common scheme in hardware-based volume rendering is a hierarchical scheme (octree) with recursive subdivision and increasing resolution. It was first introduced in volume rendering by LaMar *et al. (*2000) and extended by Weiler *et al. (*2000) to minimise discontinuities between blocks of different level of detail. During the rendering process, the resolutions of different parts in volume would be adapted to the corresponding display parameters (Guthe *et al*., 2002). Flat blocking is another commonly used representation; this originated from adaptive texture mapping technique (Kraus & Ertl, 2002) and uses bounding boxes with the same size to block and organise the original volume. Zimmermann *et al. (*2000) proposed an adaptive approach to volume rendering via 3D textures at arbitrary levels of detail. A texture map hierarchy is constructed in a way that minimises the amount of texture memory and guarantees consistent interpolation between different resolution levels. In Cheng *et al*. (2011), based on the flat blocking schema, a novel LoD-selection algorithm based on error analysis is proposed and the flat blocking multi-resolution representation can be obtained.

For mesh data, based on current simplification methods, Zhou (2005) classified LoD algorithms into vertex clustering, vertex decimation, quadric error metrics, reverse simplification, image-driven simplification and skip-strips.

Quite often, the selection of data resolution are automatically determined by user-specified error tolerances and viewing parameters. Many quality metrics, such as mean square error (MSE) and signal-to-noise ratio (SNR), are data-centric. Although these metrics have specific meanings and are simple to compute, they are not always effective in predicting the quality of rendered images due to the lack of correlation between data and image quality (Wang *et al*., 2004). In fact, finding the best LoD is a NP-complete optimisation problem, so most algorithms take a greedy approximation strategy instead.

3.9.1 Bundling

Long thin objects in the form of collections of fibres are a special case in multiscale visualisation. They occur in medical images of brains and muscles, in vector field visualisations and in InfoVis applications such as diagrams of interconnected items and densely connected edge graphs. The problem of visualising fibrous connections at multiresolution is so important that it has its own LoD technique, called bundling.

Edge bundles render large graphs via edge clustering, by collecting together long edges analogous to the way electric wires are merged into bundles along a shared mutual path segment, fanning out at ends to connect distinct endpoints. Hierarchical edge bundling (Holten, 2006) was introduced as a means to view a compound graph while reducing visual clutter. Edges are modelled as B-splines, with those following a similar path to one another within the hierarchy being grouped together for relevant subsections of their paths. Those bundles containing a greater number of edges are rendered as being brighter. Cui *et al. (*2008) visualised large graphs via edge clustering through a geographical control structure for general graphs. However, their method relies on the generation of a control mesh to guide the bundling, which frequently results in bundles that display considerable curvature-variation. Balzer & Deussen (2007) used edge bundles to simplify edges in a clustered level-of-detail graph visualisation that filtered the layout of the original graph. Flow map layouts (Phan *et al*., 2005) route edges through a binary cluster hierarchy in a flow graph and yield bundles that emphasise source-sink routes, though the biggest drawback is that all edge splits are binary. In Holten & van Wijk (2009), a force-directed technique is proposed that uses a self-organising approach to bundling, in which edges are modelled as flexible springs that can attract each other. Here, no hierarchy is used, and no control mesh. Telea & Ersoy (2010) presented an image-based simplified visualisation for edge bundles using shaded overlapping compact shapes. They emphasise coarse-scale structure and make bundle overlap explicit. Level-of-detail techniques help to select the visualisation granularity and further explore overlaps.

In medical applications, visualisation of the internal white matter structure *in vivo* has become feasible thanks to the development of Diffusion Tensor Imaging (DTI). One of the most popular techniques is to reconstruct the individual fibres from the tensor information, e.g. by tracing streamlines. However, white matter is a complex structure and the image gets easily cluttered. To reduce visual clutter and facilitate data understanding, a suitable solution is to selectively display streamlines that highlight important flow features. Clustering neighbouring fibres traced from DTI data (Moberts *et al*., 2005) allows clear observation of the fibre structure and patterns. The spatial proximity between two fibres indicates their similarity and thus can be used in fibre clustering (Zhang *et al*., 2008). Different distance measures have been proposed, including the mean of closest point distances (Corouge *et al*., 2004), the thresholded average distance (Chen *et al*., 2008), the weighted normalised sum of minimum distance (Jianu *et al*., 2009) and the distance measured in a transformed feature space. Yu *et al. (*2011) present a multiscale fashion to characterise and visualise the flow structure and patterns. Exploring the hierarchy allows a complete visualisation of important flow features. This technique is scalable and is promising for viewing large and complex flow fields due to its selective streamline display and flexible LoD refinement.

Note that edge bundling determines both edge grouping and the paths of the edges. Streamline bundles, like fibre clustering, only determine grouping. Meanwhile, streamline bundles are organised hierarchically and can capture flow features of varying scales at different levels of detail.

4 Types of data and visualisation styles

In this section, we look at basic algorithms for visualisation. For 3D spatial data, visualisation style can be loosely characterised by the dimensionality of the primitives which represent the data: points or glyphs, lines, surfaces or planes, and volumes.

4.1 Volume visualisation

A volume, or 3D image, is a grid of points or voxels in which each point is associated with one or more scalar values. These are the grey levels or colours of the image. Image grids are typically regular, but can also be rectilinear with irregular spacing, or even curvilinear. Rectilinear grids can be visualised by resampling onto a regular grid, or by modifying standard techniques to allow for the irregular spacing.

Volumes are typically visualised in one of three ways: by slice planes, isosurfaces or direct volume rendering.

4.1.1 Volume visualisation by slice planes

A widely used technique in medical routine is the cine mode, in which the individual slices of a volumetric dataset are examined consecutively. Slices are popular in medical imaging because they show detail without occlusion, and are convenient to work with in terms of accurately landmarking and measuring features. The user might choose to view a single plane, or use a multi-planar reconstruction (MPR) in which the data is presented on two or three intersecting and orthogonal slices. The additional planes can be helpful in imparting a sense of 3D structure to the view. Slices are typically aligned with the coordinate axes, but oblique slices, arbitrarily oriented within the dataset, are also possible. Oblique slices are more computationally expensive because they require interpolation from the image grid to calculate the image on the plane.

A refinement of plane-based representation is curved planar reformation (CPR). This technique addresses the problem that some medical structures are not well-represented by a single axis or plane. For example, the orientation of blood vessels changes rapidly. CPR computes a curve along a selected blood vessel and a respective curved slice, using the vessel curve as profile and the largest vessel diameter as spanning vector. For longitudinal slices, the slice itself can be curved to match the shape of the vessel (Kanitsar *et al*., 2002; 2003).

A major advantage of the slice view in multiscale visualisation is its openness; there are no occlusions (except the trivial case of being on the far side of the slice) to hide sub-scale data or placeholder tokens (Viceconti *et al*., 2011).

4.1.2 Volume visualisation by isosurfaces

Structures of interest in volumetric data are typically differentiated from the surrounding image data by a boundary or a material interface. The surface-based variation of indirect volume rendering aims at a visual representation of that boundary, which needs to be specified. An isosurface is a surface representing a contour of constant value in the image.

The specification of a contour value is equivalent to a binary transfer function and, hence, is equivalent to threshold-based segmentation. Surface-based volume rendering can produce visually impressive results, but its success depends on achieving an accurate segmentation of the image, and may require segmentation methods that are more advanced, or that need some manual intervention, than simply calculating the basic isosurface.

- 2D Contour tracing can also be used to extract a 3D contour or isosurface from a volume dataset by applying a standard 2D contour tracing algorithm to each image slice of the volume dataset individually. The major issue here is the subsequent correspondence problem between the 2D contours, particularly when there are changes in topology.
- Polygonal isosurface extraction is the direct extraction of the contour surface, with Marching Cubes (Lorensen & Cline, 1987) being the most widely used method for extracting isosurfaces. Marching Cubes can suffer from surface artefacts caused by triangulation ambiguities; methods for reducing or avoiding these include asymptotic deciders (Natarajan, 1994), various case table refinements (Cignoni *et al*., 2000) and marching tetrahedra (Shirley & Tuchman, 1990). Marching Cubes is basically a thresholding algorithm, and as such might not always produce meaningful surfaces if the materials or tissues concerned are not well-separated by a consistent contour value.

A disadvantage of isosurfaces, compared to slice planes, is that the outer surfaces can conceal any structures which might be inside, or conceal the locations of sub-scale data, if the view is multiscale. Visualising internal surfaces requires transparency, or methods to remove parts of the occluding outer surfaces to allow a view of the interior.

4.1.3 Volume visualisation by direct volume rendering

Direct volume rendering (DVR), produces an image directly from the data without an intermediate geometrical representation. The whole volume is shown as a semi-transparent cuboid, with the internal features viewed through the outer layers. The challenge of this type of visualisation is to see clearly through the outer voxels to the desired detail inside. Success depends largely on finding a good transfer function – a mapping between the voxel values (typically scalar value and image gradient) and the colour and transparency that they contribute to the rendering.

DVR algorithms fall into two groups; image-space (or backward-mapping) approaches such as ray-casting (Levoy, 1988; Westermann & Sevenich, 2001) or shear warp (Lacroute & Levoy, 1994; Schulze *et al*., 2003) and object-space (or forward-mapping) approaches such as splatting (Westover, 1990; Huang *et al*., 2000) or texture-mapping (Westermann & Ertl, 1998; Rezk-Salama *et al*., 2000).

DVR methods are especially suitable for amorphous datasets, such as fluids, smoke and gases and provide a way to see through the data, revealing complex 3D relationships.

4.2 Surface data

Because the appearance of an object depends largely on the exterior of the object, surface data is very common in computer graphics. Surface data is most often represented as a polygonal mesh, usually composed of triangles or quads. Graphics processors (GPU's) use polygons as primitives, so rendering is simple and efficient.

There are many different representations for special applications. Streaming meshes (Isenburg & Lindstrom, 2005) store faces in an ordered, yet independent, way so that the mesh can be transmitted in pieces; progressive meshes (Hoppe, 1996; Garland & Heckbert, 1997) transmit the vertex and face data with increasing levels of detail; while normal meshes (Guskov *et al.*, 2000) transmit progressive changes to a mesh as a set of normal displacements from a base mesh.

Parametric surface patches can be used to create surfaces that are much smoother than those based on polygons. The surface is normally formed by a tensor product of parametric curves; curve types include B-splines, β-splines and NURBS. Associated with the patches are weighted control points, and the shape of the surface can be varied by adjusting their positions and weights. Surfaces can be represented by many fewer patches than there would be polygons in an equivalent model. However, they introduce substantial overheads into the rendering calculations, so in most cases, polygonal models are favoured.

With the advent of 3D digitisers such as laser scanners, complex meshes are increasingly available. A simple but important idea in efficient rendering is the observation that a long triangle strip can be rendered three times more efficiently than in a straightforward triangle-by-triangle-based rendering. This idea has been adopted in graphics APIs such as OpenGL and also in graphics processing hardware. However, the problem of finding an optimal stripification for a given mesh is an NP-complete problem (Estkowski *et al.*, 2002); several heuristics-based fast algorithms have been constructed and are reported to work well in practice (Hoppe, 1999, Diaz-Gutierrez *et al.*, 2005).

Multiscale data does not often come in the form of surfaces. However, high-resolution meshes which exceed the size of the GPU memory are problematic to render and would usually be decimated to the appropriate level of detail. An interesting example of a multiscale surface is the work of Max *et al*. (2009), in which a fractal attractor is represented by a polygonal surface. The mesh is rendered at different levels of detail by subdivision.

4.3 Point data

Point data is most often encountered in simulations of clouds of interacting point-like objects. Its visualisation is frequently a multiscale problem because the number of points can be very large, and it is often desirable to see small details as well as the global shape of the cloud.

Point data is often rendered using the technique of splatting (Schroeder *et al.*, 2002b), in which the points are represented by extended point distribution functions (PDF's). The PDF's are aggregated into a density function which can then be rendered as a volume image or an isosurface. The disadvantage of this, in terms of multiscale visualisation, is that the rendering requires the point function values to be interpolated onto an image grid, resulting in loss of detail. An alternative approach, which avoids the need for an image grid, is to represent the points as glyphs, usually spheres or ellipsoids (Gribble *et al.*, 2008).

The multiscale problem has been addressed by Hopf *et al*. (2004) in an astrophysics simulation of the positions of hundreds of millions of stars. The point cloud density function was decomposed into a hierarchical LoD structure, using a splitting algorithm based on principal component analysis, allowing levels of detail to be loaded and rendered as required. Some out-of-core memory was used, though only to store pending timeslices. Achieving a stable structure from one time-slice to the next, without sudden visible changes, was an outstanding issue.

4.4 Vector field visualisation

Vector fields are commonly used to represent directional physical properties such as flow velocities or magnetic fields. Visualisation of vector fields seeks to convey a qualitative understanding of patterns of flow to the user, and is important for in diverse areas such as astronomy, aeronautics, meteorology and medicine. The need to convey both magnitude and direction makes vector fields one of the more interesting and varied areas of scientific visualisation; for example, methods for visualising direction include glyphs (arrows), streamlines, animated particles and Doppler colouring. The importance of vector field visualisation has motivated a large amount of research, and current methods can be grouped into four general classes (Post *et al*., 2003).

4.4.1 Glyph-based visualisation

Glyph-based visualisation (direct, hedgehog or icon-based visualisation) directly maps the vectors to oriented glyphs, such as arrows, cones or lines. Usually, arrows are tied to grid points and indicate both the direction and the magnitude of the vector field at these points. In 3D applications, because of occlusion and complexity, glyphs showing all vector information at the same time are hard to interpret visually. In many methods, glyphs are based on selective seeding and reduce the amount of data being displayed. The choice of seeds is usually based on simplification, clustering or some extracted features of the vector field (Telea & van Wijk, 1999).

4.4.2 Texture-based visualisation

The common texture-based techniques, such as Line Integral Convolution (LIC) (Cabral & Leedom, 1993) were initially proposed for 2D applications. By taking the vector field data and a texture (scalar field) as input, these methods then produce another texture to use for rendering, which provides a dense visualisation of the vector field. Basically, this type of approach maps the vector field onto a scalar field, which is then rendered.

For 3D LIC approaches, the texture-generation and rendering stages are both expensive. For texture generation, some proposed improvements, such as the Fast LIC algorithm proposed by Stalling & Hege (1995), separate the computation of streamlines and convolution algorithmically to reduce computational costs by an order of magnitude compared to the original algorithm. Others, such as Seed LIC proposed by Helgeland & Andreassen (2004), which takes advantage of the sparseness of the input texture in 3D LIC and achieves good computational time, are especially suitable for 3D. For the rendering stage, considerable acceleration has been produced by using the texture-mapping features in modern graphics hardware and interactive frame rates can now be achieved.

4.4.3 Geometry-based visualisation

Geometry-based visualisations are based on integrating the vector field and use geometric objects in the resulting visualisation, such as streamlines, stream tubes, stream ribbons and stream surfaces. Among these, the streamline is the simplest and most commonly used. A streamline is tangential to the vector field at every point at a given instant and is produced from a local integration process, initiated at a set of positions called seed points. Streamlines give a more quantitative representation of vector fields than do direct and texture-based visualisations. As longer integral curves, streamlines have more spatial continuity and are more suitable for describing global information relating to the behaviour of the flow than are direct and texture-based visualisations.

There are three commonly used streamline seeding strategies: density-based approaches (Mao *et al*., 1998; Mattausch *et al*., 2003), topology-based or feature-based approaches (Verma *et al*., 2000; Ye *et al*., 2005; Carmo *et al*., 2004) and interactive seeding approaches (Laramee, 2002; Laramee *et al*., 2005). The addition of animated particles to streamlines is very effective at conveying the velocity of flow, though of course such visualisation cannot be transferred to printed media.

4.4.4 Feature-based visualisation

In this class of visualisation, discussed by Post *et al. (*1993), salient features of the vector flow, such as vortices and critical points, are extracted and displayed.

4.5 Tensor field visualisation

Visualisation of tensor fields is a very important topic. The challenges are similar to vector visualisation, but with the increased difficulty of conveying three (or more) principal magnitudes and directions per point instead of one. Tensor magnitudes appear frequently in many applications. In medical imaging, the most common applications are the representation of diffusion in tissues, as obtained from the Diffusion Tensor Imaging modality (Pierpaoli *et al*., 1996) and also the stress and strain in tissues obtained from elastography modalities (Ophir *et al*., 2002).

As in the vector field case, there are several styles of visualisation: direct visualisation using glyphs; indirect visualization using scalar magnitudes derived from the tensor field; or implicit visualization using streamlines. Given the inherent complexity of this kind of data, tensor fields are sometimes reduced to vector fields, taking the largest eigenvalue of the tensor at each point of the domain, and then applying a vector visualisation, but this can only display part of the tensor, usually one principal component, at a time.

The most interesting ways of representing this high-dimensional data in order to display the maximum information available from the tensor field are the use of glyphs and colours. Several approaches have been proposed, including ellipsoids, super quadrics (Kindlmann, 2004), Box glyphs (Johnson *et al*., 2001), Reynolds glyphs (Moore *et al*., 1994), Hue-balls and Lit-tensors (Kindlmann & Weinstein, 1999).

4.6 Finite element data

Finite element modelling (FEM) is a long-established method for modelling the physics of materials and fluid flows. In FEM, the differential equations describing the properties of a material are approximated by discrete cells. Wherever models need to be accurately founded on physical principles, it is the method of choice. In the biomedical field, FEM is used to model the complex geometries, anisotropic material properties and boundary conditions associated with living tissues. FEM has been shown to give an accurate model when simulating deformations of living tissues (Mollemans *et al*., 2007). However, the approach is sometimes not applicable to real-time problems due to its high computational cost and large memory.

FEM data comes in the form of a 3D mesh, usually with tetrahedral or hexahedral cells. FEM meshes are often visualised by showing the material as a wireframe and adding cross-sectional slices to show the scalar values. Generating well-formed FEM meshes has long been regarded as difficult, requiring tedious manual intervention, and this has been a factor in discouraging its routine use amongst clinicians in creating patient-specific FEM models. More effort has been given to automatic mesh generation in recent years (Luboz *et al*., 2005; Sigal *et al*., 2008) and the vast majority of automatic mesh generators for living tissues now produce tetrahedral meshes (Molino *et al*., 2003; Si, 2006). Fernandez *et al*. (2004) described methods for the initial creation of generic models of various anatomical systems, and their subsequent registration with patient-specific data.

There is current research interest in multiscale finite element modelling and visualisation (Arnold *et al*., 2010). Podshivalov *et al*. (2009) presented an unusual visualisation of the multiscale structure of bone in the form of a continuous strip image, with scale as the long axis.

4.7 AMR data visualisation

AMR (Adaptive Mesh Refinement) is a technique commonly used in large-scale simulations involving partial differential equations. An introduction to it can be found in Berger & Oliger (1984). The concept is simple: the image grid is subdivided into higher-resolution grids where more detail is required, so the output is an image containing a hierarchy of nested sub-grids. This concentrates both the accuracy of the simulation and the density of voxels in the places where they are needed.

AMR data can be rendered conventionally if the hierarchical grid is first resampled onto a regular grid, but clearly this must either lose some detail or result in a very large image indeed, so it is preferable to use a dedicated rendering algorithm such that as that reported by Kahler *et al*. (2002). This described the rendering of a stellar evolution simulation which required 27 levels of resolution. The film, made for Discovery Channel, was directed with the aid of an immersive CAVE environment and automatic detection of features of interest.

4.8 Hybrid volume rendering

Combining surface and volume data into one visualisation is difficult. One approach is to convert one representation into the other and fuse the two. However, converting the volume data to surface data might not give the desired visualisation result in terms of revealing the internal information, whilst converting the surface into voxels might not give a good result in terms of resolution or surface illumination, especially if there is a mismatch between the resolutions of the surface and volume data.

An alternative approach is to render both representations separately and then combine the individual rendered results. For this, there are different strategies regarding the stage of the pipeline at which the results should be combined. In ray merging (Levoy, 1990), parallel rays are traced from the observer into both the polygonal and the volumetric data, which are treated separately. Colours and opacities for points on the surfaces and in the volume are determined and are collected in two vectors of samples. The final image value is formed in a compositing step that considers these two vectors.

4.9 Time-varying data visualisation

In the medical sciences, there is interest in long-term studies, measuring effects over time periods ranging from several minutes to several months or years. In addition to 3-dimensional spatial data, clinicians may use temporal data to diagnose pathological tissue based on their altered perfusion dynamics. They may track disease progress or its response to therapy over a short time or over several months or years. Further, they may examine correlations between diseases and drugs and organ function.

4D time-varying volume visualisation is not a trivial extension of 3D volume visualisation. There are two major challenges (Balabanian *et al.*, 2008): the first relates to the computational complexity and requirements, and the second concerns how to effectively represent a temporal dataset visually to enable a particular exploration task. For the first challenge, possible solutions stress the issue of compression and time coherence (by means of advanced compression/decompression techniques (Nagayasu *et al.*, 2008; Wang *et al.*, 2010) and make use of the dedicated memory of high performance processing units (Ma & Lum, 2004).

Ma and Lum (2004) give a survey of two general classes of time-varying visualisation: homogeneous, in which all dimensions are treated equivalently and inhomogeneous, in which the time dimension is considered separately from the spatial dimensions.

Time-varying data is usually explored by animation or by arrays of static images. Neither is particularly effective for classifying data by different temporal activities. Important temporal trends can be missed due to the lack of ability to find them with current visualisation methods. Change blindness can occur when visually perceptual phenomena change too slowly to be detected. So such visualisations can give a notion of structured movement; however they will be less useful for most precise analytic and quantitative tasks. In the case in which the time series consists of a small number of time-steps, it is possible to use fanning in time, which shows all time-steps next to each other (Grimm *et al.*, 2004). However, such an approach does not specifically address the visual emphasis of temporal characteristics in a time-varying dataset.

For rendering a single time step, most methods do not have a concept of mapping based on time activities or intercorrelated time activities. Only recently has scientific visualisation attempted to tackle classification based on temporal activity. For example, in medical visualisation time-varying data has been used to plot the temporal development of contrasted blood perfusion as a one dimensional polyline (Fang *et al.*, 2007). Other techniques let the user link multivariate time-varying data with temporal histograms to simplify the exploration and design of traditional transfer functions for this type of data; in Akiba *et al*. (2006) and Akiba & Ma (2007), a time histogram was given by concatenating a series of conventional 1D histograms for each step. In this system, the time histogram was used to create time-varying transfer functions based upon the time profile of a dataset. Users were able to specify transfer functions either directly or indirectly using the time histogram with interactive feedback. In the past, techniques have been proposed on the automatic generation of transfer functions by incorporating statistical analysis and coherence (Kelly & Ma, 2001; Younesy *et al.*, 2005). All of the above techniques that use automatic or interactive data analysis for exploration and transfer function design employ single-time-step renderers. This means that the individual images contain no temporal information.

By considering a time-varying dataset as a 3D array in which each voxel contains a time-activity curve (TAC), Fang *et al.* (2007) developed a system to classify and segment medical data based on a distance metric from the TAC vector. They designed and appraised three different TAC similarity measures and allowed the user to specify a transfer function on the 1D and 2D histograms and on the scatter plot, respectively.

Building on these concepts for facilitating time-varying data exploration, Woodring and Shen (2009) explore and classify the data based on multiscale temporal activities. Using the wavelet transform along the time axis, they transform data points into multiscale time series curve sets. The time curves are clustered so that data of similar activity are grouped together at different temporal resolutions. These groups of similar trends are then shown to the user, who can browse the trends present in their data, select and interact with the data and eventually visualise the explored phenomena.

There are also visualisation approaches that attempt to visually represent the temporal characteristics of the time-series directly in the physical space. Some approaches have been inspired by illustration, cartoon or comic drawing techniques where helper graphics, e.g. arrows and lines, indicate temporal developments such as movement or size changes (Joshi & Rheingans, 2005). The final image consists of several individual time-steps and the helper graphics convey the temporal information. Another approach, with the idea of chronophotography (Woodring & Shen, 2003a) integrates a time-varying 3D dataset into a single 3D volume, called a chronovolume, using different integration functions. Chronovolumes have been generalised (Woodring *et al.*, 2003b) to create 3D volumes of the time-varying data by slicing the 4D space spanned by the data with hyperplanes and integration operators.

Such methods and systems are useful for exploring data in a time-centric manner, rather than focusing on space and value. Also, in contrast to conventional time-varying visualisation, in which each time step is visualised separately, considering all time steps together and offering a non-animated visualisation of the time-behaviour can provide comprehensive information about the dynamic profiles of different parts of the data.

5 Style of Interaction

Virtual 3D environments can be immersive, such as Virtual Reality, or desktop based, such as Google Earth. Interaction is typically controlled through a 6 degree-of-freedom device, such as a bat, or a 2D device, such as a mouse. Egocentric navigation describes the interaction in which a user wants to move through a space, while exocentric navigation pertains to a user moving around an object.

Designing an interface for a 3D environment is a non-trivial task, complicated by the difficulties inherent in navigating and exploring a 3D scene, as well as the confusing spatial relationships that sometimes exist between scene objects. It needs to effectively span a huge problem domain of all forms of 3D interactions, touching on many fundamental difficulties: being inside an object *vs.* being outside, how close is the viewpoint to the object, what is the user looking at and/or is interested in, egocentric *vs.* exocentric thinking, parallel *vs.* perspective viewing projections, multiscale and level-of-detail issues, what kind of data is being examined (abstract, incomplete, engineering, CAD, entertainment, medical, simulation, etc.) and what the user's task is (authoring, inspecting, etc.). Further technical issues include the handling of clipping planes and floating-point precision problems.

Many points of user confusion when navigating in 3D are related to the multiscale nature of a 3D virtual environment. An effect described as "desert fog" occurs when a user is either too close to a low-detail object or too far from a high-detail object, causing confusion and disorientation resulting from a loss of context. In addition to the importance of detail to provide context, speed of motion has also been identified as being important to effectively navigate 3D environments. It can be helpful to vary the velocity of a user flying through an environment, depending on how close they are to elements in the scene.

More research is needed to improve the understanding of the coupling of space and scale in multiscale user interfaces. Dynamically changeable interaction scales in virtual environments provide users with opportunities to travel quickly and precisely by manipulating the scale factor of the virtual space or the relative size of the space. Zhang (2005) demonstrated the use of animation to enhance user understanding between scales. The animation was generated by interpolating the view positions, view orientations and view scales of two views. Such a dynamic view will effectively narrow the view difference. Zhang (2009) evaluated two travel techniques: scaling-then-travelling, a technique that allows users to separate scaling from moving; and scaling-as-travelling, a technique that enables them to combine scaling with moving. This stressed the importance of travel speed and explored different techniques that offer the user control over this scale-dependent parameter. Scaling-as-travelling provides users with a navigation method that does not exist in the real world; it may have the potential to greatly improve user performance in a virtual world, but there are some challenges in understanding the scene transitions in scaling and in controlling the scaling centre.

McCrae *et al. (*2009) proposed a solution that senses the size of the environment and adjusts the viewing and travel parameters in a scale-sensitive way using an image-based environment representation called a cubemap (see Section 2.1). This approach also solves other issues relevant to multiscale navigation such as the management of viewing frustum parameters and can be used for collision avoidance. It allows consistent navigation at various scales and real-time collision detection without pre-computation or prior knowledge of the geometric structures.

Exploring large structures requires users to obtain both detailed content and sufficient context information. The conflict between content and context is indeed a multiscale problem, because detailed content information and global context information are usually distributed at different scale levels. Zhang & Furnas (2005) introduced the concept of the multiscale collaborative virtual environment (mCVE) described the benefits of multiuser collaboration for navigating content and context at different scale levels, and discussed the design and implementation of multiscale tools to support the visualisation of structures, cross-scale information sharing and cross-scale action.

When sets of geometry are viewed outside a complete context, it can become difficult to determine the spatial relationships between objects. Glueck *et al. (*2009) described how a multiscale reference grid, augmented with position pegs, provides additional depth cues to users, informing them of whether an object lies above or below the grid, the distance of each object from the grid, the depth of the objects along the grid, the relative size of the objects and the approximate scale of the objects being viewed.

Interactions are limited because current 3D graphics systems and tools were primarily designed only for a single scale. When considering a new thread of research pertaining to general interaction with multiscale 3D date, many new problems will occur. User understanding of position and orientation within such an environment is examined in McCrae *et al*. (2010), which presents a design space that considers egocentric and exocentric user goals, landmark formation rules and indicators and controllers for environment visualisation. The proposed mirror ball wedge is very appealing as an orientation controller. It is an exocentric visualisation that expresses the spatial distribution of objects by showing Voronoi region boundaries on the surface of a sphere.

5.1 Fly-through interaction

The two common modes in which a user interacts with a graphical display of a spatial scene are scene-in-hand and fly-through (McCrae *et al*., 2009; Ware & Osborne, 1990). A third mode, eyeball-in-hand, is not often encountered. Scene-in-hand is the type of interaction most commonly used to interact with graphical objects on a desktop display: the user can rotate and translate the scene and zoom the display using the mouse. The fly-through interaction is that used in the street level visualisation of Google Earth (2011): the metaphor is that of the user walking or flying through the scene. The user might move about between pre-defined targets using a click-and-fly interaction, or might have unconstrained movement in the form of flight controls. This kind of interaction is well-suited for large, complex scenes, such as cities, where the scene-in-hand metaphor cannot be sustained.

The user interaction in a 3D virtual environment fly-through has been described as the integration of two activities: travel (the task of moving from one location to another) and wayfinding (the task of acquiring and using spatial knowledge) (Bowman *et al*., 2004). A major challenge of 3D travel is the mapping of control values from the input device (often only 2 degrees of freedom) to the parameters of a virtual camera model (7 or more, including 3 for position, 3 for orientation and 1 for field of view). The issue of speed control is particularly important for environments that are very large or sparse and for environments with objects that exist at diverse spatial scales. Li et al. (2006) introduced an ultra-fast “warp speed” for traversing the empty distances between star systems. Wayfinding involves cognitive activity defining landmarks and paths through the environment, which can be thought of as a mental representation of environment knowledge. Applicable wayfinding techniques for virtual environments include: providing integrated 2D maps and markers; incorporating landmarks and cues into the environments; and providing a consistent and hierarchical structure for the environments.

Multiscale virtual environments (MSVEs), have several levels of scales, smaller scales are nested within large scales. To interact with MSVEs, several issues must be addressed (Kopper *et al*., 2006):

- the size of the user at any location must be compatible with the scale at that location;
- how to tell the user which objects of the environment are levels of scale (LoS) objects;
- how to make it easy for the user to travel between different LoS while maintaining spatial orientation and understanding.

The World in Miniature (WIM) technique (Li *et al*., 2006; Stoakley *et al*., 1995) has been widely used for navigation and manipulation in VEs at different scales. It consists of a hand-held miniature of the world in which the user can select and manipulate the objects. If the user changes his/her position in the miniature, he/she will be sent to the selected position.

Zhang &Furnas (2002) explored the use of 3D multiscale environments in collaborative VEs, in which users could change their own scale (e.g. giant or ant scale) and were able to manipulate the environment as well as interact with other users at the same scale. They also presented scale-based semantic representations in which the representation of an object changed according to how close it was to the user, for example, showing its inner composition.

Two techniques for travelling through large-scale virtual environments developed by Pierce & Pausch (2004) could be applied to multiscale navigation. The Visible Landmarks technique creates points of reference that become visible by having a scale factor applied to them from any point of the environment and serve as a reference to travel. The Place Representations technique divides the VE into hierarchical semantic units, and instead of showing the distant visible landmarks, gives the user a representation of what its semantic unit contains. The combination of these two techniques allows users to travel large distances with a small number of commands.

When the user explores the immersive environment, the most common task is to fly through regions of interest and there exist many different aided navigation models for travel and wayfinding tasks.

- *Target based multiscale navigation* (Li *et al*., 2006). The user is automatically moved from the current location to the centre of the selected LoS (level of scale) object. The movement occurs by translating the user in a straight line, which is appropriate for tasks in which the user has a goal to accomplish and wants it done quickly and efficiently. During the transition, the navigation control is disabled and the user is scaled to the appropriate size.
- *Path-based multiscale navigation* (Kopper *et al*., 2006). By specifying a series of landmarks, the user can create a flight along an interpolated path passing all the landmarks in sequence. The main advantage of using a navigation path for exploration lies in the fact that the user can choose to stay on the path and let the system guide the way, or can stop at any point of interest on the path, detach from the path, explore the local context and then continue back on the guided path. Here, the path offered by the system must span enormous scales to support guided tours through MSVEs.
- *Steering-based multiscale navigation* (Li *et al*., 2006). The method of changing scale is by travelling in the direction of an LoS object and entering it. The technique is designed for exploratory multiscale navigation, when time is not the most important factor.
- *Hierarchically Structured Map (HiSMap) based navigation* (Bacim *et al*., 2009). The HiSMap shows the entire structure of the hierarchy formed by the levels of scale, so that users can view and select any level of scale at any time, as illustrated in Figure 5 in which several icons are connected by blue lines over the virtual tablet. Each icon represents a specific level of scale and the lines connect nested levels of scale. The rows in which these icons are located represent how many scales nest the level of scale that is illustrated by the icon. If the user moves the virtual hand and touches the icons, the selected scale changes to the one that is being touched.

There are also many multiscale navigation aids that help to avoid spatial disorientation in large multiscale environments.

- A You-are-Here map is a powerful tool for spatial knowledge acquisition; a map works better when aligned with the environment to avoid mental rotations.

- A miniature of the top scale is always shown in the top right corner of the display, with the same orientation as the user. A blinking dot representing the user's position is also shown in a miniature model of the environment.
- The use of 'gravity' with the steering-based technique has been suggested as a method for pulling the user towards potentially interesting scales or features (Kopper *et al*., 2006)

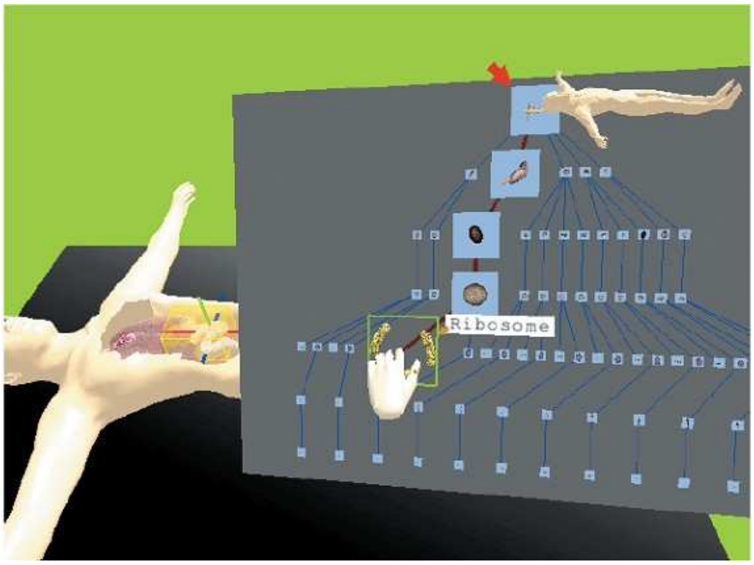

Figure 5. The HiSMap: the level of scale and its icon representation being highlighted when touched (Bacim et al., 2009)

5.2 Crop and zoom

The simplest scene-in-hand method of navigating a multiscale volume image is probably the interaction used by Agrawal *et al*. (2010), which we will define as *crop-and-zoom*. This interaction consists of selecting regions of interest with a bounding-box widget, then cropping and zooming in on the subvolume. This is very effective for zooming into subvolumes and useful in applications in which the cropping is desirable or necessary, such as volumes serialised from out-of-core. For general exploratory browsing of a volume image, the basic box widget is not a very good tool for zooming out and moving around, but could easily be made so by the addition of a translation handle and a dedicated zoom-out interaction. The interaction is scene-in-hand throughout; there is no sense of immersion in the global volume.

6 Classification and design of multiscale visualisation

As we have seen, multiscale data comes in many forms and a variety of techniques have been developed to visualise it. This section attempts to answer the question: which techniques should be deployed, given the input data?

The factors affecting multiscale visualisation can be grouped under the following headings:

- type of data and visualisation style
- nature of multiscale
- style of interaction
- choice of multiscale technique.

6.1 Type of data and visualisation style

The types of data and visualisation styles that have been discussed in this report can be summarised as follows:

- spatial data
  - volume image
  - surface
  - planar slice
  - point data
  - vector field
  - tensor field
  - finite element
  - AMR
  - time-varying data
- time series
- GIS
- InfoVis
- genomics

GIS, InfoVis and genomics have their own specialist techniques for multiscale and are beyond the scope of this analysis. Likewise, AMR data requires a dedicated importer and renderer. We will restrict this analysis to conventional spatial and temporal data.

The visualisation style affects the choice of multiscale techniques which might be deployed. For example, techniques appropriate to a direct volume rendering might include out-of-core chunking, fly-through or call-outs; vector field visualisation might require bundling of streamlines; and slice visualisations would be well-suited to click-and-zoom interactions.

6.2 Nature of multiscale

How does the multiscale manifest itself in the data and what problems are inherent? The following are important factors in deciding which multiscale techniques need to be recruited into the visualisation:

- *Range of spatial scales*. The data may be considered multiscale if the range of scales is more than about 3 orders of magnitude, i.e. the resolution of a typical computer screen.
- *Range of temporal scales*. The display resolution of time-series data is lower than that of spatial data. The resolution which can be accommodated on a static display graph is no more than about 1 part in 100, indicating the need for some sort of select-and-zoom if the scale range is more than about two orders of magnitude. If the time-series data is linked to an animated spatial display, as in Viceconti *et al*. (2011), the resolution is further limited by the human perceptual range, since the duration of the animation must be acceptable to the user, with perhaps only a factor of 10 separating too fast from too slow. Thus, even time data which is hardly “multiscale” at all in terms of orders of magnitude is likely to benefit from the temporal zooming techniques described in this report.
- *Gaps between scales*. If the scale range is not continuous, such as data captured or simulated at widely different scales, interactions which assume continuous scale, such as the fly-through, might not be suitable, or require a user interaction to jump scales, as in the transition to Street View in the Google Earth environment.

- *Salient scales*. It is possible that the data might need to be analysed to identify particular scales of interest. This is common in 2D image analysis, for determining the optimum scales of image operators (Kadir & Brady, 2001; Kadir & Brady, 2003). In 3D medical images, where high resolution is expensive to capture, it can probably be assumed that all scales are important and that no further analysis is required beyond reading the dimensions of the image. The need to extract salient scales is more likely to be of use in time-series data, where the sampling resolution might be high compared to the smallest features of interest.
- *Size of data*. Data which is larger than 1-3 GB might not fit into the RAM on a desktop PC and 1GB would probably exceed the memory of a typical GPU. The precise threshold at which data size becomes a problem depends on the hardware, which might be top-of-the-range, or several years old, depending on the target user. Large data requires the deployment of out-of-core serialisation methods.
- *Number of target objects*. The multiscale requirements of single images and scenes containing multiple objects are very different. Single very large images are typically straightforward to visualise, once the issues of out-of-core serialisation and selecting sub-volumes are dealt with. A scene containing multiple objects at different scales is a different matter, requiring an arsenal of techniques to deal with registration, placeholding, selection and navigation of multiple targets.
- *Scene structure*. A dataset of multiple objects might have the objects pre-arranged into a hierarchy, with the positions of objects registered relative to their parent nodes, or there might be no hierarchy, with all the objects sharing the same world coordinate system. If a hierarchy of relative positions exists, zooming on sub-scale objects might involve a transformation to that object's coordinate system. Methods such as the Bounding Volume Hierarchy (BVH) or Oriented Bounding Box (OBB) trees exist (Gottschalk *et al*., 1996) which can automatically create a scene hierarchy if none exists.
- *Occlusions*. If the scene contains nested or overlapping objects, there is a potential for occlusion. If data is occluded, placeholder tokens might be hard to see, so other methods should be considered for indicating the location of sub-scale data. A BVH analysis would measure the extent of occlusion in the scene.
- *Periodicity in time*. Multiscale visualisation of periodic data should ideally include functionality which allows the user to view a small feature and locate and animate all instances of it over time. This involves two timescales: the window size of the feature and the step size between instances.
- *Fibrous features*. Fibrous features such as dense connections and vector-field streamlines require bundling to handle the level of detail.
- *Ill-conditioning*. The dataset should be checked for ill-conditioning, which can arise when small objects are placed in a much larger scene and for coordinates which in absolute terms are larger or smaller than the limits of the graphics platform. The risk of ill-conditioning depends on whether the graphics software and hardware, including the user's GPU, uses double or single floating point precision.

6.3 Style of interaction

As discussed previously in Section 5.1, the two interaction modes, or metaphors, that are used for interacting with scenes are:

- scene-in-hand (user is outside the scene)
- fly-through (user is immersed in the scene).

The kinds of tools by which a user interacts with the scene, part from the standard keyboard and mouse, can be grouped under the following headings:

- in-view tokens and widgets
- external dialog buttons
- external maps, overlays and panels.

6.4 Multiscale techniques

Tables 2 and 3 together show a summary of the multiscale techniques and features which have been discussed in this report and their applicability. It is notable that the multiscale techniques described here are all add-ons, to be integrated into a scene *after* the user has selected the basic visualisation style (see Section 6.1.3). None of these techniques are visualisation styles in their own right; they are ways of navigating datasets in multiscale, but do not introduce new ways of presenting the data.

Table 2 groups the multiscale techniques by function, forming a menu for designing a multiscale visualisation – a visualisation can be constructed by selecting one technique, or none, corresponding to each component, plus any number of the miscellaneous techniques. Most functional components have more than one option, and most options should be fairly independent of each other: for example, the choice of how to indicate the location of sub-scale data should not strongly affect the choice of how to magnify it. This classification also suggests some unusual or unlikely combinations, such as magnifying call-outs in a fly-through interaction. We have included double precision as a "technique" for dealing with ill-conditioning, since the use of double instead of single precision might in some cases be the simplest solution, provided that this is possible in terms of the software platform and the target hardware.

Functional component	Techniques	Applicability
Out-of-core serialisation	Chunking	Large data.
	Pre-fetching	Large data.
Interaction mode	Scene-in-hand	Most images and scenes, but metaphor might not be sustainable in very large scenes.
	Fly-through	Immersive visualisation of very large scenes, e.g. cities, seismic data. Free-form navigation of large images with no preset targets.
Indication of sub-scale data	Placeholder tokens	Uncluttered scenes such as slice visualisations. Easily occluded.
	Annotating call-outs	Scenes in which simple placeholder tokens would be occluded.
	Hyperlinks	Metadata in uncluttered scenes.
	User-defined volumes of interest	Volumes of interest defined interactively by user. Used for free-form selection of sub-volumes for inspection or cropping. Applicable to both scene-in-hand and fly-through interactions.
	External maps, panels or overlays	Cluttered, or complex scenes
Magnification of sub-scale data	Zoom	Simple and generally applicable.
	Hyperlink	Useful for opening associated metadata. Not recommended for navigating 3D scenes.
	Split screen	Simultaneous display of scale levels, especially with time-varying data.
	Lensing	Common in 2D images, sometimes 3D. Retains context and connectivity. Can show different scalars in multiparameter data. Distortion might be a problem.
	Magnifying call-out	Generally applicable. Simultaneous display of scale levels. Link to global position is preserved. Good for non-interactive presentation. Screen space is an issue.
LoD technique	Bundling	Important LoD technique for fibres, vector-field streamlines and dense connections.
	Image resampling	Image data
	Cartographic	Maps and GIS
	Data cubes	Common data structure in InfoVis, with concept hierarchy analogous to multiscale.
	Environment textures	Efficient visualisation of distant background data.
	Decimation & subdivision	Mesh data
	Other	Application specific

Global context	External navigation panels, maps and overlays	Navigation aids in complex scenes. Multiscale models with pre-defined scale levels.
	Wayfinding widgets	Navigation in fly-through interactions. Very large scenes.
	Tree	InfoVis or hierarchical data.
Handling of ill-conditioning	Scene hierarchy with relative positioning	Objects positioned relative to parents in hierarchy, instead of world coords. May be pre-defined by user or generated automatically.
	Power-scaled coordinates	Scenes with large distances between small objects, e.g. astrophysics. Scenes with very large range of scales causing numeric overflow.
	Mobile origin	Method used in fly-through to avoid ill-conditioning.
	Double precision	Simple solution to ill-conditioning, but might not be supported by software and hardware.
Temporal multiscale	Temporal zoom	Time series data with more than 2 orders of magnitude, or with coupled animated display.
Miscellaneous techniques and features	Scale saliency	Automatic detection of scales and features of interest.
	Cube maps	Automatic detection of current scale in fly-through interaction.
	Temporal gating	Tool for periodic time-series data.
	Dialog buttons	Extra functionality for correcting navigation errors or an alternative to mouse dragging, e.g. reset, go-back and zoom-out.

Table 2. Multiscale design menu: components of a multiscale visualisation and corresponding techniques. A visualisation can be constructed by selecting one technique, or none, for each component, plus any number of the miscellaneous techniques

Some of the techniques discussed in this report are composite in terms of function, and have not been listed explicitly in Table 2; for example, a simple click-and-zoom view, such as that described in Viceconti *et al*. (2011), is a combination of two techniques: in-place tokens to indicate the sub-scale data and a zoom to magnify it. These are listed in Table 3.

Technique	Applicability
Click-and-zoom	Mouse click invokes magnification. Multi-object scenes or scenes with preset targets.
Click-and-fly	Mouse click invokes travel to target, often with no change in scale. Used in immersive environments.
Crop and zoom	Simple method for selecting and zooming into regions of interest in volume images. Useful where cropping is desirable, such as serialised large volumes.

Table 3. Composite multiscale techniques

7 Formal analysis techniques

This section considers the use of formal techniques that can assist in the design of a multiscale visualisation. The multiscale design process consists of assessing all the factors described in Section 6 and using this information to select appropriate techniques, with the assistance of Table 3. The factors that affect the multiscale design are the type of data, the visualisation style, the interaction style and the nature of the multiscale in the data. Of these, the first three are evident without any formal analysis. If there is a role for formal analysis, it is in quantifying the nature of multiscale in the data and analysing it for the presence of the features described in Section 6.2.

Techniques for quantifying the presence and structure of multiscale in a dataset would be classed as *Exploratory Data Analysis* (EDA) (Andrienko, 2005). EDA assisted by interactive visualisation is a domain of information visualisation often termed *Visual Analytics*. Multiscale is characterised by the presence of small detail and a large range of spatial or temporal frequencies in the data. Therefore frequency domain techniques such as *Fourier analysis* and *wavelets* are likely to be useful. These are applicable to both spatial and temporal data. Multiscale might also be characterised by fractal structure in some domains; one such example is the solar corona (Conlon *et al*., 2008).

For convenience, we list again, from Section 6.2, the features of the data that are most important from the point of view of multiscale visualisation:

- range of spatial scale
- range of temporal scale
- gaps between scales
- salient scales
- size of data
- number of target objects
- scene structure
- occlusions
- periodicity in time
- fibrous structures
- ill-conditioning.

The range of spatial scales normally corresponds to the resolution of the dataset and requires no special analysis for its extraction. However, we have already encountered astrophysical simulation data (Kahler *et al.*, 2002) in which the scale range was so large that the users required automated assistance to locate the scales and features of interest.

With temporal data, high resolution is cheap, so the smallest scale of interest might not be the same as the sampling resolution. In cases in which the timescales of interest are not known *a priori*, Fourier analysis would reveal the frequencies at which activity occurs. However, Fourier analysis might not tell the whole story: a step change is a high-frequency feature, but it does not follow that it is worth inspecting at the highest resolution. More sophisticated methods exist from information theory for estimating level of interest; in computer vision, a quantity called *entropy* (Gonzalez & Woods, 2008; Bishop, 2006) is widely used to quantify the amount of information in a texture or scene. An entropy-based technique called *scale saliency* was described by (Kadir & Brady, 2001; Kadir & Brady, 2003) for locating scales and features of interest in images. Such methods can be applied to graphics and time-series data where *a priori* knowledge and manual inspection are not sufficient.

The size of the data and the number of target objects should be readily determined without the need for mathematical techniques. It is possible that the exact number of targets might not be known beforehand, perhaps because they are to be located automatically in a fly-through scene, but the exact number is not critical for the design process.

The scene structure is particularly amenable to formal analysis. From the point of view of multiscale design, it can be useful to arrange the objects in a scene into a hierarchy. Methods exist to construct a hierarchical scene graph from a scene if this has not been established *a priori*; the most widely used is the *Bounding Volume Hierarchy* (BVH) or *Oriented Bounding Box Tree (*Gottschalk *et al*., 1996; de Berg & Hachenberger, 2009; Tian *et al*., 2010). The BVH is widely used in graphics for collision detection and is typically constructed by an analysis of the relative size and overlaps of the oriented bounding boxes of the objects in the scene. In Garcia *et al*. (1999), a minimal spanning tree was used to create a scene graph; the spanning tree is very easy to implement, though it does not yield information about occlusion.

Occlusions are important because they can hide embedded sub-scale data. Again, the BVH is a useful tool for quantifying the amount of occlusion.

As discussed in Section 3.5, the inspection of periodic time signals is likely to require a gating facility, which allows the user to step from one example of a given feature to the next. Gating requires some frequency-domain analysis to determine the period of the data and some template matching or heuristics for accurate location of the feature.

Fibrous structures arise from streamlines in vector field data, or in connection networks in InfoVis applications. The presence of fibrous structure would typically be known *a priori*.

Ill-conditioning in the data is indicated when objects are so small compared with their distance from the origin that the resolution of the points on the object is lower than the numerical precision of the computer. The limit depends on the floating point storage format: A 4-byte float with a 23-bit mantissa has 6-7 significant digits; an 8-byte double with a 52-bit mantissa has 15-16. Many applications, particularly those including biomolecular simulations, could exceed the single-precision limit. The double-precision limit is unlikely to be breached, except in extreme astrophysical applications. Precision problems can also arise if the values of the coordinates in the base units exceed the limits that might be imposed by the graphics platform; this was about 10^{-4} units in VTK 4, but is currently 10-^{20} in VTK 5 (Kitware, 2010).

8 WP3 Challenges

This section will relate the challenges described in WP3 to the components and techniques presented in this report.

The multiscale challenges listed in WP3 were as follows.

1. different spatial scales
2. registration issues
3. very large data
4. gaps between scales
5. heterogeneous data types
6. heterogeneous dimensionality
7. high dimensionality
8. interactive visualisation
9. time-varying issues.

Most of the WP3 challenges have been covered in some way this report. Challenges 1, 3, 4 and 9 have been listed explicitly as features of the data which directly affect the design of a multiscale visualisation. Regarding Challenge 2, registration, has not been discussed in this report, but it is a pre-requisite of most biomedical modelling and visualisation, and could pose problems in a multiscale scene. Challenge 5, heterogeneous data types, is also included in the list of data features, in the sense that it refers to datasets that consist of multiple objects, though we have not yet considered the problem of datasets in which those objects are of different types with different visualisation requirements; such datasets could pose interesting problems for the design process, particularly if their components have conflicting multiscale requirements. In Challenge 7, data with high dimensionality includes vector and tensor fields, which we have discussed, and which are likely to pose particular problems in terms of level-of-detail. Challenges 5-7 might be loosely grouped together as problems which would be complex even without multiscale; we have not encountered multiscale issues which explicitly concern these problems, but practical experience may well reveal multiscale problems which have not been anticipated.

Regarding Challenge 8, we have discussed at length the techniques that have been deployed to achieve successful user interaction; one might say that this challenge *is* multiscale visualisation. Within this challenge are a number of active problems, including the following.

- *Specifying the scale of a movement or intended interaction*. If the user is viewing a small portion of a large scene, and wants to make a very large movement to another small feature that is located very far away, how should that movement be specified?
- *Understanding position in a large dataset*. Part of the challenge of interaction within a massive dataset is to enable users to understand where they are and where they want to go relative to their current position. If the user is viewing at a small scale, this is challenging because the target destination might be out of the field of view; if the user is viewing at a

large scale, the challenge might be in handling occlusion or accurately specifying the location to which they wish to move.

- *Picking a point in 3D or moving to a desired point of view in 3D using a mouse*. This is difficult, and multiscale objects and surfaces inherently confound those tasks because of their complexity. Computer developers tend to take the mouse for granted, but recent technologies such as Microsoft Kinect, Wiimotes, 3D mice and others have shown that there are more possibilities and that these are sometimes effective and affordable.

9 Conclusions

Multiscale visualisation is used in fields such as cartography, astrophysics and information visualisation wherever the range of spatial or temporal scales exceeds the resolution of the display or, less frequently, the capabilities of the visual system of the human user, and a variety of techniques have been developed to allow datasets to be navigated at different scales. With the exception of genomics, multiscale visualisation has received rather less attention in the biomedical area, largely because of the difficulty and expense of creating high-resolution datasets.

Strictly speaking, multiscale techniques are not visualisation styles in their own right, in the manner of volumes, surfaces, slices or streamlines; they generally provide additional navigation features which are integrated into a view after the visualisation style has been decided by the user.

When we consider what will be most useful or appropriate in a given scenario, the following factors have been identified as integral to the design of the form of multiscale visualisation to be adopted:

- type of data
- nature of multiscale in the data
- style of visualisation
- style of interaction

Similarly, the following features in the data have been identified as likely to affect the applicability of particular multiscale techniques:

- range of spatial and temporal scales
- gaps between scales
- need for automatic detection of salient scales
- size of data
- number of target objects
- scene structure
- occlusions
- periodicity in time
- fibrous structures
- ill conditioning.

Most of the multiscale techniques described in this report can be grouped under 8 headings which describe the functional components of a multiscale view. The design of a multiscale view then consists of selecting an appropriate technique (if necessary) for each function. The functional components of a multiscale view are:

- out-of-core
- interaction mode
- indication of sub-scale data

- magnification of sub-scale data
- LoD technique
- global context
- handling of ill-conditioning
- temporal multiscale.

This report has presented a classification of multiscale visualisation, in which a variety of techniques have been grouped by function. A design menu has been presented showing the functional components of a multiscale visualisation, the techniques available for each and the visualisation scenarios in which they are applicable.

Some formal analysis techniques have been suggested for quantifying the nature of the multiscale in the data, in order to assist the design process. For example, the construction of a scene graph using a Bounding Volume Hierarchy could be useful.

This report has demonstrated that the multiscale visualisation literature provides a rich variety of possible solutions, but that these currently exist in isolation from each other. When they are considered together, a small number of general functionalities emerge, and the techniques are seen as a list of design options for providing them. We have shown how the design process is also influenced by the properties of the data and the way in which the user chooses to visualise and interact with it.

It is likely that there exist multiscale problems that have not yet been addressed, perhaps because resolving the visualisation issues has been beyond the remit and resources of the corresponding application. One such example is the simulation and visualisation of the clotting protein fibrin, which requires time-varying visualisation simultaneously at several scales. In this problem, the requirement for simultaneous display strongly suggests the use of call-outs, but the potentially large number of call-outs that would result implies that some novel management of the screen space would also need to be developed, in order to ensure that the user has continual access to all the relevant information across all of the scales and can immediately perform the necessary interaction at whatever scale the information indicates is currently most appropriate.

We have shown that for most applications, multiscale visualisation does not require new techniques, but rather a new unified approach. Currently it takes a great deal of research and programming effort to create a multiscale visualisation for an application, but with guidelines to provide a suitable infrastructure for design, and a software library that supports a range of multiscale methods, possibly enabling a variety of different methods to be applied within one application, with each scale using the approach most appropriate to its needs, multiscale visualisation will become more accessible, and many more problems will become amenable to solution.

References

A. Agrawal, J. Kohout, G. Clapworthy, N. McFarlane, F. Dong, M. Viceconti, F. Taddei and D. Testi (2010) Enabling the interactive display of large medical volume datasets by multiresolution bricking. *Journal of Supercomputing* **51**(1):3-19 (doi: 10.1007/s11227-009-0289-2)

H. Akiba, N. Fout and K.L. Ma (2006) Simultaneous classification of time-varying volume data based on the time histogram. In *Proc. Eurographics /IEEE VGTC Symposium on Visualization (EuroVis '06)*, pp. 171-178 (doi: 10.2312/VisSym/EuroVis06/171-178)

H. Akiba and K.L. Ma (2007) A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proc. Eurographics /IEEE VGTC Symposium on Visualization (EuroVis '06)*, pp. 115-122 (doi: 10.2312/VisSym/EuroVis07/115-122)

N. Akkiraju, H. Edelsbrunner, P. Fu and J. Qian (1996) Viewing geometric protein structures from inside a CAVE. *IEEE Computer graphics and applications* **16**(4):58-61 (doi: 10.1109/38.511855)

AMUSE Project (2011) http://www.amusecode.org

G. Andrienko (2005) *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*, Springer

S.M. Arnold, B. Bednarcyk, A. Hussein and V. Katiyar (2010) Micromechanics-Based Structural Analysis (FEAMAC) and Multiscale Visualization Within Abaqus/CAE Environment. *NASA Technical Report, NASA/TM-2010-216336*

F. Auchere, E. Soubrie, K. Bocchialini and F. LeGall (2008) FESTIVAL: a multiscale visualization tool for solar imaging data. *Solar physics,* **248**(2):213-224 (doi: 10.1007/s11207-008-9163-2)

M. Auer, H.C. Peng and A. Singh (2007), Development of multiscale biological image data analysis: Review of 2006 International Workshop on Multiscale Biological Imaging, Data Mining and Informatics, Santa Barbara, USA (BII06) *BMC Cell Biology,* **8**(Suppl 1):S1 (doi:10.1186/1471-2121-8-S1-S1)

F. Bacim, D. Bowman and M. Pinho (2009) Wayfinding techniques for multiscale virtual environments. In *Proc. IEEE Symposium on 3D User Interface*, pp. 67-74 (doi: 10.1109/3DUI.2009.4811207)

J. Balabanian, I. Viola, T. Moller and E. Groller (2008) Temporal styles for time-varying volume data. In *Proc. Fourth Int. symp. on 3D Data Processing, Visualization and Transmission (3DPVT'08)*, June, Atlanta, USA, pp. 81-89

M. Balzer and O. Deussen (2007) Level-of-detail visualization of clustered graph layouts. In *Proc. 6th Asia-Pacific Symposium on Visualization (APVIS'07)*, Feb 5-7, Sydney, NSW, pp. 133-140 (doi: 10.1109/APVIS.2007.329288)

M.J. Berger and J. Oliger (1984) Adaptive mesh refinement for hyperbolic partial differential equations. Journal *of Computational Physics*, **53**:484–512 (doi: 10.1016/0021-9991(84)90073-1)

E.A. Bier, M.C. Stone, K. Pier, W. Buxton and T.D. DeRose (1993) Toolglass and magic lenses: the see-through interface. In *Proc. ACM SIGGRAPH'93*, Aug 1-6, Anaheim, Ca, USA, pp. 73-80 (doi: 10.1145/166117.166126)

Biodigital Human (2011) http://www.biodigitalhuman.com

C.M. Bishop (2006), *Pattern recognition and machine learning*, ch. 1.6, pp. 48-58, Springer Science

D.A. Bowman, E. Kruijff, J.J. LaViola and I. Poupyrev (2004), *3D user interfaces: theory and practice.* Addison Wesley

B.S. Brook and S.L. Waters, eds. (2007) Research challenges. In *Seeding the EuroPhysiome: A roadmap to the virtual physiological human*, ch. 6, pp. 38-55, European Commission FP6-IST-2004 Co-ordination action 027642

J. Brosz, S. Carpendale and M.A. Nacenta (2011) The undistort lens. In *Eurographics/IEEE Symposium on visualization, EuroVis 2011*, pp. 881-890, June 1-3, Bergen, Norway (doi: 10.1111/j.1467-8659.2011.01937.x)

A.D. Brown, T.C. Mowry and O. Krieger (2001) Compiler-based i/o prefetching for out-of-core applications. ACM Transactions on Computer Systems **19**(2):111-170 (doi: 10.1145/377769.377774)

B. Cabral and L.C. Leedom (1993) Imaging vector fields using line integral convolution. In *Proc. ACM SIGGRAPH'93*, Aug 1-6, Anaheim, Ca, USA, pp. 263–270 (doi: 10.1145/166117.166151)

R. Cardenes, E. Munoz-Moreno, A. Tristan-Vega and M. Martin-Fernandez (2010) Saturn: a software application of tensor utilities of research in neuroimaging. *Computer methods and programs in biomedicine,* **97**(3):264-279 (doi: 10.1016/j.cmpb.2009.09.007)

B.S. Carmo, Y.H.P. Ng, A. Prugel-Bennett and G.Z. Yang (2004) A data clustering and streamline reduction method for 3d MR flow vector field simplification. In *Medical Image Computing and Computer-assisted Intervention (MICCAI'04) Part I*, Sept 26-29, St. Malo, France, pp. 451-458 (doi: 10.1007/978-3-540-30135-6_55)

M.S.T. Carpendale, D.J. Cowperthwaite and F.D. Fracchia (1997) Extending distortion viewing from 2D to 3D. *IEEE Computer Graphics and Applications: Special Issue on Information Visualization*, **17**(4):42-51 (doi: 10.1109/38.595268)

C. Chang, T. Kurc, A. Sussman and J. Saltz (2000) Optimizing retrieval and processing of multi-dimensional scientific datasets. In *Proc. of the Third Merged IPPS/SPDP Symposiums*, IEEE Computer Society Press, pp. 405-410 (doi: 10.1109/IPDPS.2000.846013)

W. Chen, S. Zhang, S. Correia and D.S. Ebert (2008) Abstractive representation and exploration of hierarchically clustered diffusion tensor fiber tracts. *Computer Graphics Forum*, **27**(3):1071–1078 (doi:10.1111/j.1467-8659.2008.01244.x)

Y. Cheng, L. Jiang, X. Ma, J. Xue and Z. Zheng (2011) Multi-resolution texture rendering for medical data. In *Proc. Workshop on Digital Media and Digital Content Management*, pp. 166-171 (doi: 10.1109/DMDCM.2011.43)

D. Chisnall, M. Chen and C. Hansen (2006) Knowledge-based out-of-core algorithms for data management in visualization. In *Eurographics/IEEE VGTC Symposium on Visualization (EUROVIS)*, pp. 107-114

P. Cignoni, F. Ganovelli, C. Montani and R. Scopigno (2000) Reconstruction of topologically correct and adaptive trilinear surfaces. *Computers & Graphics*, **24**(3):399-418 (doi: 10.1016/S0097-8493(00)00036-4)

M. Cohen and K. Brodlie (2004) Focus and context for volume visualization. In *Proc. of Theory and Practice of Computer Graphics '04*, pp. 32-39 (doi: 10.1109/TPCG.2004.1314450)

P.A. Conlon, P.T. Gallagher, R.T.J. McAteer, J. Ireland, C.A. Young, P. Kestener, R.J. Hewett and K. Maguire (2008) Multifractal properties of evolving active regions. *Solar physics* **248**(2):297-309 (doi: 10.1007/s11207-007-9074-7)

I. Corouge, S. Gouttard and G. Gerig (2004) Towards a shape model of white matter fiber bundles using diffusion tensor MRI. In *Proc. International Symposium on Biomedical Imaging*, pp. 344-347 (doi: 10.1109/ISBI.2004.1398545)

CoViz 4D (2011) http://www.dgi.com/coviz/cvmain.html

W. Cui, H. Zhou, H. Qu, P.C. Wong and X. Li (2008) Geometry-based edge clustering for graph visualization. *IEEE Trans. Visualization and Computer Graphics*, **14**(6):1277-1284 (doi: 10.1109/TVCG.2008.135)

M. de Berg and P. Hachenberger (2009) Rotated-Box Trees: A Lightweight c-Oriented Bounding-Volume Hierarchy. In Proc. 8th int. symposium on experimental algorithms (SEA '09), pp. 63-75, June 3-6, Dortmund, Germany (doi: 10.1007/978-3-642-02011-7_8)

P. Diaz-Gutierrez, A. Bhushan, M. Gopi and R. Pajarola (2005) Constrained strip generation and management for efficient interactive 3D rendering. In *Proc. Computer Graphics International (CGI)*, pp. 115-121 (doi: 10.1109/CGI.2005.1500388)

P. Doshi, E. Rundensteiner and M. Ward (2003) Prefetching for visual data exploration. In *Proc. Eighth International Conference on Database Systems for Advanced Applications,* pp. 195-202 (doi: 10.1109/DASFAA.2003.1192383)

K. Engel, M. Hadwiger, J.M. Kniss, A.E. Lefohn, C.R. Salama and D. Weiskopf (2004) Real-time volume graphics, Course notes. In *Proc. of ACM, SIGGRAPH*, New York, USA, ACM Press

B. Ernst and J.E. Brigham (1986) *The magic mirror of M.C. Escher*, Tarquin Publications

R. Estkowski, J.S.B. Mitchell and X. Xiang (2002) Optimal decomposition of polygonal models into triangle strips. In *Proc. 18th annual symposium on Computational geometry*, pp. 254-263 (doi: 10.1145/513400.513431)

Z. Fang, T. Moller, G. Hamarneh and A. Celler (2007) Visualization and exploration of time-varying medical image data sets. In *Graphics Interface 2007*, pp. 281-288 (doi: 10.1145/1268517.1268563)

G. Favalora (2005) Volumetric 3D Displays and Application Infrastructure. *IEEE Computer*, **38**(8):37-44 (doi: 10.1109/MC.2005.276)

J.W. Fernandez, P. Mithraratne, S.F. Thrupp, M.H. Tawhai and P.J. Hunter (2004) Anatomically based geometric modeling of the musculo-skeletal system and other organs. *Biomechanics and Modeling in Mechanobiology* **2**(3):139-155 (doi: 10.1007/s10237-003-0036-1)

C.W. Fu and A. J. Hanson (2006) A transparently scalable visualization architecture for exploring the Universe. *IEEE Transactions on Visualization and Computer Graphics*, **13**(1):108-121 (doi: 10.1109/TVCG.2007.2)

J. Gao, J. Huang, C. Johnson and S. Atchley (2005) Distributed data management for large volume visualization. In *Proc. IEEE Visualization*, pp. 183-189 (doi: 10.1109/VISUAL.2005.1532794)

M.A. Garcia, A.D. Sappa and L. Basanez (1999) Efficient generation of object hierarchies from 3D scenes. In *Proc. IEEE Int. Conf. on robotics & automation,* Vol. 2, pp. 1359-1364, May 10-15, Detroit, MI, USA. (doi: 10.1109/ROBOT.1999.772550)

S. Garfinkel (2007) Google Earth - how Google maps the world. *Technology Review* **110**(6):20-21

M. Garland and P.S. Heckbert (1997) Surface simplification using quadric error metrics. In *Proc. ACM SIGGRAPH*, pp. 209-216 (doi: 10.1145/258734.258849)

M. Glueck, K. Crane, S. Anderson, A. Rutnik and A. Khan (2009) Multiscale 3d reference visualization. In *Proc. the 2009 symposium on Interactive 3D graphics and games (I3D' 09)*, pp. 225-232 (doi: 10.1145/1507149.1507186)

R.C. Gonzalez and R.E. Woods (2008), *Digital image processing, 3 ed.,* ch. 8.1.4, pp. 531-534, Pearson Education

Google Earth (2011) http://google.com/earth

S. Gottschalk, M.C. Lin and D. Manocha (1996) OBBTree: a hierarchical structure for rapid interference detection. *In Proc. ACM Siggraph 1996*, pp. 171-180, Aug 4-9, New Orleans, USA (doi: 10.1145/237170.237244)

C.P. Gribble, C. Brownlee and S.G. Parker (2008) Practical global illumination for interactive particle visualization. *Computers and graphics* **32**(1):14-24 (doi: 10.1016/j.cag.2007.11.001)

S. Grimm, S. Bruckner, A. Kanitsar and E. Groller (2004) Flexible direct multivolume rendering in interactive scenes. In *Proc. Vision, Modeling and Visualization'04*, pp. 379-386

I. Guskov, K. Vidimce, W. Sweldens and P. Schroder (2000) Normal meshes. In *Proc. ACM SIGGRAPH*, pp. 95-102 (doi: 10.1145/344779.344831)

S. Guthe, M. Wand, J. Gonser and W. Strasser (2002) Interactive rendering of large volume data sets. In *Proc. IEEE Visualization*, pp. 53-60 (doi: 10.1109/VISUAL.2002.1183757)

J. Gray, A. Bosworth, A. Lyaman and H. Pirahesh (1996) Data cube: a relational aggregation operator generalizing GROUP-BY, CROSS-TAB and SUB-TOTALS. In *Proc. IEEE 12th int. conf. on data engineering, ICDE 1996*, pp. 152-159, New Orleans, USA (doi: 10.1109/ICDE.1996.492099)

T. Grossman and R. Balakrishnan (2006) An evaluation of depth perception on volumetric displays. In *Proc. working conference on Advanced visual interfaces (AVI'06),* pp. 193-200 (doi: 10.1145/1133265.1133305)

A.J. Hanson, C.W. Fu and E.A. Wernert (2000) Very large scale visualization methods for astrophysical data. In *Proc. Joint Eurographics and IEEE TVCG Symposium on Visualization, Data Visualization 2000*, pp. 115-124, May 29-31, Amsterdam, NL (doi:10.1007/978-3-7091-6783-0_12)

A. Helgeland and O. Andreassen (2004) Visualization of vector fields using seed LIC and volume rendering. *IEEE Trans. Visualization and Computer Graphics*, **10**(6):673-682 (doi: 10.1109/TVCG.2004.49)

D. Holten (2006) Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. on Visualization and Computer Graphics*, **12**(5):741-748 (doi: 10.1109/TVCG.2006.147)

D. Holten and J.J. van Wijk (2009) Force-Directed Edge Bundling for Graph Visualization, *Eurographics/ IEEE-VGTC Symposium on Visualization*, **28**(3):983-990 (doi:10.1111/j.1467-8659.2009.01450.x)

[CG186] M. Hopf, M. Luttenberger and T. Ertl (2004) Hierarchical splatting of scattered 4D data. *IEEE Computer Graphics* **24**(4):64-72 (doi: 10.1109/MCG.2004.7)

H. Hoppe (1996) Progressive meshes. In *Proc. ACM SIGGRAPH*, pp. 98-108 (doi: 10.1145/237170.237216)

H. Hoppe (1999) Optimization of mesh locality for transparent vertex caching. In *Proc. 26th annual conference on Computer graphics and interactive techniques*, pp. 269-276 (doi: 10.1145/311535.311565)

W.H. Hsu, K.L. Ma and C. Correa (2011) A rendering framework for multiscale views of 3D models, *ACM Transactions on graphics* **30**(6) (doi: 10.1145/2070781.2024165)

J. Huang, K. Mueller, N. Shareef and R. Crawfis (2000) FastSplats: Optimized Splatting on Rectilinear Grids. In *Proc. IEEE Visualization*, pp. 219-226 (doi: 10.1109/VISUAL.2000.885698)

J.A. Insley. L. Grinberg and M.E. Papka (2011) Visualizing multiscale, multiphysics simulation data: brain blood flow. *Preprint ANL/MCS-P1930-0911*, U.S. Dept. of Energy

M. Isenburg and P. Lindstrom (2005) Streaming meshes. In *Proc. IEEE Visualization (Vis'05)*, pp. 231-238 (doi: 10.1109/VISUAL.2005.1532800)

IUPS Physiome (2011) http://www.physiome.org.nz/

R. Jianu, C. Demiralp and D.H. Laidlaw (2009) Exploring 3D DTI fiber tracts with linked 2D representations. *IEEE Trans. Visualization and Computer Graphics*, **15**(6):1449-1456 (doi: 10.1109/TVCG.2009.141)

C.R. Johnson, Y. Livnat, L. Zhukov, D. Hart and G. Kindlmann (2001) Computational field visualization. In *Mathematics Unlimited - 2001 and Beyond, Vol. 2*, B. Engquist & W. Schmid, eds., pp. 605-630, Springer

R.R. Jones, K.J.W. McCaffrey, P. Clegg , R.W. Wilson, N.S. Holliman, R.E. Holdsworth, J. Imber and S. Waggott (2009) Integration of regional to outcrop digital data : 3D visualisation of multi-scale geological models. *Computers & geosciences*, **35**(1):4-18 (doi: 10.1016/j.cageo.2007.09.007)

A. Joshi and P. Rheingans (2005) Illustration-inspired techniques for visualizing time-varying data. In *Proc. IEEE Visualization '05*, pp. 679-686 (doi: 10.1109/VISUAL.2005.1532857)

K.I. Joy (2009) Massive data visualization: a survey. In T. Moller, B. Hamann and R.D. Russell, eds., *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, pp. 285-302, Springer (doi: 10.1007/978-3-540-49926-8)

T. Kadir and M. Brady (2001) Saliency, scale and image description. *International journal of computer vision,* **45**(2):83-105 (doi: 10.1023/A:1012460413855)

T. Kadir and M. Brady (2003) Scale saliency: a novel approach to salient feature and scale selection. In *IEEE int. conf. on visual information engineering, VIE 2003,* pp25-28, July 7-9 (doi: 10.1049/cp:20030478)

R. Kahler, D. Cox, R. Patterson, S. Levy, H.C. Hege and T. Abel (2002) Rendering the first star in the universe - a case study. In *Proc. IEEE conf. on visualization*, VIS 02, pp. 537-540 (doi: 10.1109/VISUAL.2002.1183824)

A. Kanitsar, D. Fleischmann, R. Wegenkittl, P. Felkel and E. Groller (2002) CPR - curved planar reformation. In *Proc. IEEE Visualization (Vis 2002)*, Oct 27 - Nov 1, Boston, MA, USA, pp. 37-44 (doi: 10.1109/VISUAL.2002.1183754)

A. Kanitsar, R. Wegenkittl, D. Fleischmann and E. Groller (2003) Advanced curved planar reformation: flattening of vascular structures. In *Proc. IEEE Visualization (Vis 2003)*, Oct 19-24, Seattle, Wa, USA, pp. 43-50 (doi: 10.1109/VISUAL.2003.1250353)

T. J. Kelly and K.L. Ma (2001) A study of transfer function generation for time-varying volume data. In *Proc. Volume Graphics '01*, pp. 51-68

G. Kindlmann and D. Weinstein (1999) Hue-balls and lit-tensors for direct volume rendering of diffusion tensor fields. In *Proc. IEEE Visualization '99*, Oct 24-29, San Francisco, Ca, USA, pp.183-190 (doi: 10.1109/VISUAL.1999.809886)

G.L. Kindlmann (2004) Superquadric Tensor Glyphs. In *Proc. EG Symposium on Visualization (VisSym 2004)*, May 19-21, Konstanz, Germany, pp. 147-154

Kitware Inc. (2003) *The VTK User's Guide, VTK 4.2*, Kitware Inc., USA

Kitware Inc. (2010) *The VTK User's Guide, 11 ed.*, Kitware Inc., USA

R. Kopper, T. Ni, D.A. Bowman and M. Pinho (2006) Design and evaluation of navigation techniques for multiscale virtual environments. In *Proc. IEEE conference on Virtual Reality (VR '06)*, pp. 174-182 (doi: 10.1109/VR.2006.47)

M. Kraus and T. Ertl (2002) Adaptive texture maps. In *Proc. ACM Siggraph /Eurographics conference on Graphics hardware*, Saarbrucken, Germany, pp. 1-10

P. Lacroute and M. Levoy (1994) Fast volume rendering using a shear-warp factorization of the viewing transform. Computer Graphics, In *Proc. ACM SIGGRAPH*, pp. 451-457

E.C. LaMar, B. Hamann and K.I. Joy (2000) Multiresolution techniques for interactive texture-based volume visualization. In *Proc. SPIE 3960, 365(2000)* (doi:10.1117/12.378913)

E. LaMar, B. Hamann and K.I. Joy (2001) A magnification lens for interactive volume visualization. In *Proc. of the Ninth Pacific Conference on Computer Graphics and Applications*, pp. 223-232 (doi: 10.1109/PCCGA.2001.962877)

R. Laramee (2002) Interactive 3D flow visualization using a streamrunner. In *Proc. CHI EA '02 CHI '02 extended abstracts on Human factors in computing systems*, pp. 804–805 (doi: 10.1145/506443.506606)

R. Laramee, C. Garth, H. Doleisch, J. Schneider, H. Hauser and H. Hagen (2005) Visual analysis and exploration of fluid flow in a cooling jacket. In *Proc. IEEE Visualization*, pp. 623-630 (doi: 10.1109/VISUAL.2005.1532850)

M. Levoy (1988) Display of surfaces from volume data. *IEEE Computer Graphics & Applications*, **8**(5):29-37 (doi: 10.1109/38.511)

M. Levoy (1990) A hybrid ray tracer for rendering polygon and volume data. *IEEE Computer Graphics and Applications*, **10**(2):33-40 (doi:10.1109/38.50671)

Y.G. Li, C.W. Fu and A.J. Hanson (2006) Scalable WIM: effective exploration in large-scale astrophysical environments. *IEEE transactions on visualization and computer graphics,* **12**(5):1005-1011 (doi:10.1109/TVCG.2006.176)

Z. Li (2007) *Algorithmic foundation of multi-scale spatial representation*, CRC Press

D.R. Lipsa, P.J. Rhodes, R.D. Bergeron and T.M. Sparr (2007) Spatial prefetching for out-of-core visualization of multidimensional data. In *Proc. of SPIE, Visualization and Data Analysis,* Volume 6495–0G, San Jose, CA, USA (doi:10.1117/12.704751)

D.R. Lipsa, R.D. Bergeron, T.M. Sparr and R.S. Laramee (2009) Dynamic chunking for out-of-core volume visualization applications. *Lecture Notes in Computer Science*, Volume 5876/2009, pp. 117-128 (doi:10.1007/978-3-642-10520-3_11)

Living Human Digital Library (2011) http://www.livinghuman.org

W.E. Lorensen and H.E. Cline (1987) Marching cubes: a high resolution 3D surface construction algorithm. In *Proc. ACM SIGGRAPH*, pp. 163-169 (doi:10.1145/37402.37422)

V. Luboz, M. Chabanas, P. Swider and Y. Payan (2005) Orbital and maxillofacial computer aided surgery: Patient-specific finite element models to predict surgical outcomes. *Computer Methods in Biomechanics and Biomedical Engineering*, **8**(4):259-265 (doi:10.1080/10255840500289921)

K. Ma and E. Lum (2004) Techniques for visualizing time-varying volume data. In *The Visualization Handbook*, pp. 511 – 531 (doi:10.1016/B978-012387582-2/50028-9)

X. Mao, Y. Hatanaka, H. Higashida and A. Imamiya (1998) Image-guided streamline placement on curvilinear grid surface. In *Proc. IEEE Visualization*, pp. 135-142 (doi: 10.1109/VISUAL.1998.745295)

O. Mattausch, T. Theubl, H. Hauser and M.E. Groller (2003) Strategies for interactive exploration of 3d flow using evenly-spaced illuminated streamlines. In *Proc. 19th Spring Conference on Computer Graphics*, pp. 213-222 (doi: 10.1145/984952.984987)

N. Max, C. Correa, C. Muelder, S. Yan, C.K. Chen and K.L. Ma (2009) Flow visualization in science and mathematics, *Journal of physics: conference series* **180**(1):1-9 (doi: 10.1088/1742-6596/180/1/012087)

J. McCrae, I. Mordatch, M. Glueck and A. Khan (2009) Multiscale 3D navigation. In *Proc. symposium on interactive 3D graphics, I3D'09* pp. 7-14, Feb 27 - Mar 1, Boston, MA, USA (doi: 10.1145/1507149.1507151)

J. McCrae, M. Glueck, T. Grossman, A. Khan and K. Singh (2010) Exploring the design space of multiscale 3D orientation. In *Proc. Int. conf. on advanced visual interfaces, AVI'10*, pp. 81-88, May 25-29, Rome, Italy (doi: 10.1145/1842993.1843008)

N. McFarlane, G. Clapworthy, A. Agrawal, M. Viceconti, F. Taddei, E. Schileo and F. Baruffaldi (2008) 3D multiscale visualisation for medical datasets. In *Proc. 5th IEEE Int. Conf on Biomedical visualization, Medivis08*, pp. 47-52, July 8-12, London, UK (doi: 10.1109/MediVis.2008.14)

B. Moberts, A. Vilanova and J.J. Wijk (2005) Evaluation of fiber clustering methods for diffusion tensor imaging. In *Proc. IEEE Visualization Conference*, pp. 65-72 (doi: 10.1109/VISUAL.2005.1532779)

N. Molino, R. Bridson, J. Teran and R. Fedkiw (2003) A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In *Proc. 12th International Meshing Roundtable,* pp. 103-114

W. Mollemans, F. Schutyser, N. Nadjmi, F. Maes and P. Suetens (2007) Predicting soft-tissue deformations for a maxillofacial surgery planning system: From computational strategies to a complete clinical validation. *Medical Image Analysis*, **11**(3):282-301 (doi: 10.1016/j.media.2007.02.003)

J.G. Moore, S.A. Schorn and J. Moore (1996) Methods of classical mechanics applied to turbulence stresses in a tip leakage vortex. *Journal of turbomachinery* **118**(4):622-629 (doi: 10.1115/1.2840917)

S. More and A. Choudhary (2000) Tertiary storage organization for large multidimensional datasets. In *8th NASA Goddard Conference on Mass Storage Systems and Technologies*, pp. 203-210, 2000.

C.B. Nielsen, M. Cantor, I Dubchak, D. Gordon and T. Wang (2010) Visualizing genomes: techniques and challenges. *Nature methods,* **7**(3):S5-S15 (doi: 10.1038/nmeth.1422)

D. Nagayasu, F. Ino and K. Hagihara (2008) A decompression pipeline for accelerating out-of-core volume rendering of time-varying data. *Computers and graphics*, **32**(3):350-362 (doi:10.1016/j.cag.2008.04.007)

B. Natarajan (1994) On generating topological consistent isosurfaces from uniform samples. The *Visual Computer*, **11**(1):52-62 (doi:10.1007/BF01900699)

NSR Physiome (2011) http://nsr.bioeng.washington.edu

S.I. O'Donoghue, A.C. Gavin and N. Gehlenborg *et al*. (2010a) Visualizing biological data - now and in the future. *Nature methods,* **7**(3):S2-S4 (doi: 10.1038/nmeth.f.301)

S.I. O'Donoghue, D.S. Goodsell and A.S. Frangakis *et al*. (2010b) Visualization of macromolecular structures. *Nature methods,* **7**(3):S42-S55 (doi: 10.1038/nmeth.1427)

C. Olston, A. Woodruff, A. Aiken, M. Chu, V. Ercegovac, M. Lin, M. Spalding and M Stonebraker (1998) DataSplash. In *Proc. 1998 ACM SIGMOD int. conf. on management of data*, pp. 550-552, June 1-4, Seattle, WA, USA (doi: 10.1145/276304.276377)

J. Ophir, S.K. Kaisar, B.S. Garra *et al*. (2002) Elastography: Imaging the Elastic Properties of Soft Tissues with Ultrasound. *Journal of Medical Ultrasonics*, **29**(4):155-171 (doi: 10.1007/BF02480847)

G.A. Petsko and D. Ringe (2009) *Protein structure and function*, Oxford UP

D. Phan, L. Xiao, R. Yeh, P. Hanrahan and T. Winograd (2005) Flow map layout. In *Proc. 2005 IEEE Symposium on Information Visualization (INFOVIS '05)*, pp. 219-224 (doi: 10.1109/INFVIS.2005.1532150)

C. Pierpaoli, P. Jezzard, P.J. Basser, A. Barnett, and G. Di Chiro (1996) Diffusion tensor MR imaging of the human brain. *Radiology*, **201**(3):637-648

J.S. Pierce and R. Pausch (2004) Navigation with place representations and visible landmarks. In *Proc. Virtual Reality 2004*, pp. 173–180, Washington, DC, USA (doi: 10.1109/VR.2004.1310071)

L. Podshivalov, Y. Holdstein, A. Fischer and P.Z. Bar-Yoseph (2009) Towards a multi-scale computerized bone diagnostic system: 2D micro-scale finite element analysis. *Communications in numerical methods and engineering* **26**(6):733-749 (doi: 10.1002/cnm.1214)

S. Pook, E. Lecolinet, G. Vaysseix and E. Barillot (2000) Context and interaction in zoomable user interfaces. In *Proc. int. conf. on advanced visual interfaces, AVI 00*, pp. 227-231, May 24-26, Palermo, Italy (doi: 10.1145/345513.345323)

F.H. Post, B. Vrolijk, H. Hauser, R.S. Laramee and H. Doleisch (1993) The state of the art in flow visualisation: feature extraction and tracking, *Computer graphics forum* **22**(4):775-792 (doi: 10.1111/j.1467-8659.2003.00723.x)

B. Preim and D. Bartz (2007a) *Visualization in medicine*, ch. 6-8, pp. 137-196, Morgan Kaufmann

B. Preim and D. Bartz (2007b) *Visualization in medicine*, ch. 10.5.1, pp. 245-248, Morgan Kaufmann

C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner and T. Ertl (2000) Interactive volume rendering on standard PC graphics hardware using multi-textures and multi-stage rasterization. In *proc. Eurographics/SIGGRAPH Workshop on Graphics Hardware*, pp. 109-118 (doi: 10.1145/346876.348238)

P.J. Rhodes, X. Tang, R.D. Bergeron and T.M. Sparr (2005) Iteration aware prefetching for large multidimensional scientific datasets. In *Proc. of the 17th international conference on Scientific and statistical database management (SSDBM'2005)*, pp. 45-54, Berkeley, CA, USA

S. Sarawagi and M. Stonebraker (1994) Efficient organizations of large multidimensional arrays. In *Proc. of the Tenth International Conference on Data Engineering*, pp. 328–336, IEEE Computer Society, Washington DC, USA (doi:10.1109/ICDE.1994.283048)

A. Sayar, G.C. Fox and M.E. Pierce (2009) Unified data access/query over integrated data-views for decision making in geographic information systems. In N. Bessis, ed., *Grid Technology for Maximizing Collaborative Decision Management and Support: Advancing Effective Virtual Organizations*, ch. 14, pp. 276-298, IGI Global (doi:10.4018/978-1-60566-364-7.ch014)

W. Schroeder, K. Martin and B. Lorensen (2002a) *The Visualization Toolkit, 3 ed.,* Kitware Inc., USA

W. Schroeder, K. Martin, and B. Lorensen (2002b) Visualising unstructured points. In *The Visualization Toolkit, 3 ed.,* ch. 9.4, pp. 337-343, Kitware Inc., USA

J. P. Schulze, M. Kraus, U. Lang and T. Ertl (2003) Integrating Pre-Integration into the Shear-Warp Algorithm. In *Proc. the Third International Workshop on Volume Graphics*, pp. 109-118 (doi: 10.1145/827051.827068)

P. Shirley and T. Tuchman (1990) A polygon approximation to direct scalar volume rendering. In *Proc. San Diego Workshop on Volume Visualization*, pp. 63-70 (doi:10.1145/99307.9932)

H. Si (2006) On refinement of constrained Delaunay tetrahedralizations. In *Proc. 15th International Meshing Roundtable*, pp. 509-528 (doi: 10.1007/978-3-540-34958-7_29)

I.A. Sigal, M.R. Hardisty and C.M. Whyne (2008) Mesh-morphing algorithms for specimen specific finite element modelling. *Journal of Biomechanics*, **41**(7):1381-1389 (doi: 10.1016/j.jbiomech.2008.02.019)

C. Silva, Y. Chiang, J. El-San and P. Lindstrom (2002) Out-of-Core algorithms for scientific visualization and computer graphics, Course notes for tutorial 4. In *IEEE Visualization*, Boston, MA, USA, IEEE Computer Society Washington, DC, USA

D. Stalling and H.C. Hege (1995) Fast and resolution independent line integral convolution. In *Proc. International Conferences on Computer Graphics and Interactive Techniques*, pp. 249-256 (doi: 10.1145/218380.218448)

R. Stoakley, M.J. Conway and R. Pausch (1995) Virtual reality on a WIM: interactive worlds in miniature. In *Proc. the SIGCHI conference on Human factors in computing systems*, pp. 265-272, New York, NY, USA (doi: 10.1145/223904.223938)

C. Stolte, D. Tang and P. Hanrahan (2003) Multiscale visualization using data cubes. *IEEE transactions on visualization and computer graphics*, **9**(2):176-187 (doi:10.1109/INFVIS.2002.1173141)

T. Taerum, M.C. Sousa, F. Samavati, S. Chan and R. Mitchell (2006) Real-time super resolution contextual close-up of clinical volumetric data. In *Proc. Eighth Eurographics/IEEE VGTC Symp. Visualization (EuroVis '06)*, pp. 347-355 (doi: 10.2312/VisSym/EuroVis06/347-354)

A. Telea and O. Ersoy (2010) Image-based bundles: simplified visualization of large graphs. *Eurographics/ IEEE-VGTC Symposium on Visualization*, **29**(3):843-852 (doi:10.1111/j.1467-8659.2009.01680.x)

A. Telea and J. van Wijk (1999) Simplified representation of vector fields. *In Proc. IEEE Visualization*, pp. 35-42 (doi: 10.1109/VISUAL.1999.809865)

J.B. Thurmond, P.A. Drzewiecki and X. Xu (2005) Building simple multiscale visualizations of outcrop geology using virtual reality modeling language (VRML). *Computers and Geosciences,* **31**(7):913-919 (doi:10.1016/j.cageo.2005.03.007)

F. Tian, W. Hua, Z. Dong and H. Bao (2010) Adaptive voxels: interactive rendering of massive 3D models, *The Visual Computer* **26**(6-8):409-419 (doi: 10.1007/s00371-010-0465-7)

M. Tory and C. Swindells (2003) Comparing ExoVis, orientation icon, and in-place 3D visualization techniques. In *29th Graphics Interface Conference*, pp. 57-64, Jun 11-13, Halifax, Canada

Uniview (2011) SCISS AB, Sweden, http://www.scalingtheuniverse.com

V. Verma, D. Kao and A. Pang (2000) A flow-guided streamline seeding strategy. In *Proc. IEEE Visualization*, pp. 163-170 (doi: 10.1109/VISUAL.2000.885690)

M. Viceconti, F. Taddei, L. Montanari, D. Testi, A. Leardini, G. J. Clapworthy and S. L. Van Sint Jan (2007) Multimod data manager: a tool for data fusion. *Computer Methods and Programs in Biomedicine,* **87**(2):148-159 (doi: 10.1016/j.cmpb.2007.05.002)

M. Viceconti, G. Clapworthy, D. Testi, F. Taddei and N. McFarlane (2011) Multimodal fusion of biomedical data at different temporal and dimensional scales. *Computer methods and programs in biomedicine,* **102**(3):227-237 (doi:10.1016/j.cmpb.2010.04.017)

J. Viega, M.J. Conway, G. Williams and R. Pausch (1996) 3D magic lenses. In *Proc. the 9th annual ACM symposium on user interface software and technology, ACM UIST '96*, pp. 51-58 (doi: 10.1145/237091.237098)

Visible Human Project (2011) http://www.nlm.nih.gov/research/visible

VPH Network of Excellence (2011) http://www.vph-noe.eu

T. Walter, D.W. Shattuck and R. Baldock *et al. (*2010) Visualization of image data from cells to organisms. *Nature methods,* **7**(3):S26-S41 (doi: 10.1038/nmeth.1431)

Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing*, **13**(4):600-612 (doi: 10.1109/TIP.2003.819861)

L. Wang, Y. Zhao, K. Mueller and A. Kaufman (2005) The magic volume lens: an interactive focus + context technique for volume rendering. In *Proc. of IEEE Visualization*, pp. 367-374 (doi: 10.1109/VISUAL.2005.1532818)

C. Wang, H. Yu and K.L. Ma (2010) Application-driven compression for visualizing large-scale time-varying data. *IEEE Computer Graphics and Applications*, **30**(1):59-69 (doi: 10.1109/MCG.2010.3)

C. Ware and S. Osborne (1990) Exploration and virtual camera control in virtual three dimensional environments**.** *ACM SIGGRAPH Computer Graphics,* **24**(2):175-183 (doi: 10.1145/91394.91442)

X. Wei, A.E. Kaufman and T.J. Hallman (2001) Case study: visualization of particle track data. In *Proc. of IEEE Visualization '01*, pp. 465-468 (doi: 10.1109/VISUAL.2001.964552)

R. Westermann and T. Ertl (1998) Efficiently using graphics hardware in volume rendering applications. In *proc. ACM SIGGRAPH*, pp. 169-177 (doi: 10.1145/280814.280860)

R. Westermann and B. Sevenich (2001) Accelerated Volume Ray-Casting Using Texture Mapping, *IEEE Computer Graphics and Applications*, pp. 271-278 (doi: 10.1109/VISUAL.2001.964521)

L. Westover (1990) Footprint evaluation for volume rendering. In *Proc. ACM SIGGRAPH*, pp. 367-376 (doi: 10.1145/97880.97919)

L. Wiechert, A. Comerford, S. Rausch and W.A. Wall (2011) Advanced Multi-scale Modelling of the Respiratory System. In M. Klass, E. Koch and W. Schroder eds., *Fundamental medical and engineering investigations on protective artificial respiration, Notes on numerical fluid mechanics and multidisciplinary design, Vol. 116/2011*, pp. 1-32, Springer (doi: 10.1007/978-3-642-20326-8_1)

J. Woodring and H.W. Shen (2003a) Chronovolumes: a direct rendering technique for visualizing time-varying data. In *Proc. Eurographics/ IEEE TVCG Workshop on Volume graphics*, pp. 27-34 (doi: 10.1145/827051.827054)

J. Woodring, C. Wang and H.W. Shen (2003b) High dimensional direct rendering of time-varying volumetric data. In *Proc. IEEE Visualization'03*, pp. 417-424 (doi: 10.1109/VISUAL.2003.1250402)

J. Woodring and H.W. Shen (2009) Multiscale time activity data exploration via temporal clustering visualization spreadsheet. *IEEE transactions on visualization and computer graphics,* **15**(1):123-137, (doi: 10.1109/TVCG.2008.69)

X. Ye, D. Kao and A. Pang (2005) Strategy for seeding 3d streamlines. *In Proc. IEEE Visualization*, pp. 471-478 (doi: 10.1109/VISUAL.2005.1532831)

H. Younesy, T. Moller and H. Carr (2005) Visualization of time-varying volumetric data using differential time-histogram table. In *4th Int. Workshop on Volume Graphics (VG05)*, June 20-21, Stony Brook, NY, USA, pp. 21-29 (doi: 10.1109/VG.2005.194093)

H. Yu, C. Wang, C. Shene and J. Chen (2011) Hierarchical Streamline Bundles. *IEEE Trans. Visualization and Computer Graphics* **17**(12) (doi: 10.1109/TVCG.2011.155)

S. Zhang, S. Correia and D.H. Laidlaw (2008) Identifying white-matter fiber bundles in DTI data using an automated proximity-based fiber-clustering method. *IEEE Trans. Visualization and Computer Graphics*, **14**(5):1044-1053 (doi: 10.1109/TVCG.2008.52)

X. Zhang (2005) Space-scale animation: enhancing cross scale understanding of multiscale structures in multiple views. In *Proc. Coordinated and Multiple Views in Exploratory Visualization (CMV '05)*, pp. 109-120 (doi: 10.1109/CMV.2005.16)

X. Zhang (2009) Multiscale traveling: crossing the boundary between space and scale. *Virtual Reality*, **13**(2):101-115 (doi: 10.1007/s10055-009-0114-5)

X. Zhang and G.W. Furnas (2002) Social interactions in multiscale CVEs. In *Proc. 4th international conference on Collaborative virtual environments (CVE '02)*, pp. 31-38, New York, NY, USA (doi: 10.1145/571878.571884)

X. Zhang and G.W. Furnas (2005) mCVEs: using cross-scale collaboration to support user interaction with multiscale structures. *Presence Teleoperators and Virtual Environments,* **14**(1):31-46 (doi: 10.1162/1054746053890288)

H. Zhou (2005) A survey on ubiquitous graphics, *PhD Qualifying Exam (PQE) Report, Computer Science Department, Hong Kong University of Science and Technology,* Dec 2005

K. Zimmermann, R. Westermann, T. Ertl, C. Hansen and M. Weiler (2000) Level-of-detail volume rendering via 3D textures. In *Proc. IEEE symposium on Volume Visualization (VV 2000)***,** Oct 9-10, Salt Lake City, Utah, USA, pp. 7-13 (doi: 10.1109/VV.2000.10010)

Zygote Body (2011) http://www.zygotebody.com